

2015

## Secure portable execution and storage environments: A capability to improve security for remote working

Peter James  
*Edith Cowan University*

Follow this and additional works at: <https://ro.ecu.edu.au/theses>



Part of the [Human Resources Management Commons](#), [Information Security Commons](#), and the [Technology and Innovation Commons](#)

---

### Recommended Citation

James, P. (2015). *Secure portable execution and storage environments: A capability to improve security for remote working*. Edith Cowan University. Retrieved from <https://ro.ecu.edu.au/theses/1707>

This Thesis is posted at Research Online.  
<https://ro.ecu.edu.au/theses/1707>

# Edith Cowan

## University

### Copyright

### Warning

You may print or download ONE copy of this document for the purpose of your own research or study.

The University does not authorise you to copy, communicate or otherwise make available electronically to any other person any copyright material contained on this site.

You are reminded of the following:

- Copyright owners are entitled to take legal action against persons who infringe their copyright.
- A reproduction of material that is protected by copyright may be a copyright infringement.
- A court may impose penalties and award damages in relation to offences and infringements relating to copyright material. Higher penalties may apply, and higher damages may be awarded, for offences and infringements involving the conversion of material into digital or electronic form.

# **SECURE PORTABLE EXECUTION AND STORAGE ENVIRONMENTS: A CAPABILITY TO IMPROVE SECURITY FOR REMOTE WORKING**

**Peter James**

BSc. (Hons) Computer Science  
MSc. System Design  
Grad. Dip. Management Studies

This thesis is presented in fulfilment of the requirements for the degree of  
Doctor of Philosophy

Faculty of Health, Engineering and Science

Edith Cowan University

October 2015

## **Abstract**

Remote working is a practice that provides economic benefits to both the employing organisation and the individual. However, evidence suggests that organisations implementing remote working have limited appreciation of the security risks, particularly those impacting upon the confidentiality and integrity of information and also on the integrity and availability of the remote worker's computing environment. Other research suggests that an organisation that does appreciate these risks may veto remote working, resulting in a loss of economic benefits. With the implementation of high speed broadband, remote working is forecast to grow and therefore it is appropriate that improved approaches to managing security risks are researched. This research explores the use of secure portable execution and storage environments (secure PESEs) to improve information security for the remote work categories of telework, and mobile and deployed working.

This thesis with publication makes an original contribution to improving remote work information security through the development of a body of knowledge (consisting of design models and design instantiations) and the assertion of a nascent design theory. The research was conducted using design science research (DSR), a paradigm where the research philosophies are grounded in design and construction.

Following an assessment of both the remote work information security issues and threats, and preparation of a set of functional requirements, a secure PESE concept was defined. The concept is represented by a set of attributes that encompass the security properties of preserving the confidentiality, integrity and availability of the computing environment and data. A computing environment that conforms to the concept is considered to be a secure PESE, the implementation of which consists of a highly portable device utilising secure storage and an up-loadable (on to a PC) secure execution environment. The secure storage and execution environment combine to address the information security risks in the remote work location.

A research gap was identified as no existing 'secure PESE like' device fully conformed to the concept, enabling a research problem and objectives to be defined. Novel secure

storage and execution environments were developed and used to construct a secure PESE suitable for commercial remote work and a high assurance secure PESE suitable for security critical remote work. The commercial secure PESE was trialled with an existing telework team looking to improve security and the high assurance secure PESE was trialled within an organisation that had previously vetoed remote working due to the sensitivity of the data it processed.

An evaluation of the research findings found that the objectives had been satisfied. Using DSR evaluation frameworks it was determined that the body of knowledge had improved an area of study with sufficient evidence generated to assert a nascent design theory for secure PESEs.

The thesis highlights the limitations of the research while opportunities for future work are also identified. This thesis presents ten published papers coupled with additional doctoral research (that was not published) which postulates the research argument that 'secure PESEs can be used to manage information security risks within the remote work environment'.

The declaration page  
is not included in this version of the thesis

## **Acknowledgements**

I would like to acknowledge and thank Don Griffiths and Peter Hannay for their contribution to the four jointly written papers forming part of this thesis. Thank you also for the support provided by my employer Secure Systems Limited and my supervisors. Finally a big thank you to my family for the support they have provided during this long and, at times demanding experience.

# Table of Contents

<b>ABSTRACT</b>	<b>II</b>
<b>DECLARATION</b>	<b>IV</b>
<b>ACKNOWLEDGEMENTS</b>	<b>V</b>
<b>TABLE OF CONTENTS</b>	<b>VI</b>
<b>1 INTRODUCTION</b>	<b>1</b>
1.1 BACKGROUND	1
1.2 OVERVIEW OF RESEARCH AND THE RESEARCH ARGUMENT	3
1.3 MOTIVATION FOR RESEARCH	4
1.4 KEY TERMS AND DEFINITIONS	6
1.5 THE RESEARCH PROBLEM AND QUESTIONS	8
1.6 RESEARCH PARADIGM AND METHODOLOGY	9
1.7 PRIOR RESEARCH AND EXISTING KNOWLEDGE	10
1.7.1 <i>Knowledge Classes</i>	10
1.7.2 <i>Knowledge Consumed</i>	11
1.8 KNOWLEDGE CONTRIBUTION AND ITS SIGNIFICANCE	11
1.8.1 <i>The Growth of Knowledge over Time</i>	11
1.8.2 <i>Positioning the Research</i>	12
1.8.3 <i>Significance of Research</i>	12
1.9 SCOPE OF RESEARCH	13
1.9.1 <i>Remote Work Categories Considered</i>	13
1.9.2 <i>Security Issues</i>	14
1.9.3 <i>Remote Workers</i>	15
1.9.4 <i>Secure PESEs</i>	15
1.9.5 <i>Limitations and Constraints</i>	16
1.10 THE RESEARCH PAPERS	17
1.11 EVOLUTION OF KEY CONCEPTS AND TERMINOLOGY	19
1.12 STRUCTURE OF THESIS	22
1.13 SUMMARY	23
<b>2 RESEARCH PROBLEM AND OBJECTIVES</b>	<b>24</b>
2.1 OVERVIEW	24
2.1.1 <i>Structure</i>	24
2.1.2 <i>Establishing the Research Problem and Objectives - Design Cycle 1</i>	24
2.2 PART 1: SEARCHING FOR A RESEARCH GAP	25
2.2.1 <i>Overview</i>	25
2.2.2 <i>Background to Papers 1, 2 and 3</i>	25
2.2.3 <i>Integrating Security Technology into Commercially Available Smartphones</i>	26
2.2.3.1 Preamble	26
2.2.3.2 Prior Research and Knowledge	27
2.2.3.3 Paper 1	28
2.2.3.4 Synopsis	48
2.2.4 <i>Preventing Data Loss from Secure Portable Storage Devices</i>	50
2.2.4.1 Preamble	50
2.2.4.2 Prior Research and Knowledge	51
2.2.4.3 Paper 2	52
2.2.4.4 Synopsis	64
2.2.5 <i>Applicability of Existing Security Models</i>	66
2.2.5.1 Preamble	66
2.2.5.2 Prior Research and Knowledge	66
2.2.5.3 Paper 3	67
2.2.5.4 Synopsis	81
2.2.6 <i>Summary</i>	81



2.3	PART 2 – LITERATURE REVIEW	82
2.3.1	<i>The Approach</i>	82
2.3.2	<i>Introducing the Concept of the Secure PESE</i>	83
2.3.2.1	Preamble	83
2.3.2.2	Prior Research and Knowledge	85
2.3.2.3	Paper 4	86
2.3.2.4	Synopsis	108
2.3.2.5	Attributes of the Initial Concept	109
2.3.3	<i>Remote Work – Categories Considered</i>	110
2.3.4	<i>Prior Research into Secure Remote Working</i>	111
2.3.5	<i>Prior Research and Development into Secure PESEs</i>	116
2.3.5.1	Bespoke Hardware based Secure PESEs with Virtualised Secure PEE	116
2.3.5.2	USB Flash Drive based Secure PESEs with Bootable Secure PEE	121
2.3.5.3	Synopsis of Prior Research and Development into Secure PESEs	125
2.3.6	<i>Holistic Review of Security Issues and the Secure PESE Countermeasures</i>	125
2.3.6.1	Background	125
2.3.6.2	Consumed Knowledge	126
2.3.6.3	Location Security	127
2.3.6.4	Personnel Security	129
2.3.6.5	Insecure Use of Technology	133
2.3.6.6	Technology Vulnerabilities.	137
2.3.7	<i>Synopsis of Analysis</i>	139
2.4	THE RESEARCH PROBLEM	140
2.4.1	<i>Establishing the Secure PESE Concept</i>	140
2.4.2	<i>Research Gap</i>	142
2.4.3	<i>Research Problem Definition</i>	143
2.4.4	<i>Research Objectives</i>	143
2.4.5	<i>Research Questions</i>	144
2.5	SUMMARY	145
<b>3</b>	<b>RESEARCH DESIGN</b>	<b>146</b>
3.1	OVERVIEW	146
3.2	RATIONALE FOR SELECTING DESIGN SCIENCE RESEARCH	147
3.2.1	CHOOSING THE RESEARCH PARADIGM AND METHODOLOGY	147
3.2.2	SELECTION OF DSR	148
3.3	DESIGN SCIENCE RESEARCH METHODOLOGY	149
3.3.1	OVERVIEW OF METHODOLOGY	149
3.3.2	DEVELOPMENT ENVIRONMENT	152
3.4	THE CONSUMPTION AND PRODUCTION OF KNOWLEDGE	154
3.4.1	KNOWLEDGE CLASSES	154
3.4.2	DESIGN CYCLES	156
3.4.3	KNOWLEDGE CONSUMED	159
3.4.4	PRODUCTION OF KNOWLEDGE	160
3.5	DEMONSTRATION	161
3.6	EVALUATION	162
3.7	SUMMARY	164
<b>4</b>	<b>DESIGN AND DEVELOPMENT</b>	<b>165</b>
4.1	OVERVIEW	165
4.1.1	<i>Structure</i>	165
4.1.2	<i>The Papers</i>	165
4.2	BASELINING THE DESIGN - DESIGN CYCLE 2	166
4.2.1	<i>Discounting the use of Virtualisation</i>	166
4.2.1.1	Preamble	166
4.2.1.2	Prior Research and Knowledge	167
4.2.1.3	Paper 5	168
4.2.1.4	Synopsis	189
4.2.2	<i>Modelling the Secure PESE Concept</i>	190
4.2.2.1	Threat Model	191

4.2.2.2	Conceptual Design Model	191
4.2.2.3	Operational Model	191
DEVELOPING THE COMMERCIAL GRADE SECURE PESE – DESIGN CYCLE 3		196
4.2.3	<i>Background</i>	196
4.2.4	<i>A Secure Up-loadable Hardened Application</i>	197
4.2.4.1	Preamble	197
4.2.4.2	Prior Research and Knowledge	198
4.2.4.3	Paper 6	199
4.2.4.4	Synopsis	221
4.2.5	<i>A Secure Hardened Bootable Operating System</i>	223
4.2.5.1	Preamble	223
4.2.5.2	Prior Research and Knowledge	224
4.2.5.3	Paper 7	226
4.2.5.4	Synopsis	250
4.2.6	<i>The Mini SDV – A Platform for a Commercial Grade Secure PESE</i>	251
4.2.6.1	Overview	251
4.2.6.2	Description in Published Papers	251
4.2.6.3	Knowledge Consumed	252
4.2.6.4	New Improved Features	252
4.2.6.5	Knowledge Contribution	255
4.2.7	<i>Combining the Mini SDV, MEE and Fireguard as a Commercial Grade Secure PESE</i>	255
4.3	DEVELOPING THE HIGH GRADE SECURE PESE – DESIGN CYCLE 4	256
4.3.1	<i>Background</i>	256
4.3.2	<i>Using Secure PESEs in Network Centric Organisations</i>	257
4.3.2.1	Preamble	257
4.3.2.2	Prior Research and Knowledge	258
4.3.2.3	Paper 8	259
4.3.2.4	Synopsis	267
4.3.3	<i>The SDV-HA – A Platform for a High Grade Secure PESE</i>	268
4.3.3.1	Preamble	268
4.3.3.2	Prior Research and Knowledge	269
4.3.3.3	Paper 9	270
4.3.3.4	Synopsis	287
4.4	SUMMARY	288
<b>5</b>	<b>DEMONSTRATION</b>	<b>289</b>
5.1	OVERVIEW	289
5.2	DEMONSTRATION PROCESS	289
5.2.1	<i>Testing</i>	290
5.2.2	<i>Trial</i>	292
5.2.3	<i>Certification</i>	292
5.2.4	<i>Commercialisation</i>	293
5.2.5	<i>Demonstration Configuration</i>	294
5.3	COMMERCIAL GRADE SECURE PESE	294
5.3.1	<i>Testing</i>	295
5.3.2	<i>Trial</i>	295
5.3.3	<i>Certification</i>	297
5.3.4	<i>Commercialisation</i>	297
5.3.5	<i>Extending the Use Case of the Bootable Secure PEE</i>	298
5.3.5.1	Preamble	298
5.3.5.2	Prior Knowledge and Research	299
5.3.5.3	Paper 10	300
5.3.5.4	Synopsis	313
5.3.6	<i>Summary of Results</i>	313
5.4	HIGH GRADE SECURE PESE	314
5.4.1	<i>Testing</i>	314
5.4.2	<i>Trial</i>	315
5.4.3	<i>Certification</i>	316
5.4.4	<i>Commercialisation</i>	316
5.4.5	<i>Summary of Results</i>	317

5.5	SUMMARY	317
<b>6.</b>	<b>EVALUATION AND DISCUSSION</b>	<b>318</b>
6.1	APPROACH	318
6.2	EVALUATION OF RESEARCH FINDINGS	318
6.2.1	<i>Demonstration Results</i>	318
6.2.2	<i>Research Questions</i>	321
6.2.3	<i>Satisfying the Research Objectives</i>	325
6.2.4	<i>Research Problem</i>	327
6.2.5	<i>Further Observations on Research Outcomes</i>	328
6.3	CONTRIBUTION TO DESIGN KNOWLEDGE AND THE AREA OF STUDY	331
6.3.1	<i>Summary of Knowledge Contributions</i>	331
6.3.1.1	<i>Design Cycle 1 - Establishing the Research Problem and Objectives</i>	332
6.3.1.2	<i>Design Cycle 2 – Baselineing the Design</i>	332
6.3.1.3	<i>Design Cycle 3 – Commercial Grade Secure PESE</i>	332
6.3.1.4	<i>Design Cycle 4 - High Grade Secure PESE</i>	333
6.3.2	<i>Classification of Knowledge Contribution</i>	333
6.3.2.1	<i>Knowledge Considered</i>	333
6.3.2.2	<i>Classifying the Knowledge Contribution</i>	334
6.3.2.3	<i>Evidence of Improvement</i>	334
6.3.3	<i>Summary</i>	336
6.4	CONTRIBUTION TO DESIGN THEORY	337
6.4.1	<i>Purpose and Scope</i>	337
6.4.2	<i>Constructs</i>	338
6.4.3	<i>Principles of Form and Function</i>	338
6.4.4	<i>Artifact Mutability</i>	338
6.4.5	<i>Testable Propositions</i>	339
6.4.6	<i>Justificatory Knowledge</i>	339
6.4.7	<i>Implementation Principles</i>	340
6.4.8	<i>Expository Instantiation</i>	340
6.4.9	<i>Summary</i>	341
6.5	DSR CONTRIBUTION TYPE	341
6.6	SUMMARY	341
<b>7.</b>	<b>CONCLUSION</b>	<b>342</b>
7.1	IMPACT AND SUCCESS OF RESEARCH	342
7.2	ORIGINALITY AND SIGNIFICANCE OF RESEARCH	343
7.3	LIMITATIONS OF RESEARCH	344
7.4	ASSESSMENT OF THE RESEARCH DESIGN	346
7.5	FUTURE WORK	346
7.6	RESEARCH SYNOPSIS	347
	<b>REFERENCES</b>	<b>348</b>
	<b>APPENDIX 1 – CO-AUTHOR STATEMENTS</b>	<b>369</b>
	<b>FIGURES</b>	
FIGURE 1.1	RELATIONSHIP BETWEEN PAPERS	20
FIGURE 3.1	DSRM PROCESS MODEL	150
FIGURE 3.2	MODEL OF DOCUMENTATION OUTPUTS FROM THE DEVELOPMENT ENVIRONMENT	153
FIGURE 3.3	MODEL OF DEVELOPMENT OF KNOWLEDGE OVER FOUR DESIGN CYCLES	157
FIGURE 3.4	GREGOR AND HEVNER KNOWLEDGE CONTRIBUTION FRAMEWORK	163
FIGURE 3.5	DSR KNOWLEDGE MATURITY MODEL	164
FIGURE 4.1	SECURE PESE THREAT MODEL	192
FIGURE 4.2	SECURE PESE CONCEPTUAL DESIGN MODEL	194
FIGURE 4.3	SECURE PESE OPERATIONAL MODEL	195

FIGURE 4.4	MINI SDV CONFIGURED AS A SECURE PESE	253
FIGURE 5.1	EXAMPLE SECURE PESE CONFIGURATION	294
FIGURE 5.2	COMMERCIAL GRADE SECURE PESE TEST CONFIGURATION	295
FIGURE 5.3	COMMERCIAL GRADE SECURE PESE TRIAL CONFIGURATION	296
FIGURE 5.4	COMMERCIAL GRADE SECURE PESE CERTIFICATION CONFIGURATION	297
FIGURE 5.5	HIGH GRADE SECURE PESE TEST CONFIGURATION	314
FIGURE 5.6	HIGH GRADE SECURE PESE TRIAL CONFIGURATION	315
FIGURE 5.7	HIGH GRADE SECURE PESE CERTIFICATION CONFIGURATION	316
FIGURE 6.1	MODEL SHOWING THE EVOLUTION OF THE RESEARCH OUTCOMES	319

## **TABLES**

TABLE 2.1	CONFORMANCE OF CSIRO TED TO INITIAL CONCEPT	117
TABLE 2.2	CONFORMANCE OF CSIRO TED TO THE REQUIRED FEATURES	118
TABLE 2.3	CONFORMANCE OF BULL GLOBULL TO INITIAL CONCEPT	119
TABLE 2.4	CONFORMANCE OF BULL GLOBULL TO THE REQUIRED FEATURES	119
TABLE 2.5	CONFORMANCE OF MXI SECURITY STEALTH MXP TO INITIAL CONCEPT	120
TABLE 2.6	CONFORMANCE OF MXI SECURITY STEALTH MXP TO THE REQUIRED FEATURES	121
TABLE 2.7	CONFORMANCE OF BECRYPT TRUSTED CLIENT TO INITIAL CONCEPT	122
TABLE 2.8	CONFORMANCE OF BECRYPT TRUSTED CLIENT TO THE REQUIRED FEATURES	122
TABLE 2.9	CONFORMANCE OF USA DOD LPS TO INITIAL CONCEPT	124
TABLE 2.10	CONFORMANCE OF USA DOD LPS TO THE REQUIRED FEATURES	124
TABLE 4.1	ASSOCIATION BETWEEN THREATS AND RISKS	193

# **1 Introduction**

## **1.1 Background**

Remote working has become an established practice (Baruch, 2000; Watts-Englert et al., 2012; Ye, 2012) primarily due to advances in information and communications technology (ICT), allowing the remote work location to both communicate with, and work effectively as an extension of the corporate office (Peacey, 2006). In Australia the work practice is forecast to substantially increase with the implementation of high speed broadband (DBCDE, 2011; Telework, 2013). Remote working is a term that encapsulates several categories including teleworking (Lister and Harnish, 2011), mobile working (Dade, 2013) and deployed working (ADoD, 2007).

The availability of highly portable computing devices, increasing Internet speeds and cloud computing (Armbrust et al., 2009; Optus, 2012; ACMA, 2013) has provided the ICT infrastructure required to enable remote working to grow. Although this ICT infrastructure allows remote access to, and processing of sensitive corporate information it has generally evolved without security being a primary requirement (Goth, 2012; Astani et al., 2013) and therefore is vulnerable to security risks. Whilst the implementation of remote working provides tangible business benefits, not ensuring that effective information security is achieved can be an omission or failing of an organisation (Clear, 2007). Without full consideration of the security threats or through the implementation of only limited, weak or no information security controls, an organisation risks losing data and/or being vulnerable to cyber-attack (Ponemon, 2012). There are several possible reasons why inadequate information security occurs:

1. Although the issues and risks within the remote work environment have been investigated (Sturgeon, 1996; Hoogendijk, 2006; Deloitte, 2011), research has shown there is a lack of awareness, understanding and/or an appreciation of these risks, particularly by small and medium sized organisations (Clear, 2007).
2. An organisation enforcing best practice information security management (ISO/IEC 27001:2013, 2013) will have appropriate physical and logical security controls established to protect information assets in the corporate office(s). However, the

enforcement of security management for remote working (NIST, 2007) can prove difficult as the information processing/computing environment resides outside the organisation's sphere of physical ICT management (Bates, 2010).

3. The standard information processing actions of a PC's execution environment (i.e. the PC's operating system and its software application suite) can result in the unintended disclosure of information as the execution environment stores temporary copies of data (often unbeknown to the user) which remain on the PC's hard disk drive (HDD) and could be recovered at a later date by an unauthorised user (Jones et al., 2008).

These three issues can result in computing devices used for remote work being vulnerable to information security risks. Security training and awareness programmes, strong ICT management of remote computing devices and configuration of the execution environment (to delete temporary data upon shutdown) can manage these respective issues. However, a single solution that can provide an appropriate level of security to address all of these issues is desirable.

Conversely, information security can be used as a barrier to remote working (GSA, 2002; Whiteman and Dick, 2006; Deloitte, 2011). Organisations that understand the risks to information processed outside the secure office environment often deny staff the opportunity to work remotely or constrain the activities that can be performed remotely due to information security concerns (Gibson et al, 2002; NIST, 2009; Kowalski & Swanson, 2005). By denying staff the opportunity to work remotely neither the organisation nor the staff accrue the benefits of remote working, possibly making the organisation uncompetitive as it may not attract the best staff nor obtain savings in reduced office and other infrastructure costs (Baruch, 2000; Davis, 2011; Ye, 2012). A solution that could be deployed, that would provide an appropriate level of assurance and manage information security risks, may lead to concerned organisations re-considering remote working.

More recently in Australia remote working (in particular teleworking) has received increased attention (DBCDE, 2011; Deloitte, 2011) due to the expected contribution that remote working could make to national economic and productivity improvements (Davis, 2011; DBCDE, 2013). It is therefore relevant that research into secure computing

technology for remote working is performed to improve security, remove a barrier and allow a concerned organisation to implement a remote working policy.

A solution for remote work security (Goslar, 2000; Pyöriä, 2003; Whiteman and Dick, 2006; NIST, 2009; OMB, 2011; OPM, 2011) would need to include:

1. Capabilities that preserve the confidentiality of information, and the integrity and availability of the execution environment from information security risks.
2. A high level of portability to enable remote working to be achieved on any PC.
3. Support for a wide range of software applications.
4. A user friendly interface that would enable efficient and effective use with no or minimal training.
5. Secure storage of data.

In this thesis a solution is designed and constructed that delivers secure storage and a secure execution environment, is highly interoperable across PCs, has a user friendly interface and allows remote work to be performed efficiently and effectively. The security features of the solution will limit the opportunities for cyber-attack, data loss and forensic discovery of data.

## **1.2 Overview of Research and the Research Argument**

This thesis describes the doctoral research (a PhD with publication) performed to develop improved secure portable computing solutions to support remote working (the research area of study). This thesis asserts that the generated body of knowledge makes a contribution to the area of study and also defines a nascent design theory underpinned by an informal concept<sup>1</sup> and improved secure portable computing devices.

The researcher has defined the term secure Portable Execution and Storage Environment (secure PESE). The term is used to represent a concept and a physical embodiment. As a concept a secure PESE is an informal model that allows the suitability of secure portable computing devices for remote work to be determined or provides a high level definition for the development of a new device. The physical embodiment of a secure PESE is a

---

<sup>1</sup> An informal concept is defined as a concept with a vocabulary/notation in English as opposed to a formal concept where a mathematical notation is used.

device that combines two capabilities - a secure portable execution environment (secure PEE) and secure data storage. The device allows the secure PEE (e.g. a hardened operating system and/or hardened software application suite) to be uploaded from the secure storage device onto a host PC to execute. Secure storage is achieved by the device providing authentication, access controls and encryption of the storage medium. The research involved the development of secure PESEs based upon augmenting existing research and improving available technologies to provide a secure computing environment to improve information security for remote working.

This PhD with publication presents a set of ten papers that together with other doctoral research documented in the thesis shows the progressive development of a body of knowledge postulating the research argument that *'secure PESEs can be used to manage information security risks within the remote work environment'*. This PhD thesis presents the outcomes of a Design Science Research (DSR) (Vaishnavi and Kuechler, 2014) project where a set of artifacts<sup>2</sup> is built to demonstrate the research argument.

The ten papers were prepared (and published) as each substantial area of investigation was progressed. In addition Chapters 2, 4, 5 and 6 present knowledge developed but not published that forms an integral part of the research. Each paper is numbered 1 to 10 to reflect the order in which the research was conducted and the paper prepared, although the respective paper publication date does not necessarily follow the chronological progression of the research. The thesis is structured into seven chapters and follows a publication schema for DSR (Gregor & Hevner, 2013) with the papers presented in Chapters 2, 4, and 5.

### **1.3 Motivation for Research**

The factors influencing the research were initially the researcher's long held personal interest in remote working (James, 1991) and his workplace responsibility for the design and development of secure portable storage technology. The Australian Government's goal (Telework, 2013) to increase the level of remote working also became a research driver as it added national context and the possibility of commercialisation opportunities.

---

<sup>2</sup> An artifact is a construct, model, process, method or instantiation developed using a research methodology; in particular the term artifact is used in Design Science Research.



The researcher is employed as a product development manager at Secure Systems Limited (SecureSystems, 2013), a security technology company that specialises in the design and manufacture of innovative secure data storage technology. In that role the researcher is responsible for leading research programmes for new products and product variations, liaising with customers and the wider market to identify new product opportunities, and keeping abreast of the relevant research literature for both secure storage and the possible applications for secure storage. Secure Systems has designed, developed and patented the Silicon Data Vault (SDV) technology, a portfolio of secure storage technologies for use in laptops, portable storage devices and embedded systems (PocketSDV, 2006; MiniSDVCert, 2012; SDV-HA, 2013).

The motivating goals for the research were:

- **Improve Remote Work Security:** To contribute to the body of research that prevents or limits the effects of information security risks in the remote work environment.
- **Increase Security Awareness:** The need to increase the awareness of information security risks in the remote work environment.
- **Develop New Solutions:** The need for a secure portable computing capability to improve information security in remote work locations.
- **Increase Remote Work:** To facilitate the increase in remote working by providing a body of knowledge and demonstrable artifacts to support both an organisation implementing remote work policies and to enable an organisation, citing information security risks as the reason for vetoing the work practice, to reconsider that position.
- **Commercialise Research:** An opportunity to develop innovative technology that might be commercialised into cost effective solutions.

It was initially the researcher's search for new innovative ideas to apply the SDV technology that provided the catalyst for the PhD. The SDV technology played an important role throughout the PhD with the constructed secure PESEs utilising the SDV technology. Conducting a PhD with publication at Edith Cowan University (ECU) enabled the researcher to apply academic rigour to his research whilst both publishing research results and using the research outcomes to potentially identify new product/solution ideas for Secure Systems and its SDV technology. The Australian Government's emphasis

on encouraging organisations to adopt remote working further validated the need for research to support secure remote working.

#### 1.4 Key Terms and Definitions

The following terms are used throughout this thesis:

**Secure Portable Execution Environment (secure PEE):** a term to define the software execution environment held on a secure PESE that is uploaded onto a PC upon successful authentication to provide a set of tools/applications. It is possible to have more than one secure PEE installed on a secure PESE. A secure PEE can be constructed from a range of software capabilities with initially (in this research) virtualisation being considered; however, as the research progressed the preferred approach is to either:

1. Use a hardened bootable operating system with the required set of tools/applications installed; or
2. Alternatively, the secure PEE can consist of a small set of hardened tools/applications that are uploaded onto a PC that is executing an operating system.

**Secure PESE:** a term used to both represent a concept for the use of secure portable computing technology for remote work and also to describe a device that implements the concept. The concept is an informal model defined by the following set of attributes:

- Prevent unauthorised access to a secure execution environment and any stored data.
- Preserve the integrity and availability of the execution environment
- Preserve the confidentiality and integrity of any stored data.
- Preserve the confidentiality and integrity of any data sent to/from the remote location.
- Provide a highly portable device with a user friendly execution environment that can be used on any available PC.
- Provide an execution environment with only the necessary secured functionality for the specific remote work activities.
- Limit the execution environment's access to the internal storage device(s) of the host PC.

As a device a secure PESE is defined as a highly portable, interoperable, secure computing device that combines a user friendly, limited functionality, execution environment together with secure storage. Upon successful authentication a secure PESE connected to a host PC will upload the execution environment and allow access to stored data. A secure PESE prevents unauthorised access to both the execution environment and stored data, preserves the confidentiality, integrity and availability of data and the execution environment, and limits access to the PC's internal disk drive. A secure PESE protects information by managing the risks of cyber-attack, data loss and forensic data discovery in the remote work environment. In Chapter 4 a conceptual design model (Figure 4.2) and an operational model (Figure 4.3) are presented. The underlying philosophy of the secure PESE is that it extends the corporate office security boundary out to the remote work location (Peacey, 2006).

A secure PESE implements access controls, authentication, cryptographic security mechanisms, anti-tamper mechanisms, a large storage medium and an up-loadable secure PEE. As a device a secure PESE is packaged into a small portable unit with external interfaces for both connection to a PC (e.g. USB, eSATA) and authentication tokens.

**SDV:** a set of security mechanisms, PC and disk interface technologies and peripheral electronics implemented in a combination of hardware and software to provide a secure storage device. The SDV security mechanisms consist of the following innovative set of security controls - strong authentication, multi-user/role capability, differentiated access controls, encryption and strong key management. In the research described in this thesis SDV technology is used to construct the hardware platform for secure PESEs.

The terms cyber-attack, malware, data loss and forensic data discovery are widely used in information security, in this thesis they are constrained to the following definitions:

**Cyber-attack:** is an offensive activity with the objective of denying, degrading, disrupting or destroying information or ICT systems (ASD, 2014). Malware is the primary tool used to launch a cyber-attack.

**Malware:** Software developed to take full or partial control of a computer through a cyber-attack. Malware can take many forms with new categories continually emerging;

however, malware that can launch a cyber-attack can currently be categorised as a virus, worm, downloader trojan, spyware, rootkit, ransom ware or a backdoor trojan (Sophos, 2009).

**Data loss:** Unauthorised access to data occurring through loss, theft or cyber-attack on a storage medium or memory. In this thesis data leakage is sometimes used to mean data loss.

**Forensic data discovery:** The (unauthorised) recovery of data from a computer storage medium or memory using forensic techniques and tools but (as defined in this thesis) not necessarily performed using a forensic methodology, i.e. forensic techniques and tools are used to gain unauthorised access to data but are not performed in a manner to recover and forensically preserve the data. N.B. This definition is specific to this thesis.

## 1.5 The Research Problem and Questions

Several investigations, a comprehensive literature review and a holistic review of the remote work security issues were performed to identify a research problem and as a consequence afforded the following additional outcomes:

- A comprehensive description of the remote work information security issues and threats was prepared.
- A set of experimental secure PESEs was constructed.
- A set of secure PESE functional requirements to counter the remote work threats was specified.
- A finalised secure PESE concept for remote working was defined.

An analysis of existing 'secure PESE like' artifacts and products using the aforementioned functional requirements identified a research gap leading to the following research problem:

**A requirement exists to develop an enhanced secure PESE that limits the exploitation of vulnerabilities by hardening the execution environment, providing a tamper detection and response capability and ensuring no data remnants are recoverable from the host PC.** These enhancements will further

strengthen the secure PESE against the risk of cyber-attack, data loss and forensic data discovery in the remote work environment.

To enable the research problem to be addressed the following research questions were formulated:

- *How can a useable and maintainable hardened operating system and/or small set of hardened applications be developed?*
- *How can anti-tamper mechanisms be implemented into a small form factor and highly portable device?*
- *How can a useable and maintainable execution environment be configured to store all temporary data on a secure PESE partition?*

The research considers these questions during the design, development, demonstration and evaluation of the research artifacts (described in Chapters 4, 5 and 6).

## **1.6 Research Paradigm and Methodology**

DSR was selected as the research paradigm as it is specifically intended for design and development based research (McKay and Marshall, 2005; Hevner and Chatterjee, 2010). An interpretive paradigm, with action research (Chiasson et al., 2008; Antill, 1986) was considered as the methodology, but was discounted (as discussed in Chapter 3). DSR is based upon a research paradigm where the ontology, epistemology and axiology philosophies are grounded in design and construction. The specific method selected was Design Science Research Methodology (DSRM) (Peppers et al., 2007) because:

1. It is highly cited (ACM-DL, 2014) in academic papers and therefore has a proven track record for the development of novel software and ICT based artifacts.
2. Its process model consists of six well defined activities also known as process elements: problem identification; defining of objectives; design and development; demonstration; evaluation; and communication, that align with constructing novel artifacts based upon improving existing technology. The methodology supports iteration from the evaluation or communication process elements back to the objectives or design/development elements.

3. The researcher's background includes developing systems using structured methodologies and the DSRM has similarities to a structured systems development methodology, thus enabling the researcher to apply the methodology rigorously.

A model of the DSRM is given in Figure 3.1 (Chapter 3). The research results of a DSR project are innovative designs (i.e. artifacts) with assessment of the results performed through showing a contribution to design knowledge and possibly to design theory. DSR complements the more traditional interpretive and positivist paradigms. A characteristic of the doctoral research is that the DSR paradigm was applied by the researcher whilst conducting research at his place of work. The outcome is that the DSR paradigm was used to design new technology that was commercialised.

## 1.7 Prior Research and Existing Knowledge

### 1.7.1 Knowledge Classes

This thesis uses an approach that classifies knowledge consumed and produced in DSR (Gregor and Hevner, 2013) as:

- ***Descriptive knowledge*** is the “what” knowledge from observations, measurements, natural laws, principles and patterns.
- ***Prescriptive knowledge*** is the “how” knowledge represented by the design attributes of innovative artifacts and can take the form of a construct, model, method or instantiation.
- ***Justificatory knowledge*** is the “why” knowledge that explains the underlying philosophy for an artifact's construction. Formal justificatory knowledge is the existing recognised theories and informal justificatory knowledge is the specialist knowledge of practitioners or other informal grounded knowledge.

A DSR project consumes descriptive knowledge and (possibly) existing prescriptive knowledge and produces predominately prescriptive knowledge but may also generate descriptive knowledge in the form of an artifact's demonstration results. Justificatory knowledge is consumed as it informs the design.

### **1.7.2 Knowledge Consumed**

The knowledge consumed during the research can be categorised into the following overarching categories:

- Security issues in the remote work environment (descriptive knowledge).
- Empirical evaluation results from earlier stages in the doctoral research (descriptive knowledge).
- Stakeholder input and market trends (descriptive knowledge).
- Existing technology/artifacts suitable for secure portable devices and secure storage (prescriptive knowledge).
- Design knowledge from earlier stages in the doctoral research (prescriptive knowledge).
- Security risk assessment, operating system, and security and cryptographic engineering design theories (formal justificatory knowledge).
- Theory of Network Centric Warfare (formal justificatory knowledge).
- Practitioner knowledge (informal justificatory knowledge).

The prior knowledge provides a baseline of knowledge to use or assist in the design of new artifacts (Gregor and Hevner, 2013).

## **1.8 Knowledge Contribution and Its Significance**

### **1.8.1 The Growth of Knowledge over Time**

In this thesis consumed and produced knowledge is presented in four design cycles (Gregor and Hevner, 2013). The concept of knowledge generation through design cycles is used to capture the descriptive and prescriptive knowledge outcomes from earlier research that can be consumed in the development of the final artifacts. The four cycles are: Establishing the Research Problem and Objectives (design cycle 1), Baselineing the Design (design cycle 2), Developing the Commercial Grade Secure PESE (design cycle 3) and Developing the High Grade Secure PESE (design cycle 4). Figure 3.3 (Chapter 3) portrays a conceptual model of the four design cycles. Presentation using design cycles was identified as the research was being finalised; however, the approach reflected how the doctoral research had been conducted and therefore provided an elegant method of

presenting the consumed and produced knowledge. The approach was particularly useful for presenting the comprehensive knowledge produced early in the research that was not reflected in published papers, e.g. the review of existing 'secure PESE like' artifacts/products and the holistic review of security issues.

### **1.8.2 Positioning the Research**

A knowledge contribution framework for DSR is used to position the research (Gregor and Hevner, 2013). The framework categorises a knowledge contribution as an improvement (a new/improved solution for a known problem), or an exaptation (extending a known solution to a new problem) or an invention (a new solution for a new problem). The body of knowledge presented in this thesis fits into the improvement category as the remote work security problem is known and the secure PESE, through a combination of innovative technology improvements and new technology, provides an improved solution. An anatomy of a design theory (Gregor and Jones, 2007) is used to assert that a nascent design theory for secure PESEs has emerged from the research.

### **1.8.3 Significance of Research**

The growing risk of cyber-attack in remote work locations (Bates, 2010), coupled with the expected growth in remote working (DBCDE, 2011), accentuates the importance of this research. The research is therefore considered significant as it provides a deep analysis of the security problem and constructs secure artifacts to address the problem. More specifically it:

1. Improves the use of secure portable computing technology for remote working by:
  - a. Presenting a holistic description of the information security issues in the remote work environment.
  - b. Providing a comprehensive identification of the information security threats.
  - c. Specifying a set of secure PESE functional requirements.
  - d. Introducing the concept of a secure PESE, where the concept:
    - i. Provides an assessment and implementation model for secure portable computing devices for remote work. The concept attributes encompass the constructs and a model that can be used to assess or construct



partially or fully compliant secure PESEs, i.e. the research artifacts can be considered as just one example set of secure PESEs.

- ii. Facilitates understanding of remote work security and identifies the technology needed to match corporate office security. Essentially the secure PESE concept models an extension of corporate office security to the remote location.
- e. Delivering demonstrable research artifacts (i.e. secure PEE, secure storage devices and secure PESE implementations) that address the research problem.

These artifacts have been:

- i. Constructed (to product quality standards) primarily through the novel improvement/enhancement of existing technology components.
- ii. Comprehensively tested and also trialled in real remote work situations.
- iii. Certified and commercialised in the case of the Mini SDV and the SDV-HA secure portable storage devices.

2. Asserts a nascent design theory for secure PESEs for use in remote working.

## **1.9 Scope of Research**

The scope of the research included:

- Considering three categories of remote work (teleworking and, mobile and deployed working).
- Information security issues grouped by location, personnel, insecure use of technology and technology vulnerabilities.
- The development of both high grade and commercial grade secure PESEs.

### **1.9.1 Remote Work Categories Considered**

The research focuses on enhancing security for the following remote work categories:

1. Teleworking (Lister and Harnish, 2011), also known as telecommuting, is an established practice defined as the external processing of information conducted predominately from home.

2. Mobile working (Dade, 2013) is defined as work performed outside the corporate office and conducted from any location as and when the work needs to be performed. Like teleworking, mobile working has become an established practice.
3. Deployed working (ADoD, 2007) is considered to be the work practice where a group of workers is assigned to work in a particular location for a period of time usually to perform a well-defined assignment.

Teleworking and deployed working receive particular attention with a telework trial being performed using transaction-oriented workers<sup>3</sup> and a deployed work trial involving network centric operations (Folks and Richard, 2011). Teleworking is expected to be the practice that will increasingly deliver economic benefits by allowing various types of work to be performed from the home (DBCDE, 2013). Deployed working in a network centric organisation (Abrams, 2009) was identified as an area where secure PESEs could enhance remote node security.

### **1.9.2 Security Issues**

The issues affecting information security were assigned to in the following groups:

1. Location – the physical security issues resulting from work being performed outside of a secure corporate office (Zbar, 2000; Protective-Management, 2011; Protective-ICT, 2011; ASIAL, 2014).
2. Personnel – the security issues that may occur due to the experience and/or behaviour of a remote worker (GSA, 2002; Hoogendijk, 2006; Joice, 2007; Crossler et al., 2013; Jilani et al., 2013).
3. Insecure Use of Technology – the security issues resulting from incorrect configuration of hardware and software and/or the way the remote work computing environment is used (Furnell, 2006; Hoogendijk, 2006; NIST, 2007; Furnell, 2009).
4. Technology Vulnerabilities – the inherent vulnerabilities in hardware and software (used by the remote worker) that can be exploited (Hoogendijk, 2006; Whiteman and Dick, 2006; Jones et al., 2008).

---

<sup>3</sup> A transaction-oriented worker can be defined as a skilled worker who performs a continuous stream of activities (transactions). Whilst a transaction may involve problem solving, the solution is often known to the worker and typically does not involve the generation of new knowledge.

Each group of security issues contains a set of threats and vulnerabilities which if exploited can lead to information security risks in the form of cyber-attack, data loss or the forensic discovery of data. A secure PESE is designed to provide countermeasures to the security issues and therefore manage the information security risks.

### **1.9.3 Remote Workers**

Remote workers will have different levels of ICT skills and experience (GSA, 2002). The secure PESE artifacts have been constructed to require minimal ICT skills for use, although system administration skills are required for configuration and deployment.

### **1.9.4 Secure PESEs**

As a device secure PESE provides the specific set of applications/tools required to perform a specific role/function within an organisation. This set of applications/tools forms the secure PEE which is installed onto a portable and highly interoperable secure storage device (to form the secure PESE). The secure PESE then enables remote working to be conducted securely from an available x86 PC. A security threat common to all remote work categories can have a different risk level for each category depending upon the sensitivity of information stored and processed. The research therefore considers the use of high grade and commercial grade secure PESEs to provide an appropriate countermeasure (depending upon the sensitivity of the information) and thus reduce the risk to an acceptable level.

A high grade security product contains technology that has been developed to rigorous development standards, subjected to a high assurance evaluation (ASD, 2014; Rae et al, 2006) and certified by a national security authority (e.g. the Australian Signals Directorate, ASD<sup>4</sup>). Such products are used to protect the confidentiality and integrity of highly classified data. A commercial grade security product is typically a product developed using industry best practices (Humphrey, 2000) and it may also have been evaluated using recognised security guidelines (Herrmann, 2002). The research considers a commercial grade secure PESE for teleworking (and to a lesser extent mobile working). Both the commercial grade and the high grade secure PESE were designed to counter the risks of

---

<sup>4</sup> ASD is the Australian Government authority for information security and also has the responsibility for the certification of security products for Australian Government use.

cyber-attack, data loss and forensic data discovery. Whilst the research focuses upon the security contribution of a secure PESE, the developmental, operational, functional and assurance aspects are also considered.

The research also considers the use scenario of the secure PEE artifacts as an execution environment that is installed onto a standard (non-secure) thumb drive, i.e. the secure PEE is considered as a capability separate from a secure PESE. Whilst this use scenario lacks the authentication, access controls and encryption capabilities of a secure PESE, it is shown that a secure PEE can provide a secure, low cost solution, computing environment where the local secure storage of sensitive information is not required.

### **1.9.5 Limitations and Constraints**

When this part-time PhD research commenced (in 2007) smartphones and tablets supporting the Apple iOS (iOS, 2014) and the Google Android (Android, 2014) operating systems had not been introduced. However, such tablets and smartphones are now widely used by remote workers, in particular mobile workers where these devices have often displaced the laptops previously used for mobile working (TTI, 2014). A secure PESE is not designed to work with a tablet or smartphone and the architecture of a secure PESE is incompatible with the iOS and Android based devices.

Both the high grade and commercial grade secure storage devices that form the hardware platforms for the secure PESEs are independently assured and certified. However, the software forming the secure PEEs is not assured. Lack of an assured secure PEE (developed by the research) could be considered to be a security weakness and may limit the use of a secure PEE for (some) high assurance applications. In this research the constructed secure PEE artifacts are positioned as research prototypes/demonstrators.

The secure PESE is not appropriate for all remote workers. A secure PESE provides an encapsulated and constrained execution environment that is ideally suited to transaction-oriented work rather than knowledge work. A knowledge worker<sup>5</sup> is likely to find that a secure PESE provides a computing environment that is too restrictive for his/her requirements, however the research showed that the commercial grade secure PESE is

---

<sup>5</sup> A knowledge worker can be defined as an individual who is highly educated and is involved in work that requires a high level of problem solving and information gathering and analysis.

particularly suited to remote workers performing transaction-oriented work, e.g. technical support personnel working in a virtual help centre.

Some information security threats to the secure PESE are considered out of scope, in particular the threat from hardware keyboard loggers and from any malware embedded in a PC's firmware.

Upon reflection, the researcher recognises that several published conference papers forming this thesis could have supported additional referencing. It should be noted that whilst some papers now appear to contain insufficient references, they were all double blind peer reviewed before acceptance into the respective conference. To improve the academic strength of each paper reference material used but not cited or new reference material that substantiates claims in the paper are identified in the pre and post paper discussions that precede and follow each paper.

#### **1.10 The Research Papers**

The ten papers that form this PhD with publication are ordered so as to reflect the research progression and to present a linked body of work rather than according to their chronological publication date. If the papers were presented in chronological order then the thesis could appear disjointed and inconsistent as the publication date is not always aligned with the conduct of the research due to the selected journal/conference taking many months from the date the paper was first submitted to the actual publication. Certain research was only published once its importance was understood, e.g. the work on security models was performed in 2008 but the Australian Government's emphasis in 2011 on increasing telework then provided the impetus to publish.

The papers presented in this thesis are exactly as published with no editing performed, and with references retained with the paper rather than collated with the thesis' citations. As each was prepared as an independent publication it was necessary to identify the background and define terms, requirements, and (where appropriate) use and operational scenarios. With the collation of the papers into this thesis, together with the description of unpublished research, there exists unfortunately a degree of repetition. The researcher has endeavoured to minimise repetitive material wherever possible so that it neither

distracts nor confuses the reader. An example of a repetitive term is the secure PESE definition which is defined in both this thesis and also in several papers, albeit using similar, but different vocabulary.

Each paper in this thesis is preceded by a preamble discussing the rationale for the paper, the research question it addresses, and an enumeration of the knowledge consumed. The paper is followed by a synopsis discussing:

- The key outcomes from the paper and any contribution to knowledge.
- The contemporary relevance of the publication, its linkage to and influence upon other papers in the thesis, whether a research question was addressed and the direction pursued subsequent to the completion of the paper.

Four papers were co-authored. Appendix 1 provides statements from the co-authors declaring that the researcher was responsible for a contribution greater than 50% of each respective paper. The ten double-blind peer-reviewed papers forming the basis of this thesis with publication are:

- **Paper 1:** James, P. (2007), Can SDV Technology be Utilised in a Smartphone to Prevent Forensic Analysis?, *5<sup>th</sup> Australian Digital Forensics Conference*, Perth, pp 164-178.
- **Paper 2:** Hannay, P. & James, P. (2007), Pocket SDV with SDGuardian: A Secure and Forensically Safe Portable Execution Environment, *5<sup>th</sup> Australian Digital Forensics Conference*, Perth, pp 154-163.
- **Paper 3:** James, P. (2011), Are Existing Security Models Suitable for Teleworking?, *9th Australian Information Security Management Conference*, Perth, pp 130-139.
- **Paper 4:** James, P. (2008), Secure Portable Execution Environments: A Review of Available Technologies, *6<sup>th</sup> Australian Information Security Management Conference*, Perth, pp 70-86.
- **Paper 5:** James, P. (2008), Preventing the Acquisition of Data from Virtual Machine based Secure Portable Execution Environments, *6<sup>th</sup> Australian Digital Forensics Conference*, Perth, pp 82-97.
- **Paper 6:** Griffiths, D & James, P. (2010), Fireguard – A Secure Browser with Reduced Forensic Footprint, *The Journal of Network Forensics*, Volume 2, Issue 2, pp 1-24.
- **Paper 7:** James, P. & Griffiths, D (2014), A Secure Portable Execution Environment to Support Teleworking, *The Journal of Information Management and Computer Security*, Volume 22, Issue 3, pp 309-330.

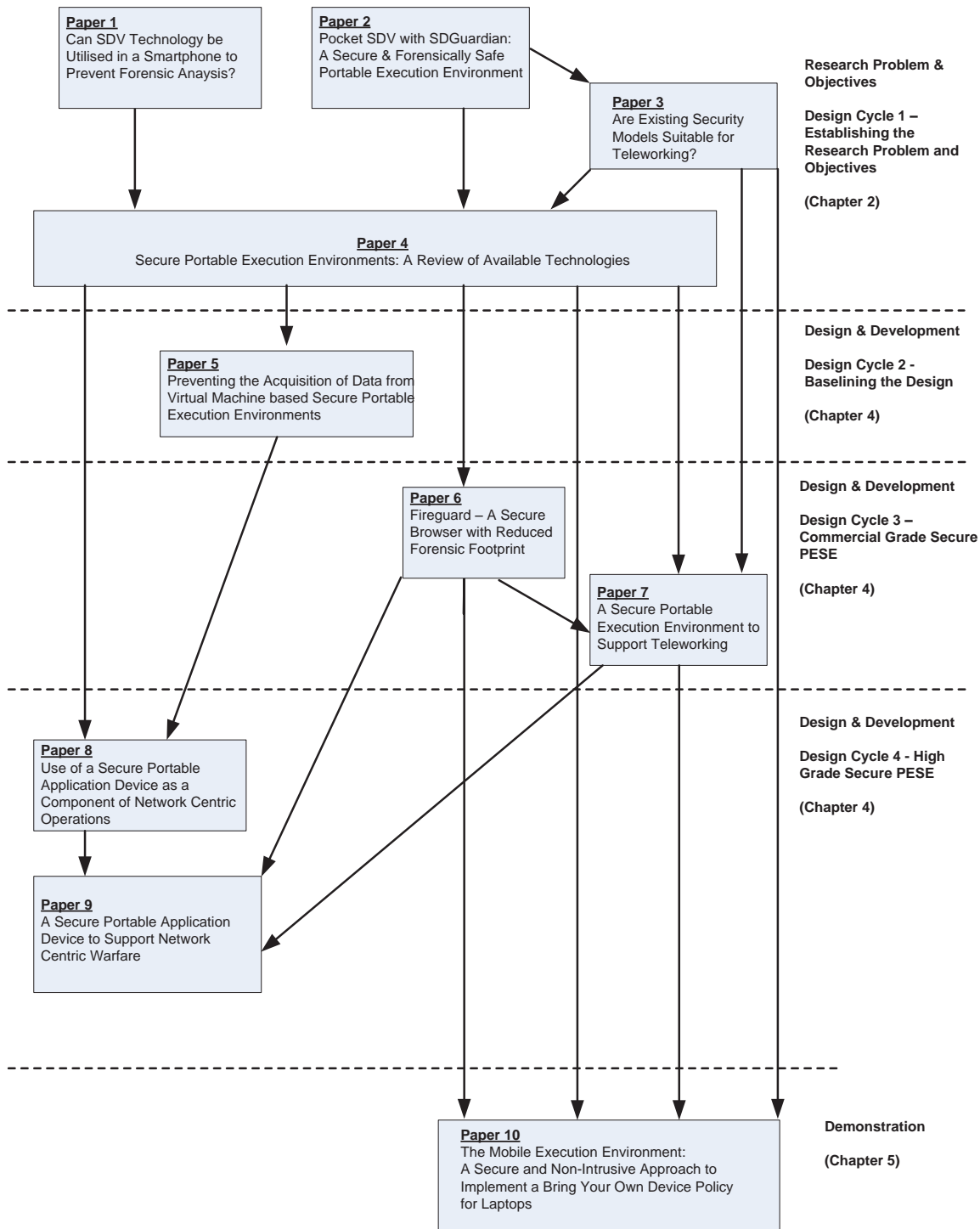
- **Paper 8:** James, P. (2009), Use of a Secure Portable Application Device as a Component of Network Centric Operations, *The Journal of Information Warfare*, Volume 8, Issue 3, pp 39-46.
- **Paper 9:** James, P (2011), A Secure Portable Application Device to Support Network Centric Warfare, *Military Communication and Information Systems Conference 2011*, Canberra, Available at: <http://www.milcis.com.au/milcis2011pdf/2.8a-paper2.pdf>
- **Paper 10:** James, P. & Griffiths, D (2012), The Mobile Execution Environment: A Secure and Non-Intrusive Approach to Implement a Bring Your Own Device Policy for Laptops, *10th Australian Information Security Management Conference*, Perth, pp 82-91.

As each paper is presented as published, multiple occurrences of figures and tables with the same figure/table number will appear. To clearly delineate a figure or table that is in the thesis (but not in a paper) the notation chapter number.figure is used, e.g. the third figure in Chapter 1 and the fourth table in Chapter 2 are labelled as 'Figure 1.3' and 'Table 2.4' respectively. The figures and tables in each paper are not listed in this thesis' table of contents. Several papers contain footnotes. As the thesis is a single Microsoft Word document any footnotes that appear in each paper have become subsumed into the whole document and therefore are sequentially numbered as they occur. To distinguish the papers the font used is 10pt Helvetica whereas the thesis uses 12pt Calibri.

Figure 1.1 presents a diagrammatic model of the relationship between the thesis' papers. The relationships modelled in Figure 1.1 show how the set of papers form a linked body of research that progresses the thesis argument that secure PESEs can be used to manage information security risks within the remote working environment. Chapter 2 discusses how four of the papers (1, 2, 3 and 4) contributed to the research problem identification whilst the five papers (5, 6, 7, 8 and 9) presented in Chapter 4 position the design or describe artifact design. Paper 10 forms part of the research demonstration in Chapter 5.

### **1.11 Evolution of Key Concepts and Terminology**

As the research progressed, papers were published and the researcher's expertise grew; the key concepts were refined and therefore the naming and semantics for key terms can vary between papers. In particular, the terminology and semantics of the following three areas evolved over time:



**Figure 1.1 – Relationship between Papers**

The papers are categorised using the DSRM process elements (i.e. problem and objectives, design and development, and demonstration) and are presented in chapters with the corresponding name. The arrows represent the influence and direction the preceding paper(s) had upon the respective papers to which the arrows point.



**Security risks:** When the literature review commenced, the risk of data loss (also referred to as data leakage in some papers) and forensic data discovery were considered important. With the establishment of the research problem and objectives the prevention of malicious software also became an important consideration. Some variations to the semantics of these aforementioned risks occur in certain papers. The risks of cyber-attack, data loss and forensic data discovery are nominated as the key risks a secure PESE should manage. However, throughout this thesis the risks to be countered by a secure PESE are often summarised as the information security risks.

**Secure PESE:** In Papers 4 and 5 a secure PESE is termed a “secure PEE” or “secure PEE device”. When the research commenced, a secure PEE or (secure PEE device) was considered an appropriate term to define the package consisting of the USB storage device and execution environment; however, as the research progressed the term secure PESE was considered more appropriate as the term encompassed the secure storage capability. In Papers 4 and 5 the terms “secure PEE OS” and “secure PEE VM” are also used for what this thesis defines as the secure PEE.

In Papers 8 and 9 the secure PESE is referred to as a secure Portable Application Device (secure PAD). Papers 8 and 9 consider the use of secure PESEs within a network centric organisation where the secure PESE (i.e. secure PAD) is used at the network’s remote nodes; these nodes may be fixed or mobile and each node could have one or more users present. The term secure PAD was selected to reflect the specific application set provided to each user to support network operations for the deployed work at a remote node. It is expected a user will have a secure PESE configured with a specific application set appropriate for the type of work to be performed at a remote network node.

**Trusted and Hardened:** In the earlier papers (e.g. Papers 4 and 5) the term “trusted OS” is used to refer to the operating system component of a secure PEE, where the trusted OS is defined as an independently evaluated and/or hardened operating system. The secure PEE artifacts design and development descriptions (presented in Papers 6 and 7) use the term “hardened” rather than “trusted” as the secure PEE component of a secure PESE consists of a hardened operating system and/or hardened applications which were not

subjected to independent evaluation. In Paper 8 a secure PEE is termed a “trusted application”.

It is recognised that the use of differing terminology in the papers can be confusing, but is a consequence of publishing research outcomes as the research was progressing. To reduce the likelihood of confusion the preamble preceding a paper identifies the use of any differing terminology.

## **1.12 Structure of Thesis**

In addition to this introductory chapter this thesis consists of a further six chapters. These six chapters are organised according to the process elements (activities) of the DSRM adopting a proposed publication schema for DSR (Gregor & Hevner, 2013). An overview of each chapter is given below.

Chapter 2 – Research Problem and Objectives: The chapter is presented in two parts. Part 1 documents the search for a doctoral research problem and an area of study. At the conclusion of Part 1 an initial secure PESE concept was identified. Part 2 documents a literature review of prior research into secure remote working and an analysis of ‘secure PESE like’ devices. Part 2 also includes a holistic assessment of the remote work information security issues. A research gap was identified and a research problem, objectives and questions defined. At the conclusion of Part 2 a finalised concept for secure PESEs had been defined. Four published papers are presented.

Chapter 3 – Research Design: The selection rationale for the DSR paradigm and the DSRM methodology is discussed including the consideration given to an alternative paradigm and methodology. The approach adopted to generate, demonstrate and evaluate secure PESEs is described. The method used to categorise the knowledge consumed and produced is discussed and the methodologies applied to determine a contribution to the area of study and define a nascent design theory are presented.

Chapter 4 - Design and Development: This chapter describes the design and development of two secure PEE artifacts (a hardened operating system and a hardened browser that would form part of an up-loadable secure PEE application set) and two secure portable

storage artifacts. These four artifacts are used to construct two secure PESEs (a high grade device and a commercial grade device). Five published papers are presented.

Chapter 5 – Demonstration – The demonstration process is presented and the results of applying the process to the commercial grade and high grade secure PESEs are described. One published paper contributes to the demonstration.

Chapter 6 – Evaluation and Discussion: This chapter presents the thesis discussion through a holistic critical appraisal and academic explanation of the research. The evaluation considers how the artifacts satisfy the research questions, objectives and research problem. The evaluation also considers how the thesis encapsulates a body of knowledge that makes a contribution to the area of study and asserts that the research has defined a nascent design theory for secure PESEs for use in remote work.

Chapter 7 – Conclusion: The success, originality, significance and limitations of the research are discussed. The chapter also reflects upon the suitability and application of the research methodology. The thesis concludes by considering the possible future directions for the research.

Appendix 1 – Statements from co-authors confirming the researcher contributed over fifty percent of each paper.

### **1.13 Summary**

In this chapter the background, rationale and motivation for the research have been presented. The research problem and questions have been enumerated and the research methodology and approach used to position and present the research summarised. The key concepts and terminology used in this thesis have been defined and the scope and limitations of the research discussed. The significance of the research and its knowledge contribution are postulated. The ten published papers are enumerated together with a diagrammatic model showing the relationship between the papers. Finally the content and structure of each chapter in the thesis is summarised.

## **2 Research Problem and Objectives**

### **2.1 Overview**

#### **2.1.1 Structure**

Chapter 2 consists of two parts. The first part presents the outcomes from investigations performed whilst searching for a research problem. The second part presents the thesis literature review where the concept of a secure PESE is defined. This chapter includes three published papers that document the results of investigating and prototyping ideas to improve the security of portable devices used by remote workers to store and process sensitive data. A fourth published paper investigates the applicability of existing security models to remote working (more specifically teleworking). These investigations made a knowledge contribution and assisted in identifying a research gap and problem.

The investigations considered how to enhance the security in portable devices (initially a smartphone and then an SDV portable storage device) before identifying the attributes of a telework security model and then introducing the initial concept of the secure PESE. The literature review considers the prior research into both secure remote working and existing 'secure PESE like' artifacts and products that could satisfy the aforementioned concept. The literature review also provides a comprehensive discussion on the remote work security issues before identifying the research gap and defining the problem and objectives.

#### **2.1.2 Establishing the Research Problem and Objectives - Design Cycle 1**

In addition to presenting the literature review, research problem and objectives this chapter also documents the first design cycle as the four papers presented in the chapter describe experimental design work. The outcomes of design cycle 1 are basic artifacts/ and prescriptive<sup>6</sup> knowledge, i.e. each outcome is either a design (Paper 1), model attributes (Paper 3) or prototype instantiation (Papers 2 and 4). The prescriptive knowledge produced in this first design cycle complements the descriptive and prescriptive knowledge identified in the literature review and the descriptive knowledge

---

<sup>6</sup> Descriptive, prescriptive and justificatory are knowledge classes used to demonstrate the epistemological basis for the research and classify knowledge utilised and generated. Chapter 3 provides definitions of the classes with a brief overview also given in Chapter 1.

provided through stakeholder discussions. Collectively this knowledge forms a knowledge baseline for the second design cycle. Figure 3.3 (Chapter 3) presents a conceptual model of the design cycles used in this doctoral research. Whilst it may appear unusual to include design activities in a chapter describing the literature review the prescriptive knowledge generated in design cycle 1 contributed to identifying the research gap, problem and questions. The researcher therefore believes that presenting design activities together with a literature review is an acceptable approach in a DSR project.

## **2.2 Part 1: Searching for a Research Gap**

### **2.2.1 Overview**

Part 1 describes investigations that were based upon the premise that portable computing and storage devices used by remote workers to store and process sensitive data required enhanced security. A further investigation seeks to determine if existing security models were applicable to remote working. The first investigation considered the introduction of security mechanisms into a smartphone (and is presented in Paper 1). The second investigation considered how to prevent the storage and recovery of data remnants from a host PC disk drive following data processed from a secure storage device (and is presented in Paper 2). The third investigation (presented in Paper 3) considered the applicability of existing security models to telework using a set of policy enforcement mechanisms and attributes identified for a telework security model. These investigations were performed while examining the literature and past research into secure remote working in the pursuit of identifying a research gap. The researcher's interest in remote working (James, 1991) and the experience/knowledge gained from these three investigations directed the research towards secure portable computing environments suitable for remote work. The three investigations provided prescriptive knowledge used to direct the research towards defining the secure PESE concept. In particular both the investigation into secure portable storage devices and up-loadable software, and the analysis of existing security models led to the introduction of the secure PESE concept.

### **2.2.2 Background to Papers 1, 2 and 3**

When the PhD commenced the use of smartphones to store and process highly sensitive data by remote workers, in particular mobile workers was identified as a potential area of

research. An investigation into integrating hardware based security mechanisms into smartphones was performed. The use of smartphones by mobile workers to store and process corporate data was starting to gather momentum (in 2007) although it was small compared to the prolific use today. However, as shown in Paper 1 there are many technical and business challenges to the concept of integrating security technology into an existing commercial off the shelf smartphone.

The challenges identified in attempting to improve smartphone security resulted in a transition in the doctoral research towards considering how secure portable storage devices (used by mobile workers) could be enhanced to prevent temporary data remaining on a host PC after data processing has occurred. An experimental artifact termed SDGuardian was developed and packaged with a secure portable disk drive (and is described in Paper 2). SDGuardian demonstrated different approaches to prevent the storage of temporary data on a PC disk drive.

As secure portable computing devices to support remote working was emerging as the likely area of doctoral research an investigation into the suitability of existing security models to support teleworking was conducted (and is described in Paper 3). It was considered that such a model could assist in the understanding of the design requirements for secure portable computing devices as a security model provides a policy enforcement and analysis tool to address a specific security problem (Liska, 2003; Jonsson, 2006). An outcome from the investigation was policy enforcement mechanisms and a set of attributes for a teleworking security model, which were used as a basis for introducing the secure PESE concept.

## **2.2.3 Integrating Security Technology into Commercially Available Smartphones**

### **2.2.3.1 Preamble**

In 2007 smartphones were emerging as an important business tool, therefore conducting research into enhancing their security appeared a possible PhD topic. An investigation commenced primarily as a result of stakeholder discussions and market monitoring (i.e. descriptive knowledge formed an important knowledge input). In particular, one of the key drivers for the research was a stakeholder requirement to protect highly sensitive data (stored on a smartphone) from unauthorised access including attempts to dismantle

the smartphone and acquire data from its memory. Another driver for the investigation was determining the viability of integrating security functionality into a commercially available smartphone as an add-on capability with the possibility of product development opportunities.

The investigation and its outcomes are presented in Paper 1. The paper considers how aspects of the SDV technology could be integrated into a commercially available smartphone (the Palm Treo). The SDV technology is designed to secure data on a PC hard disk drive (HDD) and directly integrating existing SDV technology components into a smartphone without substantial re-engineering was not feasible. Therefore six SDV security design principles were identified and used to perform an analysis resulting in a set of proposals for the inclusion of 'SDV like' technology into a commercially available smartphone.

#### **2.2.3.2 Prior Research and Knowledge**

When the research commenced smartphone security was at an early stage, particularly research into the construction of artifacts to secure data stored on a smartphone against forensic data discovery. The emergence of the malware threat to smartphones (Furnell, 2005) and the risk of data loss (Khokhar, 2006) had however, resulted in the development of anti-virus (Norton, 2007) and encryption (Pointsec, 2007) capabilities.

Security aware organisations using smartphones would implement available security capabilities (e.g. authentication, anti-virus and encryption) to protect stored data. Yet the emergence and low-cost availability of phone flasher devices to extract data from a smartphone (Al-Zarouni, 2007) could enable an unauthorised user to gain access to sensitive data despite the implemented security capabilities. Thus a potential gap was identified driven by the research question: 'How can SDV Technology be Utilised in a Smartphone to Prevent Forensic Analysis?'

The following sources provided the knowledge base for the investigation:

- Prescriptive knowledge: The SDV technology was the key knowledge driver of the investigation. The architectures of the Palm Treo (Treo, 2007), Palm OS (PalmDev, 2007) and SDIO interface (SDIO, 2007) provided the design knowledge to enable the

‘SDV smartphone’ proposals to be identified. The Palm documentation lacked definitive information on the Treo architecture and so information sourced from various web sites (identified in the paper) were used to develop an understanding of the Treo architecture.

- Descriptive knowledge: A stakeholder suggestion, a prior high level engineering proposal by a colleague of the researcher and a discussion with ECU colleagues provided descriptive knowledge. Market monitoring also contributed to the understanding of available technology and where the market was failing to address an identified problem.

### 2.2.3.3 Paper 1

**Paper 1** - James, P. (2007) **Can SDV Technology be Utilised in a Smartphone to Prevent Forensic Analysis?** *5th Australian Digital Forensics Conference*, Perth, pp 164-178.

#### Abstract

*Eliminating the opportunities to successfully acquire data from mobile devices is a critical security objective for certain organisations. In particular, Government agencies require assurance that classified data is secured against hostile forensic analysis. The Secure Systems Silicon Data Vault (SDV) is a hardware based data encryption and access control device that has been accredited by the Australian Government to secure classified information held on laptops and portable hard disk drives; hardware is recognised as a superior trusted platform to implement security mechanisms. The SDV's 128bit Advanced Encryption Standard (AES) cryptography, sophisticated key management & access controls and total disk encryption makes the SDV an extremely difficult device from which to acquire data and perform forensic analysis.*

*With the increasing functionality and storage capabilities of Smartphones strong security mechanisms are required by organisations that may hold sensitive data on these devices. Software based security applications exist for Smartphones that provide good security and severely impact the acquisition of data suitable for forensic analysis. If strong hardware based security can be integrated into a Smartphone, forensic analysis could be further constrained. This paper considers the feasibility of implementing the SDV technology into a Palm Treo. An overview of the SDV is given and six security design principles are enumerated. Implementation of the six design principles ensure the SDV provides strong security. The Treo architecture is reviewed and the concept of operation enumerated. The challenges with respect to implementing a Smartphone SDV that is conformant with the security design principles are discussed. Possible Smartphone SDV conceptual designs are presented. The concept of operation, implementation issues and conformance of each conceptual design to the SDV security design principles are discussed.*



## **Keywords**

Smartphone security, Silicon Data Vault, pre-boot authentication, encryption, PalmOS 5, Treo 650, mobile forensics.

## **Introduction**

The Secure Systems Silicon Data Vault (SDV) (Armstrong et al., 2004; SDVTech, 2006) is an award winning (iAward, 2006; SoAITI, 2005) hardware based data protection solution for mobile applications. The SDV provides protection for data at rest when the data is stored on Integrated Drive Electronics (IDE) Parallel Advanced Technology Attachment (PATA) and IDE Serial ATA (SATA) hard disk drives (HDD). The SDV technology has been implemented into a range of laptop and portable HDDs to provide amongst the strongest commercially available protection for data at rest. The SDV product range has been accredited by the Australian Government to protect classified information. A number of Secure Systems customers have asked if SDV technology could be implemented into Smartphones to provide strong security.

A Smartphone is essentially the merging of mobile phone and Personal Digital Assistant (PDA) technology into the one fully featured product. Typically, a Smartphone provides more features and functions than a standard mobile, for example Smartphones usually have a qwerty keyboard and a push email capability. Smartphones started to emerge in the late 1990s and have now become a key business communication tool for managers, executives and mobile workers. Smartphones use sophisticated operating systems to provide memory management, device control, application management & scheduling and data storage. There are five operating systems that dominate the Smartphone market; Symbian, Windows Mobile, Linux, Blackberry and PalmOS. There is little or no compatibility between the five operating systems and therefore consolidation is likely in the future.

Palm Inc (Palm 2007), traditionally a vendor of PDAs, produces a range of Smartphones branded the Treo range. Early models of the Treo range came with the PalmOS operating system; however more recent models now support the Windows Mobile operating system as an alternative to PalmOS. The particular Treo model considered in this paper is the Treo 650. The Treo 650 supports only a basic password protection mechanism as standard security. Numerous software security applications exist to provide stronger protection of data stored on the Treo; good examples include Pointsec Mobile (Pointsec, 2007) and Teallock (Teallock, 2007), both applications provide stronger authentication based access controls and encryption of data stored in internal and external flash memory.

With no standard Smartphone hardware architecture or dominant Smartphone operating system, designing a Smartphone SDV is a challenging proposition; the existing SDV design was able to rely upon established PC and HDD technology standards. Integrating SDV technology directly into a Smartphone's circuitry is considered infeasible (due to the close alliance required with a Smartphone manufacturer like Palm Inc) and it has therefore been assumed that a Smartphone

SDV would be an attachable device using an industry standard interface. A high level review performed by Secure Systems (Geddes 2004) on the possible integration of SDV technology into PDAs proposed using the Secure Digital (SD) card interface on a PDA to connect/insert a device containing SDV functionality. A number of Smartphones including the Palm Treo range have an SD card slot. This paper builds upon the idea of using the SD card interface by proposing a conceptual design for a Smartphone SDV device using the Secure Digital Input Output (SDIO) card.

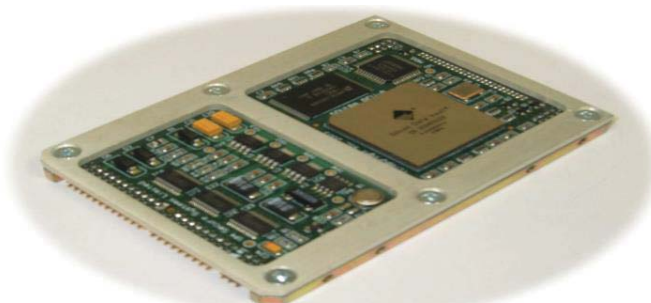
An SDIO card (SDIO 2007) has the same mechanical, electrical, power and signalling attributes of an SD card; an SDIO device can be inserted into an SD card slot and if the host device supports SDIO devices the SDIO device can be operated. Devices that support SDIO cards usually provide the single slot for both SD cards and SDIO cards. The SDIO card provides high-speed data I/O with low power consumption for mobile electronic devices. An SDIO device is able to interrupt the host (e.g. a Smartphone) when the SDIO device is inserted into an SD/SDIO card slot. While an SD card is a storage device, an SDIO card allows hardware accessories to be developed; examples include Wi-Fi and Bluetooth adapters, GPS receivers, TV tuners, cameras, RFID readers and fingerprint readers. The SDIO standard provides a suitable interface to enable an external SDV device to be attached to a Smartphone. The Palm Treo range supports SD cards and SDIO cards/devices.

## **An Overview of the SDV (Laptop SDV)**

The Laptop SDV is the core SDV unit that all other SDV models utilise; it also provides the most appropriate model to use for analysis in this paper. Only the attributes and features of the Laptop SDV necessary to support the discussion on the feasibility of a Smartphone SDV are presented.

### **Overview of Design**

The Laptop SDV (SDVTech 2006) is an alternative secure HDD for a laptop PC; it has the same form factor as a laptop 2.5" HDD. The Laptop SDV replaces the HDD in a laptop; it is connected to the host motherboards IDE controller. Figure 1 below presents a pictorial image of the Laptop SDV.



*Figure 1 – Picture of Laptop SDV*

The implementation of security mechanisms in hardware coupled with total independence of security mechanisms from the laptop's operating system ensures that successful direct attacks

and/or exploitation of operating system vulnerabilities are extremely difficult. The primary objective of the SDV is to provide strong security for data at rest<sup>7</sup>. The SDV is a cryptographic hardware device (James et al., 2004) that asserts total control over a HDD at system start-up and enforces correct user authentication before data on the HDD is accessible. Once successful authentication has been achieved the SDV allows the laptop's operating system to be loaded. The SDV supports differentiated access rights, i.e. user profiles can be defined with permissions to access different parts of the HDD. The SDV operates independently of the host computer's resources, providing real time encryption and decryption of all data transferred to and from the integral HDD; ensuring the data stored on the hard disk drive is cryptographically secured at rest, even if the SDV is physically removed from the laptop. A conceptual model of a Laptop SDV topology is given in Figure 2 below.

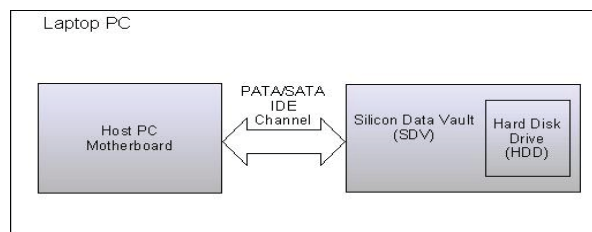


Figure 2 – Conceptual Model of Laptop SDV Topology

### Concept of Operation

At system power-up, a Laptop without an SDV installed will identify the storage devices available and load a Master Boot Record (MBR) from the main boot device; usually the primary HDD. The boot device in turn loads the operating system present on the storage device. While the operating system is running, the user typically has unrestricted access to all sections of the storage media. Conversely a laptop with an SDV inside operates as follows:

- At system power-up the laptop loads the Master Boot Record from the SDV. This in turn loads an Authentication Application (AA) stored in the SDV. While the AA is running, the user has no access to the SDV's integral HDD.
- The user is prompted to authenticate.
- The AA passes the information entered by the user to the SDV for authentication processing. Should the authentication process fail, the AA will prompt the user to re-authenticate. If the user fails to authenticate after a pre-defined number of attempts, the computer must be powered down and restarted to continue the user pre-boot authentication process.
- Once the user has successfully authenticated, the SDV decrypts the access keys and associated access rights stored in the authenticated user's profile. This information is used by the SDV to ensure protected hard disk data is accessed according to the profile for each user. The system continues the boot process and loads the OS from the SDV hard disk drive.

<sup>7</sup> Data at rest is a term that is used to refer to all data in computer storage while excluding data that is traversing a network or temporarily residing in computer memory.

- The SDV continues to operate independently of the host computer's resources, providing real time encryption and decryption of all data transferred to and from the SDV integral hard disk storage device until the computer is shut down.

### **SDV Security Design Principles**

To be considered a valid implementation of SDV technology any Smartphone SDV design would need to encompass the design characteristics that deliver strong security and hence reduce the ability to acquire data. Conformance to the following SDV security design principles will ensure opportunities to use forensic analysis techniques on acquired data are significantly reduced:

1. *Pre-boot authentication*: Performing authentication before the operating system has loaded ensures no hostile software or operating system vulnerabilities can be exploited to obtain authentication credentials.
2. *Full disk encryption*: With no data in plain text the opportunities to gain a 'starting point' to break the encryption are eliminated.
3. *Sector level encryption*: Encrypting at the lowest level of formatted storage reduces the possibility that pattern matching can be performed to break the encryption.
4. *Control of data channel*: Physically positioning the SDV between the PC motherboard and HDD ensures all writes are encrypted. Also access control to parts of the HDD can be enforced.
5. *Totally independent of PC Operating System*: The SDV behaves like a standard HDD and resides beneath the operating system so no attacks or vulnerabilities can be exploited.
6. *Security functionality implemented in hardware*: Exploiting and attacking hardware is extremely difficult.

These six SDV security design principles will be used in this paper as criteria to assess if the proposed Smartphone SDV conceptual design can provide the same level of security as the security provided by the Laptop and Portable SDVs.

### **An Overview of the Palm Treo 650 & PalmOS**

A Palm product and the Treo 650 in particular, was selected as the host for a (proposed) Smartphone SDV design due primarily to the information available, from both Palm Inc and the Internet. As with any (closed) proprietary product range, Palm does not publish extensive technical information. However, sufficient information was able to be sourced from a combination of Palm developer documentation (PalmDev, Guide 2007) and developer & hacker web sites (Treo Web Sites, 2007) that have appeared over the past few years dedicated to the Treo range.

This overview focuses on developing an understanding of the appropriate areas of the Treo 650 hardware and software architecture necessary to determine if SDV technology could be integrated into a Smartphone. In particular, the Treo storage architecture and memory management are outlined together with the PalmOS operating system capabilities.

## Overview of the Treo 650 Storage and Memory Management

The Treo, by Palm, is a family of compact Smartphones that integrates a mobile phone, wireless data applications such as messaging and web browsing, and an organiser. The Treo 650 (Treo 650, 2007) has been available since late 2004 and is managed and controlled by version 5.4.8 of the PalmOS operating system. Figure 3 below presents a pictorial image of the Palm Treo 650.



Figure 3 – Picture of Palm Treo 650<sup>8</sup>

The Treo 650 does not have an internal HDD. Prior to the 650, the Treo stored all application and data in volatile memory with the PalmOS operating system loaded from masked Read Only Memory (ROM); as a consequence power had to be supplied to the Treo all the time, if power was lost all the data and applications were lost. The Treo 650 has both non-volatile memory for storage of PalmOS, applications and data, and volatile memory for execution of PalmOS and applications. Two other storage devices are available on the Treo 650:

- Up to 2GB of non-volatile memory on an SD card.
- Variable size non-volatile memory available on the Subscriber Identity Module (SIM) card.

The Treo 650 has 32MB of non-volatile NAND flash memory (sometimes referred to as a DiskOnAChip) which is structured into two partitions. The first partition contains a boot loader and the compressed PalmOS operating system, known as the 'ROM' or 'compressed ROM', and occupies approximately 9MB. The second partition is available storage space for applications and data. The second partition is approximately 23MB and is structured into a 512 byte sector file system - the PalmOS Non-Volatile File System (NVFS). Figure 4 presents a memory map of the non-volatile memory.

---

<sup>8</sup> Image obtained from Palm Inc web site May 2007

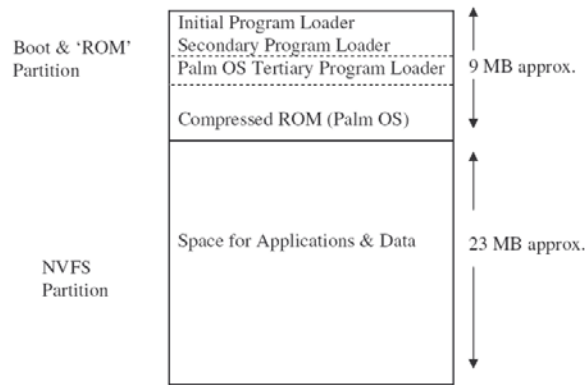


Figure 4 – Non-volatile NAND Flash Memory Map

The Treo 650 has 32MB of volatile SDRAM which is structured into three parts. Approximately 16MB of SDRAM is allocated to the executing PalmOS image, known as the decompressed ROM. A further 5MB is allocated for the PalmOS and application dynamic heap and temporary space. The remaining memory is used for the executing applications and data, known as the DBCache. The PalmOS image is protected from corruption from other executing applications by setting the area of SDRAM to Read-Only. Figure 5 presents a memory map of the volatile SDRAM memory.

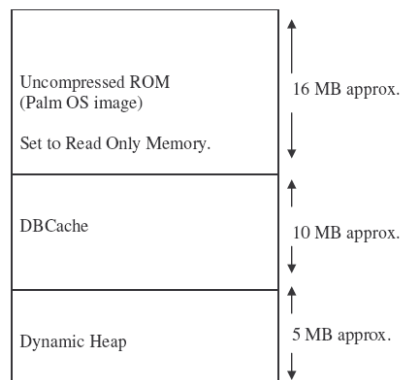


Figure 5 – Volatile SDRAM Memory Map

The Treo 650 will be automatically placed into sleep mode after a defined period to conserve power. Upon receiving a sleep notification PalmOS writes any changes to the applications/data partition. This does not, however, mean that the contents of the SDRAM are removed or PalmOS is stopped. Only a loss of power (depleted battery) or a soft or warm reset causes the SDRAM to be cleaned and a fresh reload of the PalmOS image (from the compressed ROM partition on the non-volatile NAND memory) to occur. A hard reset results in clearing of both the SDRAM and the NVFS partition of the non-volatile NAND memory.

An SD/SDIO card memory has any file system on the card mounted before data can be accessed. The Treo 650 officially supports SD/SDIO cards with up to 2GB of memory. Once the SD/SDIO file system is mounted, applications (and data) on the SD/SDIO card can be loaded into the DBCache

and executed<sup>9</sup>. The SIM card memory is accessible and available for storage via certain applications (e.g. SIMBook). SIM memory can vary in size; typically the size of a SIM card's memory is 64KB. It is assumed that an application reading or writing to the SIM card memory would process the data in the Treo's SDRAM.

Protecting data on any SIM card memory has been deemed beyond the scope of this paper.

### **Overview of PalmOS – File Systems, DBCache Management & SDIO Slot Management**

The Treo 650 comes loaded with PalmOS version 5.4.8; this is a sophisticated operating system providing comprehensive memory, device and file system management in addition to graphical input and output. An overview is given of the PalmOS file systems, SDRAM management and SDIO slot management capabilities, as these capabilities are relevant to supporting a Smartphone SDV design.

PalmOS 5.4.8 supports two file systems; NVFS for managing information stored in the non-volatile NVFS partition and Virtual File System (VFS) for managing information stored on SD/SDIO cards. SIM card memory is managed by applications that directly read and write to it and is not considered in this paper.

*NVFS:* PalmOS formats the NVFS partition into 512 byte sectors. When an application is invoked it is loaded into the DBCache in the SDRAM together with any data to be processed. Depending upon the application, as data is updated it is written back to the NVFS partition. Also certain PalmOS events (e.g. Treo going into sleep mode) will cause the DBCache to update the NVFS partition to ensure data is not lost. To ensure all available memory is utilised and avoid fragmentation in the NVFS partition, PalmOS will look for available space in NVFS sectors and allocate data to a sector from more than one DBCache record (essentially PalmOS terminology for a file) or downloaded application.

*VFS:* VFS is a unified interface that allows PalmOS to access different file systems on different media types, e.g. VFS allows a FAT 12 or FAT 16 file system on an SD card to be accessed using the same method/procedure call. There appears to be no relationship between VFS and NVFS. It is assumed that an application and data held on an SD/SDIO card is loaded into the SDRAM and that PalmOS performs updates to the SD/SDIO card as required in a similar way in which records are written from DBCache to the NVFS partition.

*DBCache Management:* As the DBCache is only 10MB the PalmOS cache manager has to manage this section of SDRAM efficiently to ensure an application can execute when invoked. Therefore PalmOS will write data back to its source location (NVFS partition or SD/SDIO card) upon an application's instruction or when space is required (typically once the DBCache exceeds 9MB).

---

<sup>9</sup> No documentation could be identified to confirm that an application on an SD card is loaded in to the DBCache to execute, but logically it would appear the viable approach as SD memory is block readable/writeable NAND memory where execute in place is not possible.



When applications start, stop, or use memory, fragmentation can occur so the cache manager continuously moves data into contiguous blocks to maximise available SDRAM.

**SDIO Slot Management:** PalmOS has a set of libraries to enable an application or PalmOS to control and read/write to an SDIO card. The PalmOS Expansion Manager detects insertion and removal of the SDIO card and mounts/unmounts any file systems. VFS manager provides the unified file system management. Both the expansion and VFS managers interface to the SDIO card through the SDIO Slot driver which manages power, interrupts, notification of events and essentially all other functionality specified in the SDIO Card Specification (SDIO 2007). Figure 6 presents a conceptual model of the interactions between the libraries and an application.

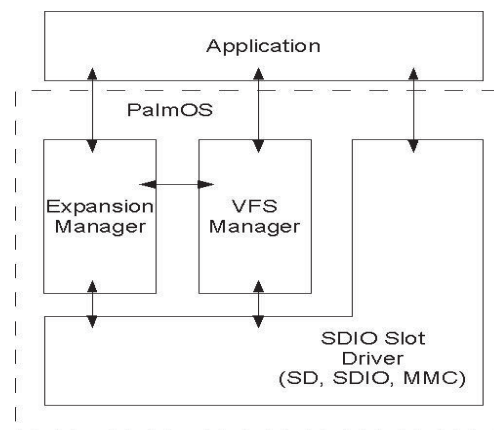


Figure 6 – Model of PalmOS libraries required to support SDIO card

An SDIO card/device can exist as a single function device or as device and storage combination. An SDIO device has its data and executable code located in the SDIO Code Storage area (CSA). The CSA is accessible as a mounted file system through the VFS manager. Once mounted, code in the CSA can be downloaded and 'autorun' in the Treo.

### Concept of Operation

Upon power being supplied for the first time or subsequent to a hard reset the following occurs:

1. The NVFS partition and SDRAM will be empty.
2. The Initial Program Loader (IPL) executes from the non-volatile memory. Whilst the IPL is located in the non-volatile block addressable NAND flash memory, a very small part of the memory allows the IPL to execute in place. The IPL performs some initialisation of the Treo 650 processor and hardware then the IPL loads the Secondary Program Loader (SPL) from non-volatile memory into the Treo's SDRAM.
3. Once loaded control is passed to the SPL, which initialises Treo devices (e.g. LCD and keyboard) and loads the Tertiary Program Loader (TPL) from non-volatile memory into SDRAM, passing control to the TPL once loaded.
4. The TPL decompresses the compressed PalmOS image held in the non-volatile memory and loads the decompressed PalmOS image into SDRAM passing control to it once loaded.



5. The NVFS partition will be available to install applications and data.
6. Any PalmOS function, or application loaded into the NVFS partition or on an SD/SDIO card, will be available for selection and execution.
7. When an application is selected, PalmOS loads the application (and its respective data) from its source location (either the NVFS or an SD/SDIO card) into the SDRAM DBCache and executes it.
8. PalmOS remains active until either power is lost or until a soft reset or system reset occurs.

Upon a soft reset/system reset the following will occur:

1. The NVFS partition will remain unchanged, but the SDRAM will be cleared.
2. Events 2 to 4 above are performed.
3. Events 6 to 8 above apply.

Upon entering and resuming from sleep mode:

1. No clearing of NVFS partition or SDRAM occurs.
2. In sleep mode certain devices are switched off (e.g. LCD screen) to reduce power consumption.

Upon SDIO card/device insertion:

1. Power is supplied to the device and it is initialised.
2. The CSA is mounted, if the device is a combo device the file system on the flash memory is also mounted.
3. Code in the CSA is downloaded and executed.

## **Challenges in Achieving the SDV Design Principles for a Smartphone SDV**

The overview of the memory capabilities & management, file systems and SDIO card management has highlighted that the Treo 650 with PalmOS 5.4.8 works differently to a laptop PC and its respective HDD. Designing a Smartphone SDV that meets the six security design principles will therefore be difficult and need to consider the following:

*Treo & PalmOS are Closed Technologies:* Whilst Palm and Access Co Ltd (a co-developer of PalmOS 5) do publish good documentation and APIs for PalmOS 5 (which is more significantly informative than documentation available from other proprietary Smartphone operating system vendors e.g. Microsoft and Symbian) detailed descriptions of PalmOS internals appear only to be available to strategic partners. No information appears to be published on the Treo hardware design. Lack of comprehensive hardware and operating system documentation presents a considerable challenge to implementing *the six security design principles* for a Smartphone SDV.

*Different Modus Operandi:* When a laptop is to be used it is turned on and the operating system is booted, work is performed and when finished the laptop operating system is shutdown. A Treo 650, however, is effectively always on; there is an on/off mode but this mode puts the Treo 650 to sleep to conserve power. Provided the battery has sufficient charge and a reset is not performed, the PalmOS image and executing applications (and data) remain active in the SDRAM even when the Treo is 'sleeping'. This different mode of operation (between Smartphone and PC) will make *the pre-boot authentication design principle* difficult to achieve.

*Different Storage Technologies:* A PC's HDD is separated from the PC motherboard and accessed through the IDE bus, hence the SDV is able to be located on the IDE bus between the PC and HDD. Whilst the internal bus structure of the Treo 650 is not known<sup>10</sup> it is highly likely that the NAND Flash and the SDRAM are closely coupled (i.e. physically connected circuitry). Interposing SDV technology (as it is currently conceived) to control the data channel, between the two memories via an SDIO card would be impossible. Therefore, it follows that fully encrypting the non-volatile NAND Flash (Disk On A Chip) memory is not possible as the boot start point could not be moved to an SDIO device. As a result performing *full disk encryption and controlling the data channel*, as per the SDV design, would not be possible for internal Treo storage.

*NVFS Partition is not Fully at Rest:* An important difference between the Treo 650 and a PC is that data in the NVFS partition (the equivalent of an internal HDD in a Treo) can never be considered to be at rest. As outlined above, PalmOS optimises storage by moving data and filling partially filled sectors in the NVFS partition. This method of storage optimisation may potentially make *sector level encryption* difficult to achieve, e.g. if a sector is encrypted by a Smartphone SDV (assuming it is possible to implement some form of internal sector level encryption beneath PalmOS) following a write request to the NVFS partition and then subsequently the PalmOS NVFS manager performs storage optimisation and changes the contents of the sector, then when the sector is re-read it will not decrypt correctly due to the changed contents of the sector.

*PalmOS & Storage Are Highly Integrated:* PalmOS provides a rich set of functionality to manage memory, file systems and devices in a compact and efficient package. Developing a Smartphone SDV that is totally independent of the operating system and implementing security functionality in hardware would require a large amount of functionality to be built to emulate some of the capabilities of expansion card manager, SDIO slot manager, VFS manager and NVFS manager.

## **Possible Smartphone SDV Design Options**

### **Packaging a Smartphone SDV as an SDIO Device**

Implementing a Smartphone SDV as an SDIO card/device provides a logical way of retrofitting SDV technology into a Treo 650. It is envisaged that a Smartphone SDV would be packaged into a

---

<sup>10</sup> No detailed documentation could be located on hardware design and schematics of the Treo 650.

“block” on the end of an SDIO card which protrudes out of the top of Treo 650 SDIO slot. Figure 7 presents a possible example of how a Smartphone SDV may be packaged.



*Figure 7 – Possible SDIO Smartphone SDV Packaging<sup>11</sup>*

On a Treo 650 the SDIO slot is located on the top of the phone (see Figure 8). It is envisaged that the SDIO Smartphone SDV protruding “block” would be approximately the same height and width as the Treo 650 external aerial (see figure 3 for frontal image of Treo 650 with external aerial). The size of the protruding “block” would, however, vary depending upon the amount of functionality and supporting circuitry required.



*Figure 8 – Top down view of SD/SDIO slot on Treo 650<sup>12</sup>*

### **Qualifications to Designs**

The proposed SDV Smartphone design options are conceptual; no qualification has been performed to confirm the:

- Treo 650 can supply sufficient power to the SDIO packaged Smartphone SDV circuitry.
- Required Integrated Circuits (ICs) and supporting circuitry can be packaged into an acceptable size SDIO “block”.
- Cost to build. Neither the development nor manufacturing costs have been estimated to qualify if any of the options are commercially feasible.

<sup>11</sup> Image obtained from SD Worldwide web site May 2007

<sup>12</sup> Image obtained from Palm Inc web site May 2007

- Market demand. No detailed market research has been performed to ascertain if a viable market exists for a Smartphone SDV. A few existing customers indicating interest would not be sufficient to commence development.
- Host Smartphones. The Treo 650 with PalmOS was selected for this research because it is a tried and tested product with good documentation available. However, if a Smartphone SDV was to proceed it would need to be a product that could work with the broadest range of Smartphones and operating systems.

### **Infeasible Functionality**

A number of challenges have been identified with respect to designing a Smartphone SDV that is conformant with the SDV security design principles. Developing functionality for a Smartphone SDV for a Treo 650 with PalmOS 5 would appear to be infeasible for the following areas:

- Hardware based encryption of the NVFS partition
- Sector level encryption of the NVFS partition
- Control of the data between SDRAM and the NVFS partition
- Full disk encryption of the internal “Disk on Chip” non-volatile NAND Flash storage.

### **Option 1 – A Full ‘SDV like’ Implementation**

This conceptual design is the most conformant to the six SDV security design principles. It would also be the most difficult to implement – it may, after further investigation, prove infeasible to implement. In this option the proposed core functionality will include:

- Pre-boot authentication.
- Access to data on external flash only possible after successful authentication.
- Hardware based encryption of the external flash memory.
- Sector level encryption of the external flash memory.
- Software based encryption of NVFS partition.

Pre-boot authentication would be achieved by replacing the standard SPL with a ‘secure SPL’ that interfaces with the inserted (SDIO) Smartphone SDV to download an authentication application. Upon successful authentication the SPL loads the standard TPL and the standard PalmOS boot process resumes. Access to data on the external flash memory and the CSA is blocked until successful authentication.

Hardware based, sector level encryption of the external flash memory would be performed on the fly by the crypto capabilities of the Smartphone SDV and would be separate and transparent to the Treo and PalmOS. Encryption key generation will be based on authentication credentials.

Software encryption of the NVFS partition would be achieved by downloading an application from the inserted Smartphone SDV (SDIO) CSA.

It is proposed a Smartphone SDV would mimic the SDV hardware architecture. Figure 9 presents a model of the SDV hardware architecture.

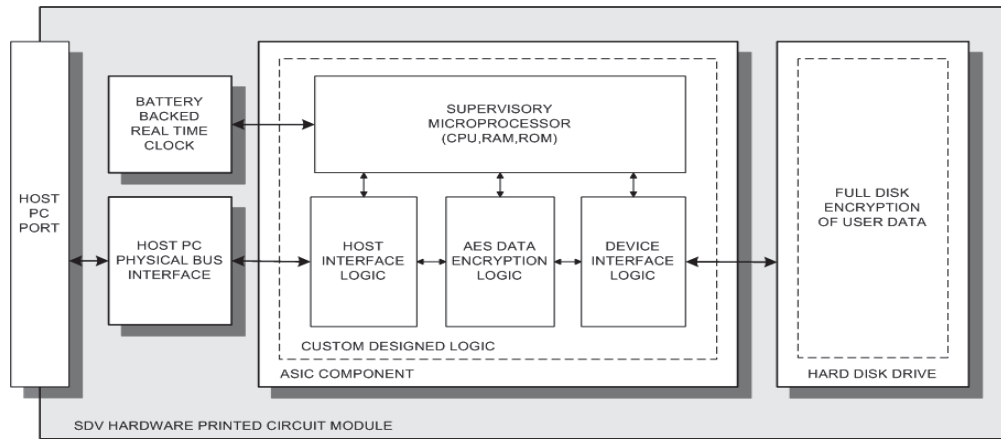


Figure 9 – Model of Key Components and Interfaces in SDV

In a Smartphone SDV the:

- Host PC physical bus interface will be an SDIO slot physical interface.
- Host interface logic will be the SDIO interface logic.
- Device interface logic will be the interface logic to flash memory.

The logic components could be packaged into a single application specific IC or a number of ICs each implementing one or more of the specialist functions.

The SDIO interface logic will work in both a pre-boot and post-boot mode. In pre-boot mode the 'secure SPL' will need to communicate with the Smartphone SDV through SDIO logic to enable the authentication application to be downloaded. In post-boot mode the SDIO logic interface will operate as a standard SDIO card. PalmOS will identify the SDIO device and mount the CSA and file system on the flash memory. When the CSA is mounted the SDIO capability to automatically download an application in the CSA will be used to load an NVFS encryption application; it is envisaged that this application will operate in a similar manner to existing software encryption applications (Teallock 2007) that are available, i.e. particular applications and data held in the NVFS partition are selected for encryption with actual encryption taking place once the Treo goes into sleep mode, with decryption occurring once the Treo is woken up.

Data will be written/read to/from the external SDIO flash memory using the VFS manager but as each sector is written/read to/from memory the Smartphone SDV will encrypt/decrypt each sector on the fly unbeknown to the Treo. The hardware and software crypto systems will adopt different key generation and management strategies to ensure that if the weaker software encryption is broken the stronger hardware encryption is not immediately vulnerable.

The downloaded encryption application will include an authentication function that will be activated when the Treo goes into sleep mode. This authentication function will communicate with Smartphone SDV to perform authentication. Only successfully authentication will allow the Treo to exit sleep mode.

#### *Concept of operation*

As the NVFS software based encryption will be weaker than the hardware based external flash memory security it would be expected that a user of a Treo will move as many applications and as much data as possible to the external flash memory in the Smartphone SDV.

Insert SDIO Smartphone SDV and immediately perform a soft reset - the following set of events will occur:

1. The IPL loads the Smartphone SDV 'secure SPL'.
2. If the 'secure SPL' does not detect a correctly inserted Smartphone SDV (N.B. for occasions when a soft reset is performed without Smartphone SDV being inserted) the secure SPL behaves like a normal SPL, otherwise the 'secure SPL' will supply power to the Smartphone SDV and load an authentication application from the Smartphone SDV, passing control to the authentication application.
3. The authentication application requests the authentication credentials from the user and passes them to the Smartphone SDV for authentication. If correct authentication occurs the TPL loads and control passes to the TPL; upon correct authentication the Smartphone will have correctly generated the encryption keys for both hardware and software based crypto systems.
4. The TPL decompresses and loads the PalmOS image into SDRAM and passes control to PalmOS
5. PalmOS will detect the Smartphone SDV and mount both the CSA and external flash memory file system. The NVFS encryption application will be downloaded from the CSA and commence execution.
6. Whenever data is written to the external flash memory it will be encrypted, likewise for selected NVFS based applications the respective data will be encrypted when written from SDRAM.

#### *Possible Implementation Issues*

Theoretically this design option can be implemented. A lot of information is available (Treo Web Sites, 2007) on how "customised ROMs" (customised boot loader and PalmOS) and Linux implementations have been installed into a Treo 650, therefore changing the boot loader to include a 'secure SPL' is entirely feasible. However, the following implementation questions arise:

Is performing a soft/system reset user friendly? On a Treo a soft reset requires the battery to be removed and then re-inserted, whilst a system reset requires the reset button positioned under the

battery cover to be pushed while pressing the up arrow on the keyboard. Neither reset option is particular elegant to perform.

Can a concise 'secure SPL' be developed that can detect, power and communicate with an SDIO device? It has been shown that SDIO device management requires comprehensive PalmOS libraries, implementing the necessary software to enable communication with an SDIO device and downloading an authentication application will be challenging.

Can a concise authentication application be developed with the drivers required to accept input from the keyboard and display output on the LCD? As authentication is performed pre-boot none of the PalmOS input/output drivers will be available.

Will performance of external flash based applications be acceptable? As 'SDV like' strong security can only be provided on external memory all, data and applications requiring protection should be located to the external flash memory. Loading from flash is noticeably slower than loading from the NVFS partition. Coupled with 'on-the-fly' encryption, performance may become a barrier to use.

Can the Smartphone SDV be removed while the software encryption application is resident in PalmOS SDRAM without corrupting the NVFS partition? Either the software encryption application will need to detect if the Smartphone SDV has been removed and then perform an orderly closure, or the encryption application is developed so that it can remain a resident application, independent of the Smartphone SDV, to provide on-going protection for applications and data held in the NVFS partition.

Will the Flash Translation Layer (FTL) prevent sector level encryption? The FTL allows NAND flash to be addressed as logical 512 byte sectors and ensures flash 'bad blocks' and 'worn out' blocks are not used. Figure 10 shows how FTL is positioned in the flash memory addressing scheme. The FTL manages the flash while providing a simple logical sector interface to the host system. It is possible that the FTL changes the location of data (FTL discussion 2007) as part of FTL management, i.e. as blocks become bad or worn data is moved; such movement of data may cause major problems for sector level encryption.

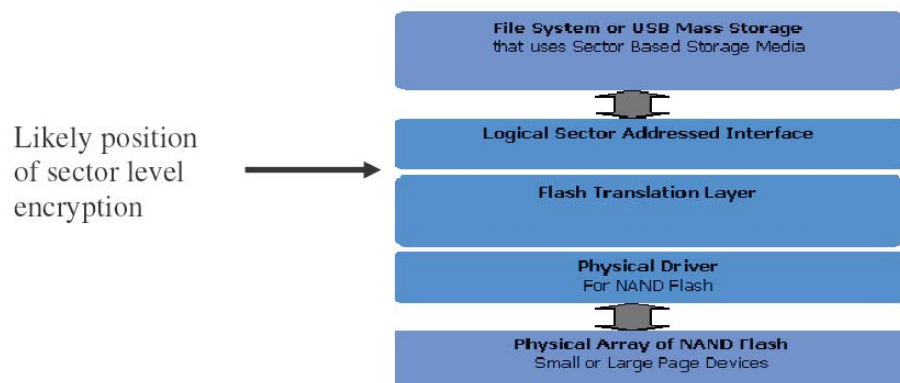


Figure 10 – Position of FTL in Flashing Memory Addressing Scheme

### *Conformance to SDV Design Principles*

- *Pre-boot authentication*: Theoretically met.
- *Full disk encryption*: Partially met, external flash memory will be fully encrypted but internal flash will not.
- *Sector level encryption*: Partially met, external flash memory will use sector level encryption but the NVFS partition will use file encryption.
- *Control of data channel*: Partially, SDV technology will be positioned between the Smartphone and external flash memory. Not possible for internal memory.
- *Totally independent of PC Operating System*: Partially, external based flash memory security will be independent of the operating system. However, the NVFS encryption application would utilise PalmOS capabilities.
- *Security functionality implemented in hardware*: Partially, the external flash memory encryption will be implemented in hardware; software encryption will encrypt data in internal memory.

### **Option 2 - Secure Authentication and Software Encryption**

In this design option the proposed functionality will include:

- Pre-boot authentication.
- Access to data on external flash only possible after successful authentication.
- Software based encryption of external flash memory located in Smartphone SDV.
- Software based encryption of NVFS partition.

Pre-boot authentication is implemented as described in option1 with access to data on the external flash memory and the CSA blocked until successful authentication.

Software encryption of the external flash memory and the NVFS partition would be achieved by downloading an application from the inserted Smartphone SDV (SDIO) CSA.

A simpler hardware architecture is required consisting of:

- SDIO interface logic.
- A simple (secure) microcontroller to process authentication credentials and perform key generation and management.

The PalmOS SDIO management capabilities will write encrypted data to the external flash memory via the encryption application running on the Treo. No complex encryption hardware is required.

The rationale for developing this option is to provide a secure separate storage device protected by strong pre-boot authentication. Whilst this option will not be as secure as option 1, it will be less complex to develop.



### *Concept of Operation*

Insert SDIO Smartphone SDV and immediately perform a soft reset - the following set of events will occur:

1. Events 1 to 4 in option 1 are performed.
2. PalmOS will detect the Smartphone SDV and mount both the CSA and external flash memory file system. The encryption application for both the internal (NVFS partition) and external flash memory will be downloaded from the CSA and commence execution.
3. Whenever data is written to the external flash memory it will be encrypted, likewise for selected NVFS based applications the respective data will be encrypted when written from SDRAM.

### *Possible Implementation Issues*

With the exception of pre-boot authentication, this option will be considerably less complex to implement. The option 1 useability and pre-boot authentication implementation issues exist, and due to software encryption of the external flash memory performance is likely to be worse than option 1.

To avoid potentially corrupting both the internal and external flash memory either the software encryption application will need to detect if the Smartphone SDV has been removed and then perform an orderly closure, or the encryption application is developed so that it can remain a resident application, independent of the Smartphone SDV, to provide on-going protection for applications and data held in the NVFS partition.

### *Conformance to SDV Design Principles*

- *Pre-boot authentication*: Theoretically met.
- *Full disk encryption*: Partially met, external flash memory would be fully encrypted, albeit using software encryption.
- *Sector level encryption*: No.
- *Control of data channel*: No.
- *Totally independent of PC Operating System*: Partially, pre-boot authentication will be performed before the operating system is loaded.
- *Security functionality implemented in hardware*: No.

### **Option 3 – Secure External Storage**

This design option is the least conformant to the SDV security design principles. It will be a simple SDIO device providing:

- Post-boot authentication.

- Access to data on external flash only possible after successful authentication.
- Software based encryption of external flash memory located in Smartphone SDV.
- Software based encryption of NVFS partition.

No soft/system reset will be required as the Smartphone SDV will be inserted into a booted Treo. The Smartphone will operate like a standard SDIO device, i.e. upon insertion into the SDIO slot the Smartphone SDV will be powered and notify PalmOS of its existence, the CSA in the Smartphone SDV will be mounted and the encryption application downloaded. In this option the Smartphone SDV relies upon the PalmOS SDIO management libraries.

This option offers comparatively little advantage over currently available software encryption applications and an SD card. The major difference is that access to the Smartphone SDV external flash memory is blocked until authentication is complete.

#### *Concept of Operation*

Insert Smartphone SDV into the SDIO slot of a full powered and running Treo 650 – the following events will occur:

1. Power is supplied to the Smartphone SDV and it is initialised.
2. The Smartphone SDV CSA is mounted together with the file system on the Smartphone SDV flash memory.
3. An authentication application is downloaded from the Smartphone SDV CSA.
4. The user will be prompted to enter authentication credentials.
5. If authentication is successful, the software encryption application in the CSA is downloaded and executed. No access to the external flash memory will be allowed until successful authentication.
6. Whenever data is written to the external flash memory it will be encrypted, likewise for selected NVFS based applications the respective data will be encrypted when written from SDRAM.

#### *Possible Implementation Issues*

There should be relatively few implementation issues. Standard SDIO hardware can be used, no specialist ICs or microcontroller will be required. The implementation issues with respect to the software encryption application identified in option 2 apply to this option.

#### *Conformance to SDV Design Principles*

- *Pre-boot authentication:* No.
- *Full disk encryption:* Partially, external flash memory would be fully encrypted, albeit using software encryption.

- *Sector level encryption: No.*
- *Control of data channel: No.*
- *Totally independent of PC Operating System: No.*
- *Security functionality implemented in hardware: No.*

## Conclusion

A comprehensive review of the hardware and software architecture of a sophisticated Smartphone has been performed to identify if SDV technology can be integrated into a Smartphone to make it more secure and restrict the opportunity to acquire data and perform forensic analysis. Three conceptual design options have been presented and assessed against SDV security design principles with varying degrees of compliance.

So, can SDV technology be utilised in a Smartphone to prevent forensic analysis? There is no clear yes or no answer. It has been shown not all of the SDV security features, as currently conceived, can be integrated into a Smartphone, e.g. control of the data channel and sector level encryption for internal storage. However, some SDV functions can be integrated into a Smartphone SDV that would strengthen security and virtually eliminate the opportunity to acquire meaningful data for forensic analysis.

If the Smartphone SDV is captured in an authenticated state (whilst in a Treo) then the opportunity exists to acquire sensitive data. If however, sensitive data and applications are held in the Smartphone SDV external flash memory and the Smartphone SDV is removed from the SDIO slot when it is not in use, acquiring sensitive data can be prevented.

Future work is planned to both consider other options for a Smartphone SDV and develop a proof of concept Smartphone SDV based on the approach proposed in this paper.

## References

- Armstrong A, Wynne M, O'Shea A 2004, Who has the keys to the vault? Protecting secrets on Laptops, IEEE Information Assurance Workshop 2004.
- FTL discussion 2007, Mobile Forensics class discussion, School of Computer and Information Sciences, Edith Cowan University, May 2007.
- Geddes 2004, Mike Geddes, PDA Security, Internal Discussion Paper, Secure Systems Limited, 2004.
- iAward 2006, Australian Information Industries Association, iAward Competition Security Category, URL <http://www.aiia.com.au/i-cms.isp?page=1346>
- James P, Wynne M 2004, Securing Data at Rest, 2nd Australian Information Security Conference, Edith Cowan University, Perth November 2004.
- Palm 2007, Palm Inc URL <http://www.palm.com>

PalmDev Guide 2007, Palm® Developer Guide, Palm OS Platform Software and Hardware Rev. F April 30, 2007

Pointsec 2007, Pointsec Mobile Technologies Inc, URL <http://www.pointsec.com>

SDIO 2007, SD Specifications Part E1, SDIO Simplified Specification, Version 2.0, 8/2/07, Technical Committee, SD Card Association.

SDVTech 2006, SDV Technical Overview, SSL-TD 0098, Version 1.4, 14/7/06

SoAITI 2005, Secrets of Australian IT Innovation Competition Security Category, URL [http://www.dcita.gov.au/\\_\\_data/assets/pdf\\_file/68179/2005\\_Secrets\\_of\\_IT\\_Innovation\\_competition\\_winners.pdf](http://www.dcita.gov.au/__data/assets/pdf_file/68179/2005_Secrets_of_IT_Innovation_competition_winners.pdf)

Teallock 2007, Teallock User Manual, Version 7.2, TealPoint Inc.

Treo 650 2007, Product description and specification of Palm Treo 650, URL <http://www.palm.com/au/products/smartphones/treo650/>, accessed May 2007.

Treo Web Sites 2007, URLs (accessed May 2007)

<http://www.grack.com/blog/articles/2006/02/27/treo-650-memory-management>

[http://www.shadowmite.com/wiki/index.php/The\\_Treo\\_650\\_Bootloader](http://www.shadowmite.com/wiki/index.php/The_Treo_650_Bootloader)

<http://www.grack.com/blog/articles/2006/02/07/the-lowdown-on-dbcache-and-rom-size>

<http://mytreo.net/archives/2005/07/living-with-nvs-on-your-treo-650.html>

<http://mytreo.net/treofaq/Treo650FileManagement>

<http://doc.trolltech.com/qtopia4.2/greenphone-integration-guide.html>

<http://hazelware.luggle.com/archive.html?2005.2>

#### 2.2.3.4 Synopsis

**Outcomes and Contribution to Knowledge:** The investigation identified the fundamentally different modus operandi between the data storage management used for smartphones and that used for PCs. The different modus operandi restricts the options for interposing hardware based PC style security mechanisms into a smartphone. A PC is based upon the x86 architecture (x86, 2013) and the Microsoft Windows operating system (Windows, 2013) which generally provides a consistent and standard platform for security research. When the investigation was performed no such standard existed for smartphones. The investigation showed it is difficult to conduct research where there is no standard smartphone architecture and where the technology is rapidly changing. When the paper was prepared (in 2007) different smartphone architectures supported five operating systems. By 2014 two completely different smartphone operating systems dominated the

market (IDC, 2013) neither of which was available in 2007. The challenges of different modus operandi, no standard architecture and the rapidly changing product market make it impractical to implement hardware based 'SDV like' add-on functionality to a smartphone.

More recently the issue of mobile phone security and lack of architecture standardisation has been recognised by the Trusted Computing Group (TCG)<sup>13</sup>. The TCG (TCG, 2013) has formed the Mobile Phone/Platform Work Group (MPWG) which is tasked with defining standards for the use of TCG technology in mobile phones/platforms. The security issues identified in Paper 1 are being considered by the MPWG. Recent publications by the MPWG include a trusted module specification (MTMSpec, 2010) and a use case document (MTMUse, 2011).

The key contribution to prescriptive knowledge was a conceptual design of a security module for a smartphone and an understanding of the strengths and limitations of the SDV technology with respect to its application within embedded systems. In addition, a high level but detailed description (derived using a range of Internet sources) of the hardware and software architecture and concept of operation of the Palm Treo 650 was prepared. This smartphone architecture had not previously been publically available in a concise, consistent and complete form. Although the product was superseded by late 2007, it can be asserted that the description provides a published prescriptive knowledge contribution to the understanding of a smartphone architecture and operation. This prescriptive knowledge has proved to be useful to other research where the proposed ideas and smartphone embedded architecture description were utilised in the following citation:

Isoaho, J., Virtanen, S., & Plosila, J. (2010). Current Challenges in Embedded Communication Systems. International Journal of Embedded and Real-Time Communication Systems (IJERTCS), Volume 1, Issue 1, pp1-21.

***Contemporary relevance, linkage with other papers and future direction:*** Smartphones have traditionally been designed with little consideration given to information security (Husted et al., 2011) and therefore it was an early input into the area of smartphone

---

<sup>13</sup> The Trusted Computing Group (TCG) is a standards organisation formed to develop vendor-neutral industry standards for interoperable trusted computing platforms.

security. The current and growing prolific use of smartphones has resulted in information security for such devices becoming a highly relevant topic (Cisco, 2014).

The paper's linkage with Paper 3 (refer to Figure 1.1) is through its contribution to the identification of improving security of portable devices used by remote workers. The challenges identified in the paper resulted in the research changing direction. An investigation into augmenting the capabilities of attachable PC secure storage devices become the focus. The prescriptive knowledge gained on the strengths and limitations of the SDV technology did assist in refining the research scope. The paper did propose ideas for future work, however as the research focus changed the proposed ideas were not investigated.

#### **2.2.4 Preventing Data Loss from Secure Portable Storage Devices**

##### **2.2.4.1 Preamble**

As a product development manager the researcher was the architect of the Pocket SDV (PocketSDV, 2006), a USB attachable portable secure storage device considered at the time of its release to contain a number of innovative features. The Pocket SDV utilises the SDV technology and provides a secure portable solution that prevents data loss if the device is lost or stolen. The product was commercialised in 2006. A customer that had procured the Pocket SDV to store highly sensitive data was concerned that the product would be connected to untrusted PCs in a mobile or deployed work environment. If the PC was used to access and process data held on the Pocket SDV then sensitive pieces of temporary data (or data remnants) could remain on the (untrusted) PC disk drive. The customer was concerned that these data remnants may be recoverable at a later date resulting in data loss. An investigation led to the development of a proof of concept toolset (known as SDGuardian) designed to prevent data remnants or their forensic recovery.

Paper 2 considers how the SDV security features and the prototype SDGuardian combine together to provide a portable solution that prevents data loss. The paper commences by presenting a detailed description of the Pocket SDV architecture, functionality, methods of use and concept of operation. The design of SDGuardian is described and three use scenarios are outlined that the toolset is required to address.

#### 2.2.4.2 Prior Research and Knowledge

The Pocket SDV was designed to address the market need to prevent data loss if the product was lost or stolen. SDGuardian was prototyped to address the research question: ‘What technology could be used to prevent the retention of sensitive temporary data (processed from an externally connected storage device) on the host PC disk drive?’

A review (in 2007) of prior literature on specifically preventing the retention of temporary data only identified a paper by Al-Zarouni describing how he used MojoPac (MojoPac, 2006), a sandboxing<sup>14</sup> tool, and a U3 flash drive (Sandisk, 2007), to provide a portable uploadable execution environment that left no forensic footprint (Al-Zarouni, 2006). Al-Zarouni used Mojopac to run malicious code which stole information from a PC but left no evidence that the code had been executed. Based on the Al-Zarouni approach the researcher (and the Paper 2 co-author) considered that sandboxing could be used to run a known and safe environment that left no sensitive data remnants.

The use of junction points (Rusinovich, 2006), to redirect where data was stored, was also considered an option. Junction points are a Microsoft file system capability to allow a directory to be defined that is an alias for an existing directory. The aim was to use junction points to change the storage location for temporary data from directories on the PC disk drive to directories on the Pocket SDV. The use of secure deletion was also considered a method to remove data where prevention was not possible.

In addition to the SDV technology, prescriptive knowledge in the form of sandboxing, junction points and secure deletion formed the knowledge baseline.

---

<sup>14</sup> Sandboxing is a technique that separates executing applications and is used to provide a secure and safe environment to execute untrusted or untested software. Sandboxing techniques include virtualisation and the use of an alternate restricted file system namespace (often termed a jail).

#### 2.2.4.3 Paper 2

**Paper 2** - Hannay, P. & James, P. (2007) **Pocket SDV with SDGuardian: A Secure and Forensically Safe Portable Execution Environment**, 5th Australian Digital Forensics Conference, Perth, pp 154-163.

#### Abstract

*Storage of sensitive and/or business critical data on portable USB attachable mass storage devices is a common practice. The ability to transport large volumes of data from the standard place of work and then access and process the data on an available PC at a different location provides both convenience and flexibility. However, use of such USB attachable mass storage devices presents two major security risks; the risk of loss of the portable storage device during transport and the risk of data remnants residing on a PC after accessing the data from the USB storage device. The latter risk is due to the way Windows and third party applications store temporary information on the host PC's hard disk. Even if every effort is made to delete temporary information it may be possible to recover this information by using forensic data recovery techniques such as header analysis and magnetic force microscopy.*

*The Pocket SDV with SDGuardian provides an elegant solution to the aforementioned security risks. The Pocket SDV is a commercially available USB attachable secure hard disk drive. Features of the Pocket SDV include hardware based encryption, strong authentication, differentiated access rights and cryptographically separate partitioning capabilities. Only a user with the correct authentication credentials can gain access to data stored on the Pocket SDV, thus providing assurance if the Pocket SDV is lost. SDGuardian is a proof of concept toolkit that minimises the remnants left on a PC if it is used to process data stored on a Pocket SDV. Forensic examination of the PC, following processing of data held on a Pocket SDV with SDGuardian, should not reveal any remnants of protected data. In this paper an overview of the Pocket SDV is given and its functionality is enumerated. The motivation for SDGuardian is outlined before discussing the design, capabilities and limitations of the Pocket SDV with SDGuardian.*

#### Keywords

Secure Portable Storage, Forensically Safe Portable Execution Environment, Digital Forensics.

#### Introduction

The Pocket SDV is a secure portable USB attachable mass storage device. The Pocket SDV enforces correct user authentication before data on the integral hard disk drive (HDD) may be accessed. Once the user has been correctly authenticated, the SDV allows access to the partitions (drives/volumes) on the Pocket SDV integral HDD. The Pocket SDV provides cryptographically enforced access to data contained on the integral HDD according to a previously configured data access profile for each user. The Pocket SDV operates independently of the host PC's resources,



providing real time encryption and decryption of all data transferred to and from it. If the Pocket SDV is lost or stolen its owner can be assured that no one can gain access to the data due to strong authentication, nor use digital forensic tools to gain access to the data due to strong encryption.

When a Pocket SDV is connected to a PC (with its own internal HDD, operating system and applications) and sensitive data is accessed (from the Pocket SDV) then temporary copies of the data may be saved on the PC's internal HDD by the operating system and/or applications. For instance, data accessed using Microsoft Word from a file stored on the Pocket SDV may leave temporary files inside temporary folders on the PC's internal HDD. As a result these folders may contain sensitive/private data of which the user may not necessarily be aware. If the PC is used by other users it may be possible for those users to find data remnants (temporary copies of files created during the processing of sensitive data) may remain on the PC's internal HDD after the user has detached the Pocket SDV. If the PC does not use encryption technology to encrypt everything written to its internal HDD then it may be possible for digital forensic tools to find sensitive data if the HDD were to be obtained by an inappropriate source.

SDGuardian (Sensitive Data Guardian) is a proof of concept toolkit aimed at addressing the issue of accessing sensitive data from a USB mass storage device like the Pocket SDV in an untrusted environment, e.g. a PC, not owned by the user, is used to process sensitive data held on a Pocket SDV, then subsequently other people use the PC and are able to find remnants of sensitive data left in temporary files. A variety of technologies are employed in order to address the aforementioned issue. SDGuardian may be commercialised, depending upon market demand, and used with the Pocket SDV (or other portable products offered by Secure Systems).

## **An Overview of the Pocket SDV: Design, Methods of Use & Concept of Operation**

### **Overview of Design**

The Pocket SDV is one of a range of SDV products; the product range also includes the Laptop SDV, the SDV Duo and SDV Plus. The primary objective of the Pocket SDV is to provide strong security for data at rest<sup>15</sup>. The Pocket SDV is a cryptographic hardware device (James et al., 2004) that asserts total control over its integral HDD at start-up and enforces correct user authentication before data on the Pocket SDV is accessible.

The encryption processes utilised by the Pocket SDV are implemented in the hardware. The hardware implementation of cryptographic functions avoids many of the inherent insecurities of a software-based approach, for example the hardware based approach ensures that keys are not present within the PC RAM; in addition the hardware implementation results in security enforcement that is transparent to the user and not dependant on the resources of the host PC.

---

<sup>15</sup> Data at rest is a term that is used to refer to all data in computer storage while excluding data that is traversing a network or temporarily residing in computer memory.

Once successful authentication has been achieved the Pocket SDV allows access to data based on pre-defined access rights. The implementation of the Pocket SDV's security mechanisms in hardware coupled with independence from the PC's operating system ensures that successful direct attacks and/or exploitation of operating system vulnerabilities are minimised. Figure 1 provides a pictorial image of the Pocket SDV.



Figure 1: Image of Pocket SDV<sup>16</sup>

The Pocket SDV supports differentiated access rights, i.e. user profiles can be defined with permissions to access different parts of the integral HDD. The Pocket SDV operates independently of the host PC's resources, providing real time encryption and decryption of all data transferred to and from the integral HDD; ensuring the data stored on the hard disk drive is cryptographically secured at rest. A conceptual model of the Pocket SDV topology is given in Figure 2 below.

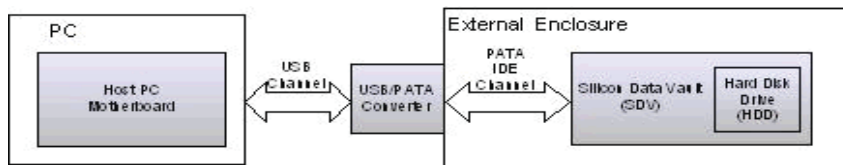


Figure 2 – Conceptual Model of Pocket SDV Topology

There are two modes of authentication supported by the Pocket SDV; pre-boot and post-boot authentication. When authenticating using the pre-boot method the host PC will boot off the attached Pocket SDV and the Authentication Application (AA) will be launched from the Pocket SDV's on-board flash memory. Once successful authentication has been performed the operating system on the PC's internal HDD is loaded. Authentication via the post-boot method requires that the Portable Authentication Application (PAA) is installed on the host PC. When the Pocket SDV is attached the user will be prompted for authentication details by PAA.

The authentication credentials of a Pocket SDV user are tied to a specific set of access rights for each partition on the Pocket SDV. These rights can be no access, read only or read/write. These user profiles could be used by different individuals or by the same individual, e.g. one user profile could be used for work and one for home. The key functionality and attributes of the Pocket SDV can be summarised as:

- *Pre-boot authentication:* The Pocket SDV achieves a high level of portability by performing authentication before the operating system has loaded. The only requirement is that the host

<sup>16</sup> Image of Pocket SDV made available by Secure Systems Limited.

PC provides the capability to allow a USB device to be the first boot device. Pre-boot authentication ensures no hostile software or operating system vulnerabilities can be exploited to obtain the Pocket SDV's authentication credentials.

- *Post-boot authentication:* A Pocket SDV can be authenticated to a PC running an operating system using the Portable Authentication Application (PAA).
- *Full disk encryption:* All data on the Pocket SDV is encrypted. With no data in plain text the opportunities to gain a 'starting point' to break the encryption are eliminated.
- *Sector level encryption:* Encrypting at the lowest level of formatted storage reduces the possibility that pattern matching can be performed to break the encryption.
- *Control of data channel:* Physically positioning the SDV technology between the PC USB controller and Pocket SDV integral HDD ensures all writes are encrypted. Also access control to parts of the HDD can be enforced.
- *Totally independent of PC Operating System:* The Pocket SDV behaves like a standard USB mass storage device and has no dependencies upon the PC operating system to which it is attached.
- *Security functionality implemented in hardware:* Implementing the SDV technology in an Integrated Circuit is recognised as a superior trusted platform; exploiting and attacking hardware is extremely difficult.
- *Multiple Partitions:* Up to 15 partitions can be defined for a Pocket SDV with each partition cryptographically separated from the other partitions by its own cryptographic key.
- *Differentiated Access Rights & User Profiles:* The Pocket SDV allows user profiles (roles) to be defined with different authentication credentials and access rights allowing different parts of the Pocket SDV integral HDD to be accessed according to the selected user profile.
- *Audit Log:* Security related events are written to an audit log only accessible by the Pocket SDV administrator role. This log can be used for forensic purposes.

## **Methods of Use**

The rich functionality of the Pocket SDV allows the device to be configured and used in a number of ways; three configurations are summarised below:

- *Highly Portable Secure Storage Device:* The ability to authenticate via pre-boot authentication results in a highly portable device that can be accessed via any PC capable of booting a USB device. The PAA provides the convenience of accessing data on the Pocket SDV on a fully booted system. Whilst the Pocket SDV provides Defence<sup>17</sup> level security for data at rest, it like all other USB mass storage devices cannot prevent data remnants remaining on a host PC's internal HDD.

---

<sup>17</sup> The SDV product range has successfully passed rigorous Australian, USA and International cryptographic and security evaluation standards.

- *Highly Portable Secure Storage Device with SDGuardian:* As this paper will show, the SDGuardian toolkit enables a user to attach a Pocket SDV to an untrusted or semi-trusted PC with the assurance that sensitive data remnants be minimised upon completion of data processing.
- *Highly Portable Secure Storage Device with USB Bootable Operating System:* A forensically safe alternative to using the tools and techniques of the SDGuardian is to use a Pocket SDV with a USB bootable operating system, e.g. a version of Linux or Windows PE. For a specific application the bootable operating system approach provides an ideal solution. A disadvantage of this approach is that USB bootable operating systems do not contain the functionality and look and feel of Microsoft Windows XP or Vista, resulting in a disincentive for many users. Current research at Secure Systems includes the development of a business toolkit as part of a bootable operating system with a Pocket SDV.

A stepwise summary of the Pocket SDV pre and post boot authentication is given below to enable a concept of operation to be acquired.

#### **Concept of Operation: Pre-boot Authentication**

The PC must be configured to boot from a USB device at power up with a Pocket SDV attached; operation then proceeds as follows:

- The PC loads a Master Boot Record from the Pocket SDV, which in turn loads an Authentication Application (AA) stored in the Pocket SDV flash memory. N.B. While the AA is running, the user has no access to the Pocket SDV's integral HDD.
- The user is prompted to authenticate.
- The AA passes the entered authentication credentials to the Pocket SDV for authentication. Should the authentication process fail, the AA will prompt the user to re-authenticate. If the user fails to authenticate after a pre-defined number of attempts the PC must be powered down and restarted to continue the user pre-boot authentication process.
- Once the user has successfully authenticated, the Pocket SDV decrypts the access keys and associated access rights stored in the authenticated user's profile. Information in the user profile is used by the Pocket SDV to ensure data on its integral HDD is accessed according to the access rights defined for the user.
- The user is then prompted to select one of the following:
  - Boot an operating system from the PC internal HDD.
  - Boot an operating system held on the Pocket PC.
  - Select to authenticate another SDV.
- If the user selects to boot an operating system from the PC's internal HDD a Master Boot Record for the operating system on the PC's internal HDD is loaded.
- The boot process continues and loads the operating system from the PC's HDD.

- The Pocket SDV continues to operate independently of the host PC's resources, providing real time encryption and decryption of all data transferred to and from the Pocket SDV integral HDD until either the Pocket SDV is detached from the PC USB port or the computer is shut down.

### **Concept of Operation: Post-boot Authentication**

The PC must have an operating system fully booted and the PAA installed and its underlying Windows service running; operation then proceeds as follows:

- When the Pocket SDV is attached to a USB port it is detected and a pop up authentication window presented to the user. N.B. The PAA can also be invoked to authenticate a Pocket SDV previously attached. While the PAA is running, the user has no access to the Pocket SDV's integral HDD.
- The user enters the authentication credentials and the PAA passes the entered authentication credentials to the Pocket SDV for authentication. Should the authentication process fail, the PAA will prompt the user to re-authenticate. If the user fails to authenticate after a pre-defined number of attempts the PAA must be restarted.
- Once the user has successfully authenticated, the Pocket SDV decrypts the access keys and associated access rights stored in the authenticated user's profile. Information in the user profile is used by the Pocket SDV to ensure data on its integral HDD is accessed according to the access rights defined for the user.
- If another Pocket SDV is detected the user is given the opportunity to authenticate the device.
- The Pocket SDV continues to operate independently of the host PC's resources, providing real time encryption and decryption of all data transferred to and from the Pocket SDV integral HDD until either the Pocket SDV is detached from the PC USB port or the computer is shut down.

The Pocket SDV makes use of proven encryption standards and strong authentication to provide strong hardware based security for data at rest. The 'set and forget' nature of the device with all encryption handled in hardware results in a secure solution that is transparent to the end user.

## **SDGuardian Development**

### **Motivation for Development**

The objective of the research was to ensure no data remnants remain on the internal HDD of a PC used to process sensitive data retrieved from a Pocket SDV. Initial investigations considered how a utility (known as SDCleaner) could remove data remnants, which had been written to the PC's internal HDD following the completion of data processing, i.e. a reactive approach was considered.

Research into file system structures identified that with a traditional file system (such as FAT32, NTFS and ext3) there is an area of the file system that provides a table of contents (the TOC). The TOC provides a list of all files located on the file system and where they are located logically (logical locations can be file paths such as C:\Program Files\). In addition to this logical file structure the

TOC provides a list of physical locations for each logical item, these physical locations can then be used to read and write data. The key issue for any SDCleaner utility is that when files are deleted the reference in the TOC is simply removed, with the physical data remaining elsewhere on the file system.

Another issue arises when a user launches a program that is used to access sensitive data on the Pocket SDV, this data would then be copied to a temporary location on the host PC's internal HDD. Upon exiting the application it is possible that this temporary data would then be deleted in an insecure fashion (i.e. physical data remains, logical construct removed). This series of events leads to a situation in which an SDCleaner utility would have no means of locating the physical data and is therefore unable to erase said data.

It was therefore decided to adopt a proactive approach and prevent the creation of data remnants on a PC's internal HDD as a result of retrieving and processing sensitive data from a Pocket SDV. The research project was redefined as the SDGuardian. The SDGuardian would act primarily as a preventative measure, with the aim of avoiding the situation where any sensitive information reached the host PC's internal HDD in the first place. Additional measures would also be employed to securely delete sensitive data in situations where prevention is not possible.

### Implementation of the SDGuardian Toolkit - Design, Capabilities and Limitations

*Implementation Scenarios:* The research considered a number of different scenarios where a Pocket SDV could be attached to a PC:

- *Scenario 1:* Data processing is to be performed on a "semi-trusted" PC where access to all installed applications is allowed and operation in Windows Administrator mode is permitted. Performance is also a requirement in this scenario, i.e. data processing needs to be performed at close to standard PC processing time.
- *Scenario 2:* Data processing is to be performed on an "untrusted" PC where the installed applications cannot be trusted. However, operation in Windows Administrator mode is permitted.
- *Scenario 3:* As per Scenario 2 but operation must be performed in Windows user mode, i.e. non administrator privileges are available.

To satisfy the requirements of the three scenarios SDGuardian was developed using Junction Points, Secure Deletion and Virtualisation to provide a toolkit for the proactive prevention of sensitive data from a Pocket SDV remaining on a PC's internal HDD. Table 1 shows the tools used to satisfy each scenario.

Scenario	Tool/Technique
1	<i>Junction points</i> are used to prevent specific temporary files from being written to the host PC's HDD. In addition to junction points, secure deletion is used to securely erase the Windows page file after use.
2	<i>Virtualisation technology running in privileged Administrator mode</i> is used to provide a virtualised environment where work can be performed without requiring the use of the PC's

Scenario	Tool/Technique
	installed applications.
3	<i>Virtualisation technology running in non-privileged user mode</i> is used to provide a virtualised environment where work can be performed without requiring the use of the PC's installed applications.

Table 1: Tools/Techniques used in SDGuardian to Meet the Requirements for Each Scenario

#### Junction Points - File System Manipulation

The NTFS file system supports 'junction points' which are similar to symbolic links under 'unix like' operating systems. These junction points allow for an empty folder on the file system to be mapped to a different physical location on the disk. The result of this is that two or more logical folders can reference the same physical data. Junction points can reference folders on a different volume or physical storage device.

An initial investigation into the currently available tools for the creation and manipulation of junction points was performed. The details of these tools can be found in the table below.

Software name	Description	License type	Source Availability
Junction(Russinovich, 2006)	A command line utility that allows for the manipulation of NTFS junction points	Proprietary	Available
Junction Link Magic (Rekenwonder, 2006)	A GUI based utility that allows for the manipulation of NTFS junction points	Proprietary	Not available

Table 2: A comparison of existing junction point manipulation software

Both Junction and Junction Link Magic were used to evaluate the premise that junction points could satisfy the requirement to remap operating system and application specific directories.

One of the shortcomings associated with the use of NTFS junction points is that when performing a delete operation on a junction point removal of the associated data from the disk occurs even if that physical data is referenced logically elsewhere in the file system. The window GUI however does not reflect this shortcoming and this deletion is unlikely to be noticed until the user next attempts to access this data. It is due to these factors that care was taken to ensure that junction points were not deleted with the standard tools provided by Windows; instead a custom utility was developed for this purpose.

SDGuardian used junction points to remap common temporary directories onto a partition of the Pocket SDV. This mapping was performed prior to the user accessing sensitive data located on the Pocket SDV. The result of the junction point would be that specific temporary data would never be written to the disk of the host PC, radically reducing the complexities associated with the standard methods of secure erasure.

SDGuardian removes these junction points and recreates the empty temporary directories after the user has finished working with any sensitive material. The temporary data stored on the Pocket SDV is then erased.



As a minimum junction points are used to ensure the security of the Windows temporary directories and the temporary internet directories present on a Windows system. A basic batch file capable of performing these actions in a somewhat limited fashion can be seen below:

```
mkdir %sdvTemp%\TMP
mkdir %sdvTemp%\TEMP
junction %TMP% %sdvTemp%\TMP\
junction %TEMP% %sdvTemp%\TMP\
```

The batch file has been tested with success, all data that would have been written to the Windows temporary directories was physically written to the SDV, the logical location of this data remained unchanged.

### *Secure Deletion*

Secure deletion tools allow for the forensically sound erasure of data from a hard disk or other storage device, this is achieved by overwriting the data in question several times with different sets of data. Typically the data being written will be all zeros, all ones or the output of a pseudo random number generator (Gutmann, 1996). This is often accomplished by using the Windows disk defragmentation API, which allows a logical file location to be resolved into a physical location (MSDN, 2007). Once the physical location of the data is known it is possible to overwrite this data as needed.

An initial investigation into some of the most common tools for secure erasure was performed. The details of these tools can be found in the table below.

Software name	Description	License type	Source availability
Sdelete (Russinovich, 1999)	Command line secure erasure utility	Proprietary	Available
Eraser (Tolvanen, 1997)	Graphical secure erasure utility	GPL	Available

*Table 3: A comparison of existing secure deletion software*

Both Sdelete and Eraser were used to determine the best strategy to adopt for the implementation of a secure deletion solution in the SDGuardian toolkit.

### *Virtualisation - Application Sandboxing*

Application sandboxing attempts to isolate running processes from performing modifications to the host system on which they are being executed. The type of isolation depends heavily on each sandboxing application's specific implementation. There are two main types of application sandboxes, the first attempts to create multiple isolated environments on a system, while the second attempts to limit or prevent specific processes from making changes to the host environment.

There are two main approaches to implementing sandboxing, the first of these is the use of a full virtualised environment. This environment has its own operating system that runs on top of the native operating system. The second approach to implementation involves the use of kernel hooks



to isolate a specific application or set of applications from accessing specific system resources (Gibson, 2006).

The use of application sandboxing utilities was investigated. The aim of such utilities is to create an environment which runs on the Pocket SDV to prevent any data from being written to the internal HDD of the host PC, the virtualisation software directs all writes to a specific partition on the Pocket SDV. Unfortunately virtualisation in itself cannot be used as a complete solution due to the nature of Windows virtual memory.

Windows makes use of a 'page file' which acts as virtual memory when adequate physical memory is unavailable. This 'page file' is located on the host PC's internal HDD (Mallery, 2006). The issue arises when a user accesses data stored on the Pocket SDV, the sandbox application can prevent all user level hard disk writes, however the Windows memory management system operates at a kernel level and as such it is not possible to prevent Windows from storing sensitive data in the Windows page file.

The use of a full virtualised environment such as those provided by VMWare or Qemu would allow a user to create a complete operating system environment that would run on top of the host machine's operating system. The advantage of this is that the majority of file system writes would be contained within the virtual machines disk image file, this image would be stored on the Pocket SDV. An advantage of this implementation is that the user would have the ability to install software such as a word document viewer within the disk image, thus negating the need for this software to be present on the host system.

A range of application sandboxing/virtualisation solutions can be found in the table below.

Software name	Description	License type	Source availability
Sandboxie (Tzur, 2006)	An application sandbox utility capable of redirecting file system writes to a specified location.	Proprietary	Not available
Vmware (Vmware, 2007)	A full virtualization application capable of emulating a host computer.	Proprietary	Not available
Parallels (Parallels, 2006)	A full virtualization application capable of emulating a host computer.	Proprietary	Not available
Mojopac (MojoPac, 2006)	A sandboxing utility capable of creating an isolated environment on a host computer. MojoPac is intended to be installed on a portable storage device.	Proprietary	Not available
Qemu (Bellard, 2006)	A full virtualization application capable of emulating a host computer completely in software, as such administrator rights are not needed on the host computer.	GPL	Available

*Table 4: A comparison of existing sandboxing / virtualisation software*

The full range of application sandboxing/virtualisation software specified in Table 4 were tested. Qemu was selected for SDGuardian due to its ability to execute in both Windows Administrator and User modes.

## Implementation Life Cycle

The proposed solution went through a phase of requirements specification and development over the period of several weeks. The SDGuardian was written primarily in the C# programming language. The following features were implemented:

- Junction Points
- Secure Deletion
- Virtualisation

A series of screenshots of the SDGuardian application itself are provided below:

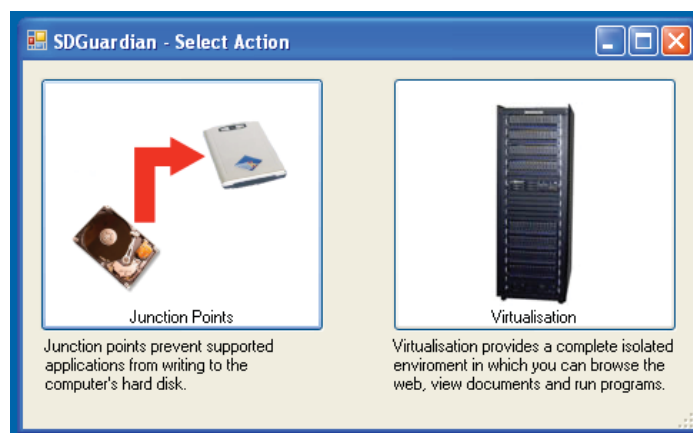


Figure 3: Main screen

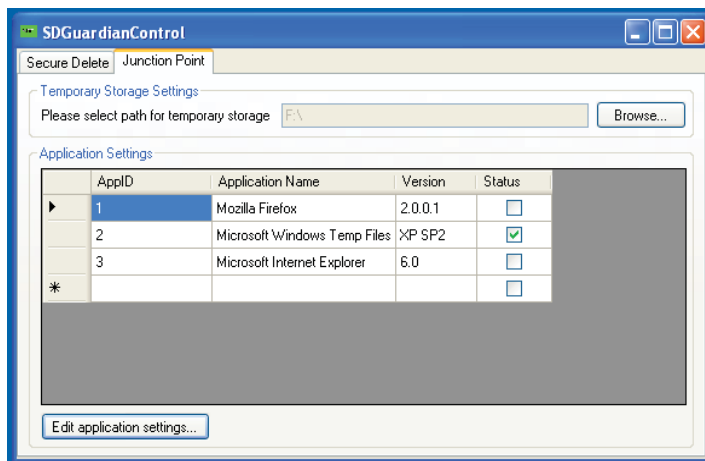


Figure 4: Junction points options dialog

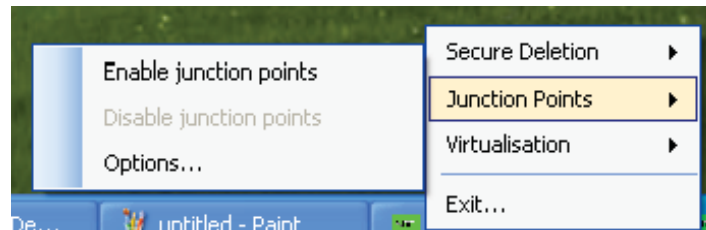


Figure 5: Junction points menu

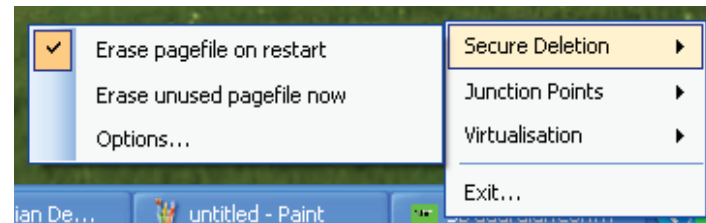


Figure 6: Secure deletion menu

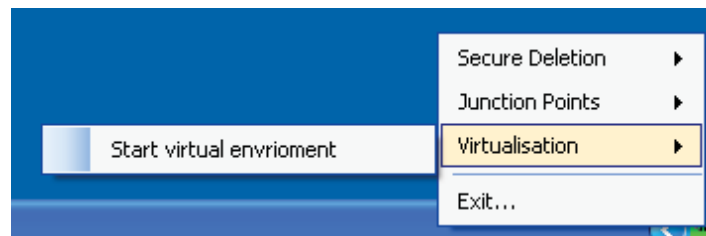


Figure 7: Virtualisation menu

## Conclusion & Future Development

The research and development of the SDGuardian was successfully achieved. A proof of concept implementing junction points, secure deletion and virtualisation was developed which meets the original goals of the project. Plans have been made for the continued development of the SDGuardian. These plans include improvements to the virtualisation system employed, additional focus on portability and an in depth forensic evaluation of the software.

## References

- Bellard, F. (2006). "QEmu." Retrieved January 11, 2007, from <http://fabrice.bellard.free.fr/qemu/about.html>.
- James, P. & Wynne, M 2004, Securing Data at Rest, 2nd Australian Information Security Conference, Edith Cowan University, Perth November 2004.
- Gibson, S. (2006, Oct 26, 2006). "Security Now - Transcript of Episode #63." Retrieved 11 January, 2007, from <http://www.grc.com/sn/SN-063.htm>.
- Gutmann, P. (1996). Secure Deletion of Data from Magnetic and Solid-State Memory. Sixth USENIX Security Symposium, San Jose, California.
- Mallery, J. R. (2001, December 6, 2006). "Secure File Deletion: Fact or Fiction?" Retrieved January 11, 2007, from [http://www.cybercrimelaw.org/documents\\_secure\\_delete.pdf](http://www.cybercrimelaw.org/documents_secure_delete.pdf).
- MojoPac. (2006). "What is MojoPac?" Retrieved January 11, 2007, from <http://www.mojopac.com/portal/content/what/>.

- MSDN. (2007). Defragmenting Files. Retrieved 8th of January, 2007, from <http://msdn2.microsoft.com/en-us/library/aa363911.aspx>
- Parallels. (2007). "Parallels Workstation." Retrieved January 11, 2007, from <http://www.parallels.com/en/products/workstation/>.
- Rekenwonder Software. (2007). Junction Link Magic, Rekenwonder Software.
- Russinovich, M. (1999). SDelete - Secure Delete, Systems Internals, Retrieved May 2007, from <http://www.systeminternals.com>
- Russinovich, M. (2006). Junction, Systems Internals, Retrieved May 2007, from <http://www.systeminternals.com>
- Tolvanen, S. (1997). Eraser, Heidi Computers Limited.
- Tzur, R. (2006, 14 December 2006). "Sandboxie." Retrieved January 11, 2007, from <http://www.sandboxie.com/>.
- VMware. (2007). "VMware: Virtualization, Virtual Machine & Virtual Server Consolidation." Retrieved January 11, 2007, from <http://www.vmware.com>.

#### 2.2.4.4 Synopsis

**Outcomes and Contribution to Knowledge:** The investigation showed that the Pocket SDV with the SDGuardian toolkit could be used to protect against data loss, either through actual loss of the Pocket SDV or through data remnants. SDGuardian demonstrated an application of sandboxing and junction points in data remnant prevention and secure deletion for temporary data removal where prevention was not possible.

Further post paper investigation found that the use of junction points to prevent data remnants was neither robust nor a holistic solution because:

- If the Pocket SDV is disconnected from the host PC when the junction point batch file is executing the host PC's Windows configuration may be corrupted.
- SDGuardian demonstrated how junction points could be used for operating system directories that hold temporary data, but to provide a holistic solution junction points would have to be established for every application's temporary folder. Whilst it is feasible to identify the application set that a remote worker with a Pocket SDV is expected to use, if the remote worker decides to execute an application not in the defined set (for which no junction points have been created) then there is a possibility the host PC may contain data remnants after data processing.

- Some applications provide little or no documentation on the location of any temporary data created therefore making it difficult to identify the directories holding any temporary data.

Whilst virtualisation appears to provide a more robust and complete solution, as the host PC's Windows operating system controls the execution of the virtual environment, there remains the possibility that the Window's pagefile (Pagefile, 2013) could contain sensitive data remnants. Also (post the paper's publication) the much slower performance of the up-loadable virtual machine relative to the performance of an operating system installed on a PC's internal disk drive was highlighted as a concern.

When completed (in 2007) the investigation resulted in an experimental artifact (SDGuardian). This artifact, constructed as a toolkit, can be considered a knowledge contribution as in itself it provided a new capability. However, to claim each individual tool is a knowledge contribution may be an overreach as:

- The use of the sandboxing concept (to enable the execution of a known and trusted up-loadable virtual environment) to ensure all temporary data created is retained within the virtualised environment is just an inverse application of its standard use.
- The application of junction points to alias directories (that store temporary data) to external directories on the Pocket SDV is just a different application of its standard use.
- The secure deletion capability is clearly not a contribution to knowledge as it uses existing tools and techniques as intended.

***Contemporary relevance, linkage with other papers and future direction:*** Preventing sensitive temporary data remaining on a PC's disk drive is an on-going security issue, albeit that there is growing awareness of the problem (Garfinkel & Shelat, 2003; DSD, 2011). This paper added to the body of research on data remnant prevention.

The use of SDV technology with virtualisation to provide a secure execution environment lead the researcher to consider how such technology could be used more broadly to provide a secure computing solution for remote workers. Paper 2 is linked to Paper 4 as it

provides prescriptive knowledge considered in the generation of the secure PESEs described in Paper 4.

## **2.2.5 Applicability of Existing Security Models**

### **2.2.5.1 Preamble**

A security model (also sometimes called a security policy model) provides a conceptual representation for a security policy (Jonsson, 2006). A security model can be used to educate and communicate security policy in addition to providing a high level model for security system design (Liska, 2003; Trivedi et al., 2009). The researcher's prior experience includes working as a software designer on a secure operating system development that implemented the Bell-LaPadula security model (Bell and LaPadula, 1976), developing and implementing formal specifications (James, 1987) and specifying and proving a formal security policy model for a secure diplomatic messaging network (Lindsay, 1998). The researcher therefore has a good appreciation of security models and is an advocate for their use. An investigation into existing security models to gauge their applicability to remote working was conducted because if an appropriate model was identified it could support the doctoral research in the design of secure PESEs for remote working.

The investigation presented in Paper 3 was performed in 2008, but a paper was not prepared and submitted to a conference until 2011. The Australian Government's emphasis on encouraging organisations to utilise telework as a work practice to improve national productivity (Telework, 2013) provided the motivation to take the findings of the initial investigation and place an emphasis upon teleworking in a conference paper.

### **2.2.5.2 Prior Research and Knowledge**

A literature analysis (in 2008) did not identify any prior research into the specific use of security models for teleworking. A search for existing security models identified four models that could be applicable to telework. To provide criteria for analysing the four models two use scenarios were defined based upon the researcher's prior experience. The research question directing the investigation was: *Are Existing Security Models Suitable for Teleworking?*

The prior knowledge used in the investigation can be categorised as:

- Formal justificatory knowledge: The four selected security models were Chinese Wall (Brewer and Nash, 1989), Clark-Wilson (Clark and Wilson, 1987), Eggshell (Bragg et al., 2004) and Onion (Bragg et al., 2004).
- Formal justificatory knowledge: Design and engineering theory for security systems (Anderson, 2008) allowing the identification of policy enforcement mechanisms. Security risk assessment theory (Landoll, 2005) providing context for risk identification.
- Descriptive knowledge: The researcher's prior knowledge on how telework is conducted from a remote location (James, 1991) to enable two use scenarios to be specified to enable threats, risks and model attributes to be identified.

#### 2.2.5.3 Paper 3

**Paper 3** - James, P. (2011), **Are Existing Security Models Suitable for Teleworking?**, 9th Australian Information Security Management Conference, Perth, pp 130-139.

#### Abstract

*The availability of high performance broadband services from the home will allow a growing number of organisations to offer teleworking as an employee work practice. Teleworking delivers cost savings, improved productivity and provides a recruitment policy to attract and retain personnel. Information security is one of the management considerations necessary before an effective organisational teleworking policy can be implemented. The teleworking computing environment presents a different set of security threats to those present in an office environment. Teleworking requires a security model to provide security policy enforcement to counter the set of security threats present in the teleworking computing environment.*

*This paper considers four existing security models and assesses each model's suitability to define security policy enforcement for telework. The approach taken is to identify the information security threats that exist in a teleworking environment and to categorise the threats based upon their impact upon confidentiality of data, system and data integrity, and availability of service in the teleworking environment. It is found that risks exist to the confidentiality, integrity and availability of information in a teleworking environment and therefore a security model is required that provides appropriate policy enforcement. A set of security policy enforcement mechanisms to counter the identified information security threats is proposed. Using an abstraction of the identified threats and the security policy enforcement mechanisms, a set of attributes for a security model for teleworking is proposed. Each of the four existing security models is assessed against this set of attributes to determine its suitability to specify policy enforcement for telework. Although the four existing models*

*were selected based upon their perceived suitability it is found that none provide the required policy enforcement for telework.*

## **Keywords**

Teleworking, secure teleworking, security model, security policy enforcement, information security.

## **Introduction**

There are a number of variations to the definition of what constitutes teleworking (Lister & Harnish, 2011; Access Economics, 2010a). In this paper teleworking is defined as the work practice involving remote computing conducted predominately from homes and occasionally from organised telecentres. Teleworking has a number of distinct advantages for both employee and employer. The employee benefits from reduced travelling time and travel costs, and possibly the opportunity to have flexibility for when the work is to be performed. The employer benefits from reduced office space and attracting talented personnel that would not otherwise be possible due to these individuals not being available to commute and/or work standard business hours. Teleworking can also deliver national economy benefits and improved productivity through reduced traffic congestion, reduced infrastructure maintenance and reduced carbon dioxide emissions (Access Economics, 2010b).

Australia presently lags internationally in levels of teleworking (DBCDE, 2011). With the increase in Internet bandwidth, teleworking has grown more rapidly in countries like the UK and USA (Lister & Harnish, 2011) where it is not uncommon for organisations to offer a telework option to staff. The UK in particular has a long history of teleworking. In the 1980s companies like Xansa (now part of Steria) and ICL (now part of Fujitsu) both established teleworking divisions to enable staff to perform software development from home. A good example of an international company utilising teleworking was Sun Microsystems (now part of Oracle). From the late 1990s Sun Microsystems identified the opportunity to accrue the benefits of teleworking for both the company and its employees and restructured its workforce through the 'Open Work' program (Computer World, 2008). At one stage over 20,000 employees were teleworking part-time or full-time. There are some notable examples of Australian organisations that have more recently implemented teleworking; these include iiNet (Australia's second largest Internet service provider) who has over 150 call centre staff working from home (Contact News, 2010), and the Queensland Government (Telecommuting, 2009) that allows public servants in some agencies to telework.

The recently published National Digital Economy Strategy (DBCDE, 2011), prepared by the Australian Government Department of Broadband, Communications and the Digital Economy, defines eight digital goals. One of these digital goals aims to increase the teleworking participation rate from 6% of the working population to 12% by 2020. Despite growth of teleworking in many countries there appears to be little research that specifically considers security models and their application to telework.. The Government lead emphasis on telework in Australia necessitates research is conducted into achieving secure teleworking. The research presented in this paper



forms part of an investigation into the methodology, tools and techniques required to achieve secure teleworking.

It is proposed in this paper that a teleworking security model is required and the attributes for such a security model are identified. This paper commences by outlining the rationale for a security model and identifying four existing models that could possibly support secure teleworking. The threats that can arise in a teleworking computing environment are described and categorised according to each threat's impact upon confidentiality, integrity and availability. Possible security policy enforcement mechanisms to counter the proposed threats are enumerated. Using the threats and security policy enforcement mechanisms, the attributes of a teleworking security model are defined. The paper concludes by reviewing the applicability of each of the four identified security models as being a suitable model for secure teleworking.

## **Security Models**

A security model provides a security design and analysis tool as it defines the basis for security policy enforcement in a system. A security model is a philosophy that directs the way an organisation approaches security (Liska, 2003). Numerous security models have been defined, with each model addressing one or more of the security policy requirements for confidentiality, integrity and availability of information. Some models are primarily focussed upon defining security policy that is appropriate in defence and government environments whilst others have a more business focussed approach to policy enforcement. Teleworking is a work practice that can be utilised by any organisation; defence, government, commercial or not-for-profit. However, as teleworking is unlikely to be used as a work practice involving the processing of classified defence and government information the existing security models considered in this paper are business oriented rather than defence/government oriented.

Four existing security models have been identified as being possibly suitable for policy enforcement in a teleworking environment. The four security models considered have been described (by their respective authors) at different levels of abstraction which has hindered meaningful comparative analysis, however the required attributes have been identified to enable each models suitability for teleworking to be assessed. Table 1 tabularises the security objectives of each model with respect to confidentiality, integrity and availability of information. A high level overview of each model follows together with rationale on why the security model is considered to potentially provide a policy enforcement philosophy for telework.

Security Model	Confidentiality	Integrity	Availability
Chinese Wall Model	X	X	
Clark-Wilson Model		X	X
Eggshell Model	X		X
Onion Model	X		X

Table 1: Objective of Security Model

Using the SANS definitions (SANS, 2011), confidentiality, integrity and availability are defined as:

- Confidentiality is the need to ensure that information is disclosed only to those who are authorised to view it.
- Integrity is the need to ensure that information has not been changed accidentally or deliberately, and that it is accurate and complete.
- Availability is the need to ensure that the information system is accessible to those who need to use it.

The *Chinese Wall* Model (Brewer and Nash, 1989) deals with both confidentiality and integrity. The objective of the model is to ensure the information of two different users is kept separate, regardless of the relationship and sensitivity/classification of the data, i.e. the model enables access to information to be prevented where a conflict of interest exists. The Chinese Wall model may be a possible enforcement model for teleworking due to its approach to user and data separation.

The *Clark-Wilson* Model (Clark and Wilson, 1987) focuses primarily on preserving the integrity of information, although a secondary attribute of the model is to support availability. The model was defined specifically for business applications. Data integrity is achieved by denying unauthorised modification. The model implements the concept of separation of duties (role based) to enforce information integrity, i.e. information access is controlled by a user's privilege(s) to execute application software that processes the information. The Clark-Wilson model is considered as a possible teleworking security model because of its role based business focus.

The *Eggshell* Model (Bragg et al, 2004) is focussed upon protecting information from unauthorised network access. The security policy for confidentiality is enforced in the model through a perimeter, i.e. access to information is controlled by rules that either allow or prevent entry through the perimeter. As teleworking is a network based activity the Eggshell model may provide a suitable security model.

The *Onion* Model (Bragg et al, 2004) defines layers of control to preserve the confidentiality of information. The model implements the concept of security in depth. Need to know controls are

enforced at each layer to control access to information. The layers of security enforced by the Onion model may be appropriate to enforce security for a network based activity like telework.

### **Information Security Threats in the Teleworking Computing Environment**

The lack of physical security in remote locations (compared to a corporate office) and the data processing actions of PC operating systems and applications can result in information security threats that are different to those present in an office based computing environment. The logical and physical security of teleworking PCs will vary between teleworking locations and are unlikely to meet the stringent security commonly in place at office locations. Weaker logical security may make the PC vulnerable to unauthorised access, network and malicious software attacks. Weaker physical security may make the PC vulnerable to unauthorised access, tampering, theft or damage.

The two main information security threats in a teleworking environment (Deloitte, 2011) are considered to be:

- **Breach of Data Confidentiality on the Internet:** Data travelling over the Internet may pass through a number of network nodes (e.g. nodes managed by an Internet service provider) before the data reaches its destination. Such nodes provide points of unauthorised access to data. If the data is not protected (e.g. through encryption) it is possible a breach of confidentiality could occur.
- **Breach of Data Confidentiality in Teleworking Environment:** The weaker physical and logical security controls, compared to a corporate environment, may result in authorised access to data.

Organisations that recognise these two threats to data confidentiality tend to establish a telework computing environment based upon the secure thin client concept. In a secure thin client configuration data processing occurs predominately on a corporate server, with the teleworker's PC providing a terminal interface to the server and all communication conducted securely over an encrypted virtual private network (VPN) connection. The following scenario provides an example of a secure thin client configuration:

*A teleworker will initiate a VPN connection over the Internet to a corporate server when commencing work. The teleworker may then use a virtual machine client or a remote desktop client to enable processing to be performed using a set of software applications installed on the corporate server. Very little data processed on the corporate server will be intentionally stored on the teleworker's PC, thus reducing the risk of data loss.*

However, in the above scenario the teleworker's PC operating system and the virtual machine client or the remote desktop client will store and retain temporary copies of data on the local PC, often unbeknown to the user, potentially resulting in the unintended storage of corporate data on the PC. Therefore this thin client approach to telework presents unforeseen information security threats that must be considered.

Alternatively, an organisation may allow a teleworker to process data locally but store information securely on a corporate server. The following scenario provides an example of local data processing:

*A teleworker will use a set of software applications installed on the PC and process data locally but ensure the data remains stored on a corporate server. A VPN connection to the corporate server is used to access/transfer data as required. The teleworker is expected to follow company policy and ensure all data is stored on the corporate server and no sensitive data is stored on the teleworker's PC.*

In this (local processing of data) scenario, even if the teleworker is diligent in ensuring data is not stored on the PC, the local processing of data will result in the operating system and software applications creating temporary copies.

As a result of the above two scenarios the following information security threats need to be considered:

- **Sensitive Data Remnants Remaining on a PC:** The PC operating system and software applications will store temporary data in the form of virtual memory and temporary files on the PC's hard disk drive. Much of this temporary data will remain on the drive after the teleworker has finished work and powered-off the PC. Such data can be readily retrieved through the use of freely available data retrieval/computer forensic tools; a concern if the PC is stolen or if it is disposed of with the hard disk drive still resident in the PC.
- **Introduction of Malicious Software:** As teleworking PCs reside outside an organisation a PC may be used by any number of people other than the teleworker (e.g. members of the teleworker's family) who could perform a range of activities that may compromise the PC. Such activities may lead to the PC becoming infected with malicious software which could exploit sensitive data processing and network transactions performed by the teleworker.
- **Use of Unsecured Portable Storage Media:** Teleworkers are likely to periodically commute to their employer's offices where they may also wish to transfer sensitive information onto or from portable storage media whilst having access to the organisation's secure network. Portable storage media may be lost or stolen in transit resulting in a possible breach of data confidentiality. Additionally, the portable storage media may become infected with malicious software when used outside the corporate environment. The malicious software could then be transferred to the secure network when the portable storage device is plugged into a networked PC in a corporate office.
- **System Integrity:** The integrity of the PC operating system and applications used for teleworking may be affected by accidental or inappropriate actions of the teleworker causing a denial of service and preventing work being performed.

With no specialist IT support on-hand to remediate the impact of the aforementioned threats the consequences can result in lost productivity through teleworkers not being able to work. This set of threats necessitates the identification of a teleworking security model that can be used as the basis for the design of secure teleworking systems.

## Categorisation of Threats

Traditionally a security model has been defined in terms of its ability to enforce security policy with respect to confidentiality, integrity and availability. To enable the suitability of the four identified security models to be assessed and the most appropriate security model to be identified (from the four or otherwise), the set of teleworking threats enumerated above are categorised (in Table 2) based upon their impact to the confidentiality, integrity and availability to information.

Threat	Confidentiality	Integrity	Availability
Breach of Data Confidentiality on the Internet	X		
Breach of Data Confidentiality in Teleworking Environment	X		
Sensitive Data Remnants Remaining on a PC	X		
Introduction of Malicious Software (Malware)	X	X	X
Use of Unsecured Portable Storage Media	X	X	X
System Integrity		X	X

Table 2: Threat Categorisation

The threat categorisation (in Table 2) shows that the set of threats in the teleworking environment can impact upon the confidentiality, integrity and availability of information. A security model is required that is capable of supporting a range of security policy enforcement mechanisms that will preserve both the confidentiality and integrity of information, and maintain availability to information.

## Policy Enforcement Mechanisms

To support the identification and definition of the attributes of an appropriate teleworking security model the potential security mechanisms that could be used for policy enforcement are presented in Table 3.

Threat	Policy Enforcement Mechanisms
Breach of Data Confidentiality on the Internet	Network encryption.
Breach of Data Confidentiality in Teleworking Environment	Authentication, access controls, data encryption.
Sensitive of Data Remnants Remaining on a PC	Data encryption, separation and protection of temporary data, access controls.
Introduction of Malicious Software (Malware)	Separation and protection of computing environment, access controls.
Use of Unsecured Portable Storage Media	Authentication, access controls, data encryption.
System Integrity	Separation and protection of computing environment, access controls.

Table 3: Possible Policy Enforcement Mechanisms

Existing security models are often presented using a formal or semi-formal notation for policy enforcing mechanisms. Usually the notation uses the subject/object concept introduced in the definition of the Bell-LaPadula Model (Bell & LaPadula, 1976). Using the concept of subject (e.g. teleworker, software application) and object (i.e. data file, information source) a definition for each of the policy enforcement mechanisms is given in Table 4 below.

Policy Enforcement Mechanism	Definition
Network encryption	Asymmetric cryptographic technique based upon each subject having a public/private key pair. An object to be transmitted is encrypted with the private key and the receiving subject decrypts using the sending subject's public key.
Authentication	Confirming the correctness of something the subject possess and/or something the subject knows and/or a unique attribute of the subject.
Access Control	Pre-defined permissions assigned to the subject that specify access rights to an object.
Data Encryption	Symmetric cryptographic technique based upon the application of the subject's encryption key to an object.
Separation and protection	Objects are protected within a partition where a subject can only access the partition if it has the correct access rights.

Table 4: Definition of Policy Enforcement Mechanisms

The subject and object definitions for policy enforcing mechanisms presented in Table 4 will enable comparative analysis of the selected four security models to be performed against the required attributes and policy enforcing mechanisms necessary for a teleworking model.

## Attributes of a Secure Teleworking Model

It is proposed that a teleworking security model is required to provide policy enforcement for the confidentiality, integrity and availability of information, as conceptually modelled in Figure 1. In this proposed security model, confidentiality will be the primary policy objective, with integrity and availability being the respective secondary and tertiary objectives. All of the identified security policy enforcement mechanisms are required to preserve the confidentiality. Subsets of the mechanisms are required to preserve integrity and maintain availability.

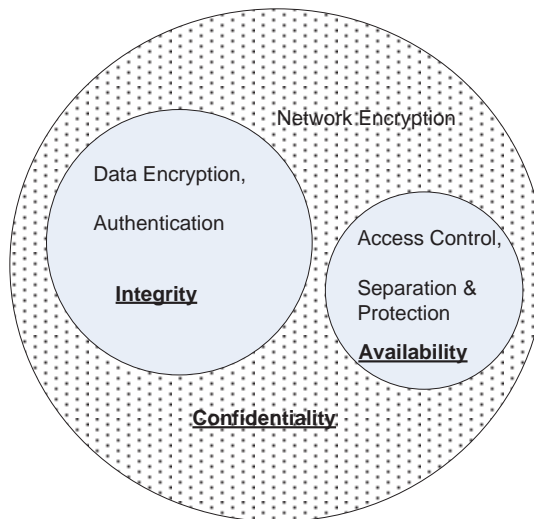


Figure 1: Conceptual Model of Teleworking Security Policy Enforcement

The identified threats and possible policy enforcement measures enable attributes for a teleworking security model to be defined. In summary a security model with the following attributes is required:

- Protect data transmitted over a network.
- Ensure only authorised access to the teleworker's computing environment is achieved.
- Protect the confidentiality of data processed by the teleworker.
- Prevent temporary data and data remnants (created through data processing) being accessed by an unauthorised user.
- Protect the integrity of the teleworker's computing environment and stored data from malicious software and/or user induced damage.
- Protect the confidentiality and integrity of any software and data stored on a portable storage device.
- Ensure the availability of the teleworker's computing environment.

## Application of an Existing Security Model to Teleworking

In this paper it has been hypothesised that a security model appropriate for teleworking is required to enforce policy for confidentiality, integrity and availability. Each of the four existing business oriented security models selected as possibly teleworking security models are considered using the defined teleworking security model attributes and policy enforcement mechanisms presented in this paper.

The Chinese-Wall Model		
Model Synopsis: The model supports policy enforcement for confidentiality and integrity to ensure users and data are separated and protected to ensure no conflict of interest occurs.		
Attribute of a Secure Teleworking Security Model	Possible Policy Enforcement Mechanism	Comments on suitability of the model for teleworking
Protect data transmitted over a network.	Network encryption.	Model does not consider network security.
Ensure only authorised access to the teleworker's computing environment is achieved.	Authentication, access controls, separation & protection.	The Chinese Wall model is data (object) centric, however it can be implied that a computing environment is protected.
Protect the confidentiality of data processed by the teleworker.	Authentication, access controls, data encryption.	The model supports this attribute.
Prevent temporary data and data remnants (created through data processing) being accessed by an unauthorised user.	Authentication, access controls, separation & protection.	The model supports the intent of the attribute and would enforce policy through the appropriate mechanisms.
Protect the integrity of the teleworker's computing environment and stored data from malicious software and/or user induced damage.	Access controls, separation & protection.	Whilst not a stated attribute of the model's policy enforcement definition the model specifies controls that should support the attribute.
Protect the confidentiality and integrity of any software and data stored on a portable storage device.	Authentication, access controls, data encryption, separation & protection.	The model considers data protection at a system level rather than at the portable device level, however its policy enforcement definition should support the attribute.
Ensure the availability of the teleworker's computing environment.	Access controls, separation & protection.	Availability is not a policy objective of the model.
Conclusion: The Chinese Wall model appears to support a number of the attributes of a teleworking security model. However, its key objective is to protect data where a conflict of interest exists, therefore designing a teleworking system using this model is unlikely to produce an appropriate implementation.		

Table 5: Suitability of Chinese Wall Model as a Security Model for Teleworking



<b>The Clark-Wilson Model</b>		
Model Synopsis: This model supports policy enforcement for the integrity of data through a role based approach.		
<b>Attribute of a Secure Teleworking Security Model</b>	<b>Possible Policy Enforcement Mechanism</b>	<b>Comments on suitability of the model for teleworking</b>
Protect data transmitted over a network.	Network encryption.	The model does not consider network security.
Ensure only authorised access to the teleworker's computing environment is achieved.	Authentication, access controls, separation & protection.	The model supports this attribute.
Protect the confidentiality of data processed by the teleworker.	Authentication, access controls, data encryption.	The model does not support confidentiality.
Prevent temporary data and data remnants (created through data processing) being accessed by an unauthorised user.	Authentication, access controls, separation & protection.	The model's integrity controls provide support for this attribute.
Protect the integrity of the teleworker's computing environment and stored data from malicious software and/or user induced damage.	Access controls, separation & protection.	The integrity controls of the model support this attribute.
Protect the confidentiality and integrity of any software and data stored on a portable storage device.	Authentication, access controls, data encryption, separation & protection.	The model does not support confidentiality, but does support the integrity aspects of the attribute.
Ensure the availability of the teleworker's computing environment.	Access controls, separation & protection.	The model's integrity controls provide support for this attribute.
Conclusion: The Clark-Wilson model supports many of the attributes required for a teleworking security model, however it does not support the enforcement of confidentiality. As protecting the confidentiality of information is important in the teleworking environment the Clark-Wilson model is not considered to be a suitable model.		

Table 6: Suitability of Clark-Wilson Model as a Security Model for Teleworking

The Eggshell Model		
Model Synopsis: The model enforces confidentiality through a perimeter to control access to data. The model is designed for network boundary protection.		
Attribute of a Secure Teleworking Security Model	Possible Policy Enforcement Mechanism	Comments on suitability of the model for teleworking
Protect data transmitted over a network.	Network encryption.	Although the model is network focussed it does not consider the confidentiality of data 'on the move'. It specifies policy to control access to network nodes.
Ensure only authorised access to the teleworker's computing environment is achieved.	Authentication, access controls, separation & protection.	The model provides boundary protection but if access is permitted no further protection is afforded.
Protect the confidentiality of data processed by the teleworker.	Authentication, access controls, data encryption.	The model provides boundary protection but if access is permitted no further protection is afforded.
Prevent temporary data and data remnants (created through data processing) being accessed by an unauthorised user.	Authentication, access controls, separation & protection.	The model provides boundary protection but if access is permitted no further protection is afforded.
Protect the integrity of the teleworker's computing environment and stored data from malicious software and/or user induced damage.	Access controls, separation & protection.	The model provides some boundary protection against malicious software but provides no protection against user induced damage.
Protect the confidentiality and integrity of any software and data stored on a portable storage device.	Authentication, access controls, data encryption, separation & protection.	The model does not support this attribute.
Ensure the availability of the teleworker's computing environment.	Access controls, separation & protection.	The model's boundary protection provides only limited support for this attribute.
Conclusion: The Eggshell model is a relatively simple model that specifies policy protection for a system against external unauthorised access. Such protection supports some, but not all of the teleworking confidentiality and availability attributes, but does not support the integrity attributes. This model is not considered appropriate for teleworking.		

Table 6: Suitability of Eggshell Model as a Security Model for Teleworking

The Onion Model		
Model Synopsis: The Onion model enforces confidentiality through multiple layers of security.		
Attribute of a Secure Teleworking Security Model	Possible Policy Enforcement Mechanism	Comments on suitability of the model for teleworking
Protect data transmitted over a network.	Network encryption.	Although the model is network focussed it does not consider the confidentiality of data 'on the move'.
Ensure only authorised access to the teleworker's computing environment is achieved.	Authentication, access controls, separation & protection.	The model supports this attribute.
Protect the confidentiality of data processed by the teleworker.	Authentication, access controls, data encryption.	The model supports this attribute.
Prevent temporary data and data remnants (created through data processing) being accessed by an unauthorised user.	Authentication, access controls, separation & protection.	The model supports this attribute.
Protect the integrity of the teleworker's computing environment and stored data from malicious software and/or user induced damage.	Access controls, separation & protection.	The model focuses primarily on enforcing confidentiality, although the layers of security may provide some integrity protections.
Protect the confidentiality and integrity of any software and data stored on a portable storage device.	Authentication, access controls, data encryption, separation & protection.	The model considers data protection at a system level rather than at the portable device level, however its policy enforcement definition should support the attribute.
Ensure the availability of the teleworker's computing environment.	Access controls, separation & protection.	The model provides limited support for this attribute.
Conclusion: This high level comparative analysis has shown that the Onion model may be suitable as a security model for teleworking; it certainly appears to be the 'best fit' of the four models considered. However, the layers of security are likely to interfere with the effectiveness of telework. Teleworking does necessitate a 'defence in depth' approach to security.		

Table 7: Suitability of Onion Model as a Security Model for Teleworking

## Conclusion

This paper has identified and categorised the set of information security threats that exist in a teleworking computing environment and proposed possible policy enforcement mechanisms to counter the threats. Using the categorised threats and policy enforcement mechanisms, attributes of a teleworking security model have been proposed. Each of the four selected security models were analysed using the attributes and policy enforcement mechanisms.

None of the four models were deemed suitable, although all four models displayed some of the attributes required for secure teleworking. The Onion model, in particular, has many of the attributes required of a teleworking security model. However, the multiple levels of security in the Onion model are likely to result in an unwieldy implementation of a security teleworking system.

A security model provides direction for the implementation of a secure information system. The author believes a specific security model for teleworking is required. The research presented in this paper provides the basis for the definition of a teleworking security model. Such a model should be presented in a formal or semi-formal notation to enable the correctness, consistency and completeness of the model to be determined. The author intends to continue the research to define a security model for teleworking.

## References

- Access Economics (2010a). Impacts of Teleworking under the NBN, prepared for the Department of Broadband, Communications and the Digital Economy, Access Economics, July 2010.
- Access Economics (2010b) Australian Business Expectations for the National Broadband Network, prepared for the Department of Broadband, Communications and the Digital Economy, Access Economics, November 2010.
- Bell, D.E, and LaPadula, L.J. (1976), Secure Computer Systems: Unified Exposition and Multics Interpretation, ESD-TR-75-306, MTR 2997 Rev. 1, The MITRE Corporation, March 1976.
- Bragg R., Phodes-Ousley M., et al (2004), Network Security: The Complete Reference, McGraw-Hill/Osborne, 2004.
- Brewer D., Nash M. (1989), The Chinese Wall Security Policy, pp.206, 1989 IEEE Symposium on Security and Privacy, 1989.
- Clark D. D., Wilson D. R. (1987), A Comparison of Commercial and Military Computer Security Policies, Proceedings of the 1987 Symposium on Security and Privacy, IEEE.
- ComputerWorld (2008), Sun's 'Open Work' program sheds light on telecommute savings, June 2008, URL: [http://www.computerworld.com/s/article/9105218/Sun\\_s\\_Open\\_Work\\_program\\_sheds\\_light\\_on\\_telecommute\\_savings](http://www.computerworld.com/s/article/9105218/Sun_s_Open_Work_program_sheds_light_on_telecommute_savings), accessed October 2011.
- Contact News (2010) iiNet hails teleworkers' performance, September 2010, URL: <http://www.contactcentres.net/CALLCENTRES/LIVE/me.get?site.sectionsshow&CALL1121>, accessed October 2011.
- Deloitte (2011), Next Generation Telework: A Literature Review for the Department of Broadband, Communications and the Digital Economy, a report prepared by Deloitte Access Economics, July 2011.
- Liska A. (2003), The Practice of Network Security: Deployment Strategies for Production Environments, Prentice Hall PTR, Pearson Education Inc., 2003.
- SANS, (2011), Glossary of Security Terms, URL: <http://www.sans.org/security-resources/glossary-of-terms>, Accessed October 2011.
- Telecommuting, (2009), Telecommuting Human Resources Policy, Queensland Government, October 2009, URL: [www.health.qld.gov.au/qhpolicy/docs/pol/qh-pol-242.pdf](http://www.health.qld.gov.au/qhpolicy/docs/pol/qh-pol-242.pdf), accessed October 2011.

#### **2.2.5.4 Synopsis**

**Outcomes and Contribution to Knowledge:** The investigation determined that no existing security model captured the policy requirements for teleworking. The key outcomes and contribution to (prescriptive) knowledge were the identification of a set of attributes for a telework/remote work security model. These attributes were formulated from analysing the threats and policy enforcement mechanisms identified in the paper.

**Contemporary relevance, linkage with other papers and future direction:** With the Australian Government encouraging public and private sector organisations to implement remote working policies to enhance productivity, this Paper 3 provides a relevant contribution. The paper has been cited in the following publications; confirming its contemporary relevance:

Ampomah, M., De Silva, Y., Li, H., Pahlisa, P., Yang, Q., & Zhang, Q. (2013), Information security strategy and teleworking (in) security, Department of Computing and Information Systems, University of Melbourne.

Jilani, U., Ahimmat, A., Raso, A., Thorpe, D., & Tran, M. (2013), Ready, Steady Telework – Information Security essentials for the teleworker, Department of Computing and Information Systems, University of Melbourne.

The threats, policy enforcement mechanisms and telework security model attributes identified in the paper became guidelines for defining the initial attributes of the secure PESE concept. The paper is linked directly to Papers 4, 7 and 10, although is not referenced by Paper 4 as Paper 3 has a later publication date.

The paper proposed that a security model for teleworking should be designed. The first step towards such a model design, in this doctoral research, is the definition of a secure PESE concept.

#### **2.2.6 Summary**

Part 1 has described the initial work into improving smartphone security, preventing data remnants occurring from processing data held on an external storage device and assessing

the suitability of existing security models for telework. This initial work involved a combination of literature analysis, design, prototyping, market monitoring and considering stakeholder observations. The work provided the prescriptive knowledge and the direction to consider if a gap existed for a secure portable computing environment (i.e. a secure PESE) to support remote working. As a result the doctoral area of study was identified as the use of secure PESEs to support remote working.

## **2.3 Part 2 – Literature Review**

### **2.3.1 The Approach**

The requirement of a literature review is to provide the foundation for the research through both the demonstration of a thorough analysis of the literature in the area of study and by identifying gaps and issues to enable the research problem to be determined; specifically the outcomes of a literature review are:

1. A thorough analysis of the area of study literature to evaluate prior theories, approaches and (in the case of DSR) artifacts (Cresswell, 2009; Gregor and Hevner, 2013).
2. Identification of existing knowledge (classified as descriptive, prescriptive and justificatory knowledge in this thesis) to be utilised in the research (Gregor and Hevner, 2013).
3. The identification of gaps and issues in prior knowledge (Cresswell, 2009); and
4. Development of expertise in the research area (Standing, 2008).

These outcomes enable the research problem to be identified. To achieve the above outcomes the literature review draws upon the justificatory, prescriptive and descriptive knowledge in the area of secure remote working and the use of 'secure PESE like' devices. The relevant prescriptive knowledge identified and/or generated in Part 1 coupled with additional descriptive knowledge in the form of stakeholder observations and the researcher's own experience and interest in remote working (James, 1991) is also utilised. In particular, the design of a toolset to prevent data remnants coupled with the identification of the attributes for a telework security model identified in Part 1 directed

the research towards developing both a secure PESE concept and artifacts that implement the concept.

The approach was to first conceive an initial concept for the secure PESE using the attributes specified for a telework security model as the starting point. To help shape the researcher's ideas four experimental secure PESEs were implemented using readily available technology and the initial concept to direct the construction. The experimental secure PESEs allowed the concept to be refined and expertise to be developed before conducting a literature analysis. Having introduced an initial concept a literature analysis was conducted to identify:

- The prior research that considers how to improve security for remote working;
- Existing and emerging 'secure PESE like' research artifacts and products; which are assessed using the security requirements and desired functionality identified for the secure PESE concept; and
- A holistic analysis of the security issues in the remote work location. For each issue the relevant threat(s) is defined together with a functional requirement(s) that provides a countermeasure(s). The set of requirements specifies the functionality that a secure PESE is expected to provide.

An analysis of both the functional requirements and the initial concept's attributes resulted in the establishment of a finalised secure PESE concept. Using the concept a gap was identified and a research problem, objectives and questions were defined. Figure 6.1 in Chapter 6 diagrammatically models the development of the finalised secure PESE concept together with other knowledge outcomes. This approach establishes the boundaries, structure and direction taken for the research presented in this thesis.

## **2.3.2 Introducing the Concept of the Secure PESE**

### **2.3.2.1 Preamble**

The impetus to develop a secure PESE and an underlying concept emerged as a result of:

- Stakeholder requirements for such a capability;

- The researcher's observation that the outcomes from the SDGuardian toolkit could be utilised to provide such a capability; and
- The realisation that the attributes of the telework security model attributes identified in Paper 3 could be used as the basis of a concept for secure portable computing.

The researcher therefore formulated an initial concept for a secure PESE. Using the threats, policy enforcement mechanisms and security model attributes identified in paper 3 as guidance, security requirements and desired functionality were defined that represent the attributes of an initial secure PESE concept - the initial concept attributes were refined and finalised upon completion of the literature review.

To test the viability of the concept the researcher constructed four experimental secure PESEs from available technology components. The use of junction points (to prevent data remnants) was not included as a technology component due to the issues identified in the Paper 2 synopsis. The research was directed towards both virtualisation and portable operating systems (i.e. live CDs) to provide an execution environment and a range of storage devices to provide the hardware platforms. Paper 4 presents the investigation into constructing four different secure PESEs using a remote work business scenario (focussed towards telework and mobile work) and a simple threat environment. The initial secure PESE concept attributes (although not defined as such in the paper) are used to assess each of the constructed secure PESEs.

When Paper 4 was prepared (in 2008) the terms secure PESE and secure PEE (as defined in Chapter 1) were both still evolving. Paper 4 does not use the term secure PESE, which did not become the term used to describe both a concept and an implementation of the concept until a later stage in the research. Instead Paper 4 uses the terms secure PEE or secure PEE device to refer to what this thesis defines as a secure PESE. The terms secure PEE OS (operating system) and secure PEE virtual machine (VM) to refer to what this thesis defines as a secure PEE are also used. It is recognised the dual use of secure PEE may be confusing for the reader but it is hoped this clarification will avoid any confusion. Also at the time the paper was prepared and published, configuring a PC to boot from an external device was not a user friendly activity, however by 2014 PCs often come



configured with a boot order that has an external device as the first boot device or allows selection of the boot device upon powering the PC via a user friendly interface.

### **2.3.2.2 Prior Research and Knowledge**

In early 2008 a live CD imaged onto a USB flash drive to provide a portable executable solution was mainly used as a utility for IT technicians and administrators (e.g. a disk recovery tool). These utilities were typically based on Linux live CDs (LiveCD, 2013) and the Microsoft portable Windows PE (WinPE, 2013). However, the use of a portable execution environment for mobile workers and teleworkers was starting to receive attention (SourceWire, 2007; LPS, 2008). Similarly, virtualisation was used mainly to maximise the use of hardware platforms (Burger, 2012); however, its use as a portable execution environment had also started to gain momentum (Kwan and Durfee, 2007; Radhakrishnan and Solworth, 2007; Ford and Cox, 2008). Live CDs (also referred as bootable execution environments) and virtualisation are the two execution environments to be packaged with existing security technology to generate the four experimental secure PESEs described. The prior knowledge utilised in Paper 4 can be summarised as:

- Descriptive knowledge in the form of stakeholder discussions and the researcher's knowledge of known threats and typical business use scenario for mobile working.
- Descriptive knowledge gained from the observations and feedback received on the SDGuardian toolkit.
- Prescriptive knowledge emerging from the security model analysis in Paper 3. The threats, policy enforcement mechanisms and security model attributes identified in Paper 3 provide the basis to define threats, security requirements and desired functionality for a secure PESE. The influence and relationship between the Paper 3 knowledge outcomes and the Paper 4 knowledge development is shown below:
  - Paper 3 threats -> Paper 4 threats and security requirements.
  - Paper 3 policy enforcement mechanisms -> Paper 4 desired functionality.
  - Paper 3 security model attributes -> Paper 4 security requirements.
- Prescriptive knowledge in the form of a live operating system (i.e. Slax), a virtualisation package (i.e. Qemu), existing secure and non-secure storage devices and a software encryption package (i.e. Truecrypt).

- Prescriptive knowledge in the form of prior research on trusted portable secure computing environments (Chan et al., 2007; Nepal et al., 2007) and research into the use of virtual machines to provide trusted execution environments (Garfinkel et al., 2003; Kwan and Durfee, 2007).

#### 2.3.2.3 Paper 4

**Paper 4** - James, P. (2008) **Secure Portable Execution Environments: A Review of Available Technologies**, 6th Australian Information Security Management Conference, Perth, pp 70-86.

#### Abstract

*Live operating systems and virtualisation allow a known, defined, safe and secure execution environment to be loaded in to a PC's memory and executed with either minimal or possibly no reliance on the PC's internal hard disk drive. The ability to boot a live operating system or load a virtual environment (containing an operating system) from a USB storage device allows a secure portable execution environment to be created. Portable execution environments have typically been used by technologists, for example to recover data from a failing PC internal hard disk drive or to perform forensic analysis. However, with the commercial potential of portable execution environments becoming realised the requirement for such environments to be secure is becoming increasingly important. To be considered truly secure a portable execution environment should require authentication prior to loading the executing environment (from the USB mass storage device) and provide full encryption of the whole mass storage device.*

*This paper discusses the outcomes from building four portable execution environments, using commercially available and/or freeware technologies. An overview is given of the emerging commercial requirement for secure portable USB execution environments, the security threats addressed and research performed in the area. The technologies and products considered in the review are outlined together with rationale behind the selection. The findings from the implementation of the four portable execution environments are discussed including successes, failures and difficulties encountered. A set of security requirements is defined which is used to gauge the effectiveness of each of the four environments.*

#### Keywords

USB boot, Live Operating Systems, Virtualisation, Encryption, Pre-boot Authentication, Post-boot Authentication, U3 technology.

#### Introduction

Live operating systems (OS) are designed to be loaded from a Compact Disk (CD) or a Universal Serial Bus (USB) storage device into a PC's Random Access Memory (RAM) and execute without necessarily having access to the PC's Hard Disk Drive (HDD). A live OS enables a user to rapidly boot an execution environment and execute an application on an available PC without having to

install the application on the respective PC HDD; a live OS can provide a platform for a Portable Execution Environment (PEE). Using a USB storage device to hold and load the OS (and applications) provides a number of advantages over a CD based live OS (and applications); advantages include a faster load time, the ability to store data generated during a session, configuring the storage device to provide OS swap space (page file/virtual memory) and a more convenient size and shape to a regular CD. However, a notable disadvantage of a USB storage device over a CD is the inconsistent support provided by the different manufacturer's PC Basic Input Output System (BIOS) to allow the booting of a live OS from a USB storage device.

Virtualisation (often referred to as virtual machines or virtual environments) provides an abstract execution environment separate from the physical PC. There are a number of different types of virtual machine (VM) (CSIRO, 2008). The type of virtualisation considered in this paper is a VM that runs within (on top of) the PC operating system; often referred to as a type 2 hosted VM. A "guest" OS is hosted within the type 2 VM and the application is executed within the guest OS. A VM and its guest OS and application can be loaded from a USB storage device into an executing PC to provide a PEE.

The prolific growth in public Internet access centres (e.g. Internet cafes, airport lounges, wireless hot spots, etc) allow an individual to perform sensitive transactions, like Internet banking, on PCs for which no level of trust can be assumed, i.e. these PCs and/or any communication infrastructure may contain malicious software. Also a lack of best practice security (e.g. anti-virus and anti-spyware software and modem/router enabled firewall capabilities) in many homes can result in malicious software residing on PCs unbeknown to the owner/user. Malicious software is capable of capturing and exploiting user credentials (enabling identity theft/fraud to occur) and/or stealing sensitive data when a user is conducting an Internet transaction.

Secure PEEs provide an opportunity for organisations to distribute an Internet application on a USB storage device that can be used safely on PCs which may contain malicious software. In this paper the following business scenario is presented to provide the context for the provision of a secure PEE.

*A compact and easily transportable secure PEE device is distributed by an organisation to individuals for a specific purpose. The secure PEE contains an application that interacts with a server over the Internet. The individual will use the secure PEE device and its application on PCs for which no level of trust can be assumed, therefore the secure PEE device must provide an environment that can protect against malicious software, i.e. the integrity of the secure PEE must be preserved. It is important that the PEE can be reliably loaded into any PC.*

*The secure PEE device should also allow the application or individual to save data on to the secure PEE device. The data may be sensitive and therefore protecting the confidentiality of the data is important. Stored data should be accessible by the individual (when the*

*secure PEE is not being used) from a PC running Windows XP. Any stored data is likely to be sensitive and therefore protecting the confidentiality of the data is important.*

*The secure PEE device, due to its compact nature, could easily be mislaid and therefore ensuring its contents cannot be exploited is paramount.*

To satisfy the above business scenario a secure PEE device would utilise authentication prior to loading the PEE, data encryption, information separation, user/application authentication and device attestation. In this paper a range of available technologies are considered to construct USB based secure PEE devices. The approach adopted is to consider a secure PEE device constructed using a standard thumb drive and available open source technologies and then progressively consider increasingly more sophisticated and expensive technologies that provide additional security capabilities.

Four secure PEEs were constructed that provide the underlying security infrastructure to enable applications to perform secure Internet transactions. The security technologies required by an application to perform Internet transactions are not considered to be within the scope of this paper. The following four PEE devices were constructed:

- Secure PEE Device 1 - Standard Flash Memory Device: A low cost thumb drive with bootable OS, an OS in a VM, partitioning and software based encrypted storage.
- Secure PEE Device 2 - U3 Flash Memory Device: A U3 thumb drive (SanDisk, 2008) with bootable OS, an OS in a VM, partitioning and encrypted storage.
- Secure PEE Device 3 - Hardware Encrypted Flash Memory Device: The IronKey thumb drive (IronKey, 2008) incorporating hardware based encryption with an OS in a VM.
- Secure PEE Device 4 - Portable Hardware Encrypted Hard Disk Drive: The Pocket Silicon Data Vault (SDV) (SecureSystems, 2008) incorporating hardware based encryption with bootable OS, an OS in a VM and cryptographically separated partitions.

Each secure PEE device is assessed to gauge the effectiveness of the technologies utilised to counter a set of key threats and satisfy a set of security requirements.

The following terms are used throughout the paper, a definition is therefore given for each:

- **secure PEE device:** the secure platform/infrastructure consisting of a USB mass storage device configured with a secure PEE, secure storage space and possibly hardware based security mechanisms/technologies (e.g. encryption and secure partitions if available).
- **secure PEE:** the trusted OS, trusted application(s), security technologies (e.g. authentication and software encryption) and the appropriately configured hardware security mechanisms (if any) of the secure PEE device.
- **secure PEE OS:** the trusted OS component of the secure PEE.

- **trusted OS:** an OS that has been acknowledged as secure by the supplier and users. To be considered trusted the OS may have a reduced set of hardened functionality and/or been subjected to independent rigorous evaluation and testing.
- **PEE:** a portable execution environment that does not necessarily have any security technology nor has been specifically configured to be secure.
- **portable storage device:** a USB flash device (often know as a thumb drive or pen drive) or a USB HDD packaged in a portable enclosure.

## Secure Portable Execution Environments – The requirement

### Emerging Business Requirement

Live OS', VMs and other portable software executable from CDs and USB storage devices have been available for a number of years; enabling PEEs to be built. Traditionally such PEEs were typically used by technicians to perform PC maintenance activities (e.g. data recovery from a corrupt or failing HDD) or by specialist IT security/forensics specialists as 'tools of the trade'.

The large data and financial losses organisations and individuals are suffering through malicious software exploiting Internet transactions is well known and documented. Organisations are increasingly looking for assurance that Internet based transactions can be performed securely and that user credentials and data will not be compromised. However, when an organisation allows a service to be provided over the Internet it is not able to control the environment from where the service is initiated, and therefore gaining assurance that a transaction has been performed securely is not possible. To combat the data and financial losses organisations are starting to consider the distribution of secure PEE devices to their staff and/or customers to protect against malicious software and enable secure Internet transactions to be performed.

Notable research on the provision of secure portable applications and secure PEEs includes the work conducted by the Commonwealth Scientific & Industrial Research Organisation (CSIRO) and its proof of concept Trust Extension Device (TED) (Chan et al., 2007; Nepal et al., 2007). The CSIRO work focussed on trust portability, where a VM is used to encapsulate a trusted computing environment into a USB storage device that can be plugged in an untrusted PC and used to perform secure Internet transactions. The key requirement of the TED research was to provide an 'on demand' secure PEE where the user and device identity, and the integrity of the execution environment could be confirmed. The research and development focussed on user, application and device attestation to ensure identities could be confirmed before an application performing an Internet transaction was initiated. Similar research into the use of VMs to provide trusted/secure PEEs has been performed at Stanford (Garfinkel et al., 2003) and Princeton (Kwan, 2007).

The CSIRO work was based on requirements gathered from a user forum with members drawn from the finance sector, government agencies and other service organisations all of whom had mobile workers and/or customers. The user forum identified the need for secure PEEs that enable

secure Internet transactions to be performed. The CSIRO TED proof of concept device did not implement the secure infrastructure considered (in this paper) as vital underlying security technology in the provision of a secure PEE.

### **Threat Environment Addressed**

It is proposed in this paper that a secure PEE device provide countermeasures to address the following three key threats:

1. Malicious software (residing on a host PC) capturing user credentials and data.
2. Sensitive data remnants (resulting from an application storing temporary information) residing on a host PC's HDD following the completion of an Internet transaction; and
3. As a result of loss or theft, unauthorised access is gained to (sensitive) data held on an unsecured USB storage device that was used to store data generated during an Internet transaction.

### **Security Requirements**

It is proposed in this paper that a secure PEE needs to provide functionality to address the following security requirements:

- The secure PEE shall only allow authorised users to load the resident OS/application(s) and access stored data.
- The secure PEE shall preserve the integrity of the resident OS and application(s) from malicious software and other external attacks.
- The secure PEE shall preserve the confidentiality of any stored data from external attacks including theft of the device.
- The secure PEE shall leave no data remnants on the host PC hard disk drive (HDD) following the completion of a user session.
- The secure PEE shall prevent the acquisition of data from the device through the use of forensic analysis techniques, if the device were to be lost/stolen.
- The secure PEE shall confirm both the identity of the user and the device before allowing an Internet transaction to occur.

These security requirements will be used to gauge the security strength of the four constructed secure PEEs.

### **Secure PEE – An Overview of the Required Functionality**

To address the security requirements and threats identified above, and to enable the aforementioned scenario to be satisfied it is proposed in this paper that a secure PEE would ideally utilise the following functionality:

- **Bootable secure PEE OS (pre-boot secure PEE OS):** Cold booting a (trusted) live OS from a USB secure PEE device onto a host PC provides a secure platform as the user can have a high degree of assurance that no malicious software will be present, i.e. any malicious software that was active on the host PC would be eliminated by the power-off required prior to the cold boot. However, booting a live OS from a USB device is neither user friendly nor reliable. To boot a live OS from a USB device requires the boot order to be changed. Some new PCs allow the user to select the boot device at power up whilst other PCs require a user to set the boot order in the PC BIOS. Whilst setting the boot order is not a technically complex task the following aspects make the process unfriendly and unreliable:
  - There are many different types of BIOS and there is no standard set of keystrokes to enter the BIOS and change the boot order.
  - Some older PC's do not support booting from a USB device.
  - Older PC's usually only support USB1.1 which can make the booting of a live OS a slow process.
  - Some public access PCs password protect the BIOS which can prevent a user changing the PC boot order.

Although loading an OS from a cold boot provides a secure platform, given the unfriendly aspects of USB booting means it should not be considered as the only method of loading a secure PEE into a host PC.

- **Secure PEE in VM (post-boot PEE):** As booting a secure PEE directly from a USB storage device has some usability problems an alternative approach to providing a secure platform is to load a VM containing an OS. VMs provide security through isolation, however VMs are susceptible to the following vulnerabilities:
  - Keyloggers: The host OS may contain malicious software that can capture and store keystrokes.
  - Screen shot logging: The host OS may contain malicious software that can capture and store screens.
  - Memory Probing/Attacks: The host OS may contain malicious software that can capture the contents of RAM utilised by the VM.

The business scenario calls for a secure PEE that can be reliably loaded on a range of PCs and protect against malicious software. A bootable OS protects against malicious software but cannot be reliably loaded. Conversely VMs are susceptible to malicious software but can be easily and reliably loaded. Therefore secure PEEs were built that have both a bootable OS and a VM with a guest OS; allowing the user to select the secure PEE that suits the user and/or operating environment. Note – it should be assumed that the bootable OS and VM guest OS



are identical and support the same (trusted) application which executes identically on both bootable OS and VM guest OS.

- **Authentication:** A secure PEE device should not be accessible until the user has entered authentication credentials. The following two authentication methods are considered:
  - *Pre-boot:* The most secure way to authenticate a user is before the live OS or host OS is loaded; it is highly unlikely that malicious software (such as keyloggers) can be present as the respective OS will be loaded from a cold boot. In pre-boot authentication an application is loaded into the PC upon power up and the user authenticates with the secure PEE device. However, pre-boot authentication will experience the same set of pre-boot problems identified above for a bootable secure PEE OS, i.e. pre-boot authentication can be secure but unfriendly to initiate.
  - *Post-boot:* A more convenient approach to authenticate a secure PEE device is to plug it into a PC that has a booted and executing OS. In post-boot authentication a PC either uploads an authentication application from the secure PEE device or a pre-installed authentication application is resident on the PC. Through the authentication application a user authenticates with the secure PEE device. However, post-boot authentication can be subject to attack by malicious software (e.g. keyloggers) resident within the host PC.

Where possible, secure PEE devices will be built that can support both pre-boot and post-boot authentication.

- **Device encryption:** To preserve the confidentiality of the secure PEE and data residing on the USB device, on-the-fly encryption is required. On-the fly-encryption can be provided by software or hardware; this paper will consider both software and hardware encryption.
- **A secure PEE that does not store data on the host PC HDD:** To prevent data remnants residing on the host PC HDD following the use of a secure PEE, the secure PEE device must be configured to:
  - provide swap space (virtual memory/page file) for the secure PEE OS on the secure PEE device itself; and
  - ensure the secure PEE OS and application(s) write all temporary information to available allocated space on the secure PEE device.
- **Ability to separate the PEE OS and data:** To preserve the integrity of the PEE OS and any stored data the secure PEE device should support storage partitioning and role based differentiated access rights to partitions. Such partitioning allows separation and isolation to be achieved.



- **Protection of secure PEE OS and data:** In addition to the provision of a partitioning capability a secure PEE should also allow a partition to be defined as Read-Only; such partitions can be used to protect the integrity of the secure PEE OS from malicious software. Read-Only partitions can also be used to protect the integrity of 'valued' data.
- **User, application and device authentication and attestation:** To enable a secure Internet transaction to occur an application on the secure PEE device needs to mutually authenticate with a remote server providing the service. Assurance is required that the application and secure PEE device are genuine; therefore device and application attestation technology is required. The techniques and technology required to perform application and device mutual authentication and attestation with a remote server are considered beyond the scope of this paper.

## The Technologies/Products Selected

The range of technologies that satisfy the functionality requirements and enable secure PEE devices to be constructed is growing rapidly. The technologies considered in this paper represent what the author considers to be amongst the best available at the time of writing (May/June 2008). Both freeware and proprietary technologies were considered. A summary is given below of the technologies and products utilised to build the four secure PEE devices.

### The Live Operating System

A range of live OS' were considered, including Windows MiniPE, Ubuntu, Puppy Linux and Slax. Following an evaluation of a set of live OS' the Linux distribution Slax (Slax 2008) was selected. Slax is a cut down Linux distribution based on Slackware and has been developed primarily as a live OS. Slax was selected for the following reasons:

- The ease with which Slax could be installed on a USB storage device as a bootable OS (PendriveLinux.com-Slax, 2008).
- The quality of documentation available (Wielenge, 2008).
- The compact yet functionally rich distribution which rapidly loads into a PC's RAM.
- The ability to boot first time every time on a range of PCs and allow Internet access without any configuration of the OS.
- Slax was rated best live OS on "The LiveCD List" (Brand, 2008).

The other OS' considered presented a range of problems including inability to boot from a USB device without time consuming configuration, slow to load, would not load into a PC with limited RAM and would not load consistently into a range of different PCs.

For the purposes of this paper Slax is assumed to be secure and is considered to be a trusted OS. In practice the OS selected would be subject to analysis to harden and remove functionality considered unnecessary for a secure PEE. Version 6.0.6 of Slax was used.

## Virtual Environment

The virtual environment Qemu was selected for the review. Qemu (Bellard, 2008) is a freeware type 2 VM. Qemu has gained widespread recognition as an effective VM due to the ease by which it can be used and configured (Hannay & James, 2007). Qemu was selected for the following reasons:

- It is an extensively tried and tested open source VM.
- It can be loaded from a USB storage device on to a PC executing Windows XP without requiring any installation or Windows XP administrator privileges; although this approach to using Qemu does result in slower execution times for the guest OS and application(s).
- A Qemu image containing Slax was readily available from [www.pendrivelinux.com](http://www.pendrivelinux.com) (PendriveLinux.com-Qemu, 2008).

## Software Encryption

The open source, freeware application Truecrypt (TrueCrypt 2008) was selected to provide software encryption. Truecrypt is a comprehensive application providing encryption for PC HDDs and portable storage devices. Truecrypt has a traveller mode option that allows a number of encrypted volumes to be placed onto a USB storage device and then accessed from any PC. When a USB storage device is configured with traveller mode a Truecrypt autoexec application (known as the 'traveler disk') is placed on to the USB device; the Truecrypt application is obviously in unencrypted form. When the USB device is plugged into a PC the user is given the option to execute Truecrypt where upon, following successful authentication, a Truecrypt volume on the device is mounted and appears as a Windows removable device.

Truecrypt was selected for the following reasons:

- It is an extensively tried and tested open source package.
- It provides strong 'on the fly' encryption, allowing selection from a variety of crypto and hashing algorithms.
- It is well documented (TrueCrypt-Foundation, 2008).
- It is one of only a few freeware packages that support encrypted volumes on portable storage devices.
- The autoexec/automount feature uses the standard Windows XP popup.
- A Truecrypt volume can be mounted Read-Only.

Some of the limitations of the Truecrypt traveller mode option include:

- the requirement for the host PC to be executing with Windows administrator mode.
- it is not possible to encrypt a whole storage device partition (in traveller mode).
- Truecrypt volumes can be identified on a storage device.

- only one volume can be automounted.
- the autoexec application is easily identifiable on the storage device and therefore susceptible to attack.

### U3

A U3 flash drive was selected as one of the platforms for the secure PEE device. U3 technology (SanDisk, 2008) allows a user to load a set of applications, and launch them from a U3 flash drive whenever the U3 flash drive is plugged into a PC executing Windows XP. The seamless launch of applications is achieved by the U3 hardware (the U3 chip)

reserving a small part of the device (containing an ISO image) that is interpreted by Windows XP as a CDROM.



Figure 1: An example of a U3 USB flash memory device

When the U3 device is plugged into a PC executing Windows XP an autoexec of the ISO image occurs. The U3 ISO provides an interface that allows applications to be selected and launched. As can be seen from Figure 1, a U3 flash device looks like a standard thumb drive<sup>18</sup>. U3 flash devices are widely used and some interesting applications of the technology have been achieved (Al-Zarouni 2006). A U3 flash drive was included in the research because:

- it provides a more sophisticated USB flash drive
- given the wide scale adoption of U3 devices it was considered important to determine how effective a U3 device would be when configured and used as a PEE device.

### Hardware Encryption

Whilst software based encryption provides adequate data protection to preserve the confidentiality for many organisations it is considered insufficient to protect highly sensitive data and applications. The main vulnerability is that the data encryption key for a software encryption package is stored in a PC's RAM (during operation) and therefore potentially subject to capture. Recent research performed by a team from Princeton (Haldermany et al., 2008) has shown that under certain circumstances it is possible to retrieve an encryption key from a PC's RAM even after the PC has been powered down. Hardware based encryption usually stores the encryption key in the hardware crypto engine on the storage device.

It is considered particularly important that a secure PEE device support hardware encryption due to the way secure PEE devices will be distributed and used. Two storage devices that use hardware based encryption were selected; IronKey and the Pocket Silicon Data Vault (SDV). Both devices

<sup>18</sup> Image from U3 website

have a crypto engine implemented on an integrated circuit located within the enclosure or the storage device.

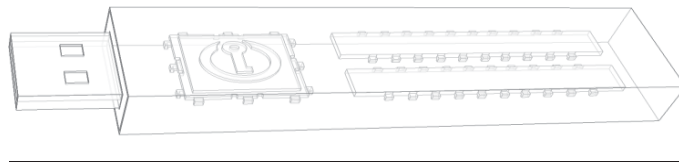
### IronKey

Since mid-2007 an increasing number of USB flash storage devices (packaged as thumb drives) have become available with full storage encryption performed by an integral hardware crypto engine; IronKey is an example of such a device. An IronKey is slightly larger than a typical thumb drive.



*Figure 2: An example of a IronKey flash memory device*

Figure 2 presents a pictorial image of an IronKey and Figure 3<sup>19</sup> presents a cross sectional image of an IronKey showing the crypto & access control chip and flash memory. The IronKey chip controls all access to the flash memory.



*Figure 3: Pictorial cross-section of IronKey*

Some of the important features of IronKey include:

- Post-boot authentication – the IronKey has implemented a “U3 like” feature where the IronKey chip reserves a small part of the device that is interpreted by Windows XP as a CDROM. When the IronKey is plugged into the PC an autoexec of an authentication application occurs.
- Strong encryption – the IronKey crypto engine implements the strong 128 bit Advanced Encryption Standard (AES).
- Key escrow (key recovery) – when the IronKey is first initialised a key escrow capability ensures that if the authentication credentials are forgotten they can be recovered from the IronKey website.
- Key destruction – after ten failed authentication attempts the IronKey encryption key(s) are destroyed and the device becomes unusable.

IronKey was selected due to its compact form factor (it is only slightly larger than a standard USB thumb drive (pen drive)). The IronKey does however have the following limitations:

---

<sup>19</sup> Images obtained from IronKey website.

- Support for Windows XP/Vista only – an IronKey can only be authenticated with XP/Vista. As Windows only recognises the first partition on a thumb drive the IronKey can only be recognised as a single partitioned device.
- Key escrow limitation – the IronKey key recovery facility requires the user to remember a set of web authentication credentials. Also as the recovery key and authentication credentials are held outside the user's control the assurance afforded to their security cannot be guaranteed.
- No pre-boot support - an IronKey can only be authenticated by post-boot authentication.

### **Pocket SDV**

The Pocket SDV is a portable 2.5 inch form factor HDD. The Pocket SDV contains an 'SDV chip' which controls access to an internal 1.8 inch HDD. The SDV chip encrypts all data written to the 1.8 inch HDD. There are two modes of authentication supported by the Portable SDV; pre-boot and post-boot authentication.



*Figure 4: An example of a Pocket SDV HDD*

When authenticating using the pre-boot method the host system will boot off the attached Portable SDV and the Authentication Application (AA) will be launched from the Portable SDV's on-board flash memory. Once successful authentication has been performed the secure PEE is loaded. Authentication via the post-boot method requires that the Portable Authentication Application (PAA) is installed on the host PC. Also once successful authentication has been achieved the Pocket SDV allows access to data based on pre-defined access rights. The Pocket SDV supports differentiated access rights, i.e. user profiles can be defined with permissions to access different parts of the integral HDD.

Figure 4 provides a pictorial image of the Pocket SDV<sup>20</sup>. The key functionality and attributes of the Pocket SDV can be summarised as:

- Full disk encryption - all data on the Pocket SDV is encrypted; with encryption performed at the sector level which reduces the possibility that pattern matching can be performed to break the encryption.
- Totally independent of PC Operating System - the Pocket SDV behaves like a standard USB mass storage device and has no dependencies upon the PC operating system; it works with Linux as well as Windows.
- Multiple Partitions - up to 15 partitions (drives/volumes) can be defined for a Pocket SDV with each partition cryptographically separated from the other partitions by its own cryptographic key.

<sup>20</sup> Image obtained from Secure Systems web site.

- Differentiated Access Rights & User Profiles - a Pocket SDV allows user profiles (roles) to be defined with different authentication credentials and access rights allowing different parts of the Pocket SDV integral HDD to be accessed according to the selected user profile.

The Pocket SDV was selected due to its comprehensive functionality; however it does have the following limitations:

- Form factor – the shape and size of the Pocket SDV is not as convenient as the USB thumb drive.
- Pre-installation of PAA – post-boot authentication can only be performed after the PAA has been installed onto the host PC.

## **Building Secure Portable Execution Environments**

### **Host PC Technology Constraints**

The following constraints in the host PC technology influence how secure PEEs can be constructed:

1. Unfriendly PC BIOS' and a lack of user experience changing a PC boot order can result in problems when assigning a secure PEE device as the first boot device.
2. Windows XP only supports FAT (File Allocation Table) file systems for USB flash devices.
3. Windows XP only supports access to the first FAT partition (volume/drive) on a USB flash drive<sup>21</sup>.

It is again worth reviewing both the BIOS boot and VM vulnerability issues as building secure PEEs that address these issues is an important proposition in this paper. Changing a PC BIOS setting can be difficult, inconvenient and in some instances impossible to change (i.e. the BIOS is locked). Building a secure PEE based purely on a bootable OS is impractical. As noted above, to provide both a high level of security and convenience the secure PEEs constructed will contain both a bootable (trusted) OS & application(s) and a loadable VM containing a (trusted) OS & application. The bootable OS provides the most secure execution environment but it is an unfriendly activity to enable a secure PEE device to be the first boot device. Alternatively loading a VM with a guest OS is a user friendly action but is vulnerable to attacks from malicious software that may be resident on the host PC. By providing a secure PEE with multiple partitions with a bootable OS in one partition and a VM with a guest OS in another partition, allows the user to select either execution environment based upon:

- user skill level.
- allowed access to the PC BIOS.
- access to PC power on/off switch.
- the level of trust that can be placed in the host PC.

---

<sup>21</sup> This is not the case for a USB portable HDD where Windows drivers will mount all FAT partitions on the HDD.

- user convenience.

The Windows limitation of only supporting FAT file systems on a flash drive and also only recognising the first partition identified on a flash device limits the opportunity to provide an elegant secure PEE solution. Whilst a single partition could be used to hold a bootable OS, a VM (with OS) and user generated data. However, a single partition solution would not readily support encryption and data separation and therefore the aforementioned secure PEE requirements for confidentiality, integrity and preventing data remnants would be difficult (probably impossible) to satisfy. One solution to overcome this Windows limitation and satisfy the secure PEE requirements is to use multiple partitions (a mixture of FAT and Linux file systems) as described below.

### **Secure PEE Configuration Decisions**

Secure PEE devices were built using multiple partitions; this approach enables certain secure PEE requirements and aspects of the business scenario to be achieved for secure PEE devices built using USB flash drives.

It is important the first partition on a secure PEE device has a FAT file system so that it is recognised by Windows XP, if the first partition were to be a Linux partition Windows would not be able to mount it and as only the first partition is recognised by a USB flash device no other partitions on the device would be mounted. Whilst the selected OS (Slax) can be installed as a bootable OS on a FAT partition it must however be the first partition. If the bootable OS is in the first partition then it will not be possible to have a VM in a second partition that can be loaded into a PC executing Windows XP. Therefore the first partition on a secure PEE device must be a FAT file system containing the VM and guest (trusted) OS and application(s).

A Linux ext3 partition was required as the second partition to hold a bootable copy of Slax. The boot loader 'lilo' was used to configure the Slax Master Boot Record (MBR) so that Slax would boot from the second partition.

An outcome from the use of a Linux partition is that when the user is accessing the secure PEE device from Windows XP the Linux partition is not mounted and therefore the possibility of the bootable OS being attacked or corrupted is reduced.

## Secure PEE Configuration Model

Based on the above configuration decisions the configuration model presented in Figure 5 defines a target configuration for a secure PEE device. For each secure PEE constructed an attempt was made to implement the target configuration model. The configuration model presents four partitions. The first is a Windows (FAT 16/32) partition containing a VM hosting an OS with additional space for operational data. The second partition is a Linux partition with a bootable OS.

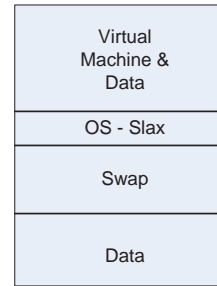


Figure 5: Target Configuration Mode

Ideally the first and second partitions will be Read-Only partitions with the Read-Only mechanism enforced by the secure PEE device infrastructure. The third (Linux) partition will be swap space for both the VM guest OS and the bootable OS. Finally the fourth partition (with a FAT file system) will provide storage space for user generated data; whilst the fourth partition is a target requirement in practice it is only possible for a secure PEE implemented on a USB portable HDD. All partitions will be primary partitions and ideally fully encrypted.

## Secure PEE Device 1 - Implementation on a Standard Flash Memory Device

The flash memory device used was a Buffalo 2GB high speed thumb drive. The target configuration model could not be fully achieved as only one FAT file system can be mounted; therefore the first partition was used for the VM and user created data. Figure 6 models the implemented configuration. Truecrypt was used to protect the VM and user generated data; two truecrypt volumes (containers) were generated in the first partition, one for the VM (containing Slax and an application) and another for user generated data – the truecrypt volumes provided separation.

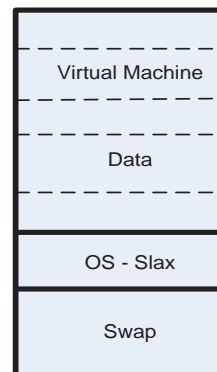


Figure 6: Configuration Model for Secure PEE Device

The second and third partitions were Linux partitions; the second an ext 3 (journaling) file system for a bootable Slax and the third an ext2 file system for OS swap space – the swap partition was used by both the VM running Slax or the bootable Slax. The following observations were made during the construction and testing of secure PEE device 1:

- The Slax distribution included the script “liloinst.sh” which allowed the second partition on the secure PEE device to be easily configured as a bootable partition.
- Booting Slax from the second partition was tested on a range of PCs and only once (on a Dell 5400 PC) did it fail to boot.
- Ubuntu (on a live CD) with fdisk provided a simple system to configure the device’s partition table, file systems and bootable version of Slax.



- The truecrypt volumes were easy to create and use, they could however be identified on the secure PEE device.
- When truecrypt was configured for autoexec and the secure PEE device was plugged into the host PC a standard Windows 'removable device' popup appeared, from which the truecrypt option could be selected resulting in the truecrypt authentication screen appearing. Upon successful authentication the first partition on the secure PEE device is opened as a 'removable device' with a truecrypt label.
- When truecrypt was configured for 'automount upon authentication' the truecrypt volume containing the VM was automatically mounted. Only one truecrypt volume could be automounted. The automounted truecrypt volume is presented as a 'removable device' by Windows XP.
- The VM and its guest OS executed from a truecrypt volume without problems, however when the truecrypt automount Read-Only feature was used for the truecrypt volume containing the VM, the VM encountered problems booting Slax (the guest OS). Therefore it may not be possible to protect the integrity of the VM & its guest OS by using the truecrypt Read-Only mechanism.
- When the truecrypt volume used to store and protect user generated data was mounted it was mounted as a 'local disk' rather than a 'removable device', which could confuse a user.
- Limited testing showed that both the bootable OS and VM guest OS appeared to use the swap partition successfully, albeit with some degradation of performance.

### Secure PEE Device 2 - Implementation on a U3 Flash Memory Device

The U3 device used was a SanDisk 2GB flash drive. Like the secure PEE device 1, the target configuration model could not be fully achieved as only one FAT file system can be mounted. Therefore the first partition was used for the VM and user generated data. Figure 7 models the implemented configuration. Also, like secure PEE device 1, truecrypt was used to protect the VM and user generated data. The second and third partitions were Linux partitions and configured exactly like secure PEE device 1.

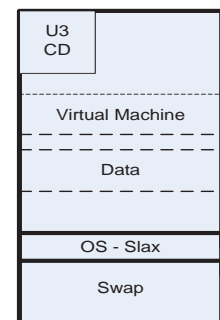


Figure 7: Configuration Model for Secure PEE Device 2

The following observations were made during the construction and testing of secure PEE device 2:

- Most of the observations made above with respect to secure PEE device 1 were found to apply to the U3 based secure PEE device 2. The notable differences occurred when the U3 authentication feature was enabled, as highlighted below.
- The truecrypt automount works with U3. The U3 red cruzer icon appears in addition to the windows popup enabling truecrypt to be selected.

- The U3 (post-boot) Windows authentication feature prevents any upload from the U3 device until successful authentication. Therefore the truecrypt autoexec and automount features could not be utilised if the U3 authentication feature was enabled.
- Although a post-boot authentication mechanism, if the U3 authentication is enabled the secure PEE bootable OS was blocked from loading. Similarly if the U3 secure PEE device was plugged into a host PC running Linux no access to any partitions was possible, i.e. the U3 authentication feature appears to block all access to the device until successful authentication.
- When the two truecrypt volumes (in the first partition) were mounted each volume was labelled as a 'local disk', whilst the U3 secure PEE device first partition is labelled as a 'removable disk'; this approach to labelling truecrypt volumes differs from secure PEE device 1. As each truecrypt volume 'local disk' label only differs by the drive letter it may be difficult for a user to distinguish the VM volume and the data volume.

### Secure PEE Device 3 - Implementation on a Hardware Encrypted Flash Memory Device (IronKey)

The IronKey could not be configured to achieve the target configuration model due to its Windows only based authentication feature. The IronKey authentication feature prevents any access to the IronKey until successful authentication; therefore neither could a bootable OS be installed nor was it possible to use Ubuntu to set up separate partitions for the VM, user generated data and swap space.

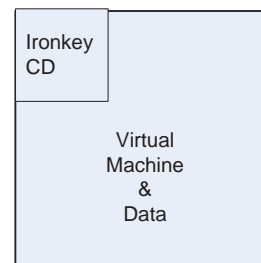


Figure 8: Configuration Model for Secure PEE Device

Figure 8 models the IronKey secure PEE device configuration; essentially it is the standard IronKey containing a VM and separate directory structure for user generated data. The following observations were made during the construction and testing of secure PEE device 3:

- Establishing the encryption keys (via the creation of the authentication credentials) both for the IronKey and the key recover capability (via the IronKey web site) was a user friendly seamless set of actions.
- The IronKey was tested on numerous host PCs running Windows XP and successfully authenticated (based on the insertion of the correct credentials).
- As would be expected the secure PEE VM and guest OS loaded from the IronKey correctly.

#### Secure PEE Device 4 - Implementation on Hardware Encrypted Hard Disk Drive (Pocket SDV)

The Pocket SDV was able to be configured to fully satisfy the target configuration model. As the Pocket SDV is a USB HDD, Windows is able to detect and mount all FAT partitions on the HDD; therefore both the first and fourth partitions are accessible on the Pocket SDV secure PEE device. The following configuration could be achieved for the secure PEE device 4:

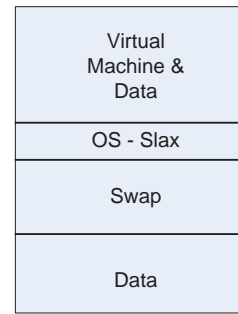


Figure 9: Configuration Model for Secure PEE Device 4

- The first and second partitions were set as Read-Only, to protect the integrity of the VM and bootable OS.
- A separate Read-Write data partition was created to separate and protect user generated data as required by the reference target configuration.
- As a further integrity protection measure, different user profiles were created with different access rights to partitions; a pre-boot profile denied access to partition 1 and a post-boot profile denied access to partition 2.

#### A Qualitative Assessment of the Secure PEE Devices

The required features and functionality of the four secure PEE devices was driven primarily by the capabilities required to satisfy the business scenario and the countermeasures required to address the identified threats. A set of security requirements were defined to allow a comparative analysis of the four secure PEE devices to be performed; the comparative analysis is given below. As would be expected the security requirements overlap with security aspects of the business scenario and threat countermeasures. Before performing the comparative analysis, the capabilities of the four secure PEE devices are summarised against the requirements of the business scenario and countermeasures required for the threats.

##### Satisfying the Business Scenario

The key aspects of the scenario are identified together with a summary of how the four secure PEEs provide appropriate functionality to address the scenario.

**A compact and easily transportable secure PEE device:** Three of the four secure PEEs were implemented on thumb drives. The Pocket SDV whilst compact is in a less transportable form.

**The individual will use the secure PEE device and its application on PCs for which no level of trust can be assumed, therefore the secure PEE device must provide an environment that can protect against malicious software:** Two secure PEE OS environments are provided.

Where a user is concerned about the trustworthiness of a PC the bootable OS can be used. Read-Only partitions were configured (where possible) to protect the secure PEE against corruption by malicious software.

***It is important that the PEE can be reliably loaded into any PC:*** The less secure, but easy to load VM based secure PEE provides a high level of reliability.

***The secure PEE device should also allow the application or individual to save data on to the secure PEE device:*** Three of the four secure PEEs were able to be configured to provide a separate secure volume/partition for user generated data.

***Protecting the confidentiality of the data is important:*** All of the secure PEEs enforced user authentication to deny access to unauthorised users coupled with storage encryption to protect the confidentiality of data.

***Stored data should be accessible by the individual (when the secure PEE is not being used) from a PC running Windows XP:*** All four secure PEEs were configured to enable user generated data to be easily stored and retrieved.

***The secure PEE device, due to its compact nature, could easily be mislaid and therefore ensuring its contents cannot be exploited is paramount:*** User authentication and encryption protect the contents of the secure PEE device.

#### **Counter Measures to Address Threats**

***Threat - Malicious software (residing on a host PC) capturing user credentials and data:***

Countermeasure - Using the bootable OS capability of the secure PEE device will ensure no malicious software is present to capture user credentials and data. If the VM capability of the secure PEE is used it is more difficult to prevent the capture of user credentials and data, therefore it is assumed a user will not use the VM if the host PC is considered to be likely to have been compromised by malicious software.

***Threat - Sensitive data remnants (resulting from an application storing temporary information) residing on a host PC's HDD following the completion of an Internet transaction:*** Countermeasure - The secure PEE device swap partition will prevent data remnants in the form of page files being written to the host PC HDD. It is assumed that a secure PEE OS and Internet based application which are considered to be trusted will have been hardened to prevent any temporary data being written to the host PC HDD.

***Threat - As a result of loss or theft, unauthorised access is gained to (sensitive) data held on an unsecured USB storage device that was used to store data generated during an Internet transaction:*** Countermeasure - User authentication and encryption will prevent access by unauthorised personnel to user generated data following the loss or theft of a secure PEE device.

## Comparative Analysis Security Strength of Secure PEE

Table 1 provides a comparative analysis of the security strength of the secure PEEs by presenting for each security requirement a statement of compliance for each secure PEE; with the strongest secure PEE highlighted.

Requirements	Secure PEE Features that Address the Requirements			
	Device 1: Std Flash	Device 2: U3	Device 3: IronKey	Device 4: SDV
Only allow authorised users to load the resident OS/application(s) and access stored data	Post-boot authentication via truecrypt password. Only authorised users can gain access to the 2 truecrypt volumes; however access to non-encrypted storage space is possible. No pre-boot authentication therefore unauthorised users can gain access to bootable OS.	Post-boot authentication via truecrypt password. Only authorised users can gain access to the 2 truecrypt volumes; however access to non-encrypted storage space is possible, unless U3 password used. No pre-boot authentication therefore unauthorised users can gain access to bootable OS.	Post-boot authentication via IronKey password. No access can be gained to the device until successful authentication. No pre-boot authentication, however no bootable OS is available.	<b>Both strong pre-boot and post-boot authentication. No access can be gained to the device until successful authentication.</b>
Preserve the integrity of the resident OS and application(s) from malicious software and other external attacks	Partitioning provides separation and isolation. Although truecrypt can mount a volume Read-Only, running a VM and OS inside a truecrypt Read-Only volume failed. Further analysis and testing of the Read-Only feature is required.	Partitioning provides separation and isolation. Although truecrypt can mount a volume Read-Only, running a VM and OS inside a truecrypt Read-Only volume failed. Further analysis and testing of the Read-Only feature is required.	No partitioning and no Read-Only mechanism. Once successful authentication has been achieved the device is open to any read and write access.	<b>Partitioning provides separation and isolation. Integrity can be preserved by the Read-Only capability. Read-Only and No-Access permissions can be set per partition per user profile, i.e. different profiles can have different access rights to the same partition.</b>

Requirements	Secure PEE Features that Address the Requirements			
	Device 1: Std Flash	Device 2: U3	Device 3: IronKey	Device 4: SDV
Preserve the confidentiality of any stored data from external attacks including theft of the device	User generated data is stored in a truecrypt encrypted volume.	User generated data is stored in a truecrypt encrypted volume.	User generated data is protected by fully hardware based encryption.	<b>User generated data is protected by fully hardware based encryption. Each partition has its own encryption key, if one is broken then only one partition at most is exposed.</b>
Leave no data remnants on the host PC hard disk drive (HDD) following the completion of a user session	<b>By utilising the swap partition on the device no page files are written to the host PC.</b>	<b>By utilising the swap partition on the device no page files are written to the host PC.</b>	The IronKey cannot be configured with a swap partition.	<b>By utilising the swap partition on the device no page files are written to the host PC.</b>
Prevent the acquisition of data from the device through the use of forensic analysis techniques, if the device were to be lost/stolen	256 bit AES truecrypt volumes will prevent acquisition; however software encryption is not as strong as hardware encryption <sup>22</sup> .	256 bit AES truecrypt volumes will prevent acquisition; however software encryption is not as strong as hardware encryption.	<b>128 bit AES hardware based encryption will stop even the most sophisticated highly resourced forensic investigation.</b>	<b>128 bit AES hardware based encryption will stop even the most sophisticated highly resourced forensic investigation.</b>
Confirm both the identity of the user and the device before allowing an Internet transaction to occur	This requirement is considered out of scope in this paper.			

Table 1

## Conclusion

Four secure PEEs were built and tested using a range of freeware and commercial off the shelf technologies. A brief summary of each secure PEE device is given together with areas that could be subject to further investigation.

*Secure PEE Device 1 - Implementation on a Standard Flash Memory Device:* Secure PEE 1 was the cheapest solution to construct, it provides a reasonable capability that addresses many of the security requirements, counters the threats and could be used under certain circumstances within

<sup>22</sup> Software encryption keys reside in the host PC RAM and therefore could be captured by a determined, highly skilled and well-resourced forensic analyst (Haldermany et al., 2008).

the given business scenario. It would not be suitable as a turnkey solution that could be supplied to an organisation's customers to perform secure transactions. It is conceivable however that such a secure PEE device could be distributed to a 'controlled audience' for a specific application. For instance an organisation could distribute the secure PEE device to a certain group of employees with fixed operational instructions.

*Secure PEE Device 2 - Implementation on a U3 Flash Memory Device:* The secure PEE features and functionality of this device are identical to secure PEE device 1, but instead uses a U3 thumb drive as its platform which provides a strong titanium enclosure, the 'U3 chip' and a post-boot authentication mechanism. As per secure PEE device 1, the device is probably not suitable for a turnkey application; it would however provide a more robust platform than secure PEE device 1 for a 'controlled audience' application.

*Secure PEE Device 3 - Implementation on a Hardware Encrypted Flash Memory Device (IronKey):* The secure PEE built using the Ironkey had fewer features than the other secure PEE devices. It did provide a strong platform with hardware based encryption and authentication. Secure PEE device 3 did not provide the best platform for use in the business scenario presented.

*Secure PEE Device 4 - Implementation on a Hardware Encrypted Hard Disk Drive (Pocket SDV):* Secure PEE device 4 provided the best functionality to satisfy the security requirements and counter the threats. It would be the best secure PEE to use as a solution for the business scenario presented in this paper. The Pocket SDV does have a number of disadvantages, including its size, shape and weight compared to the other secure PEE devices considered, and it is also not as robust as platforms like the IronKey and U3.

The work presented in this paper was defined to be a contained and complete piece of research. However, a number of areas for further investigation arose during the research including:

- Determine if a VM and guest OS can be configured to execute within a truecrypt Read-Only volume.
- Determine if truecrypt can be launched from the U3 CD partition as other applications have been configured to do (Al-Zarouni & Al-Harji 2007).

## References

- Al-Zarouni, M. (2006). The Reality of Risks from Consented use of USB Devices. 5th Australian Digital Forensics Conference, Perth, Edith Cowan University.
- Al-Zarouni, M. Al-Harji. H. (2007). A Proof of Concept Project for Utilising U3 Technology In Incident Response. 6th Australian Digital Forensics Conference, Perth, Edith Cowan University.
- Bellard, F. (2008). "Qemu Open Source Processor Emulator." Retrieved May, 2008, from <http://bellard.org/qemu/>.
- Brand, N. (2008). "The LiveCD List sponsored by FrozenTech." Retrieved April, 2008, from <http://www.frozentech.com/content/livecd.php>.



- Chan, J. N., S. Moreland, D. Hon Hwang, Shiping Chen, Zic, J. CSIRO ICT Centre, Marsfield (2007). User-Controlled Collaborations in the Context of Trust Extended Environments. WETICE 2007. 16th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2007. .
- CSIRO (2008). Virtual Machines: An Initial Analysis of Threats and Remedial Actions. CSIRO, 2008.
- Garfinkel T, P. B., Chow J, Rosenblum M, Boneh D (2003). Terra: A Virtual Machine Based Platform for Trusted Computing. Proc. 9th ACM Symposium on Operating Systems Principles SOSO'03.
- Hannay, P., James, P. (2007). Pocket SDV with SDGuardian: A Secure & Forensically Safe Portable Execution Environment . 5th Australian Digital Forensics Conference, Perth, Edith Cowan University.
- IronKey. (2008). "IronKey Technology." Retrieved May, 2008, from <https://www.ironkey.com/technology>.
- Haldermany A.J., S. D. S., Nadia Heningery, William Clarksony, William Paulx, and A. J. F. Joseph A. Calandrinoy, Jacob Appelbaum, and Edward W. Felten (2008). Lest We Remember: Cold Boot Attacks on Encryption Keys. Proc. 2008 USENIX Security Symposium.
- Kwan P, D. G. (2007). Practical Uses of Virtual Machines for Protection of Sensitive Data. Proc. 3rd Information Security Practice and Experience Conference (ISPEC).
- Pendrivelinux.com-Qemu (2008). "Qemu Persistent SLAX Linux tutorial." Retrieved May, 2008, from <http://www.pendrivelinux.com/2007/04/02/qemu-persistent-slax-linux/>.
- Pendrivelinux.com-Slax (2008). "SLAX USB flash drive installation using Windows." Retrieved April, 2008, from <http://www.pendrivelinux.com/2006/09/20/all-in-one-usb-slaxzip/>.
- SanDisk. (2008). "What is a U3 Smart Drive." Retrieved May, 2008, from <http://www.u3.com/smart/default.aspx>.
- SecureSystems. (2008, May 2008). "Portable SDVs." Retrieved May, 2008, from [http://www.securesystems.com.au/pages/04\\_news/brochure\\_pdf/Portable-Specs.pdf](http://www.securesystems.com.au/pages/04_news/brochure_pdf/Portable-Specs.pdf).
- Slax. (2008). "Slax - your pocket operating system." Retrieved May, 2008, from <http://www.slax.org/>.
- Nepal S, Hon Hwang and David Moreland (2007). Trust Extension Device: Providing Mobility and Portability of Trust in Cooperative Information Systems Springer Berlin / Heidelberg
- TrueCrypt. (2008). "TrueCrypt - Free Open Source On-The-Fly Encryption." Retrieved May, 2008, from <http://www.truecrypt.org/>.
- TrueCrypt-Foundation. (2008). "TrueCrypt Users Guide." Version 5.1a. Retrieved May, 2008, from <http://www.truecrypt.org/docs/>.
- Wielenge, D. (2008). "Slax Guide." Retrieved May, 2008, from <http://www.geocities.com/slaxfansite>.

#### 2.3.2.4 Synopsis

**Outcomes and Contribution to Knowledge:** The paper documented a set of security requirements and desired functionality for a secure PESE which provided the initial



attributes for the concept. The security requirements were used to gauge the effectiveness of four experimental secure PESEs, providing validation of the emerging initial secure PESE concept. The experimental secure PESEs were constructed to suit both different operational and security situations occurring within the use scenario. Both the bootable operating system and the virtual machine were shown to provide an appropriate secure PEE component of a secure PESE, albeit the virtual machine being susceptible to malware and also having performance and data remnant issues. The experimental secure PESE known as 'secure PEE device 4' is the instantiation that best encapsulates the initial concept.

***Contemporary relevance, linkage with other papers and future direction:*** The paper provided a contribution to the growing academic research and commercialisation of secure portable computing environments and also established the direction for the doctoral research enabling the research problem to eventually be defined. The paper sets the direction for research into secure PESE and secure PEE design with Papers 5, 6, 7, 8 and 10 drawing upon or being influenced by the developed knowledge (as shown diagrammatically in Figure 1.1).

The paper proposed two further possible areas of investigation, but neither was progressed as both involved experimenting with technology components subsequently discounted in the research.

#### **2.3.2.5 Attributes of the Initial Concept**

Using the outcomes from Paper 4 the attributes of the secure PESE initial concept are:

- Only allow authorised users to load the resident OS/application(s) and access stored data.
- Preserve the integrity of the resident OS and application(s) from malicious software and other external attacks.
- Preserve the confidentiality of any stored data from external attacks including theft of the device.
- Leave no recoverable data remnants on the host PC hard disk drive (HDD) following the completion of a user session.

- Prevent the acquisition of data from the device through the use of forensic analysis techniques, if the device were to be lost/stolen.
- Require the following features:
  - Bootable secure PEE OS (pre-boot secure PEE OS).
  - Secure PEE in VM (post-boot PEE).
  - Pre-boot Authentication
  - Post-boot Authentication
  - Device encryption.
  - A secure PEE that does not store data on the host PC HDD.
  - Ability to separate the PEE OS and data.
  - Protection of PEE OS and data.

These attributes are the Paper 4 security requirements and desired functionality with the exception of the attestation capability, which was deemed not to be a necessary attribute because neither stakeholder consultation nor a review of security issues identified it as a required feature of a secure PESE.

### **2.3.3 Remote Work – Categories Considered**

Remote working is a growing practice, for instance in the United Kingdom legislation has recently been enacted (Seebacher, 2014) to allow any employee (after a defined period of employment) to request to work flexibly (including remote working). The PhD research considered the use of secure PESEs in three categories of remote work: teleworking, and mobile and deployed working.

Teleworking (Baruch, 2000; Lister and Harnish, 2011; Deloitte, 2011; DBCDE, 2011; Telework, 2013), also known as telecommuting, is defined (in this thesis) as the external processing of information conducted predominately from home. As identified in Paper 3 compared to similar sophisticated economies, Australia has yet to embrace the teleworking category of remote working, but it is expected to grow dramatically in the coming years (AccessEconomics, 2010). ICT enabled teleworking gained recognition in the early 1980s when companies like ICL and Xansa employed home-based programmers (Kinsman, 1987). The emergence of the Internet, digital subscriber line technology, digital

cellular networks and more recently superfast broadband has enabled teleworking to grow (DBCDE, 2013; Deloitte, 2011). Papers 3 and 7 provide a good overview of telework and its benefits and therefore no further description is provided here.

Mobile working (Dade, 2013) is defined (in this thesis) as work performed outside the corporate office and conducted from any location as and when the work needs to be performed, providing productivity benefits to organisations and convenience for the employees. Like teleworking, mobile working has become an established practice. Mobile working started to grow rapidly once cellular networks began providing data broadcast capabilities and faster transmission speeds which allowed laptops to connect from any location (within broadcast range) using a cellular network modem (Optus, 2012). However, more recently the prolific rise in the use of smartphones and tablets that access corporate networks and cloud services from both WI-FI hotspots and cellular networks has resulted in mobile working becoming a standard practice (Optus, 2012; ACMA, 2013).

Deployed working is considered (in this thesis) to be a practice where a group of workers is assigned to work in a particular location for a period of time usually to perform a well-defined assignment. Examples of deployed work are emergency and relief work following a catastrophic event, military deployment into a conflict zone (ADoD, 2007) or the assignment of a team to a strategic partner to work on a joint project (Corbett et al., 1999). The increasingly popular work practice of activity based working (Colliers, 2011) could, in some circumstances, be considered to be deployed working. The ICT infrastructure associated with deployed working is typically in existence in advance of the deployment and purpose built/configured for expected deployment scenarios. An organisation benefits from deployed working by being able to respond to an event using a specialist team with the ICT infrastructure ready for deployment to achieve the desired outcomes (Folks and Richard, 2011).

#### **2.3.4 Prior Research into Secure Remote Working**

A literature analysis was conducted to identify the prior research performed into secure remote working. The pertinent literature has been grouped into the following areas: research focus, lack of risk management, management attitudes, poor implementation of remote working, technology configurations and guidelines for secure remote working.

Although research was identified that considered remote work security no prior work comprehensively reviewed the information security issues and proposed tangible capabilities to manage the issues. The intent of this section is to summarise at a high level the notable published work addressing remote work security. A further section below documents a holistic assessment of the security issues identifying the literature that supports the assessment.

This literature analysis identified considerably more publications for telework security than for the other two remote work categories considered in this thesis. However, the research is summarised generically as remote work security unless specifically identified as being particularly applicable to a certain category.

**Research focus:** The literature shows that an organisation implementing a remote work policy has typically focused upon the managerial, organisational, behavioural and human resource aspects of remote working (Baruch, 2000; Gibson et al., 2002; Kowalski and Swanson, 2005; Ye, 2012). Information security policies for remote working have traditionally received either less attention (Clear, 2007) or the security risks are cited as the reason for not implementing a remote work policy (Garner and Dick, 1997; Pyöriä, 2003; Whiteman and Dick, 2006).

**Lack of Risk Management:** Early work by Sturgeon identified that organisations were implementing telework arrangements (in North America) without performing a security threat and risk assessment tailored for the remote work environment (Sturgeon, 1996). Sturgeon also identified that organisations were allowing telework to occur with limited or no network (Internet) security in place and surprisingly as late as 2007 a comprehensive review of telework in small and medium sized organisations continued to identify a lack of network (Internet) security being enforced (Clear, 2007). Goslar found that management complacency in organisations implementing network security led to external attackers exploiting connections reserved for remote workers (Goslar, 2000), e.g. an unauthorised individual entering a network through a poorly configured firewall and VPN. Even where good network and PC security is enforced Pyöriä (Pyöriä, 2003) highlights how backup drives and USB flash drives (used to transport data) can be security vulnerabilities when encryption and access controls are not enforced on these drives.

A risk analysis methodology for secure teleworking, developed using information gathered from Dutch Government agencies (Hoogendijk, 2006), was identified. The methodology has four phases and consists of a highly structured template driven approach with the definition of a comprehensive set of threats associated with user actions, location and technology, together with very specific controls (countermeasures) to manage the threats. This methodology was trialled within a small focus group (Hoogendijk, 2006) but no evidence was identified of its application to organisations planning or reviewing remote working arrangements.

**Management Attitudes:** In Australia information and physical security were cited as a reason for the slower embracement of remote working (Whiteman and Dick, 2006). A survey of executives in Australian organisations (Garner and Dick, 1997) identified that both the security of corporate assets in the remote work location and remote access to corporate servers was a major concern and prevented the implementation of remote work policies. Conversely, the USA General Services Administration commissioned a comprehensive review of barriers impacting the adoption of telework (GSA, 2002) and found that the executive management of USA government agencies did not see information security as a barrier to implement remote working. This review (GSA, 2002) found that whilst the executive management were concerned about information security when implementing remote work policies they were confident adequate secure solutions were available. The review did highlight that the poor implementation of such secure solutions, a lack of resources and expertise to manage the implemented solutions and/or a lack of security training for remote workers were the main contributors to poor information security.

**Poor Implementation of Remote Working:** Carelessness by remote workers when handling sensitive information in both soft and hardcopy is cited as a reason for security breaches (Deloitte, 2011). A lack of both an appropriately designed security solution and security awareness training are identified reasons for carelessness (Whiteman and Dick, 2006; Godlove 2012). However, remote working has on occasions been unfairly cited as an insecure work practice (Joice, 2007). Joice identifies how a number of security breaches were deemed to be caused through remote working and the technology used, but analysis of the breach typically showed poor organisation and management practices and/or lack

of security training for remote workers. Furnell found a strong security culture throughout an organisation contributes to a positive attitude by remote workers in enforcing security (Furnell, 2006), a finding also supported by an examination of the security behaviour of over 150 teleworkers (Godlove, 2012).

**Guidelines for Secure Remote Working:** The best examples of detailed and comprehensive guidelines for implementing secure remote working have been produced by USA government agencies. In 2007 the USA National Institute of Standards and Technology (NIST) published guidelines (NIST, 2007) on how to secure PCs and other devices used for remote work (predominately focussed towards telework). The NIST guidelines were comprehensive covering security technology for both network security (through the use of encryption on all networks including the Internet) and PC security (through encryption, backups, anti-virus, anti-malware, personal firewalls, limited privileges, content filtering and application configuration). A further enhanced set of NIST guidelines focussing upon enterprise telework were published in 2009 (NIST, 2009). The USA Office of Management and Budget and the Office of Personnel Management also issued generic information security guidelines and handling procedures (OMB, 2011; OPM, 2011) and the USA General Service Administration has issued a federal management regulation bulletin that addressed telework technology with a focus on IT security (GSA, 2007). However, whilst these comprehensive guidelines identify the processes, procedures and technology to adopt there is no identification and correlation to the specific security threats, vulnerabilities and risks; it is left to each organisation to decide which technologies to apply to address threats and manage risks.

**Technology options for secure remote working:** The NIST guidelines (NIST, 2007; NIST, 2009) provide details on how to secure various technology configurations used for remote work. The technology configurations typically used (where a PC is used by the remote worker) are:

- **Thin client to corporate server:** In this configuration a software application on the remote worker's PC is used to connect to a corporate server or cloud service; the PC becomes a terminal on a networked server. The application could be a browser using https (Kyrnin, 2014), a VPN (Mitchell, 2014), a remote desktop (Rouse, 2014), an

application hosting client (Citrix, 2014) or application streaming client (Rouse 2011b; Citrix, 2013). The advantages of this configuration are that all processing is performed on a server requiring little or no local data storage and (as the PC is being used as a dumb terminal) if the PC becomes infected or compromised then the impact on remote work has to some extent been minimised. The disadvantages are the requirement for expensive servers (to host many remote workers) and possibly performance and productivity issues if the Internet/network connection is slow or unavailable.

- **Local processing with remote applications on PC:** In this configuration the remote worker's PC is used to perform the work and process data. The software applications to be used for remote work are directly installed onto the PC's operating system. Data may be held on a server/cloud (with access via a VPN) or data may be stored locally. The main advantage of this configuration is that a PC (either furnished by the organisation or the remote worker) needs only the required applications installed to enable remote work to occur. The main disadvantage is if the PC becomes infected or compromised then the impact on remote work and corporate data could be severe.
- **Local processing using virtualisation:** The remote worker uses a virtualised environment for remote work in this configuration. The virtual machine may be held on the PC's local disk or a virtual machine client (Rouse, 2011) is used to load the virtual machine from a server/cloud when remote work is to be performed. The main advantage of this configuration is that the virtualisation provides separation between the remote work and any other activities performed on the PC. The main disadvantage is the virtual machine can be compromised using a number of (often unforeseen) techniques (James, 2008); these virtual machine vulnerabilities are explored in detail in Chapter 4, Paper 5.
- **Up-loadable execution environment from portable storage device:** This technology configuration is the subject of the PhD research described in this thesis. It is the hypothesis of this thesis that 'secure PESEs can be used (i.e. provide a secure configuration) to manage the information security risks in the remote work environment'. The secure PESE concept provides a reference model for the research.

The use of existing secure PESE 'like' technology for remote working is considered separately in the next section below.

### **2.3.5 Prior Research and Development into Secure PESEs**

Using the attributes of the initial secure PESE concept as criteria a literature search was performed (in 2008). A number of interesting research artifacts and products were identified; an overview of each is given below with its respective capabilities and features presented (using the search criteria) in a tabular form. The identified artifacts/products were categorised as having bespoke hardware or using standard USB flash drives. Interestingly all artifacts/products using bespoke hardware also had an execution environment that used virtualisation whilst the artifacts/products using USB flash drives as the hardware platform all had bootable (live CD) execution environments.

#### **2.3.5.1 Bespoke Hardware based Secure PESEs with Virtualised Secure PEE**

The following three 'secure PESE like' artifacts/products were identified that addressed a number of the criteria and utilised purpose built hardware and storage platforms. These three artifacts/products had similarities to the experimental secure PESEs known as 'secure PEE device 3' and 'secure PEE device 4' (presented in Paper 4).

***The Commonwealth Scientific and Industrial Research Organisation (CSIRO) Trust Extension Device (TED) research project:*** The CSIRO TED research project commenced in 2006 with the objective of providing a secure portable computing device (Chan et al., 2007; Nepal et al, 2007; Nepal et al., 2011). The TED concept was based upon a purpose built portable hardware artifact that contained a TCG Trusted Platform Module (TPM) and a secure PEE implemented as an up-loadable virtualised environment containing an operating system and applications. The TPM (TPM, 2008) is a microcontroller that provides secure generation and storage of encryption keys. The TED used the encryption capabilities of the TPM for authentication and attestation<sup>23</sup>, which if successfully confirmed allowed the virtualised execution environment to be up-loaded onto a PC and executed. In 2008 the TED was a prototype concept device (i.e. a research artifact) using a standard flash drive with a TPM implemented as a software emulator and the virtual machine implemented using the Qemu open software (QEMU, 2014). Surprisingly the TED

---

<sup>23</sup> Attestation is the process of confirming the identity and integrity of a computing device or application.



was not designed to provide secure storage, i.e. the TPM capability was not used to encrypt the storage medium. Although the TED was produced in a number of different prototypes (including a version with a hardware TPM) CSIRO never commercialised the TED. It is understood that work on the project was stopped in 2011. Tables 2.1 and 2.2 present an assessment of the TED using the secure PESE initial concept attributes.

**Table 2.1 Conformance of CSIRO TED to Initial Concept**

<b>Secure PESE Security Requirement</b>	<b>CSIRO TED Feature/Capability</b>
Only allow authorised users to load the resident OS/application(s) and access stored data	An authentication and attestation capability ensures only authorised users can access the virtualised secure PEE.
Preserve the integrity of the resident OS and application(s) from malicious software and other external attacks	No access controls or write protection mechanisms provided and therefore integrity of the secure PEE is not preserved.
Preserve the confidentiality of any stored data from external attacks including theft of the device	Content of TED is only protected by authentication and attestation controls. No encryption provided.
Leave no data remnants on the host PC hard disk drive (HDD) following the completion of a user session	The secure PEE is implemented as an up-loadable virtual machine so there is the possibility of data recovery from the PC's paging system held on the PC's disk drive.
Prevent the acquisition of data from the device through the use of forensic analysis techniques, if the device were to be lost/stolen	The content of the TED is not encrypted so there is limited protection against forensic analysis.

**Table 2.2 Conformance of CSIRO TED to the Required Features**

Secure PESE Desired Functionality	CSIRO TED
Bootable secure PEE OS (pre-boot secure PEE OS).	No
Secure PEE in VM (post-boot PEE).	Yes
Pre-boot Authentication	No
Post-boot Authentication	Yes
Device encryption.	No
A secure PEE that does not store data on the host PC HDD.	Yes
Ability to separate the PEE OS and data.	No
Protection of PEE OS and data.	No

Whilst the TED is a secure PESE like device with similar operational goals the primary security objective was to authenticate a user and attest the application and device before allowing a secure PEE to be uploaded on to a PC and executed. Lack of encryption, secure partitions and the implementation of its secure PEE in a virtual machine results in the TED only satisfying a few of the secure PESE security requirements and desired functionality.

**The Bull globull product:** In 2008 Bull (Weber, 2009; Bull, 2014), a French IT company, announced globull (BullDirect, 2008; globull, 2010; globull, 2011), a secure portable computing device. The globull combines secure storage with an execution environment; it is a ‘secure PESE like’ device. The globull encrypts all data held on the device including the execution environment. The device enforces authentication via a touch screen before the execution environment or data can be accessed. The execution environment utilises virtualisation to enable a virtual machine (containing an operating system and applications) to be up-loaded onto a host PC executing Windows or Linux. The product is a commercial success and continues to be marketed in 2014. No trial of commercial use of the TED was identified. Tables 2.3 and 2.4 present an assessment of the globull using the secure PESE initial concept attributes.

**Table 2.3 Conformance of Bull globull to Initial Concept**

<b>Secure PESE Security Requirement</b>	<b>Bull globull Feature/Capability</b>
Only allow authorised users to load the resident OS/application(s) and access stored data	Only authorised users can gain access via touch screen authentication
Preserve the integrity of the resident OS and application(s) from malicious software and other external attacks	Secure partition preserves the integrity of the virtualised secure PEE.
Preserve the confidentiality of any stored data from external attacks including theft of the device	Full encryption of device's storage medium protects stored data.
Leave no data remnants on the host PC hard disk drive (HDD) following the completion of a user session	The secure PEE is implemented as an up-loadable virtual machine so there is the possibility of data recovery from the PC's paging system.
Prevent the acquisition of data from the device through the use of forensic analysis techniques, if the device were to be lost/stolen	Encryption of the storage medium protects against forensic analysis.

**Table 2.4 Conformance of Bull globull to the Required Features**

<b>Secure PESE Desired Functionality</b>	<b>Bull globull</b>
Bootable secure PEE OS (pre-boot secure PEE OS).	No
Secure PEE in VM (post-boot PEE).	Yes
Pre-boot Authentication	No
Post-boot Authentication	Yes – Touch Screen
Device encryption.	Yes
A secure PEE that does not store data on the host PC HDD.	Yes
Ability to separate the PEE OS and data.	Yes
Protection of PEE OS and data.	Yes

A case study published in 2009 describes how the French Defence Ministry used the globull to support mobile working. The globull was supplied to Defence personnel to transport data securely between sites and enable processing to be performed from an available PC using the up-loadable virtualised secure PEE. The globull is a good example of a product that gets close to implementing the initial secure PESE concept; however, its

secure PEE is implemented as an up-loadable virtual machine and therefore it is susceptible to attacks from an infected host PC operating system possibly limiting its use for the processing of highly sensitive information.

***MXI Security Stealth MXP:*** The Stealth MXP (Smith, 2008) is a secure PESE like device from MXI Security (a Canadian secure storage company) (MXI, 2008) and was introduced into the North American market in 2008. The Stealth MXP provides secure storage using strong encryption, a secure PEE based upon the MojoPac sandbox (MojoPac, 2006; BusinessWire, 2008) and biometric authentication. A secure partition protects the secure PEE. In 2011 Imation (Imation, 2014) acquired MXI Security and subsequently replaced the Stealth MXP with the Ironkey range (Ironkey, 2014). Tables 2.5 and 2.6 present an assessment of the Stealth MXP using the secure PESE initial concept attributes.

**Table 2.5 Conformance of MXI Security Stealth MXP to Initial Concept**

<b>Secure PESE Security Requirement</b>	<b><i>Stealth MXP</i> Feature/Capability</b>
Only allow authorised users to load the resident OS/application(s) and access stored data	Biometric authentication (fingerprint reader) ensures only authorised users can gain access.
Preserve the integrity of the resident OS and application(s) from malicious software and other external attacks	Secure partition preserves the integrity of the virtual machine based secure PEE.
Preserve the confidentiality of any stored data from external attacks including theft of the device	Full encryption of device's storage medium protects stored data.
Leave no data remnants on the host PC hard disk drive (HDD) following the completion of a user session	The secure PEE is implemented as an up-loadable virtual machine so there is the possibility of data recovery from the PC's paging system held on the PC's disk drive.
Prevent the acquisition of data from the device through the use of forensic analysis techniques, if the device were to be lost/stolen	Encryption of the storage medium protects against forensic analysis.

**Table 2.6 Conformance of MXI Security Stealth MXP to the Required Features**

<b>Secure PESE Desired Functionality</b>	<b><i>Stealth MXP</i></b>
Bootable secure PEE OS (pre-boot secure PEE OS).	No
Secure PEE in VM (post-boot PEE).	Yes
Pre-boot Authentication	No
Post-boot Authentication	Yes - biometric
Device encryption.	Yes
A secure PEE that does not store data on the host PC HDD.	Yes
Ability to separate the PEE OS and data.	Yes
Protection of PEE OS and data.	Yes

A case study published in 2008 describes how the Stealth MXP was used by the Canadian Government to implement teleworking (MXI Security, 2008). The Stealth MXP is a product that implements a number of the secure PESE security requirements. However, like the globull it uses a virtualised secure PEE which makes it vulnerable to attacks from an infected host PC operating system and therefore could limit its use for some remote work applications where highly sensitive information is processed.

#### **2.3.5.2 USB Flash Drive based Secure PESEs with Bootable Secure PEE**

The two experiential secure PESEs known as ‘secure PEE device 1’ and ‘secure PEE device 2’ (in Paper 4) utilised standard flash drives with bootable live CD based operating systems. In 2008 a research artifact and a product were identified with similarities to the two aforementioned secure PESEs, but also included features that made them more compatible with the initial concept.

***The Becrypt Trusted Client (TC):*** In 2007 Becrypt (Becrypt, 2013) released TC a secure PESE that provides an up-loadable secure PEE on to a PC from a standard flash drive (SourceWire, 2007; TrustedClient, 2009). TC incorporates authentication, (software implemented) encryption and a bootable secure PEE. The TC is a secure PESE like implementation using a standard flash drive as the hardware and storage platform. The encryption capability is an integral component that implements on-the-fly encryption. The TC is based upon a tailored version of Linux and allows an organisation to disable and remove functionality to reduce the application set that is provided to a remote worker.

Tables 2.7 and 2.9 present an assessment of the Becrypt TC using the secure PESE initial concept attributes.

**Table 2.7 Conformance of Becrypt Trusted Client to Initial Concept**

Secure PESE Security Requirement	Becrypt TC Feature/Capability
Only allow authorised users to load the resident OS/application(s) and access stored data	Pre-boot (password) authentication ensures only authorised users gain access.
Preserve the integrity of the resident OS and application(s) from malicious software and other external attacks	No access controls or write protection mechanisms provided and therefore integrity of the secure PEE is not preserved.
Preserve the confidentiality of any stored data from external attacks including theft of the device	Software encryption preserves the confidentiality of both the operating system and stored data.
Leave no data remnants on the host PC hard disk drive (HDD) following the completion of a user session	Bootable execution environment prevents data remnants on host PC disk drive.
Prevent the acquisition of data from the device through the use of forensic analysis techniques, if the device were to be lost/stolen	Encryption of the storage medium protects against forensic analysis.

**Table 2.8 Conformance of Becrypt Trusted Client to the Required Features**

Secure PESE Desired Functionality	Becrypt TC
Bootable secure PEE OS (pre-boot secure PEE OS).	Yes
Secure PEE in VM (post-boot PEE).	No
Pre-boot Authentication	Yes
Post-boot Authentication	No
Device encryption.	Yes
A secure PEE that does not store data on the host PC HDD.	Yes
Ability to separate the PEE OS and data.	No
Protection of PEE OS and data.	No

A case study published in 2009 explains how the TC was used by the UK Government Foreign and Commonwealth Office to allow embassy personnel to work remotely on any available PC (e.g. from home or at another non-UK embassy location) and access and process sensitive information securely (Becrypt, 2009). The Becrypt TC is a secure

portable execution and storage environment that satisfies a number of the attributes of the secure PESE concept. It is available as a low cost solution to support remote working, but its main weakness is there is no protection of its execution environment.

***The USA Department of Defence (DoD) Lightweight Portable Security (LPS):*** The LPS (LPS, 2008) was developed as part of the USA DoD software protection initiative (Hughes and Stytz, 2003). The LPS became available as a test release (i.e. a research artifact) in 2008 although it was not until 2010 (Harris, 2010) that the first formal release was available. The LPS is a secure PEE that is imaged onto a standard flash drive to form a 'secure PESE like' device. The LPS is a tailored version of Linux and is available as a thin network client (known as LPS-Remote Access), allowing only remote connection to corporate servers, or is available as a local execution environment (known as LPS-Public), with a small fixed set of applications. The limited set of applications available with LPS-Public include a browser that supports a smart card capability for authentication, an office automation suite, remote desktop and a text editor. The LPS is designed primarily for use as a network client in a remote location with data stored on a network server in a corporate office. However, local data storage is possible and an encryption option enables data to be protected. Encryption can be turned on and off as required. The LPS itself is not encrypted. Tables 2.9 and 2.10 present an assessment of the LPS using the secure PESE initial concept attributes.

**Table 2.9 Conformance of USA DoD LPS to Initial Concept**

Secure PESE Security Requirement	USA DoD LPS Feature/Capability
Only allow authorised users to load the resident OS/application(s) and access stored data	The LPS' OS and applications can be loaded without authentication, but a browser enables a https connection to a server through smartcard based authentication.
Preserve the integrity of the resident OS and application(s) from malicious software and other external attacks	No access controls or write protection mechanisms provided and therefore integrity of the secure PEE is not preserved.
Preserve the confidentiality of any stored data from external attacks including theft of the device	User has ability to encrypt files held on the secure PESE.
Leave no data remnants on the host PC hard disk drive (HDD) following the completion of a user session	Bootable execution environment prevents data remnants on host PC disk drive.
Prevent the acquisition of data from the device through the use of forensic analysis techniques, if the device were to be lost/stolen	File encryption prevents recovery of sensitive data.

**Table 2.10 Conformance of USA DoD LPS to the Required Features**

Secure PESE Desired Functionality	USA DoD LPS
Bootable secure PEE OS (pre-boot secure PEE OS).	Yes
Secure PEE in VM (post-boot PEE).	No
Pre-boot Authentication	No
Post-boot Authentication	Yes – browser based
Device encryption.	File encryption only
A secure PEE that does not store data on the host PC HDD.	Yes
Ability to separate the PEE OS and data.	No
Protection of PEE OS and data.	No

An article published in 2010 claims 30 USA Defence agencies were actively using the LPS-Remote Access version and over 35,000 licences for the LPS-Public version have been downloaded (Jackson, 2010). The LPS is a freely distributed, albeit a limited functionality remote work portable execution environment. The LPS is similar to the experimental 'secure PEE device 1' but with all unnecessary functionality removed to make it a more secure device.



### **2.3.5.3 Synopsis of Prior Research and Development into Secure PESEs**

The literature analysis showed that a number of research artifacts and commercially available products partially satisfied the initial secure PESE concept; however, no single artifact/product fully satisfied it and none could match the conformance achieved by the experimental 'secure PEE device 4'. Despite their weaknesses, the five artifacts/products identified would provide a suitable computing environment for various remote working scenarios although it is unlikely any would be considered suitable for use with highly sensitive data.

No artifact/product combined a bootable secure PEE with bespoke hardware (like the 'secure PEE device 4'). The use of virtualisation makes a secure PEE vulnerable to a host PC infected with malware whilst the use of a standard flash drive as a hardware lacks the mechanisms to protect the integrity of secure PEE. The artifacts/products did provide some features and ideas that could augment the initial secure PESE concept, e.g. the LPS was designed as a network client providing only minimal applications and thereby reducing the likelihood of exploitation of vulnerabilities if the LPS contained the unnecessary applications.

The literature analysis did not identify any published underlying reference model, concept or theory used to assist in the design and development of the five artifacts and products. The initial secure PESE concept provided a basic reference model to analyse the five artifacts/products; however this concept was based upon a partial review of the security issues for remote working. A grounded concept requires a full review of the security issues together with identification of the capabilities necessary to counter the issues.

### **2.3.6 Holistic Review of Security Issues and the Secure PESE Countermeasures**

#### **2.3.6.1 Background**

A corporate office in an organisation implementing best practice information security management (DSD, 2012; ISO/IEC 27001:2013, 2013) will have physically secure offices that are also resilient against environmental conditions, have appropriate security procedures that employees should follow and implement security awareness programs for employees. Such an organisation will have logical network boundary controls to prevent both unauthorised external access and cyber-attacks, and logical access controls to

protect the computing environment and data should an unauthorised individual be able to gain physical access to a PC. The corporate security environment is likely to have been designed and configured to respond to a threat and risk model developed through a regularly performed (ISO/IEC 27005:2011, 2011). An organisation implementing best practice security is likely to fully consider all of the security issues when implementing a remote working policy. However, many organisations do not implement best practice information security management (CertAustralia, 2013; Cisco, 2014) and therefore will not be in a strong position to implement secure remote working. One of the aims of this doctoral research was to develop a secure PESE that can be used for remote work by both security conscious organisations and organisations that are less aware of the remote work security issues. A secure PESE that provides countermeasures to remote work security issues can be considered to be extending best practice security management to the remote location (Peacey, 2006).

The initial secure PESE concept presented in this chapter was identified through a prototyping exercise and the definition of a set of high level security requirements and desired functionality. A literature review then considered both the prior research performed to address remote work security concerns and the prior research into 'secure PESE like' artifacts/products that could satisfy the initial concept. However, a complete review of the remote work security issues is required to enable the identification of the functionality that a secure PESE could provide. This section therefore considers the full set of security issues.

#### **2.3.6.2 Consumed Knowledge**

A number of prior publications have identified and assessed the security issues to be considered when implementing a remote work policy (Sturgeon, 1996; Hoogendijk, 2006; Clear, 2007; Pyöriä, 2011; Ampomah et al., 2013). Two of these publications provided a comprehensive description of the issues. The first was a risk analysis methodology for secure teleworking (Hoogendijk, 2006) which identified threats (supported by a description of the associated security issue) and the corresponding countermeasures. The threats/issues were grouped into user actions, location and technology and were technically focussed with specific countermeasures proposed. Although comprehensive, the security issue descriptions were prescriptive. The second publication considered

security issues using people, physical equipment, organisation and environment as categories (Ampomah et al., 2013). Whilst the publication did consider network security (as a physical equipment issue) neither the insecure use of technology (by remote workers) nor the vulnerabilities inherent in technology were considered. These two publications are drawn upon to provide a broader, enhanced and holistic description, with the issues grouped by location, personnel, insecure use of technology and technology vulnerabilities. Established publications on security threat and risk assessment (Landoll, 2005) provide justificatory knowledge for the approach taken and the threat definitions.

For each security issue a threat definition(s) is given together with the functional requirement(s) for a secure PESE and its integral secure PEE to prevent the exploitation of vulnerabilities identified by the issue. These requirements provide a specification that can be analysed to identify a research gap. Explanatory text supporting a requirement's specification is presented in parenthesis. A requirement may be appropriate to address one or more issues. The review assumes a remote worker uses a PC to access and process data stored on corporate server or cloud service.

#### **2.3.6.3 Location Security**

**Weak physical security:** The physical security controls of a work location form an important and integral part of an overall information security policy (ISO/IEC 27001:2013, 2013). The remote work location is obviously different to a corporate office particularly within a security conscious organisation. Such a corporate office is likely to have appropriate physical access controls (that are commensurate with the sensitivity of the information stored and processed) to prevent an unauthorised individual gaining access to the building. These controls will vary depending upon the type of organisation and the sensitivity of the work performed. Physical access controls to a data processing area range from a high security building using (human) security guards, individual access swipe cards, biometric systems, sophisticated door locks and escorted access (Protective-Management, 2011; Protective-ICT, 2011) to a lower security environment where access is controlled by a reception desk and locked doors.

The security controls in the remote work location are likely to be very different to the corporate office (Zbar, 2000) and will also vary depending upon the type of remote work

to be performed. The physical security controls for home based telework could range from monitored security alarms and strong quality door and window locks (ASIAL, 2014) to just low cost door locks. In the mobile environment where work may be performed from any potential location no physical security can be assumed. Depending upon the organisation and the work to be performed the deployed environment is more likely to have better physical controls. Military deployments are likely to have guards and controls similar to a high security building (Protective-Management, 2011; Protective-ICT, 2011), whilst a commercial organisation that has deployed staff working in a complementary (or possibly competing) host organisation would have to rely upon the host organisation's physical security.

Weak physical security in the remote work location could allow unauthorised access resulting in the theft of a PC, subverting or tampering with a PC and its computing environment, and/or unauthorised access to information (Hoogendijk, 2006).

*Threat Definitions:*

- Remote work PC is destroyed or stolen.
- Remote work PC is tampered with and subverted.
- Intruder is able to gain access to data.

*Secure PESE requirements:*

- A secure PESE shall have a small highly portable form factor (allowing it to be disconnected from the PC and stored in a lockable draw/cabinet when not in use).
- A secure PESE shall implement anti-tamper mechanisms to render the device inoperable if attempts are made to tamper with it.
- A secure PESE shall encrypt its whole storage medium to protect stored information (if the device is stolen).
- A secure PESE shall implement pre and post boot authentication, and access controls (to ensure unauthorised users cannot gain access).

**Environmental factors:** A corporate office is likely to have infrastructure implemented to account for environmental conditions like power failure, excessive heat and fire. Such infrastructure will include uninterruptable power supplies on servers, possibly an

emergence power generator, air-conditioning and fire monitoring and response systems (Protective-ICT, 2011). Depending upon the category of remote work being performed, a remote work location is less likely to have the same level of infrastructure as a corporate office. A loss of power, a failure in air-conditioning (or no air-conditioning installed) in extreme heat or a fire in the remote location could damage the computing environment and/or prevent work from being performed (Hoogendijk, 2006).

*Threat Definition:* Remote work cannot be performed due to environmental condition.

*Secure PESE requirements:*

- A secure PESE shall have a small highly portable form factor (allowing it to be taken to another location and used with any available PC if the remote work location experiences fire, a loss of either power or air-conditioning).
- A secure PESE shall operate with any X86 PC.

#### **2.3.6.4 Personnel Security**

***Remote workers knowledge/experience of information security:*** Remote workers will have differing levels of information security knowledge and experience (Furnell, 2006; Godlove, 2012). Evidence shows that the type of employing organisation, the mode of work performed and the sensitivity of information processed is expected to influence the security education, training and awareness (SETA) received by the remote worker (GSA, 2002; Godlove 2012). However, it should be noted that SETA is a critical component of any remote work policy irrespective of the sensitivity of the information processed (Joice, 2007; Jilani et al., 2013). Remote workers receiving little or no SETA may perform actions that inadvertently make their computing environment vulnerable to cyber-attack, data loss or forensic data discovery (Joice, 2007; Crossler et al., 2013).

*Threat Definition:* Remote workers have no or insufficient information security skills/experience resulting in insecure information handling and processing.

*Secure PEE requirements:*

- A secure PEE shall only have the applications required to securely perform the remote work.

- The operating system and all applications forming the secure PEE shall be hardened to remove or disable insecure functionality.
- A secure PEE shall have a simple user interface (requiring minimal training).

A secure PEE with a simple user interface and reduced and hardened functionality can provide a countermeasure to no or insufficient SETA as it will be easy to use and limit vulnerable actions.

***An organisation's inability to control access to a PC used for remote work:*** A PC used for remote work may be personally owned by the worker or furnished by the employing organisation. As the PC is outside the physical control of the organisation it is possible that family and/or friends of a remote worker (in the case of teleworkers and mobile workers) or allies/joint-venture workers (in the case of deployed workers) could use the PC (Whiteman and Dick, 2006), irrespective of its ownership. One of the predominant issues for an organisation implementing a remote work policy is maintaining the integrity and availability of the computing environment so that work can be performed as and when required (Riswadkar and Riswadkar, 2009). If the PC is used by anyone other than the remote worker the integrity and possibly the availability of the computing environment may be affected as a result of accidental inappropriate actions including the unintended introduction of malware (Whiteman and Dick, 2006; Antonopoulos, 2007). With many organisations implementing a bring your own PC/laptop policy (Overby, 2009) the PC provided for remote work is increasingly likely to be owned by the remote worker further compounding the issue as a personally owned PC is more likely to be used by the worker's family and friends (WSJ, 2013).

***Threat Definition:*** Unauthorised user accidentally/deliberately performs action that impacts integrity or availability of the PC's execution environment.

***Secure PESE requirements:*** No specific requirements are necessary to address this issue as the use of the secure PESE for remote work means that it does not matter if the integrity of the remote work PC's computing environment is compromised as the secure PESE uses only the PC's hardware, e.g. processor, memory, graphics and network hardware. If an

inappropriate action causes the PC to become unavailable then a remote worker can use the secure PESE on another PC (depending upon the availability).

***Preparing and enforcing compliance to security procedures:*** An organisation implementing a remote work policy needs to prepare appropriate security procedures to guide remote workers on how to minimise information security risks (GSA, 2002; Deloitte, 2011). Such procedures should be supported by a SETA programme (Godlove, 2012). Lack of security procedures will make the whole organisational computing environment vulnerable (Joice, 2007).

***Threat Definition:*** No or poor security procedures provided to the remote worker resulting in insecure information handling and processing.

***Secure PESE requirements:***

- A secure PEE (shall only have the applications required to securely perform the remote work (necessitating that only a small set of security procedures is required to support the distribution of secure PESEs).
- The operating system and all applications forming the secure PEE shall be hardened to remove or disable insecure functionality.

***Remote worker behaviour:*** The behaviour of a remote worker can have an impact upon security. It has been determined that approximately half of all reported organisational security breaches occur from within the organisation (Baker et al., 2010). Interestingly, an examination of the willingness of over 150 remote workers (predominately US Federal Government workers) to comply with information security guidelines identified good information security awareness and behaviour (Godlove, 2012); possibly reflecting the commitment to ensuring the practice is a success. However, evidence exists that security breaches have occurred due to poor security behaviour by remote workers (Ng and Rahim, 2005). Behaviour can be categorised as misbehaviour or deviant behaviour (Crossler et al., 2013). Misbehaviour by a remote worker could include not adhering to security procedures, performing unsafe Internet activities (e.g. clicking on phishing web links and emails) and inadvertently storing sensitive corporate data on unsecured storage.

Deviant behaviour by a remote worker could include industrial/political espionage, theft of information and sabotage of systems.

*Threat Definitions:*

- Misbehaviour or deviant behaviour of remote worker causes breach of information confidentiality.
- Misbehaviour or deviant behaviour of remote worker impacts integrity and availability of the PC's execution environment.

*Secure PESE requirements:*

- A secure PEE (component of the secure PESE) shall only have the applications required to securely perform the remote work (thereby limiting the opportunity to use a secure PESE for any inappropriate behaviour).
- The operating system and all applications forming the secure PEE shall be hardened to remove or disable insecure functionality.

***Storage and Disposal of Hardcopy Material:*** Best practice security (ISO/IEC 27002:2013, 2013) for hardcopy material recommends procedures for printing, storage and disposal of sensitive information. Such information handling procedures are required for the remote work location as weak storage and disposal practices or carelessness by the remote worker can lead to a breach of confidentiality.

*Threat Definition:* Inappropriate handling of hardcopy material causes breach of confidentiality.

*Secure PESE requirements:*

- No specific requirements can be defined for a secure PESE as this issue is an information hardcopy issue.
- The secure PEE applications could be configured to prevent the local printing of documentation and/or the downloading (onto the secure PESE) of information.
- The secure PESE could be used as a thin client that ensures all information processing is performed on the server and also prevents information from being downloaded.



#### 2.3.6.5 Insecure Use of Technology

Typically a remote worker is furnished with a computing environment that replicates the office computing environment; however the information threat and risk model has changed as the remote work environment presents a different set of security risks (Deloitte, 2011). When implementing a remote working policy it is not uncommon for an organisation to fail to conduct a technical threat and risk assessment (Sturgeon, 1996; Ampomah et al., 2013; Jilani et al., 2013) for the remote work environment. Such an omission can lead to security risks through technology vulnerabilities that were not present in the corporate environment. If the computer technology used in the more physically and logically secure office is utilised in the remote work location without specific configuration changes and/or augmented security mechanisms to account for the different threat profile (of the remote work location) vulnerabilities may be exploited.

***Computing environment not configured nor secured for remote location:*** Some organisations allow remote working (particularly teleworking) to occur without specifically applying a security configuration to the remote worker's PC and/or without using security enforcing technology (NIST, 2007). A security configuration will reduce inherent vulnerabilities by hardening the computing environment (i.e. disabling and or removing insecure functionality) (BerkeleySecurity, 2014). Utilising security technology (e.g. anti-malware software, disk encryption, small office unified threat management (UTM) firewalls) will reduce the exposure to external and internal attacks (NIST, 2007). Lack of a security configuration or not using security technology can occur when a PC is furnished by the remote worker (Sternstein, 2007), or if a corporate PC configured for use in the secure corporate office is used without any changes (Hoogendijk, 2006). Lack of hardening and/or use of security enforcing technology can leave the PC vulnerable as the user may perform actions that inadvertently expose vulnerabilities (Furnell, 2006; Hoogendijk, 2006; Sternstein, 2007). The NIST has defined best practice guidelines to secure the remote work computing environment (NIST, 2007; NIST, 2009).

***Threat Definition:*** The PC's execution environment is neither configured nor secured for remote work location.

*Secure PESE requirements:*

- A secure PEE shall only have the applications required to securely perform the remote work.
- The operating system and all applications forming the secure PEE shall be hardened to remove or disable insecure functionality.
- A secure PESE shall encrypt its whole storage medium to protect stored data. Appropriate anti-malware and firewall software shall form part of the secure PEE.

**Insecure Remote Location Wireless Networks:** A wireless (Wi-Fi) local area network may be used by teleworkers and deployed workers. If the Wi-Fi node is employing no or weak encryption (e.g. WEP (Ciampa, 2007)) then it is possible that an unauthorised user will be able to monitor the network traffic and gain access to sensitive data (Furnell, 2009; Ciampa, 2007). For mobile workers using cellular networks and Wi-Fi hotspots all traffic will be unencrypted and therefore sensitive data should not be processed.

*Threat Definition:* No (or weak) Wi-Fi\_\_\_33 encryption is enabled.

*Secure PESE requirements:* The secure PEE shall support the strongest available wireless cryptographic network protocols.

**Failure to backup computing environment:** Regular backups are an essential aspect of any security management policy (ISO/IEC 27001:2013, 2013). Failure to backup/image the remote worker's computing environment can significantly delay recovery if the computing environment becomes accidentally or deliberately corrupted (Crossler et al., 2013).

*Threat Definition:* No backup of the remote work PC execution environment is performed.

*Secure PESE requirements:*

- A secure PEE shall be a hardened, locked down image protected by a read-only partition on the secure PESE.
- A secure PEE image shall be protected within a secure PESE read-only partition to protect the image's integrity.
- A corrupted secure PEE shall be replaceable through re-imaging the secure PESE with a new copy of secure PEE.

***Remote worker unable to interpret, respond and/or configure security applications:*** If an organisation has implemented security technology within the computing environment it may have to rely upon a remote worker to interpret and respond to a security event if the application cannot be managed over a network (by corporate ICT personnel). Such a response may require the remote worker to reconfigure the computing environment. Research has shown that the user may not be able or may be recalcitrant in interpreting and responding to security events identified by security applications (Furnell and Clarke, 2012).

*Threat Definitions:*

- Remote worker lacks experience or inclination to respond to security event.
- Remote worker lacks experience to reconfigure PCs security features.

*Secure PESE requirements:*

- A secure PEE shall have a simple user interface (to all applications including security applications).
- Appropriate anti-malware and firewall software shall form part of the secure PEE.
- A secure PEE shall be designed to be a network client through the provision of a secure hardened browser, virtual private network or secure hardened remote login client all implementing network encryption (and therefore only limited security applications like anti-malware and firewall are required to form part of the secure PEE image).

***Insecure storage of sensitive corporate data in the remote work location:*** Sensitive corporate information processed by the remote worker may be stored locally on the PC disk drive. The local storage of corporate information may be an authorised action or could occur inadvertently or deliberately if the computing environment is not configured to prevent storage (LPS, 2008). If the PC disk drive is not protected by access controls and encryption then unauthorised access to sensitive information could occur if the PC is accessed by a user other than the remote worker or the PC is lost or stolen (Riswadkar and Riswadkar, 2009).

*Threat Definition:* Sensitive data is stored on unprotected/insecure storage media.

*Secure PESE requirements:*

- A secure PESE shall encrypt its whole storage medium to protect stored data.
- A secure PESE shall implement pre and post boot authentication, and access controls (which will prevent unauthorised access to data stored on the secure PESE).

***Use of computing environment for non-work activities:*** With the PC allocated for remote work residing outside the physical control of the organisation's ICT management it could be used by the remote worker (or the worker's family/friends) for non-work activities (Whiteman and Dick, 2006). Such activities may result in the introduction of malware affecting the integrity and availability of the PC's execution environment (Antonopoulos, 2007).

***Threat Definition:*** Remote worker uses PC for non-work activities and inadvertently affects the integrity and/or availability of the PC's execution environment.

***Secure PESE requirements:*** A secure PESE and its secure PEE shall prevent all access to the PC disk drive (it will not matter if the remote work PC is subsequently used for non-work activities and becomes infected by malware as all remote work will be performed using the secure PESE; any malware on the PC disk drive will have no effect on the secure PESE as the PC's disk drive is inaccessible).

***Securing data during transport:*** A remote worker may need to transport sensitive data to/from the remote work location from/to the corporate office. If the data is stored on unprotected storage medium a breach of data confidentiality could occur if the storage medium is lost or stolen (Furnell, 2009; Riswadkar and Riswadkar, 2009; Deloitte, 2011).

***Threat Definition:*** Sensitive data is transported on unprotected storage media.

*Secure PESE countermeasures:*

- A secure PESE shall implement pre and post boot authentication, and access controls.
- A secure PESE shall encrypt its whole storage medium to protect stored information (if the device is lost or stolen during transport).

#### **2.3.6.6 Technology Vulnerabilities.**

**Internet security:** With few exceptions, teleworking and mobile working are conducted over the Internet and therefore the risk of data loss over the Internet is ever present (Whiteman and Dick, 2006). Client server applications (3Com, 1999; David, 2002) and particularly voice over IP telephone systems may be implemented without encryption (James and Woodward, 2007) capabilities allowing sensitive information to be intercepted/eavesdropped on the Internet.

*Threat Definition:* Applications lacking strong network encryption are used to transmit sensitive data over the Internet leading to possible breach of confidentiality.

*Secure PESE requirements:* A secure PEE shall be designed to be a network client through the provision of a secure hardened browser, virtual private network or secure hardened remote login client all implementing network encryption (to prevent the interception of information).

**Embedded malware through successful attack:** A successful targeted attack upon a remote work PC may result in embedded malware impacting the integrity of the execution environment. This malware may affect the availability of the execution environment and/or the confidentiality of stored data.

*Threat Definition:* A targeted attack introduces malware into the execution environment.

*Secure PESE requirements:*

- A secure PEE image shall be protected within a secure PESE read-only partition to protect the image's integrity.
- Appropriate anti-malware and firewall software shall form part of the secure PEE.
- A secure PEE shall be a hardened locked down image on the secure PESE.
- A corrupted secure PEE shall be replaceable through re-imaging the secure PESE with a new copy of secure PEE.

**Storage of temporary data:** As described in Part 1, operating systems and applications store temporary data in hidden directories that can remain on the PC disk drive after data processing. These data remnants may contain sensitive data that could be retrieved

(Jones and Valli, 2011) by an unauthorised user if the PC is lost or stolen or if the PC is not sanitised prior to disposal (Jones et al., 2008).

*Threat Definitions:*

- Disk sanitation is not performed prior to PC disposal.
- Unauthorised user gains access to lost or stolen PC and is able to retrieve sensitive data remnants.

*Secure PESE requirements:*

- A secure PESE shall have a partition for the storage of temporary data.
- A secure PEE's operating system and applications shall be configured to store all temporary data on the secure PESE (ensuring no data remnants can reside on the host PC's disk drive).

**Functionally rich computing environment:** The computing environment used for remote work may contain applications that are either part of a standard distribution or are personal applications that have no relevance to the remote work (particularly where a bring your own PC policy for remote work is implemented). The greater the number of applications available the higher the risk that vulnerabilities can be exploited if the applications are used (LPS, 2008).

*Threat Definition:* The remote work PC's execution environment contains applications not required for remote work that if used may contain exploitable vulnerabilities.

*Secure PESE requirements:*

- A secure PEE shall only have the applications required to securely perform the remote work.
- The operating system and all applications forming the secure PEE shall be hardened to remove or disable insecure functionality.

**Hardware keyboard logger:** An attacker may be able to connect a keyboard logging device to the PC and conceal it from the remote worker. This device is able to capture data entered on the keyboard which can be retrieved by the attacker.

*Threat Definition:* Hardware-based keyboard logger compromises remote work PC.

*Secure PESE requirements:* There is no countermeasure that a secure PESE can feasibly provide to mitigate a hardware-based keyboard logger. This threat is therefore considered out of scope.

***Compromised PC firmware:*** An attacker may be able to embed malware into a PC's firmware (i.e. the Basic Input Output System (BIOS) (Fisher, 2014) or the Unified Extensible Firmware Interface (UEFI) (Anthony, 2011). Whilst a BIOS is true to its name and provides basic functionality, which limits the attack opportunities, the UEFI is essentially a small operating system and therefore has a greater attack surface.

*Threat Definition:* Remote work PC's firmware is compromised.

*Secure PESE requirements:* There is no countermeasure that a secure PESE can feasibly provide to mitigate a PC's compromised firmware. This threat is therefore considered out of scope.

Using these identified threats a secure PESE threat model is presented in Figure 4.1 (Chapter 4).

### **2.3.7 Synopsis of Analysis**

The literature review has:

1. Introduced the initial concept of a secure PESE and described the construction and assessment of four experimental secure PESEs.
2. Considered the prior research into secure remote working and categorised the research areas.
3. Identified 'secure PESE like' artifacts and products were assessed (using the initial concept attributes) for their suitability to be used for secure remote working.
4. Identified the complete set of security issues associated with remote working and for each identified issue a threat definition(s) is given together with one or more requirements for secure PESE functionality to counter the threat.

Whilst products like the Bull globull and the MXI Security Stealth MXP did provide capabilities that satisfied a number of the initial concept attributes none of the identified artifacts/products provided a fully conformant solution; highlighting a need for the design and development of enhanced secure PESEs. The review of the remote work security issues identified both threats and functional requirements not addressed by the initial concept. Although the initial concept provided a good foundation to commence the doctoral research a well-defined secure PESE concept was required that was supported by fully conformant instantiations.

## 2.4 The Research Problem

### 2.4.1 Establishing the Secure PESE Concept

The initial secure PESE concept was imbalanced with some attributes being abstract and others being implementation-oriented. The set of functional requirements identified in the review of security issues provide an implementation-oriented definition for a secure PESE. A secure PESE concept is therefore required that is underpinned by constructs that allow an abstract model to be defined with symmetrical attributes, and that is complete and consistent with respect to the security problem it addresses.

The secure PESE concept evolved as the research progressed and was finalised following artifact design<sup>24</sup>. As prescriptive knowledge was developed the initial concept attributes were refined allowing the following constructs<sup>25</sup> of a secure PESE concept to be identified: “preventing unauthorised access”, “preserve confidentiality”, “preserve integrity”, “availability”, “portability”, and “execution environment”. Using these constructs a set of abstract attributes (presented below) were defined for the (finalised) secure PESE concept; for each attribute the associated secure PESE functional requirements<sup>26</sup> are given to provide perspective and enable an understanding of what the abstraction represents:

- ***Prevent unauthorised access to the execution environment and any stored data.***
  - A secure PESE shall implement pre and post boot authentication, and access controls.

---

<sup>24</sup> The finalised concept is presented in Chapter 2 as its origins were substantially established in the first design cycle.

<sup>25</sup> Constructs are the vocabulary or notation used to define the basic entities of a concept.

<sup>26</sup> The functional requirements represent just one possible specification of a secure PESE.



- ***Preserve the integrity and availability of the execution environment.***
  - A secure PEE image shall be protected within a secure PESE read-only partition to protect the image's integrity.
  - Appropriate anti-malware and firewall software shall form part of the secure PEE.
  - A secure PEE shall be a hardened, locked down image on the secure PESE.
  - A corrupted secure PEE shall be replaceable through re-imaging the secure PESE with a new copy of secure PEE.
- ***Preserve the confidentiality and integrity of any stored data.***
  - A secure PESE shall encrypt its whole storage medium to protect stored data.
  - The secure PEE shall support the strongest available wireless cryptographic network protocols.
  - *A secure PESE shall implement anti-tamper mechanisms to render the device inoperable if attempts are made to tamper with it.*
- ***Preserve the confidentiality and integrity of any data sent to/from the remote location.***
  - A secure PEE shall be designed to be a network client through the provision of a secure hardened browser, virtual private network or secure hardened remote login client all implementing network encryption.
- ***Provide a highly portable device with a user friendly execution environment that can be used on any available PC.***
  - A secure PESE shall have a small highly portable form factor.
  - A secure PESE shall operate with any X86 PC.
  - A secure PEE shall have a simple user interface.
- ***Provide an execution environment with only the necessary secured functionality for the specific remote work activities.***
  - A secure PEE shall only have the applications required to securely perform the remote work.

- *The operating system and all applications forming the secure PEE shall be hardened to remove or disable insecure functionality.*
- **Limit the execution environment's access to the internal storage device(s) of the host PC.**
  - A secure PESE and its secure PEE shall prevent all access to the PC disk drive.
  - *A secure PESE shall have a partition for the storage of temporary data.*
  - *A secure PEE's operating system and applications shall be configured to store all temporary data on the secure PESE.*

These attributes can be used:

1. To build (or qualify an existing device as) a secure PESE
2. As principles for designing a secure remote work computing solution.
3. To define a conceptual security design model for secure portable computing.
4. As a basis to design a security model for remote working.

A secure PESE can be produced (and currently exists in products like the Bull globull) that partially conforms to the concept attributes and satisfy many of the functional requirements. However, three of the attributes reflect four functional requirements (highlighted using blue italics) that have not been addressed by existing products.

#### **2.4.2 Research Gap**

The following four secure PESE functional requirements represent a knowledge gap and an opportunity to develop innovative research artifacts.

- A secure PESE shall implement anti-tamper mechanisms to render the device inoperable if attempts are made to tamper with it.
- The operating system and all applications forming the secure PEE shall be hardened to remove or disable insecure functionality.
- A secure PESE shall have a partition for the storage of temporary data.
- A secure PEE's operating system and applications shall be configured to store all temporary data on the secure PESE.

### 2.4.3 Research Problem Definition

As defined in Chapter 1 (and stated again here to support the research problem definition), a secure PESE is as a highly portable, interoperable, secure computing device that combines a user friendly, limited functionality, execution environment together with secure storage. Upon successful authentication a secure PESE connected to a host PC will upload the execution environment and allow access to stored data. A secure PESE prevents unauthorised access to both the execution environment and stored data, preserves the confidentiality, integrity and availability of data and the execution environment, and limits access to the PC's internal disk drive. A secure PESE protects information by managing the risks of cyber-attack, data loss and forensic data discovery in the remote work environment. In Chapter 4 a conceptual design model (Figure 4.2) and an operational model (Figure 4.3) are presented. The underlying philosophy of the secure PESE is that it extends the corporate office security boundary out to the remote work location (Peacey, 2006).

The research seeks to expand the capabilities of technology available to implement an enhanced secure PESE. The research problem for this doctoral research is therefore:

**A requirement exists to develop an enhanced secure PESE that limits the exploitation of vulnerabilities by hardening the execution environment, providing a tamper detection and response capability and ensuring no data remnants are recoverable from the host PC.** These enhancements will further strengthen the secure PESE against the risks of cyber-attack, data loss and forensic data discovery in the remote work environment.

### 2.4.4 Research Objectives

The research methodology selected for this doctoral research nominates the definition of research objectives as an early research activity (Peppers et al., 2007). Defining objectives allows the researcher to specify the desired outcomes of the research. The refined and revised (i.e. finalised) secure PESE concept attributes translate readily into research objectives. Input from stakeholders identified two further useability and performance objectives which were more implementation-oriented than the seven concept attributes. The full set of research objectives are:

- Prevent unauthorised access to a secure execution environment and any stored data.
- Preserve the integrity and availability of the execution environment.
- Preserve the confidentiality and integrity of any stored data.
- Preserve the confidentiality and integrity of any data sent to/from the remote location.
- Provide a highly portable device with a user friendly execution environment that can be used on any available PC.
- Provide an execution environment with only the necessary secured functionality for the specific remote work activities.
- Limit the execution environment's access to the internal storage device(s) of the host PC.
- The execution environment shall have a look and feel similar to Microsoft Windows.
- The execution environment shall have similar performance to a PC operating system executing from an internal disk drive.

Artifact design was directed through a tailoring of the above objectives and the secure PESE functional requirements.

#### **2.4.5 Research Questions**

An analysis of the research gap highlighted a number of unknowns necessitating the following research questions:

- *How can a useable and maintainable hardened operating system and/or a small set of hardened applications be developed?* Hardening an operating system and/or applications involves removing/disabling access to functionality, particularly privileged functionality and as a result could impact both useability and maintainability. It is important that the hardening of software does not render it difficult to maintain nor limit its usefulness.
- *How can anti-tamper mechanisms be implemented into a small form factor and highly portable device?* An anti-tamper system is achieved by protecting security critical functionality from any access. Typically a tamper boundary is created that if

penetrated causes a tamper detection system to render the device inoperable. The requirement for a secure PESE to be highly portable with a small form factor makes the anti-tamper design a greater challenge.

- *How can a useable and maintainable execution environment be configured to store all temporary data on a secure PESE partition?* It is important that temporary data created by the execution environment is not written to the PC's disk drive as it may be recovered through the application of forensic data discovery applications used by an unauthorised user. To address this question the execution environment must be changed or configured to write temporary data to a secure PESE partition, such changes may impact usability and maintainability.

## **2.5 Summary**

This chapter has described how a research gap was identified was defined using a design approach which resulted in the generation of artifacts and prescriptive knowledge, that together with descriptive knowledge identified in a literature review enabled the research problem, objectives and questions to be defined. The design research described in this chapter forms the first design cycle and generated the following knowledge:

- A conceptual design of a security module for a smartphone.
- A toolkit for a secure portable disk drive that prevents data remnants.
- The identification of the attributes for a teleworking security model.
- A set of experimental secure PESEs.
- A comprehensive description of the remote work security issues and the respective threats.
- Functional requirements for a secure PESE.
- A secure PESE concept.

The first design cycle provides a knowledge baseline for design cycle 2. A problem and questions have been defined to direct the research to confirm that 'secure PESEs can be used to manage information security risks within the remote work environment'.

## 3 Research Design

### 3.1 Overview

Research design is a plan defining how the research will be conducted. It provides a framework and theoretical basis for the research (Creswell, 2009; Explorable, 2013; Jalil, 2013). A research design identifies the research paradigm, the methodology to be used, how the research outcomes are generated, demonstrated and evaluated, and how the contribution to knowledge and theory are presented.

A research paradigm can be defined as a collection of concepts and beliefs that guide research actions (Explorable, 2013; Creswell, 2009). The two frequently used research paradigms for IS/IT research are positivist (based upon quantitative research actions) and interpretive (based upon qualitative research actions), but other research paradigms suitable for IS/IT research have evolved including design science research (Vaishnavi and Kuechler, 2014).

The research methodology consists of a set of activities that are followed to address the research problem and validate the research argument (Palvia et al, 2003; Avison & Fitzgerald, 2006). The methodology directs how the research is conducted, the data is collected, the argument is tested and the analysis of the research outcomes is performed. It is important that the appropriate research methodology is selected to provide the evidence and results necessary to address the research problem and deliver the desired outcomes to satisfy the research argument.

The paradigm selected for the research presented in this thesis is design science research (DSR) (DESRIST, 2014) and the specific design science methodology is the Design Science Research Methodology (DSRM) (Peffers et al., 2007). The DSRM has six process elements: identify problem, define objectives, design and develop, demonstrate, evaluate and communicate. In this doctoral research the research problem and objectives are specified following a set of investigative design activities (forming a design cycle) together with a literature review which includes an analysis of the remote work security issues. The research is conducted in three further cycles resulting in the design and development of a set of artifacts. The artifacts are subjected to testing, trialling, certification and commercialisation to demonstrate the correctness of their design. Evaluation of the

research outcomes involves analysing the research findings, identifying how the body of knowledge generated makes a contribution to the area of study and providing an academic explanation of the research. This thesis represents the research communication.

In this chapter the alternative research paradigm and methodology considered is also summarised and the rationale for not selecting it is given before an overview of the selected DSR paradigm and the DSRM methodology is presented. The four design cycles used to conduct the research are described. The demonstration approach is discussed and the chapter concludes by explaining the evaluation approach used to interpret the research findings and identify a knowledge and theory contribution.

## **3.2 Rationale for Selecting Design Science Research**

### **3.2.1 Choosing the Research Paradigm and Methodology**

Both interpretive (qualitative) and DSR were considered as suitable paradigms for the research. In IS/IT research an interpretive approach is typically concerned with the study of user behaviour with a system or technology (Myers and Avison, 1997). An interpretive approach was initially considered as it could be used to construct secure PESEs and then study user views and preferences with respect to the implemented security mechanisms and their ability to counter information security risks.

If the interpretive paradigm had been selected then the action research methodology would have been used. Action research (Chiasson et al., 2008; Antill, 1986; Baskerville et al., 1998) is a methodology based upon a cyclic approach where each cycle involves a change action and an observation of the impact of the change. Further cycles of action/observe are performed until the research problem is addressed and the research hypothesis validated. Action research was considered a possible methodology as it could be used to create a secure PESE from existing technology components (i.e. the action is modifying technology components to create a secure PESE) and observing the suitability of the secure PESE to manage information security risks.

Although an interpretive research paradigm coupled with the action research methodology may have provided a suitable framework for the research it was decided that DSR was more appropriate because:

1. It is important when using an interpretive paradigm and qualitative methodology that the largest possible sample of users (experimental group) is available for the research. It may have proved difficult to recruit suitable representative experimental groups (organisations and individuals) that were agreeable to being subjected to information security research. Typically organisations are not prepared to be the subject of an experiment to determine its exposure to cyber-attack, data loss and forensic data discovery. With the possibility of a limited experimental group the qualitative research results may not have been sufficiently rigorous.
2. For a PhD being conducted part-time the numerous social interactions occurring from the multiple change and observe action research cycles (required to gauge the effectiveness of a secure PESE) may have proved both difficult to manage and to finalise in a satisfactory time.
3. An interpretive action research approach may have directed the research towards the development of secure PESEs that address user concerns rather than focussing upon innovative security design research. Whilst user focussed security research is important the researcher was particularly interested in design focussed research.
4. DSR enabled the development environment the researcher has established for his employer to be used effectively to develop artifacts whilst still using the DSR research paradigm to guide the research.

### **3.2.2 Selection of DSR**

DSR (Hevner et al, 2004; McKay & Marshall, 2005; Nunamaker et al., 1990; Peffers et al., 2007) was selected as it was specifically conceived as a design and development research paradigm that can be used effectively in IS/IT research. DSR is concerned with the design and evaluation of artifacts that address a specific problem. An artifact can range from an IT framework/methodology through a design to a prototype product (Hevner et al, 2004).

DSR has become an established IS/IT research methodology with both a strong academic community advocating its use (DESRIST, 2014) and an international conference dedicated



to the methodology now in its ninth year (DESRIST-2014, 2014). The paradigm has been used to develop artifacts across the whole spectrum of IS/IT research, interesting examples showing the breadth of its application include:

- A conceptual data model for complex resource requirements (Ouyang et al., 2010).
- A digital forensic process model (Adams, 2013).
- A method to assess competitiveness gained from IS capabilities (McLaren et al., 2011).
- A network security visualisation tool (Luse et al., 2011).

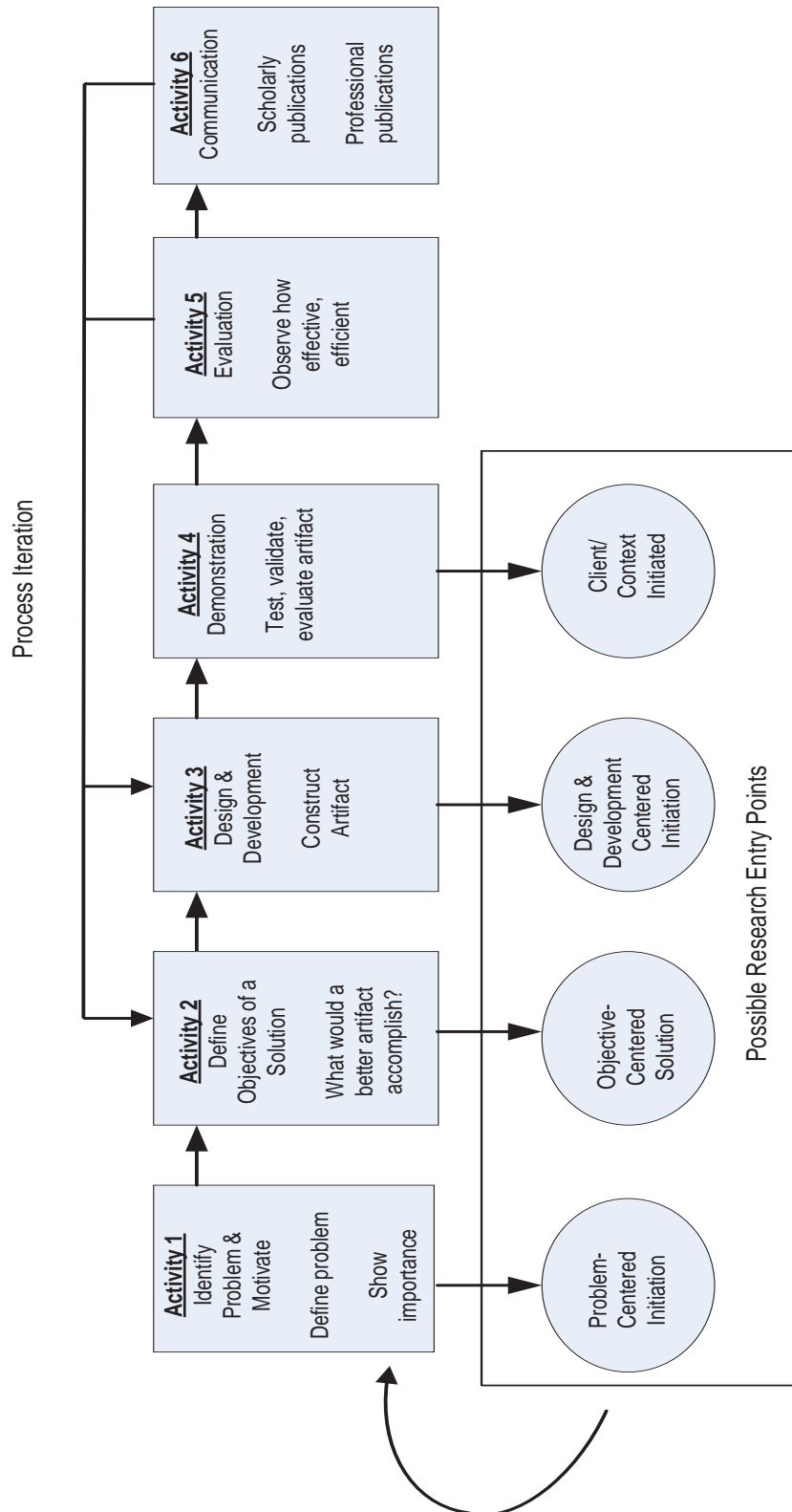
As a relatively new paradigm in IS/IT research DSR is itself the subject of on-going research (McKay and Marshall, 2007; Venable, 2006; Gregor and Jones, 2007; Gregor and Hevner, 2013). A definitive summary of the research into DSR for IS, particularly with regard to the philosophical and theoretical basis of the paradigm is provided in a periodically updated paper (Vaishnavi and Kuechler, 2014) available on the DSR web site (DESRIST, 2014).

### **3.3 Design Science Research Methodology**

#### **3.3.1 Overview of Methodology**

A number of DSR methodologies (process models) have evolved that are targeted at IS/IT research (Nunamaker et al., 1990; Hevner et al., 2004; Peffers et al., 2007). The design science methodology selected for the research is DSRM (Peffers et al, 2007); a popular methodology with over 100 citations identified in the Association of Computing Machinery digital library (ACM-DL, 2014). The methodology defines a progressive series of activities (also known as process elements) that provide a structured approach to address a specific IS/IT research problem and allow validation of a research argument. This DSRM process model consists of the following six process elements (activities) with the model presented diagrammatically in Figure 3.1.

*Activity 1: **Problem identification and motivation*** requires the encapsulation and presentation of the research problem for which an artifact is required and an explanation of why there is a need to find a solution to the problem.



**Figure 3.1 – DSRM Process Model (Peppers et al., 2007)**  
The model depicts each research activity, the possible research entry point and where iteration can occur.

*Activity 2: **Defining objectives*** for the artifact allows the researcher to specify the desired outcome(s) of the research.

*Activity 3: **The design and development*** process element involves the specification, design and implementation of an artifact that satisfies the defined objectives.

*Activity 4: **The demonstration*** of the artifact requires the test, trial or verification of its capabilities or attributes.

*Activity 5: **Evaluating*** the artifact involves assessing the observations/measurements collected during the demonstration to determine the effectiveness of the artifact with respect to addressing the defined problem and satisfying the objectives. The evaluation will identify the contribution to knowledge and theory. The evaluation may determine iteration is required back to either activity 3 to redevelop an aspect of the artifact and/or activity 2 to refine an objective.

*Activity 6: **The communication*** process element of the DSRM requires the publication of the research and the respective outcomes to relevant audiences. The publication may identify future work requiring iteration back to activities 2 and/or 3.

The DSRM process model allows research to commence at different entry points as shown in Figure 3.1. The entry point for the research presented in this thesis was ‘problem centred’. The structure of this thesis follows the sequential progression of the DSRM activities as Chapter 2 presents the research problem (activity 1) and objectives (activity 2), Chapter 4 presents the design and development (activity 3), Chapter 5 presents the demonstration (activities 4) and Chapter 6 presents the evaluation (activity 5). The thesis itself is the research communication (activity 6). This approach to structuring a DSR thesis is consistent with recent research on presenting DSR research (Gregor and Hevner, 2013).

DSRM could be considered to be similar to an IT system design methodology (Microsoft, 2013) used for routine system design rather than a methodology used to direct IT/IS research. However, the distinguishing feature is that the DSRM is concerned with addressing unsolved problems or making an existing artifact more efficient and effective (Hevner et al., 2004) whereas routine design applies existing knowledge to the design of a solution. The doctoral research utilised the DSRM within a commercial environment.

### 3.3.2 Development Environment

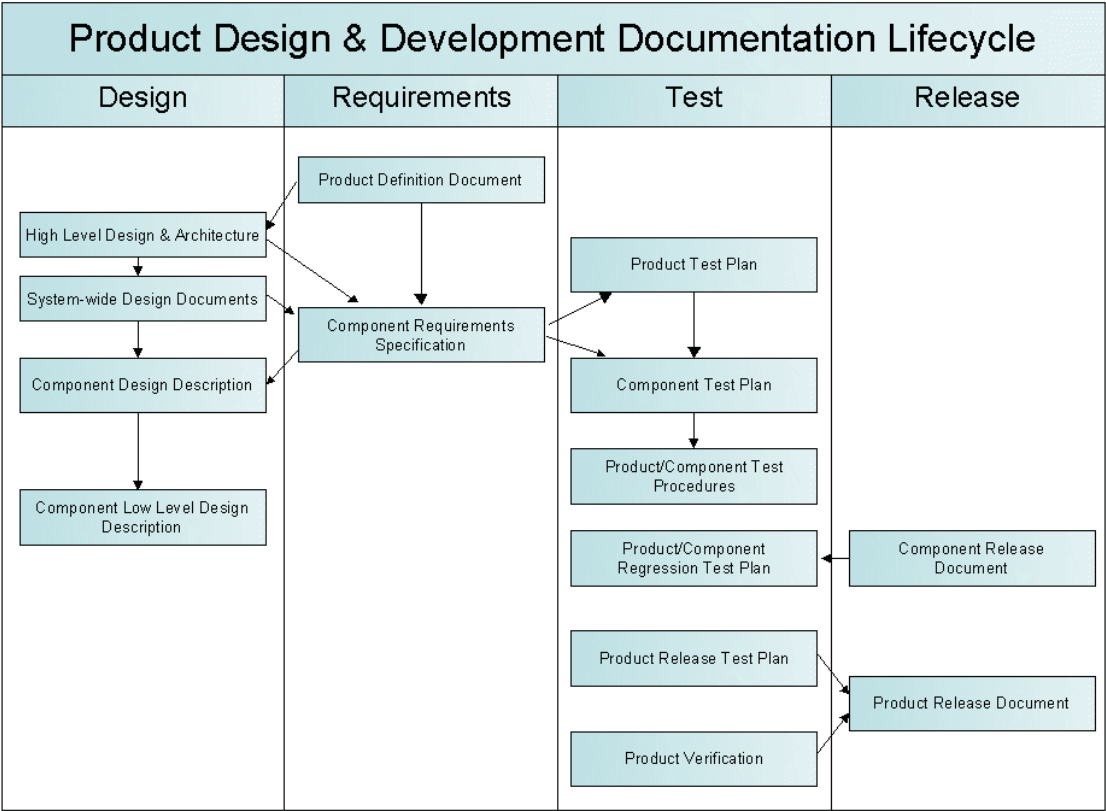
The researcher has designed and implemented a structured and rigorous technology and product development environment for his employer (SSL-WP-17, 2006) based upon the Waterfall model (SDLC-W, 2014). This development environment was used effectively with the DSRM during the research. The development environment allows for rigorous development of design (including constructs, models and methods/algorithms) and software/hardware technology (i.e. instantiations) components. Each component can be maintained as a separate configuration item that can then be used for product development. The development environment provided the necessary support for a research project using DSR. Important features of the development environment include:

- It is based upon a traditional Waterfall model, i.e. development stages consisting of: requirements, architecture and high-level design, detailed design, implementation, test and release. However, the environment is structured to be fit for purpose with the ability to enter and exit different phases depending upon the work being performed, e.g. a prototype may have requirements defined and a high-level design prepared, but then implementation is performed without detailed design.
- Separation of design/development and test functions to ensure totally independent testing. Testing is performed hierarchically based upon the V model (SDLC-V, 2014), (i.e. system testing confirms correct implementation of the requirements, integration testing confirms correct implementation of the architecture and high-level design, etc.). However, testing can also be tailored to meet the needs for the rapid development of a product or the development of a research artifact/initial prototype product.
- A structured peer review process is applied to all design activities based upon a formal review meeting. The review meeting consists of relevant stakeholders attending with agreed minutes taken to correct/improve the review item. After the meeting the creator of the review item corrects/improves the item according to the minutes with a nominated reviewer (from the meeting) confirming the minutes have been correctly applied (after the creator has fully addressed the actions in the minutes). In addition to enforcing correctness and quality the review process contributed to innovation as it

allowed reviewers to offer suggestions to improve/enhance the review item. The review process also provided a mechanism to enable iteration in the DSRM.

- A set of design and implementation guidelines that are not prescriptive and hence allows for flexibility in the design process.
- An integrated set of issue management, configuration management and test management systems that record and track all aspects of the design and development lifecycle.

Figure 3.2 below depicts the documentation outcomes from the development environment activities. The diagram shows the development stages and the inter-dependencies between the stages. The approach allows for comprehensive documentation to be prepared at all stages of the development lifecycle.



**Figure 3.2 – Model of Documentation Outputs from the Development Environment**

DSR and in particular the DSRM process elements provided a research framework that allowed the utilisation of the development environment. The process elements corresponded to design activities defined in the environment. The review process in particular provided both a mechanism to support the iteration feature of the DSRM and

allow for knowledge development. For design focussed research the use of a rigorous development environment enabled quality research results (i.e. the artifact test results) to be achieved that demonstrated the correctness of the design.

### **3.4 The Consumption and Production of Knowledge**

A successful PhD research project will both consume and produce knowledge (Standing, 2008). Consumed knowledge forms the knowledge baseline and in this doctoral research it is derived from literature analysis, existing theories, related previously developed artifacts, stakeholder discussions, market monitoring and the knowledge produced in an earlier design cycle. Consumed knowledge is the informing knowledge for the research.

The produced knowledge emerges from the artifacts' innovative design and may consist of constructs, models, methods, instantiations and design theory. Produced knowledge is the research's contribution to knowledge.

This thesis utilises the ideas presented in 'Positioning and Presenting Design Science Research for Maximum Impact' (Gregor and Hevner, 2013) to understand and categorise the consumed knowledge and to position and present new knowledge. Also the approach proposed in 'The Anatomy of a Design Theory' (Gregor and Jones, 2007) is used to nominate a design theory for secure PESEs.

#### **3.4.1 Knowledge Classes**

Gregor and Hevner classify knowledge consumed and produced as prescriptive and descriptive, however this thesis adopts a minor variation to their presentation style with knowledge consumed and produced categorised descriptive, prescriptive, formal justificatory and informal justificatory knowledge. Gregor and Hevner (Gregor and Hevner, 2013) strongly emphasise the importance of justificatory knowledge but position it as informing knowledge to the consumed and produced prescriptive knowledge. A definition of each knowledge class is given below.

***Descriptive Knowledge*** is the "what" knowledge and has two primary forms; phenomenon and its sense making relationships. Phenomena are composed of observations, classifications and measurements. The sense making relationships of phenomena are represented by natural laws, principles, regularities, patterns and

theories. A DSR project predominately consumes descriptive knowledge. Whilst a DSR project may acquire descriptive knowledge from various disciplines the DSR project itself may also create descriptive knowledge. This created descriptive knowledge emerges from the results captured during the demonstration and evaluation of an artifact (constructed during the research). This new descriptive knowledge can be used within the research project or in future research. Examples of descriptive knowledge used in this thesis are:

- Security risks (i.e. a sense making relationship arising from the phenomena of a threat and its likelihood); and
- Remote worker behaviour (e.g. a remote worker's adherence to security policy, i.e. the sense making relationship to security policy through the phenomena of adherence or lack of it).

Stakeholder discussions and the monitoring of market trends were also forms of descriptive knowledge as they represent phenomena in the form observations or requirements with the sense making relationship based upon experience. In particular descriptive knowledge was sourced from:

1. The researcher had access to a wide range of stakeholders from which to gather ideas and information including work colleagues, conference delegates, the reviewers of the published papers and customers (both existing and prospective) of the products that the researcher is responsible for developing.
2. Monitoring market trends for both security technology and how/where technology is used.

***Prescriptive Knowledge*** is the “how” knowledge of constructed artifacts consisting of constructs, models, methods and instantiations. Constructs are the vocabulary/symbols used to understand problems. Models are the conceptual representations of problems and possible solutions. Methods are the algorithms and specifications for a solution. Instantiations are the implementations (software/hardware) that embody design knowledge. Whilst a DSR project produces prescriptive knowledge it may also consume prescriptive knowledge. Examples of prescriptive knowledge created in this thesis are:

- A secure PESE concept (both of construct and model)

- Conceptual models to assist the secure PESE design (model)
- An anti-tamper mechanism (instantiation)
- The Mobile Execution Environment (an instantiation of a secure PEE).
- The high grade secure PESE (an instantiation of a secure PESE).

**Formal justificatory knowledge** is the underlying theories that explain the basis for an artifact's construction and "why" the design works. Justificatory knowledge may consist of theories from other disciplines. Examples of formal justificatory knowledge used in this thesis are:

- Secure system design theory and security engineering practice from established publications, e.g. Security Engineering by Anderson (Anderson, 2008).
- Cryptographic design theory from established publications, e.g. Cryptography Engineering: Design Principles and Practical Applications by Ferguson, Schneier and Kohno (Ferguson et al., 2011).

**Informal justificatory knowledge** is the knowledge of practitioners and other informal knowledge used in the research to explain "why" a design works. Examples of informal justificatory knowledge used in this thesis are:

- The researcher's knowledge of the SDV technology design and capabilities, and how it can be utilised in the construction of secure PESEs.

### 3.4.2 Design Cycles

The doctoral research is structured into four design cycles (Gregor and Hevner, 2013) which are modelled diagrammatically in Figure 3.3. Each cycle will consume and produce knowledge, with knowledge produced in an earlier cycle being available for consumption in a later cycle. The publication describing design cycles (Gregor and Hevner, 2013) was not available until the latter stages of the doctoral research, but the approach reflected how the research had been conducted and therefore provided an elegant structure for the thesis presentation.



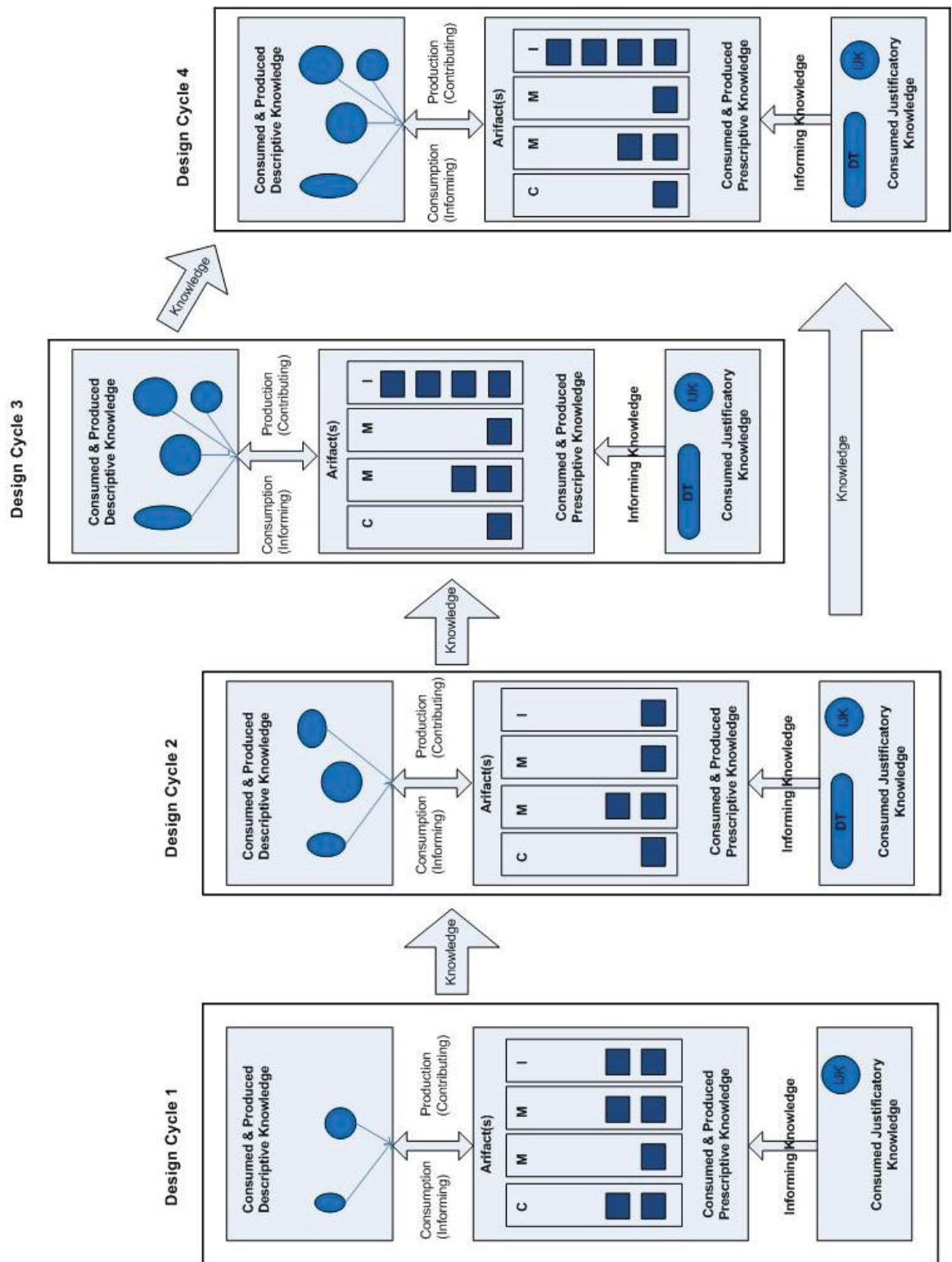


Figure 3.3 – Model of Development of Knowledge over Four Design Cycles

Figure 3.3 (based upon the Gregor and Hevner model) diagrammatically presents a conceptual model of the development of knowledge through the four design cycles that were applied during the research. The difference between Figure 3.3 and the Gregor and Hevner model (Gregor and Hevner, 2013) is that Figure 3.3 places greater emphasis upon highlighting the consumption of justificatory knowledge where it informs the design. The vertical two-way arrow in each design cycle represents the consumption and production of both descriptive and prescriptive knowledge during artifact development. The vertical one way arrow represents the use of justificatory knowledge to inform the design. The bi-directional arrow represents knowledge consumed in a design cycle that was produced in a preceding cycle. The circles and ovals represent the different types and growth in consumed and produced descriptive knowledge. The rectangle boxes labelled C, M, M and I (which correspond to construct, model, method and instantiation categories respectively) containing squares represent the type of artifact and prescriptive knowledge consumed and produced. A rectangle with no squares represents no prescriptive knowledge for that specific category was consumed or produced in the respective design cycle. The consumption of justificatory knowledge is only shown in each cycle where it informs the design. The rounded rectangle labelled DT represents design theory (or formal justificatory knowledge) and the oval labelled IJK represents informal justificatory knowledge.

As the model shows, both descriptive and prescriptive knowledge grows with each cycle. During the first design cycle knowledge is identified and created that is used to identify the research problem, objectives and questions. In the second design cycle the design focus is narrowed and produces conceptual design models to assist with the design of the artifact instantiations produced in the second and third cycles. Design cycles 2 and 3 consume knowledge from both cycles 1 and 2. Design cycle 4 also consumes knowledge from cycle 3. Design cycle 3 produces artifacts that are used to construct a commercial grade secure PESE artifact. Design cycle 4 produces an artifact and consumes artifacts from design cycle 3 to produce a high grade secure PESE artifact. The first two design cycles do not follow every activity of the DSRM whereas the final two do follow the DSRM rigorously.

### 3.4.3 Knowledge Consumed

The informing knowledge consumed during the research can be categorised into the following overarching categories:

- ***Security Issues in the Remote Work Environment (Descriptive Knowledge):*** The protection of sensitive information and maintaining the integrity of the computing environment are known security issues for an organisation implementing remote working (Bates, 2010; Sturgeon, 1996; NIST, 2009). These security issues are categorised as threats, vulnerabilities and risks. The threats and vulnerabilities occur due to physical, human and technological phenomena/events. A risk arises when a threat has a likelihood of occurring (ISO 31000:2009, 2009). This descriptive knowledge provides a contextual environment in which an artifact is required to operate securely.
- ***Empirical Results from Earlier Design Cycles (Descriptive Knowledge):*** The test and trial results from the artifacts produced in the early design cycles provide descriptive knowledge for subsequent artifact development.
- ***Stakeholder Input and Market Trends (Descriptive Knowledge):*** Contributions to the research from the stakeholder and the monitoring of market trends all provided a knowledge contribution.
- ***Existing Technology Suitable for Secure Portable Devices, Secure Storage and Open Source Systems and Applications (Prescriptive Knowledge):*** Knowledge created from prior novel artifacts (external to this research) was analysed and considered during the design of secure PESEs. In particular, the research draws upon design knowledge present in previous 'secure PESE like' artifacts, from existing security technology (e.g. SDV technology) and prior novel operating systems and browser instantiations.
- ***Design Knowledge from Earlier Design Cycles (Prescriptive Knowledge):*** The prescriptive knowledge produced through the creation of artifacts in earlier design cycles provides knowledge for artifact development in subsequent design cycles.
- ***Security Risk Assessment, Operating System, and Security and Cryptographic Engineering Design Theories (Formal Justificatory Knowledge):*** The established theories for the design of secure systems using proven security risk assessment

methodology, conceptual design practices, operating system design, and tried and tested security mechanisms and cryptographic primitives.

- ***Theory of Network Centric Warfare (Formal Justificatory Knowledge):*** A theory of network centric warfare (Fewell and Hazen, 2003) from the defence science discipline is used to gauge the suitability of a secure PESE to provide a capability at a remote deployed node within a network centric organisation.
- ***Practitioner Knowledge (Informal Justificatory Knowledge):*** The researcher and colleague's (paper co-authors and work colleagues) informing knowledge on how to use the SDV technology and how execution environments are designed and configured.

#### **3.4.4 Production of Knowledge**

The contributing knowledge developed during the research was accumulated over the four design cycles; although it is in the third and fourth design cycles that the tangible knowledge contribution occurs. The following knowledge was produced in each respective design cycle.

***Design Cycle 1 - Establishing the Research Problem and Objectives:*** In this cycle a conceptual secure smartphone design, a toolkit to prevent data remnants, the identification of attributes for a security model and the development of a set of experimental secure PESEs, provides the knowledge to define an initial secure PESE concept. A holistic review of the remote work security issues enables remote work threats and functional requirements for a secure PESE to be defined. Collectively the design cycle 1 knowledge outcomes facilitates the definition of a finalised secure PESE concept.

***Design Cycle 2 – Baselining the Design:*** Descriptive knowledge is produced that ascertains the vulnerabilities in virtualisation. A set of conceptual design models provides the prescriptive knowledge to assist in the design of secure PESE artifacts.

***Design Cycle 3 – Developing the Commercial Grade Secure PESE:*** In this cycle three artifacts (two secure PEEs and a storage device) are produced that are combined to construct a commercial grade secure PESE artifact.

**Design Cycle 4 – Developing the High Grade Secure PESE:** A high assurance portable storage artifact is produced that is combined with the two secure PEE artifacts constructed in design cycle 3 to enable a high grade secure PESE artifact to be produced.

### **3.5 Demonstration**

The artifacts developed in design cycles 3 and 4 were demonstrated (and described in Chapter 5) through extensive formalised independent testing, user trials, certification and commercialisation.

**Testing:** The artifact test environment enabled independent, systematic and exhaustive testing to be conducted. The research objectives were used to test propositions with test procedures prepared for each objective. The test procedures covered functional (with error trapping), exception, vulnerability, stress/soak, sociability, compatibility, environmental and useability testing. Testing utilised a test management toolset consisting of a test management system to manage and capture test results, and an issue management system to report both problems/bugs and suggested enhancements. The test environment enabled a structured, consistent and repeatable approach to be applied to the research artifacts and achieve the level of testing necessary to ensure the innovative design features (i.e. the new prescriptive knowledge) operated securely and correctly.

**Trials:** Two trials were conducted to demonstrate the artifacts, one with a commercial organisation and the other with a government agency. The trial at the commercial organisation involved the use of a secure PESE by teleworking technical support staff. Whilst the commercial organisation did not want its identity revealed it did agree that the researcher could use the results of the trial in this thesis. The trial at the government organisation involved the use of a secure PESE by deployed personnel working in a net-centric environment. Due to the sensitivity of the work performed the results of the trial cannot be published; permission was only given for an outline of the trial to be published. The trials confirmed the usefulness and effectiveness of a secure PESE for remote working.

**Certification:** The Mini SDV and the SDV-HA storage products were evaluated and certified for use by Australian Government agencies. The Mini SDV was certified to protect

sensitive data and the SDV-HA was subjected to a high assurance evaluation and was certified to protect highly classified data. Certification demonstrated the strength of the security mechanisms implemented in the SDV storage technology.

**Commercialisation:** The Mini SDV and the SDV-HA were commercialised as secure storage products with details provided describing how to configure the product as a secure PESE. To validate the knowledge contribution of the products certain commercialisation activities were selected and described to support the demonstration process. Activities were selected where an independent assessment was involved and confirmed the innovation and knowledge contribution of the research.

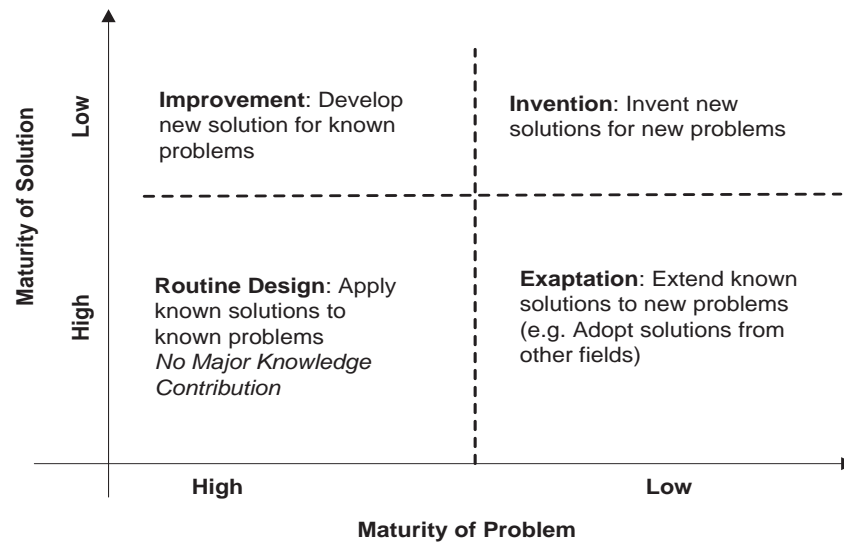
### 3.6 Evaluation

The artifacts developed in design cycle 3 and 4 are evaluated (and described in Chapter 6) by assessing the research findings, providing an academic explanation of the research and highlighting the contribution made (by the generated the body of knowledge) to the area of study.

The research findings were evaluated by assessing:

- The demonstration results.
- If the research questions were satisfactorily answered.
- If each of the nine research objectives is satisfied.
- If the research problem was addressed.

An academic explanation is achieved through the use of a knowledge contribution framework (Gregor and Hevner, 2013) and a structure for determining a design theory (Gregor and Jones, 2007).



**Figure 3.4 – Gregor and Hevner Knowledge Contribution Framework**

In addition to supporting an academic explanation the knowledge contribution framework is used to assert a contribution by the body of knowledge to the area of study. The framework presents three categories to define a knowledge contribution with a fourth category representing routine design which is not a contribution. Figure 3.4 presents and describes the categories forming the contribution framework. It is shown that the knowledge contribution is in the improvement category.

An anatomy of a design theory (Gregor and Jones, 2007) developed specifically for DSR is used to show that the research has resulted in a design theory for secure PESEs to support remote working. The anatomy has the following eight components:

1. **Purpose and Scope** requires the research to be well defined with a generalised definition allowing a range of artifacts to be built that conform to the definition.
2. **Constructs** define the basic entities (building blocks) of the theory.
3. **Principles of Form and Function** define the architecture of the artifact to which the theory applies.
4. **Artifact Mutability** is the level of design change an artifact can support that is encompassed in the theory.
5. **Testable Propositions** are the testable truth statements about the design theory.

6. **Justificatory Knowledge** is the underlying informing knowledge and/or theories that provide an explanation for the design. In this thesis justificatory knowledge is categorised as formal or informal (see definitions above).
7. **Principles of Implementation** define the processes to be used to implement the artifact and hence demonstrate the theory.
8. **Expository Instantiation** is an example implementation of an artifact representing the theory.

The anatomy postulates that if the research addresses the eight components it can be claimed that the research has made a contribution to IS design theory. Gregor and Hevner (Gregor and Hevner, 2013) have also defined a knowledge maturity model (shown in Figure 3.5) to gauge a research contribution type. The model defines three maturity levels ranging from the less abstract contribution type to the more abstract contribution type. It is shown that the research presented in this thesis is a level 2 contribution type, i.e. a nascent design theory underpinned by a secure PESE concept, design models and instantiations.

	Contribution Types	Example Artifacts
<p>More abstract, complete, and mature knowledge</p> <p>↑ ↑ ↑ ↑</p> <p>↓ ↓ ↓ ↓</p> <p>More specific, limited, and less mature knowledge</p>	Level 3. Well-developed design theory about embedded phenomena	Design theories (mid-range and grand theories)
	Level 2. Nascent design theory-knowledge as operational principles/architecture	Constructs, methods, models design principles, technological rules.
	Level 1. Situated implementation of artifact	Instantiations (software products or implemented processes)

**Figure 3.5 – DSR Knowledge Maturity Model**

### 3.7 Summary

In this chapter the paradigm and methodology used are described together with rationale for their selection. The generation of knowledge through design cycles and the methods used to demonstrate and evaluate the research were enumerated. The framework and anatomy selected to present knowledge and design theory contributions were described. A research design has been presented that allows for the design of novel artifacts to demonstrate that secure PESEs can be used to manage information security risks within the remote work environment.



## **4 Design and Development**

### **4.1 Overview**

#### **4.1.1 Structure**

To satisfy the research problem and address the research questions four novel research artifacts were designed and constructed which were used to build both a commercial grade and a high grade secure PESE. Five published papers are presented in this chapter. Like the papers presented in Chapter 2 each paper is preceded with a background discussion, identification of the research questions addressed and an enumeration of the knowledge consumed. A post paper discussion considers the outcomes and knowledge contribution, the paper's contemporary relevance, its linkage with other papers, the future direction for the research and whether the research question was addressed. This chapter is structured in accordance with the three further cycles of design (modelled diagrammatically in Figure 3.3) conducted in this DSR project, namely:

- Design Cycle 2: In this second cycle the use of virtualisation for secure PEEs is analysed and deemed insufficiently secure and the secure PESE concept is modelled diagrammatically.
- Design Cycle 3: In the third design cycle the construction of a commercial grade secure PESE is considered. A hardened secure browser, a hardened secure operating system (containing a small set of applications) and a secure storage device were the artifacts developed to enable the construction of a commercial grade secure PESE.
- Design Cycle 4: A high assurance storage device is the artifact developed in the fourth design cycle. A high grade secure PESE is then constructed using the high assurance storage device, and the browser and operating system developed in design cycle 3. A review of the use of a secure PESE within a net-centric organisation is considered before discussing the design of a high grade secure PESE.

#### **4.1.2 The Papers**

The five papers presented in this chapter (Papers 5 to 9 inclusive) document the research conducted to scope and set the direction, and the design and development of the artifacts. Given the potential security vulnerabilities in virtual machine based secure PEEs

(identified in Chapter 2), a comprehensive analysis of virtualisation is presented in paper 5. As a result of the analysis the focus for secure PEE research was directed towards the use of either a set of hardened applications (that execute upon the host PC operating system) or a bootable operating system with a limited hardened application set installed. Two secure PEE artifacts were developed; a hardened browser presented and discussed in Paper 6 and a hardened bootable operating system presented and discussed in Paper 7. To position the requirement for a high grade secure PESE, the use of such a device by deployed workers at the remote nodes of a net-centric organisation is considered and discussed in Paper 8. Paper 9 then discusses the development of a high grade secure PESE and how it improves security for remote workers in a net-centric organisation.

## **4.2 Baselineing the Design - Design Cycle 2**

This design cycle examines the vulnerabilities of virtual machine based secure PEEs and as a result of a comprehensive analysis virtualisation was discounted. Design cycle 2 also models diagrammatically the concept of the secure PESE through: a threat model showing how the concept is designed to manage the information security threats in the remote work environment; a conceptual design model providing a generic architecture for the secure PESE; and an operational model showing the process of security enforcement starting from the first action of connecting a secure PESE to a PC. Collectively, the three models convey the security properties of the secure PESE concept and provide a reference model for the artifact design in cycles 3 and 4.

### **4.2.1 Discounting the use of Virtualisation**

#### **4.2.1.1 Preamble**

It was shown in Chapter 2 how virtualisation was used to provide a secure PEE; however, it was also highlighted that as a virtual machine must execute on (a possibly compromised) host PC's operating system it could be subjected to malware attack. Virtualisation also lacked the performance of a bootable operating system and could possibly leave sensitive data remnants in the host PC's paging system.

The operational requirement of a secure PESE could range from a 'fit for purpose' level of security for commercial teleworking to an integrated set of applications used to process classified data within a sophisticated military network centric organisation. Given the

different operational uses it was considered necessary to fully investigate the vulnerabilities of a virtual machine based secure PEE. Paper 5 describes both an analysis of the vulnerabilities and possible mechanisms to strengthen virtual machine based secure PEEs. The paper was presented at a digital forensics conference and therefore considers many of the vulnerabilities from a data acquisition perspective.

As the analysis considers whether a virtual machine provides a sufficiently secure platform the paper makes a contribution to addressing the doctoral research question: *How can a useable and maintainable secure hardened operating system and/or a small set of secure hardened applications be developed?* In particular the analysis considers the sub-question: *How can virtualisation provide a secure platform for a secure PEE?*

This paper is similar to Paper 4 in that it does not use the term secure PESE, as it did not become the term used to describe both a concept and an implementation of the concept until a later stage in the research. Like Paper 4 the term secure PEE or secure PEE device is used to refer to what this thesis defines as a secure PESE; similarly, the terms secure PEE OS and secure PEE VM are used to refer to what this thesis defines as a secure PEE.

#### **4.2.1.2 Prior Research and Knowledge**

Prior research performed by CSIRO (CSIRO, 2008), Ormandy (Ormandy, 2007) and Ferrie (Ferrie, 2007) provided a strong foundation for the analysis. The knowledge (available in 2008) consumed in the analysis is summarised as:

- Descriptive knowledge in the form of technical threats derived from the threat “A targeted attack introduces malware into the execution environment” identified in Chapter 2.
- Descriptive and prescriptive knowledge on virtualisation vulnerabilities and remedial actions described in publications by CSIRO, Ormandy and Ferrie.
- Descriptive knowledge in the form of input from stakeholders, coupled with the researcher’s experience of remote work use scenarios involving highly sensitive data.
- Informal justificatory knowledge in the form of the researcher’s knowledge of malware and security countermeasures developed over many years through keeping well-informed (whilst working as a consultant and product developer) from reading a range

of industry publications e.g. annual/bi-annual IBM (IBM, 2008), Symantec (Symantec, 2008) and Sophos (2008) cyber and information threat and risk reports.

#### 4.2.1.3 Paper 5

**Paper 5** - James, P. (2008) **Preventing the Acquisition of Data from Virtual Machine based Secure Portable Execution Environments**, 6th Australian Digital Forensics Conference, pp 82-97.

#### Abstract

*A Virtual Machine (VM) based secure Portable Execution Environment (PEE) provides a safe and secure environment that can be loaded into a host PC and an application executed with a degree of confidence that the application is separated, protected and little or no forensic evidence remains after the application has executed. A VM based secure PEE is characterised as a USB storage device containing a VM with a trusted guest operating system and application(s) which is stored in a protected partition, strong authentication to only allow an authorised user to load the VM into the host PC, and full storage device encryption to protect the confidentiality of the contents of the device. Secure PEEs provide an opportunity for organisations to issue a portable device to an individual (to perform a secure transaction on an available host PC) with the reduced risk to the organisation that neither malicious software (resident on the host PC) will infect the secure PEE device, nor sensitive data remnants (resulting from the transaction) will remain on the host PC hard disk drive after the secure PEE device has been removed.*

*A VM based secure PEE significantly reduces the opportunity to use dead forensic analysis techniques to acquire evidence of the occurrence of a transaction. However, VM based secure PEEs are susceptible to the acquisition of data through monitoring software and live forensic techniques. This paper considers the mechanisms that can be used to prevent various monitoring and live forensic techniques acquiring data from a VM based secure PEE. An attack scenario is presented to provide the context for the analysis of VM based secure PEE device vulnerabilities and why it is important that such a device would be required to counter hostile monitoring and forensic analysis. An overview is given of the security mechanisms provided by the type of VM based secure PEE under consideration and how those mechanisms combine to limit the opportunity for data acquisition through dead forensic techniques. The vulnerabilities of VM based secure PEEs with respect to malicious software and live forensic techniques are enumerated and discussed. A comprehensive set of countermeasures are proposed and analysed. The paper concludes by considering the most appropriate countermeasures to include in a VM based secure PEE to prevent the live acquisition of data.*

#### Keywords

Secure Portable Execution Environments, Virtualisation, Virtual Machine Vulnerabilities, Securing Virtual Machines, Digital Forensics.

## Introduction

The convenience provided by public Internet access centres has increasingly resulted in such centres being used to perform sensitive Internet transactions by individuals unaware of the associated risks. Given the totally open access to the PCs in these Internet access centres no level of confidence can be assumed in PC security; it is readily conceivable that such PCs have been compromised by malicious software which can exploit any Internet transactions. In addition PCs often “considered safe” (e.g. the home PC), which are used for a variety of tasks and by a number of different users often lack best practice security (e.g. anti-virus & anti-spyware software and modem/router enabled firewall capabilities). These “considered safe” PCs may contain malicious software unbeknown to the user which may be able to exploit Internet transactions. An option considered by some organisations is to issue employees and/or customers with secure portable execution environments (secure PEEs) to be used to perform sensitive Internet transactions on PCs for which no level of trust can be assumed. Secure PEEs provide trusted functionality to reduce the opportunity of malicious software exploiting sensitive data processing or an Internet transaction.

In this paper a secure PEE device is considered to be a portable Universal Serial Bus (USB) storage device containing a trusted operating system and application that can be uploaded into a PC and used to perform a transaction with a high degree of confidence that the transaction will not be exploited. A secure PEE device will also provide space to store data, strong authentication to prevent unauthorised access, device encryption to protect the confidentiality of information on the device and partitioning with differentiated access rights to separate and protect the secure PEE.

In “Secure Portable Execution Environments: A Review of Available Technologies” (James 2008) a comprehensive review and analysis of secure PEE technologies and products is given. The paper finds that a secure PEE that utilises a bootable OS provides a strong secure PEE that is resistant to malicious software. However, booting a USB device often requires a user to change the PC Basic Input Output System (BIOS) boot order which can be an unfriendly and sometimes a non-trivial activity. An alternative to a bootable OS that requires no interaction with a PC BIOS is for a secure PEE to utilise a virtual machine (VM) with a guest OS, i.e. a VM based secure PEE.

A VM based secure PEE is simpler to load and execute than a bootable OS based secure PEE because the user can load and execute the VM when the secure PEE device is plugged into a host PC running a fully booted OS. A VM provides an abstract execution environment separate from the physical PC. There are a number of different types of VM (Smith 2005). The type of VM considered in this paper is one that runs within (on top of) the PC operating system; often referred to as a type 2 hosted VM. A “guest” OS is hosted within the type 2 VM and the application is executed within the guest OS. However, a VM based secure PEE may be susceptible to any malicious software that maybe resident on the host PC OS and also susceptible to certain data acquisition techniques, e.g. live forensics and monitoring software. In this paper techniques are considered to limit the opportunities of malicious software, monitoring software and live forensic techniques to acquire data from a VM based secure PEE.

The optimal features for a secure PEE device are considered in “Secure Portable Execution Environments: A Review of Available Technologies” (James 2008). It will be assumed that the type of secure PEE considered in this paper will have all of the protection measures of the “secure PEE device 4” defined in James 2008; in summary these protection measures are:

- **Authentication:** The device will prevent access to its contents and the VM cannot be loaded until an authorised user has entered the correct authentication credentials. The most convenient approach to authenticate a secure PEE device is to plug it into a PC (that has a booted and executing OS) and an authentication application is uploaded from the secure PEE device. Through the authentication application a user authenticates with the secure PEE device.
- **Device Encryption:** The secure PEE device will be fully encrypted using on-the-fly encryption to preserve the confidentiality of the secure PEE and data residing on the USB device. In this paper it will be assumed that on-the fly-encryption will be implemented in hardware.
- **Swap space (virtual memory/page file) and space for temporary files:** The VM guest OS will require swap space. The usual default approach is to create the swap space on the host PC hard disk drive (HDD). Also applications executing on the guest OS may write temporary files to a ‘temp’ directory/folder on the host PC HDD. To prevent data remnants residing on the host PC HDD, following the use of a secure PEE, the device will be configured to:
  - provide swap space (virtual memory/page file) for the secure PEE VM guest OS; and
  - ensure the secure PEE VM guest OS and application(s) write all temporary information to available allocated space on the secure PEE device.
- **Partitioning with Differentiated Access Rights:** The secure PEE will support storage partitioning and role based differentiated access rights to partitions to preserve the integrity of the VM and any stored data on the secure PEE device. Such partitioning allows separation and isolation to be achieved. In addition to the provision of a partitioning capability the secure PEE device will also allow a partition to be defined as Read-Only. A Read-Only partition will be used to protect both the integrity of the VM (from malicious software) and ‘valued’ data.

Figure 1 presents a conceptual model of the configuration for the secure PEE device to be considered in this paper. The secure PEE device will be configured with three partitions. The first partition containing the VM will be set to Read-Only, to protect the VM's integrity. The second partition will have Read-Write access and will be used as swap space for the VM guest OS and for any temporary files created by applications. A separate third partition with Read-Write access will be used to separate and protect any user generated data.

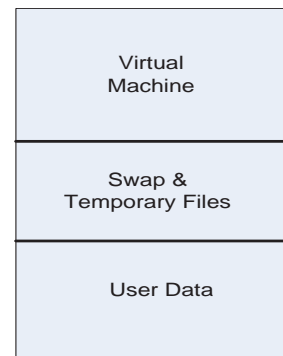


Figure 1

The secure PEE configuration presented in Figure 1 provides strong security to protect against the acquisition of data from dead forensic analysis techniques. This configuration and the aforementioned protection measures are however, unable to protect a VM based secure PEE from malicious software once the VM has been loaded into a PC.

In this paper techniques are considered to limit the opportunities of malicious software, monitoring software and live forensic techniques to acquire data from a VM based secure PEE. Through the presentation of a threat model and an attack scenario the proposition for the identification of additional protection measures for a VM based secure PEE is given. The vulnerabilities of VMs are enumerated and discussed. A set of VM protection measures are presented and an improved VM based secure PEE proposed.

The following terms are defined in James 2008 and are restated below as they are used throughout this paper:

- **secure PEE device:** the secure platform/infrastructure consisting of a USB mass storage device configured with a secure PEE, secure storage space and possibly hardware based security mechanisms/technologies (e.g. encryption and secure partitions if available).
- **secure PEE:** the trusted OS, trusted application(s), security technologies (e.g. authentication and software encryption) and the appropriately configured hardware security mechanisms (if any) of the secure PEE device.
- **secure PEE OS:** the trusted OS component of the secure PEE.
- **VM based secure PEE:** a secure PEE that utilises a VM to host the secure PEE OS.
- **trusted OS:** an OS that has been acknowledged as secure by the supplier and users. To be considered trusted the OS may have a reduced set of hardened functionality and/or been subjected to independent rigorous evaluation and testing.
- **PEE:** a portable execution environment that does not necessarily have any security technology nor has been specifically configured to be secure.

- **portable storage device:** a USB flash device (often know as a thumb drive or pen drive) or a USB HDD packaged in a portable enclosure.

## Threat Model and Attack Scenario

### Threats

The secure PEE protection measures discussed above provide countermeasures to the following threats:

- Acquisition of sensitive data remnants (resulting from an application storing temporary information) residing on a host PC's HDD following the completion of a network transaction through the use of dead forensic techniques; and
- As a result of loss or theft, unauthorised access is gained to the VM and sensitive data held on the device.

However, a VM based secure PEE is susceptible to the following threat:

- Malicious software or monitoring software (inserted by a hostile user or organisation) is able to capture information through the application of
  - memory acquisition including the use of live forensic techniques
  - keyboard logging
  - screenshot capture; and
  - reverse engineering.

### Attack Scenario

Numerous different attack scenarios are possible. The attack scenario presented in this paper provides the context and rationale for the analysis of VM vulnerabilities and the identification of tools and techniques to be applied to limit the exploitation of the identified vulnerabilities when a VM is used in a secure PEE. Rather than consider attack options possible from an easily accessible and open Internet access centre, the attack scenario presented below is focussed on a very specific area of use.

*VM based secure PEE devices are issued by an organisation to its employees to enable work to be performed remotely using a virtual private network (VPN) connection over the Internet. The VM based secure PEE will connect (through the VPN) to remote servers that contain sensitive/classified information, also the work performed by the employee using the VM based secure PEE may result in the generation of sensitive/classified data.*

*An employee will use the VM based secure PEE device and its applications on remote host PCs that the employee considers to be safe and secure. The host PCs are executing the Windows OS. However, due to the nature of the business conducted by the organisation it*



*is likely that technically sophisticated and capable hostile organisations will attempt to acquire data through:*

- *Embedding monitoring and data acquisition software on the remote PCs that can transmit data to the hostile organisation.*
- *Performing dead forensic analysis on the remote host PC subsequent to the use of the VM based secure PEE.*
- *Performing live forensic data acquisition either (possibly) during or subsequent to the use of the VM based secure PEE.*

The attack scenario may appear infeasible or extreme but for certain organisations (e.g. government agencies) considering home and/or remote working for employees such an attack scenario needs to be considered as hostile organisations (e.g. foreign intelligence agencies) would use any means available to gain access to sensitive/classified data. It is conceivable that such hostile agencies will have access to sophisticated monitoring and dead & live forensics tools and techniques to enable the analysis of:

- the host PC used for processing (assuming access to the PC by the attacker is possible).
- a captured VM based secure PEE device and the host PC used for processing.

### **How the VM based Secure PEE Limits the Opportunity for Dead Forensic Analysis**

The protection measures of the VM based secure PEE device considered in this paper provide strong mechanisms to protect the confidentiality of data and the integrity of the VM based secure PEE. These protection measures limit both the opportunity to acquire data using dead forensic analysis techniques when the device is at rest and to compromise the integrity of the VM when the device is operational.

Dead forensic analysis is the analysis of a digital device once processing has stopped and power has been removed. The analysis involves the examination of any aspect of the digital device's storage media to acquire data using a tool set that does not require the OS of the device to be booted. The protection measures of the VM based secure PEE device limit the opportunity to acquire data from both the device itself and the host PC HDD.

To assist the reader to appreciate the capabilities of the type of VM based secure PEE proposed in this paper an overview is given below of how the device's protection measures limit both the opportunity to:

- acquire data using dead forensic analysis techniques; and to
- compromise the integrity of the VM.

## **Encryption**

Full hardware based device encryption prevents the acquisition of meaningful data from the VM based secure PEE storage medium. Access to decrypted data is only possible once the device has been powered and successfully authenticated. Even if the device can be successfully dismantled to bypass the authentication mechanism and allow access to the storage medium, time consuming brute force attacks would be required. Whilst brute force attacks are possible, if a strong cryptographic algorithm like AES is used then significant computing resources (i.e. very high end supercomputing centres) and large amounts of time are required to identify the cryptographic key(s) and thus acquire data.

Unlike software based encryption, hardware based encryption does not store any encryption key(s) in the host PC's random access memory (RAM) and therefore it is not possible to acquire the secure PEE device key(s) from memory. Recent research has shown it is sometimes possible to acquire keys from a PC's RAM (Haldermany et al., 2008) after the PC has been powered off, however in practice such key recovery is difficult in the extreme (Hannay et al., 2008).

## **Strong Authentication**

The authentication component of a VM based secure PEE device prevents access to the contents of the device and the loading of the VM into the host PC until a user has entered the correct authentication credentials. Once successful authentication has occurred access to the contents of the device is possible with the device performing on-the-fly encryption/decryption so the user can read all available data (subject to partitioning access controls). The strong authentication prevents the application of dead forensic tools (running on the host PC) accessing any data on the VM based secure PEE; as access to data is blocked until the authentication process has been completed.

## **Swap Space/Virtual Memory**

Configuring the VM guest OS of the secure PEE to write its swap file direct to the secure PEE device and also to configure applications that run under the control of the VM guest OS to write any temporary data to the device prevents data remnants residing on the host PC HDD after completion of a VM based secure PEE session. Any analysis of the host PC HDD will not reveal data generated during the use of a VM based secure PEE; although the host PC OS logs may indicate a VM was loaded and executed.

## **VM & Guest OS State Information**

The VM and guest OS may generate data on their respective states that is required to be stored so that when another VM based secure PEE session is initiated the system state from the previous session is restored. Like the swap space and other temporary data, the files that store the VM and guest OS system state can be stored on the VM based secure PEE device; to avoid leaving data remnants on the host PC HDD.

## **Partitioning**

Partitioning allows the separation and protection of software and data; a partition need only be mounted as required and differentiated access rights ensure only authorised access is permitted to a partition. As shown in Figure 1 the VM based secure PEE device considered in this paper utilises three partitions. The first partition containing the VM will have Read-Only access to prevent any malicious software running on the host OS compromising the integrity of the VM.

The second partition will have Read-Write access and will be used as swap space for the VM guest OS, for any temporary files created by applications and system states. As identified above using a dedicated partition for all temporary data generated by the VM guest OS significantly limits the opportunity to acquire data remnants from the host PC HDD upon completion of a VM based secure PEE session.

A separate third partition with Read-Write access will be used to separate and protect any user generated data. A partition dedicated to storing user data ensures the partition containing the VM can be set to Read-Only without preventing user generated data being stored on the device.

## **Acquiring Data from a VM based Secure PEE**

The type of VM considered in this paper utilises the OS of the host PC as an execution platform. The host PC OS can provide a platform for malicious/monitoring software. For a host PC for which no level of trust can be assumed the possibility of the host PC OS supporting attacks (through embedded malicious software) on the VM and its guest OS (& applications) is feasible. Techniques like keyboard logging and screenshot capture are typical of the type hostile acquisition methods used.

Attackers will, if logistically possible, utilise live forensic techniques to acquire sensitive data. Live forensic analysis involves the acquisition of data from a PC whilst it is still executing and therefore enables data to be acquired from the PC's memory. Another advantage of live forensic analysis is that any mounted HDD partitions that are encrypted when the data is at rest (i.e the PC is powered off) will provide full access to the plain text data when the PC is executing.

The work performed by (CSIRO, 2008), (Ferrie, 2007) and (Ormanday, 2007) on VM vulnerabilities have provided excellent information sources to support and supplement the author's experience and knowledge.

## **Detecting the Presence of a VM Based Secure PEE**

An attacker or embedded malicious software needs to avoid detection and therefore should only launch an attack if a VM based secure PEE has been identified. The following methods could be used to identify if a VM has been loaded and is executing:

- *Using Standard OS Features:* Malicious software could query any or all of the standard Windows Autorun, Task Bar, Task Manager or Windows data structures to identify if a VM is executing.
- *Tools to Detect Processes:* A VM could be detected by a range of specialist detection tools (e.g. Process Explorer) that have been created to identify executing processes; particularly if the processes are using concealment techniques.
- *Checking OS Capabilities:* Malicious software could search:
  - the Windows registry for entries that may identify the presence of a VM.
  - the list of loaded Windows DLLs to identify DLLs used by a VM.
  - the list of installed drivers to identify drivers used by a VM.

## Memory Acquisition

The memory management function of an OS:

- controls access to, and the allocation of, the PC RAM
- implements a virtual memory system (often known as swap space or paging) which extends and optimises the use of the PC RAM by using part of the PC HDD; and
- provides memory isolation for processes.

As outlined above, to prevent virtual memory pages being placed on the host PC HDD, a VM based secure PEE device allows the VM guest OS to be configured to utilise a dedicated partition on the device for virtual memory pages. Whilst the VM based secure PEE device prevents the VM's guest OS leaving sensitive data remnants (held in virtual memory pages) on the PC HDD, the device can do nothing to protect the guest OS virtual memory from a live attack when the VM based secure PEE is executing.

As the host PC OS provides the execution environment for the VM, the PC memory (RAM and virtual memory) utilised by the VM will therefore be managed and controlled by the host PC OS. Although OS' are designed to provide process isolation (i.e. a process should not be able to interfere with the memory space of another process) it is possible for a hostile OS to interfere with a process' allocated memory; the hostile OS (or an executing hostile process) could access and modify or acquire the contents of the memory. To be successful in modifying or acquiring the contents of the memory used by the VM the attacker would need to have both a detailed knowledge of how the OS implements memory management and the design and structure of the VM and the executing application. Information on how to modify and acquire memory is becoming increasingly available due to gamers publishing details on memory hacks for games.

Techniques to modify and acquire the contents of a PC's memory include:

- **Inserting Malicious Memory Management Software:** An attacker, who has access to the host PC OS and has a detailed knowledge of the OS memory management architecture and design, may be able to rewrite the memory management software to allow access to the VM's allocated memory. For such an attack to be successful a detailed knowledge of the use of memory (RAM and virtual memory) by the VM is required. The advantage of embedding malicious memory management software in the OS kernel is that the user will be unaware of its existence, the disadvantage is that the attacker needs undetected access to the host PC and a highly sophisticated knowledge of the OS memory management and VM used in the secure PEE.
- **Utilising the Memory Management API:** The OS virtual memory management API, which allows developers to allocate, release and modify virtual memory could be used by malicious software to capture the contents of the VMs allocated memory. The advantage of using the virtual memory API is that it is a published interface that malicious software can utilise, the disadvantage is that a well written VM or guest application can use access rights on virtual memory pages to prevent malicious software acquiring data.
- **Memory Hacking Tools:** Generated predominately by gamers, the increasing number of available memory hacking tools can be used to gain access to memory; such tools include:
  - Memory Hacking Software – MHS (Spiro 2008): allows a user to search and change data via a graphical interface.
  - Tsearch (Corsica Productions 2008): provides a similar capability to MHS.
  - FU Rootkit (FU Project 2008): allows kernel data structures to be accessed via a device driver that gets installed.

The advantage of memory hacking tools is that they are readily downloadable from various web sites, however the main disadvantage is that it is difficult for the attacker to use these tools without detection as concurrent access (with the VM based secure PEE user) to the host PC is required.

- **Memory Dump:** The standard OS memory dump upon process termination can reveal sensitive data being processed by the VM. Malicious software can cause the VM process to terminate and a process memory dump to occur. An advantage of causing memory dumps to occur is that relatively simple malicious software can cause a dump, however the main disadvantage is knowing when to trigger the memory dump to acquire sensitive data; causing multiple dumps to occur (i.e. numerous memory dumps will increase the probability of acquiring sensitive data) will make the user suspicious and/or end the VM based secure PEE session due to the multiple process terminations.
- **Firewire Access:** Although an overt action, it is possible to acquire the contents of a PC's RAM using a Firewire direct memory access function (Woodward et al 2008). The technique involves connecting another PC to the target PC via a Firewire port and executing a Firewire

memory access tool. The tool will allow the contents of the target PC's memory to be copied to the PC running the Firewire memory access tool. The advantage of this approach is that the whole contents of the RAM can be acquired. The disadvantage is that it is an overt action that would be extremely difficult to conceal from the user.

- **Cold Boot Memory Access:** As discussed above, recent research (Haldermany et al., 2008) has shown it is possible to acquire the contents of a PC's RAM after the PC has been shut down if the acquisition is performed soon after the power-off has occurred. To work successfully the memory needs to be kept cold. The advantage of this approach is that it is possible to analyse the PC after the VM based secure PEE has been used and in theory acquire potentially sensitive information. The disadvantages include being able to gain access to the PC in a timely fashion to acquire meaningful data and the need to have cold RAM.

### Keyboard Logging

Keyboard logging is most often used to obtain authentication credentials and is implemented by malicious/monitoring software executing on the host PC OS and intercepting keyboard input which is either:

- stored in an unused area of the host PC HDD and retrieved by the attacker at a later time; or
- transmitted directly to the attacker as the keyboard input is received from the malicious/monitoring software.

Attackers are able to embed malicious software into a process and receive input from a keyboard due to the way OS's allow the interception of messages (using hooks) sent by concurrently executing processes. Attackers also use the information gathering capabilities available in application programming interfaces (APIs) to understand the capabilities of executing processes. Attackers are able to abuse the process communication and API features of OS' to introduce keyboard logging capabilities. Keyboard logging can be implemented in the following ways:

- **Changing Device Drivers:** A device driver is part of the OS kernel and has unrestricted access to the host PC hardware. A device driver based keyboard logger is able to communicate directly with the keyboard and capture all input. An advantage of this type of key logger is that it is very hard to detect its presence within the OS kernel. However, disadvantages include:
  - the requirement for administrator privilege and access to the OS kernel to be able to install the device driver (into the kernel); and
  - due to the low level nature of the key logger it is not possible to determine the context of the input, i.e. all input is captured and filtering authentication credentials (or other sensitive information) from other input at the device driver level is not possible.
- **Utilising Application Programming Interface Hooks:** The Windows OS provides the capability, through API hooks (Ivanov 2002), to intercept inter-process messages. The API

hook allows a malicious/monitoring dynamic link library (DLL<sup>27</sup>) to be inserted into a process which can intercept and forward messages to a keyboard logger process. The large amount of documentation on the use of API hooks, available on the Internet, enables an attacker to readily implement this type of key logger. Another advantage is that under certain circumstances it is possible to introduce an API hook based key logger without access to the host PC. A disadvantage in the use of an API based key logger is that it can be detected as it is an executing process.

- **Exploiting Debug Code Injection:** Debug code injection is a Windows facility provided to enable debugging of executing programs (where the source code is not available). The technique involves modifying the program's binary code (in the PC RAM) through the injection of interceptor code to output specific information. Debug code injection can be exploited by attackers to inject key logging code. Like API hooks an advantage of debug code injection based key loggers is that lots of documentation exists to enable an attacker to build such a key logger; Microsoft supports the Detours DLL to enable code injection. An important disadvantage is that the attacker is required to understand the executing process into which the key logger code is to be injected.
- **Replacing Dynamic Link Libraries:** Replacing a DLL with a DLL with the same name is another approach to installing a key logger. The key logger is implemented within a DLL function that a process utilises in 'good faith'. An advantage is that a malicious substituted DLL is hard to detect, however a disadvantage for the attacker is that every function in the original DLL needs to be present in the malicious replacement DLL.

The following techniques can be used to prevent a key logger process/application from being detected:

- **Covert Existence:** Methods include removing the process from the taskbar, hiding the application window, hiding tracing information of the application installing and preventing the process being listed in the task manager.
- **Alternative PC:** The key logger can be installed on an alternative networked PC.
- **Renaming:** Methods include modifying the key logger process' signature and renaming files used by the key logger.

### Screenshot Capture

A captured image of the screen can be used to acquire data from a VM based secure PEE. Windows provides a number of capabilities to capture and store screen images, also it is possible to use the processing capabilities of a PC graphics card to acquire screenshots. Screenshot capture can be implemented in the following ways:

---

<sup>27</sup> A DLL provides a capability to allow multiple programs to share a set of library functions. In the case of a VM based secure PEE the library function is linked to the VM at run-time with the host PC OS performing the binding.



- **Changing Graphics Card Device Drivers:** A PC uses a specialist graphics card to manage and present the output on the PC screen. A graphics card device driver based screenshot capturer can capture all screen output and save as bit streams in a file(s). To successfully implement such a malicious device driver requires the attacker to have a detailed knowledge of the host PC's graphic card design. An advantage of this type of screenshot capture is that it is very hard to detect its presence within the OS kernel. However, a major disadvantage is that due to the low level nature of the device drivers it is not possible to determine the context of the output and therefore large volumes of screen displays will be captured.
- **Standard Print Screen Capability:** Windows provides a capability that allows the print screen key (present on almost all keyboards) to be pressed and an image of the screen is captured and placed on the Windows 'clipboard'. Malicious software can use the print screen capability by emulating the key press, once a screenshot image has been written to the clipboard the malicious software can use the standard clipboard API to move the image to a file. An advantage in using the print screen capability is that it is a reliable existing feature that can be easily implemented. A disadvantage is that the capability can be disabled.
- **Graphics Device Interface (GDI+):** The Windows GDI+ API is responsible in the Windows OS for managing the presentation of output to the screen. GDI+ presents each screen as a pixel map. The GDI+ API can be used by malicious software to make copies of the pixel maps and move the map to an alternative part of the PC RAM and then be exported as an image to a file. An advantage of using the GDI+ based screenshot capture is that it cannot be disabled. A disadvantage is that non-trivial malicious software needs to be developed (in comparison with the print screen approach) to exploit the capability.
- **DirectX API:** The DirectX multimedia library is a comprehensive capability used for presenting videos and graphics. DirectX uses buffers to store the screen display before sending it to a graphics card for display on the screen. Malicious software can use the DirectX API to retrieve the contents of the buffers, convert into a bitmap and export as an image to a file. Like GDI+, the DirectX capability cannot be disabled, but similarly it requires relatively complex malicious software to produce useful output and of course can only be exploited if the VM guest application utilises DirectX.

## Reverse Engineering

Reverse engineering a VM based secure PEE will enable an attacker to understand how the VM (and its OS and applications) work by reconstructing the source code from binary code. Once a VM (and its OS and applications) has been reconstructed into source code the attacker can identify vulnerabilities and compile a plan of attack. Reverse engineering a VM based secure PEE will obviously require a highly skilled and experienced attacker.

Reverse engineering is generally performed by a range of techniques including:

- debug software to step through the VM code.



- emulation software which will allow snapshots of the state of the VM to be taken and analysed
- process memory dumps that can be analysed.
- forensic analysis and memory hacking tools to understand how the VM utilises the HDD and memory.
- packet sniffing software to analyse network traffic.

The advantage of reverse engineering is that it enables the attacker to build a comprehensive understanding of the VM based secure PEE implementation to enable attacks to occur. The disadvantages include the need to acquire a VM based secure PEE device to reverse engineer, and of course reverse engineering by itself does not allow data to be acquired; the technique needs to be used in collaboration with other techniques to mount a successful attack.

### **Preventing the Live Acquisition of Data from A VM based Secure PEE**

It has been shown above that there are potentially a number of exploitable vulnerabilities in VMs that the protection measures of the secure PEE device cannot counter. However, a range of techniques exist that can be constructed into a set of countermeasures to address the vulnerabilities. The work performed by CSIRO (CSIRO 2008) provided valuable input in the identification of many of the techniques presented.

### **Preventing the Detection of a VM**

If the presence of a VM based secure PEE executing on a host PC can be successfully hidden then an attacker and/or embedded malicious software will be ineffective. Most of the techniques to prevent detection counter the detection techniques identified above. Techniques to hide the presence of a VM include:

- **OS Features Avoidance:** The VM should avoid execution by Autorun and prevent an icon appearing in the Task Bar. Using an application and process naming convention that has no association to VMs may avoid detection by both the Task Manager and searching through Windows data structures.
- **Hiding from Scanning/Monitoring Tools:** Preventing the detection of a VM by sophisticated scanning/monitoring tools may be difficult and will depend upon the capabilities of each tool. For instance, the freeware tool Process Explorer (SysInternals, 2006) provides detailed information about a process icon, command-line, full image path, memory statistics, user account and security attributes. Some of these process attributes would be hard to conceal by the VM.
- **OS Capability Avoidance:** The VM should be constructed to avoid placing entries in the Windows registry, and where possible not use the host PC OS DLLs. However, avoiding the use of installed OS drivers is extremely unlikely, for obvious reasons.

## Memory Acquisition

The following techniques can be used to prevent or limit the opportunity from memory acquisition based attacks:

- **VM and Memory Address Obfuscation:** Obfuscation is a technique used to prevent the reverse engineering of software. Software obfuscation is implemented by creating hard to interpret code by masking the language syntax and grammar. The principles of obfuscation can be used to make it difficult for an attacker to locate sensitive data in memory. Obfuscation techniques can be applied as follows:
  - *Memory Address Obfuscation:* By changing (randomising) the address space of memory an attacker cannot acquire meaningful data from consecutive memory space. Address obfuscation can be implemented by the guest OS virtual memory management system. The disadvantages are the degraded performance (due to retrieving data from non-consecutive memory locations) and complexity to implement.
  - *VM Obfuscation:* By implementing a virtual memory management system within the VM (in addition to the host PC OS and guest OS virtual memory management systems) the level of complexity for the attacker and obscurity of data will increase. The disadvantages of VM obfuscation are both the complexity to implement and the performance degradation due to multiple virtual memory management systems operating simultaneously.
- **Complex Data Structures:** Approaches to making data structures complex and therefore difficult for an attacker to understand include:
  - *Mixing variables:* The methodology involves placing parts of one variable into another. Records are kept of where each of the various parts of the variables are located to enable reassembly of the correct value to occur. The disadvantage of this approach is that the attacker may be able to read the variables before they are mixed or obtain the record of mixing locations and then perform reassembly; there is also the degradation in performance due to mixing and reassemble of variables.
  - *Assigning the wrong data type:* By storing integers as strings, strings as integer arrays, etc, it is possible to make some memory hacking tools become confused and possibly crash. The disadvantage is that the sophisticated hacker is able to circumvent this technique.
- **Memory Encryption:** Encryption can be used to protect the contents of memory. Whilst encryption is probably the strongest mechanism available to prevent the attacker gaining meaningful data from memory, the cryptographic algorithm and encryption keys must be stored in plaintext in memory to enable execution. The attacker therefore may be able to acquire the algorithm and keys and reverse engineer the VM. Performance is also likely to be an issue.

- **Preventing Overflow Attacks:** Causing memory to overflow is a technique used by attackers to allow inserted malicious code to execute, i.e. code is inserted into available process memory space, then the code is able to execute when a buffer overflow is invoked. Overflow attacks can be prevented by setting the virtual memory pages with no execution rights. The advantage of this technique is that it is a well-documented technique, however a disadvantage is that an OS has the capability to mark memory pages as “no execute”.

## Keyboard Logging

The following approaches can be used to prevent or limit the opportunity from keyboard logging based attacks:

- **On Screen Keyboard:** A very popular technique to counter keyboard logging is to use an on screen keyboard (the keyboard is a screen image) where keystrokes are entered by pointing to the respective character with the mouse pointer and “clicking”; other methods (e.g. a stylus) can be used to select the required character on the screen. An on screen keyboard does not mean that it has to be used for all keyboard input, for instance the on screen keyboard could be used just for the input of authentication credentials; a number of Internet banking applications use this approach. The on screen keyboard can be implemented as:
  - a feature in the VM.
  - a feature in the guest OS; or
  - part of an application.
- **Embedded Authentication Credentials:** A technique to protect authentication credentials from capture is to embed them in a string of random text (Herley et al 2006). This technique can be used with passwords and personal identification numbers (pin) as follows:
  - when the password/pin dialogue box appears on the screen then perform the following steps:
    1. move out of password/pin dialogue box focus
    2. type any random input
    3. move into focus for the password/pin dialogue box and type the next character of the password/pin
    4. repeat the above steps 1 to 3 until the password/pin has been entered.

Any keyboard logger will gather a large string of characters, as the keyboard logger will capture all input, however the password/pin dialogue box will only receive the characters entered whilst in focus. Any parsing of the input by malicious software will prevent the password/pin being identified.

- **Preventing the use of malicious API Hooks:** A technique that works to counter an API hook based key logger is to enter as the very first hook in the hook list a trusted hook to a trusted

DLL. The trusted DLL will be responsible for passing only input to the VM and/or its guest OS. The trusted hook will only work if it is the very first hook.

- **Simple Encryption:** To counter a keyboard logger an application expecting input can issue a simple encryption key to the user which is used to encrypt the input. The keyboard logger would intercept meaningless characters but the application would be able to decrypt the input because it has the encryption key. The encryption key would need to be very simple to enable the user to calculate and enter data in a timely manner.

## Screenshot Capture

The following approaches can be used to prevent or limit the opportunity of data acquisition from screenshot capture:

- **Disabling the Standard Print Screen Capability:** Windows allows the print screen capability to be disabled. However, this approach will not prevent sophisticated attackers as disabling does not prevent GDI+ and DirectX based malicious software.
- **Use of Graphic Card Overlays:** An approach to counter GDI+ and DirectX based screenshot capture is to use the overlay capabilities available in most PC graphics cards. By using the graphics card API code can be written that bypasses the OS and directly modifies the screen space to produce an overlay. An overlay allows graphics to be placed over the graphics being displayed by an application but the overlayed graphics cannot be captured by screenshot loggers because the functions performing the overlay are not part of the OS.
- **Scrambling Screen Output:** To prevent a screenshot logger capturing useful information the output on the screen can be scrambled with the exception of a small restricted area. The user accesses only the restricted area of the screen for sensitive operations. Whilst a screenshot capturer will capture the restricted area, if this area is kept small it may be difficult to identify meaningful information amongst the scrambled data.

## Reverse Engineering

The main countermeasure adopted to prevent reverse engineering is software obfuscation (Ogiso et al 2003). As outlined above obfuscation is the process making software hard to read. Software obfuscation techniques include software compression, keyword substitution and the use/non-use of whitespace to mask language syntax and grammar.

In theory encryption could be used to encrypt the whole VM executable with only the encrypt/decrypt algorithm in plaintext, however in practice such an approach would require the cryptographic algorithm to execute under the control of the host PC OS and could therefore be susceptible to attack and the cryptographic algorithm compromised.

## Detection of Malicious Software

Performing checks to detect malicious software could be performed as an alternative, or as an additional measure to the use of the aforementioned complex countermeasures. The issue with using detection as the sole countermeasure is that only known malicious techniques will be identified; also kernel level malicious software is unlikely to be detected. The best strategy is to use detection techniques to support other countermeasures. The following detection techniques could be used:

- **Detection of Malicious DLLs:** The host PC OS DLLs and API hooks (used by the VM based secure PEE) could be monitored by checking the modules loaded prior to execution and comparing against known modules for the DLL. Any identified unknown modules can be treated as malicious and action taken. This detection technique would require a DLL examination tool to be executed prior to loading the VM.
- **Use of Anti-Virus/Anti-Spyware Tools:** Prior to loading the VM, anti-virus and anti-spyware tools can be executed to identify and remove any known malicious software. The tools would need to be appropriate to identifying the type of malicious software that can subvert VMs, which may mean specific bespoke tools are required.

The advantage of the above two detection techniques is that an untrusted host PC OS can be identified before the VM is loaded. However, the disadvantages include:

- the time required to perform OS scanning and DLL examination
- the tools will only be as good as the database of known problems
- the inconvenience of having to determine the course of action to take if an untrusted environment is identified, i.e. is it feasible to remediate the host PC OS or should an alternative host PC be used?

An alternative detection technique is:

- **In Parallel Monitoring:** This technique involves a detection process(es) running in parallel with the VM. Such a process could execute on the host PC OS or within the VM, and would monitor:
  - any unexpected changes to the VM
  - if another process attempts to access the VM
  - the commencement of any suspicious processes on the host PC OS.

The advantage of this technique is that it does not require time consuming pre-processing detection software to be executed before the VM can be loaded. The disadvantage of parallel monitoring is that it is conceivable that the VM could be exploited before the detection process has identified any malicious software.

## **An Improved VM Based Secure PEE that Limits the Opportunity for Live Acquisition of Data**

As the specific brand of VM has not been explicitly stated the reader may have assumed a commercial off the shelf (COTS) product would be used, which would also be the author's desired approach. Access would be required to the VM source code to implement many of the countermeasures proposed in this paper. To preserve its intellectual property most commercial organisations do not publish a products source code. Therefore to achieve the desired security for a COTS VM the product vendor would be required to change the VM. Alternative approaches to using a COTS VM could include:

- developing a bespoke VM that implements all of the required countermeasures; or
- using a freeware VM and adding the countermeasure, however most freeware requires any added functionality to be published and made freely available – which would obviously then be available to an attacker.

In proposing an improved VM based secure PEE the commercial and logistical viability of implementing the required countermeasures are not consider. Observations may be made on the feasibility of implementing a countermeasure, otherwise it is assumed that all required functionality is implementable.

### **Summary of Attack Scenario**

Important aspects of the attack scenario described above can be summarised as:

- The issuer of the VM based secure PEE has to assume the host PC is not secure even though the user may consider it to be secure.
- If the user considers the host PC to be safe, then it will most likely be located in an environment where physical access will be difficult. However, it must be assumed the attacker is able to gain physical access and have sufficient time to examine the HDD using dead forensic tools, install malicious software and possibly use live forensic tools if the PC has been left powered on.
- The host PC will be connected to the Internet and therefore malicious software can be pushed to the host PC.

### **An Improved VM Based Secure PEE**

Table 1 below summarises the set of countermeasures that can limit the opportunity to perform live acquisition of data from a VM based secure PEE. For each countermeasure:

- confirmation is given on whether the countermeasure will be used in an improved VM based secure PEE; and
- comments provided on the effectiveness, useability and implementation.

The decision to include a countermeasure is based upon:

1. does the countermeasure address a vulnerability that could be exploited within the given attack scenario.
2. the level of inconvenience the implemented countermeasure will cause the user, i.e. will the VM based secure PEE now be slower and more difficult to use.
3. in practice could the countermeasure be implemented, ignoring the commercial and logistical viability.

<b>Required Countermeasures for Improved VM Based Secure PEE</b>		
<b>Countermeasure</b>	<b>To Be Used</b>	<b>Comments with respect to effectiveness, useability and implementation in an improved VM Based Secure PEE</b>
<b>Preventing the detection of a VM – OS features avoidance</b>	<b>Yes</b>	Relatively simple to implement and effective.
Preventing the detection of a VM – hiding from monitoring tools	No	Counters an unlikely attack, also difficult to implement.
<b>Preventing the detection of a VM – OS capability avoidance</b>	<b>Yes</b>	Even if a bespoke VM is developed it maybe infeasible to avoid the use of DLLs & registry entries by a VM.
<b>Memory acquisition – VM &amp; memory address obfuscation</b>	<b>Yes</b>	Only possibly if a bespoke VM developed.
Memory acquisition – complex data structures	No	Complex to implement and unlikely to provide an effective measure.
Memory acquisition – memory encryption	No	Likely to be difficult to implement and resultant implementation slow to execute.
<b>Memory acquisition – preventing overflow attacks</b>	<b>Yes</b>	For COTS product it may not be possible to implement.
<b>Keyboard logging – on screen keyboard</b>	<b>Yes</b>	Can be built into secure PEE application.
<b>Keyboard logging – embedded authentication credentials</b>	<b>Yes</b>	Implemented through user procedure.
<b>Keyboard logging – preventing the use of malicious API hooks</b>	<b>Yes</b>	For COTS product it may not be possible to implement.
Keyboard logging – simple encryption	No	Likely to be complex for user and easy to break by attacker.
<b>Screenshot capture – disabling the print screen capability</b>	<b>Yes</b>	Can be implemented with guest OS
Screenshot capture – use of graphic card overlays	No	As graphic card APIs will differ any overlay countermeasure would not be portable across a range of different host PCs.
<b>Screenshot capture – scrambling screen output</b>	<b>Yes</b>	Can be implemented by the secure PEE application.
<b>Reverse engineering - obfuscation</b>	<b>Yes</b>	Only possible if a bespoke VM developed.
Reverse engineering - encryption	No	Whilst in theory this countermeasure may be possible, in practice it is likely only parts of the binary can be encrypted.
<b>Detection of malicious software – detection of malicious DLLs</b>	<b>Yes</b>	Implement within a detection application.
<b>Detection of malicious software – use of anti-virus tools</b>	<b>Yes</b>	Likely to require bespoke tools as standard anti-virus & anti-spyware are unlikely to find all the malicious software designed to attack a VM based secure PEE.
<b>Detection of malicious software – in parallel monitoring</b>	<b>Yes</b>	Likely to require bespoke tools as standard anti-virus & anti-spyware are unlikely to find all the malicious software designed to attack a VM based secure PEE.

Table 1



## Conclusion

A VM based secure PEE device provides a secure platform for portable computing, however it is susceptible to attack and the live acquisition of data due to the VM's reliance upon the underlying host PC OS. In this paper a range of countermeasures have been proposed that can address the VM's vulnerabilities. An improved VM based secure PEE that incorporates countermeasures that are effective, usable and implementable (in theory) has been proposed. The field of virtualisation is developing rapidly. It is possible that VM vendors may consider implementing some of the countermeasures identified in this paper for security applications.

Future work could include a detail review of available VMs to determine the VM most likely to be suitable for use in a secure PEE either because it has some of the countermeasures required or could readily be changed to implement the required countermeasures. Other work could include a proof of concept of the countermeasures in a VM based secure PEE.

## References

- Corsica Productions (2008). "Tsearch - a memory scanner/debugger utility." Retrieved October, 2008, from <http://duckduckgo.com/TSearch>.
- CSIRO (2008). Virtual Machines: An Initial Analysis of Threats and Remedial Actions.
- Ferrie, P. (2007). Attacks on more virtual machine emulators. Symantec Technology Exchange. Available from: <http://index-of.es/Windows/attacks2.pdf>
- FU Project (2008). "FU Rootkit." Retrieved October, 2008, from <http://www.rootkit.com/project.php?id=12>.
- Hannay, P. W., A (2008). Cold Boot Memory Acquisition: An Investigation into Memory Freezing and Data Retention Claims. The 2008 International Conference on Security & Management, Las Vegas, Nevada.
- Herley, C. F., D (2006). How To Login From an Internet Cafe Without Worrying About Keyloggers. Symposium on Usable Privacy and Security CMU.
- Ivanov, I. (2002). "API Hook Revealed." Retrieved October, 2008, from <http://www.codeproject.com/KB/system/hooksys.aspx>.
- J. Alex Haldermany, S. D. S., Nadia Heninger, William Clarkson, William Paulx, and A. J. F. Joseph A. Calandrinoy, Jacob Appelbaum, and Edward W. Felten (2008). Lest We Remember: Cold Boot Attacks on Encryption Keys. Proc. 2008 USENIX Security Symposium.
- James, P. (2008). Secure Portable Execution Environments: A Review of Available Technologies. 6th Australian Information Security Conference, Perth.
- OGISO, Y. S., M SOSHI, A MIYAJI (2003). "Software Obfuscation on a Theoretical Basis and Its Implementation." IIEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences Vol.E86-A (1): 176-186.
- Ormandy, T. (2007) An Empirical Study into the Security Exposure to Hosts of Hostile Virtualized Environments. CanSecWest Volume, DOI:
- Smith, J. (2005). "The Architecture of Virtual Machines." Computer 38(5): 32-38.



Spiro, L. (2008). "Memory Hacking Software (MNS)." Retrieved October, 2008, from <http://www.memoryhacking.com/>.

SysInternals. (2006). "Process Explorer." Retrieved October, 2008, from <http://live.sysinternals.com/>.

Woodward, A. H., P (2008). Forensic implications of using the FireWire memory exploit with Microsoft Windows XP. The 2008 International Conference on Security & Management, Las Vegas, Nevada.

#### **4.2.1.4 Synopsis**

**Outcomes and Contribution to Knowledge:** The paper provides a knowledge contribution through a comprehensive analysis of the vulnerabilities and potential countermeasures for a virtual machine based secure PEE. It can be concluded that the analysis highlighted that a virtual machine is not an optimal technology platform for a secure PEE because it is vulnerable to attacks from a compromised host PC operating system. The paper identifies countermeasures to such attacks but they impact upon the performance of a computing environment already noticeably slower than a non-virtualised environment. The proposed countermeasures can either form part of the functionality or configuration of a virtual machine, or can be implemented into an existing virtual machine. Whilst source code changes to implement the countermeasures may be possible in a freeware virtual machine (QEMU, 2014), changes to commercial off-the-shelf virtual machine products (e.g. VMware (VMware, 2014)) are highly likely to be infeasible as the source code will be proprietary.

As described in Chapter 2, virtualisation has been successfully utilised in products such as the Bull globull (BullDirect, 2008) to provide an adequate secure PEE component of a secure PESE. To attack a product like the globull, the remote work PC would have to be specifically targeted to exploit the virtual machine vulnerabilities. The effort and resources required to attack a host PC may be considered excessive for some remote work situations. A virtual machine based secure PEE may provide an adequate computing environment where an assessment has been performed which identifies that the risk of data loss from a compromised remote work PC is either negligible or acceptable. However, the PhD research gap identifies the need for a secure PESE that utilises hardening to minimise any vulnerabilities in a secure PEE. Although the guest operating

system and applications of a virtual machine based secure PEE can be hardened if the virtual machine itself is vulnerable, the whole secure PEE is potentially vulnerable.

**Contemporary relevance, linkage with other papers and future direction:** With the increased use (from 2008 onwards) of virtualisation for portable computing environments (globull; 2010; MXI, 2008) the paper provided a relevant and timely analysis. The paper complements the existing research performed into virtual machine vulnerabilities (CSIRO, 2008; Ormandy 2007; Ferrie, 2007) through the development of countermeasures to the vulnerabilities with a focus on preventing data acquisition. The paper is linked to Paper 8 as it provides justification for a secure PEE consisting of a set of up-loadable applications<sup>28</sup>.

The analysis made a partial contribution to the research question ‘How can a useable and maintainable secure hardened operating system and/or small set of secure hardened applications be developed?’, through the elimination of virtualisation (due to security concerns). In this thesis virtualisation is considered to be an insufficiently secure tool from which to create secure PEEs. The paper suggested future work could include implementing the proposed countermeasures, however as the research moved away from the use of virtual machine based secure PEEs the work was not progressed.

#### **4.2.2 Modelling the Secure PESE Concept**

Three models are presented that collectively capture the security properties of the secure PESE concept. Justificatory knowledge for the models is derived from established conceptual design theory and expertise, in particular:

- Formal justificatory knowledge on threat modelling and information systems security design (Baskerville, 1993).
- Formal justificatory knowledge on conceptual design modelling of embedded systems (Gajski et al., 1994).
- The researcher’s systems design expertise, specialising in the area of high assurance and critical systems, providing informal justificatory knowledge.

---

<sup>28</sup> The limitations of virtual machines with respect to providing a secure PEE resulted in the research concentrating on up-loadable hardened applications and hardened bootable operating systems as the basis for secure PEEs.

The development of each model was initiated early in the research and finalised as the research progressed. The models form part of design cycle 2 as each was substantially developed before the design of the secure PESE artifacts. These models facilitate a high level design baseline through the diagrammatic refinement of the secure PESE concept and provide shape/form for a design. The benefits of the models are they:

- Can be used as a general reference model for the implementation of a remote working 'secure PESE like' system.
- Enable the understanding of the secure PESE concept through abstract modelling.
- Conceptually communicate the secure PESE design and mode of operation.

#### **4.2.2.1 Threat Model**

In Figure 4.1 a threat model portrays the threats identified as part of the holistic review of the remote work security issues (presented and discussed in Chapter 2). The threat model shows how the attributes of the secure PESE concept symbolise the security properties that counter vulnerability exploitation. A threat results in a security risk if a vulnerability is exploited.

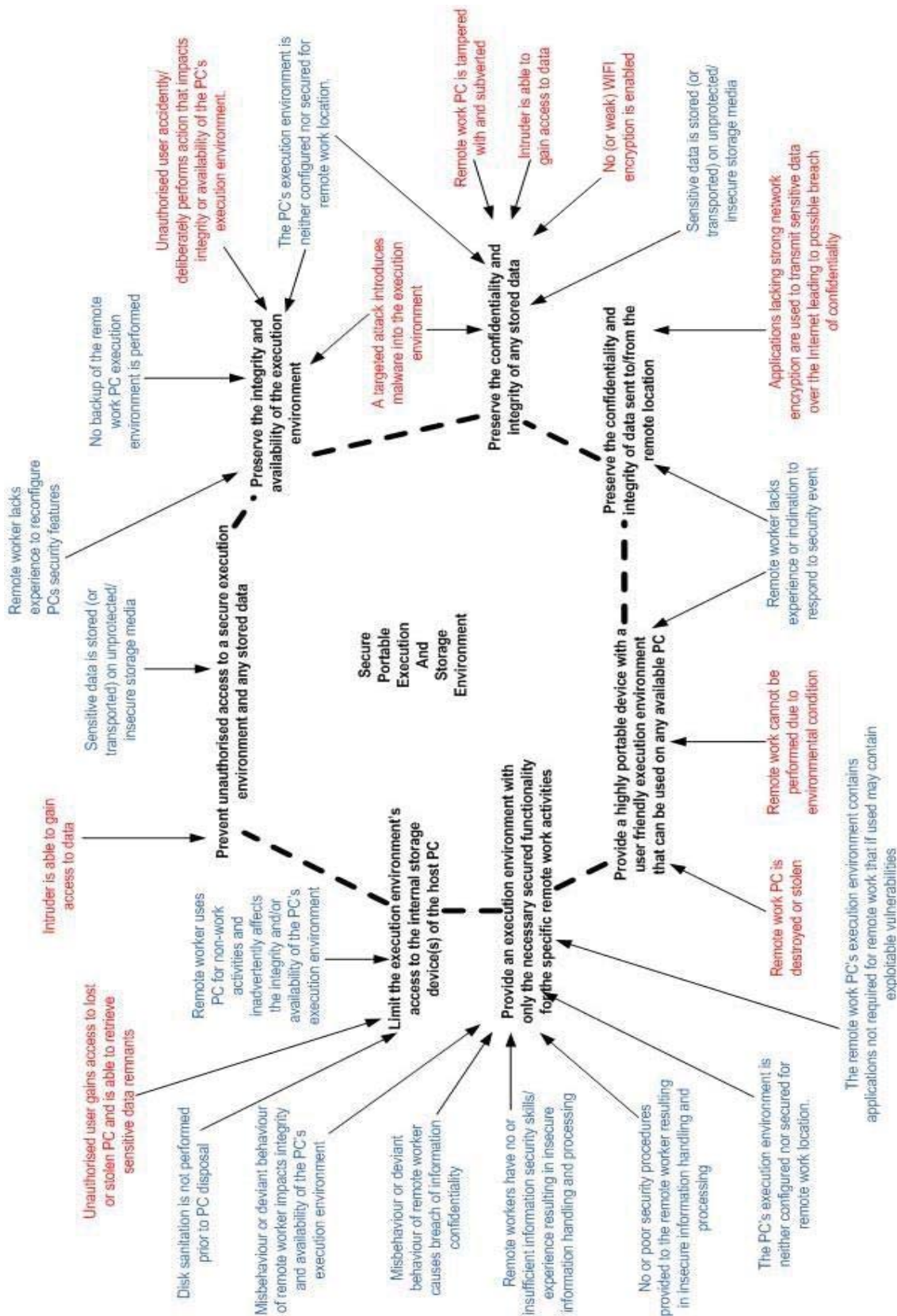
This research is focussed primarily upon managing the three risks of cyber-attack, data loss and forensic data discovery. In addition some threats can result in the risks of 'physical attack' and 'system unavailable'. Table 4.1 shows the association between the threats and the risks; with the concept attribute identified to highlight the security property that manages the risk.

#### **4.2.2.2 Conceptual Design Model**

Figure 4.2 diagrammatically presents a conceptual design model for a secure PESE. The model identifies the technology modules forming a secure PESE and provides architecture for secure PESE design.

#### **4.2.2.3 Operational Model**

Figure 4.3 diagrammatically demonstrates the secure PESE concept of operation, it models the enforcement of security prior to, and during operation.

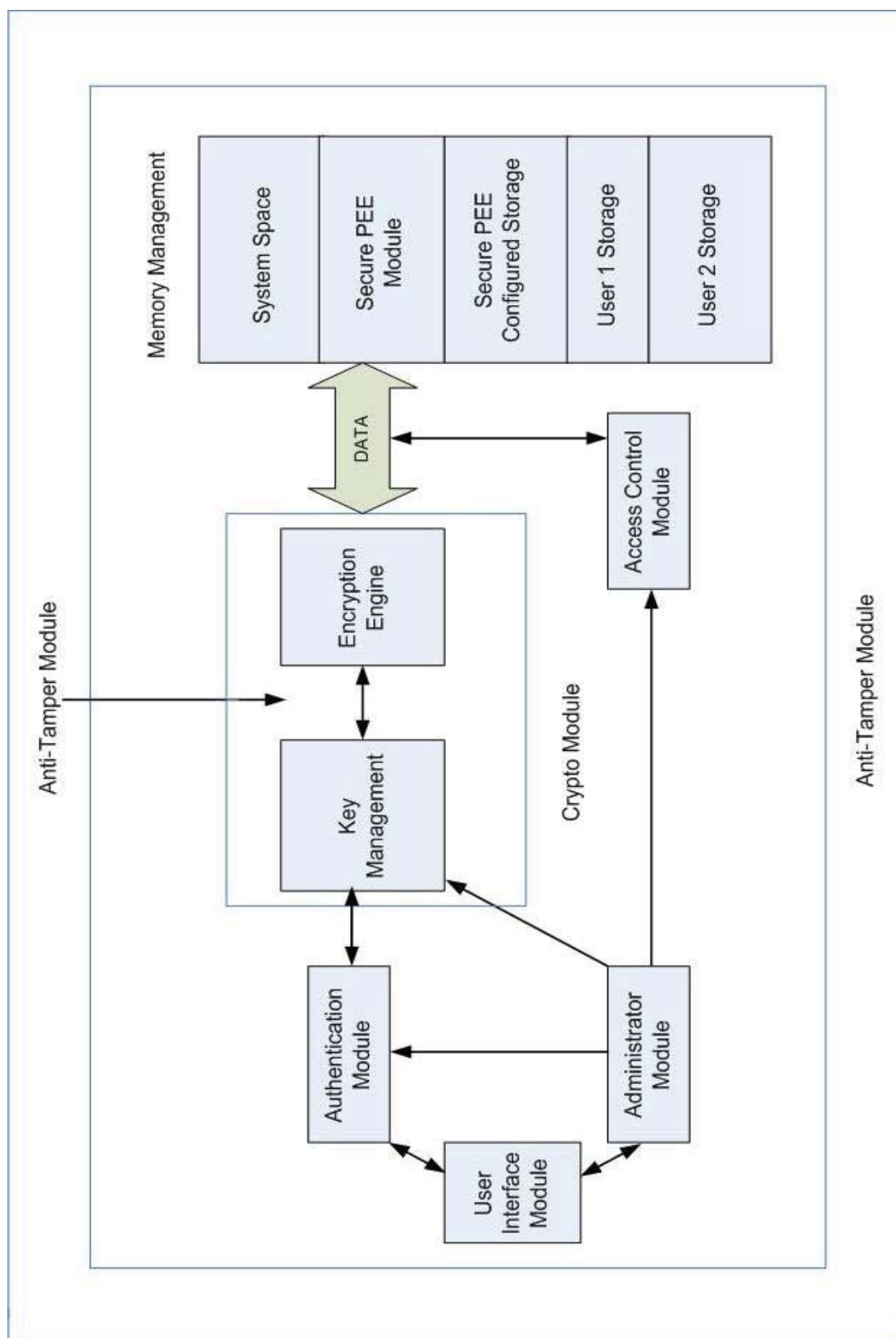


**Figure 4.1 Secure PESE Threat Model**

The remote work security threats are categorised as either external (in red) or internal (in blue) to the organisation. The concept attributes show how a secure PESE can be used to manage the information security risks in the remote work environment.

**Table 4.1 Association between Threats and Risks**

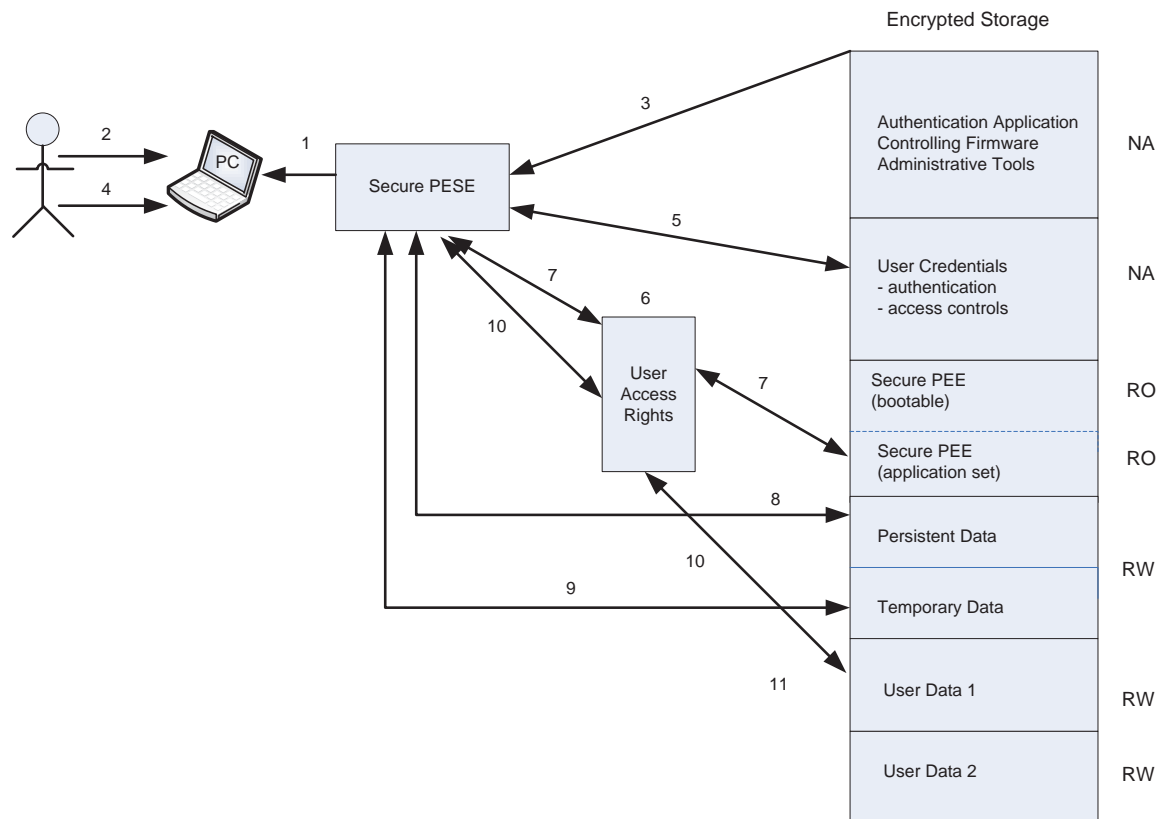
Threat	Risk	Attribute to Counter Vulnerability Exploitation
Sensitive data is stored (or transported) on unprotected/insecure storage media.	Data loss	Prevent unauthorised access to a secure execution environment and any stored data. Preserve the confidentiality and integrity of any stored data.
Remote worker lacks experience to reconfigure PC security features.	Cyber-attack	Preserve the integrity and availability of the execution environment
No back up of the remote work PC execution environment is performed.	Cyber attack System unavailable	Preserve the integrity and availability of the execution environment.
Unauthorised user accidentally/deliberately performs action that impacts integrity or availability of the PC's execution environment.	Cyber-attack System unavailable	Preserve the integrity and availability of the execution environment.
The PC's execution environment is neither configured nor secured for the remote work location.	Cyber-attack	Preserve the integrity and availability of the execution environment. Preserve the confidentiality and integrity of any stored data. Provide an execution environment with only the necessary secured functionality for the specific remote work activities.
A targeted attack introduces malware into the execution environment.	Cyber-attack	Preserve the integrity and availability of the execution environment. Preserve the confidentiality and integrity of any stored data.
Remote work PC is tampered with and subverted.	Physical attack	Preserve the confidentiality and integrity of any stored data.
Intruder is able to gain access to data.	Data loss	Preserve the confidentiality and integrity of any stored data. Prevent unauthorised access to a secure execution environment and any stored data.
No (or weak) WIFI encryption is enabled.	Cyber-attack Data loss	Preserve the confidentiality and integrity of any stored data.
Applications lacking strong network encryption are used to transmit sensitive data over the internet leading to possible breach of confidentiality.	Data loss Forensic data discovery	Preserve the confidentiality and integrity of data sent to/from the remote location. Provide a highly portable device with a user friendly execution environment that can be used on any available PC.
Remote worker lacks experience or inclination to respond to security event.	Data loss Cyber-attack	Preserve the confidentiality and integrity of data sent to/from the remote location.
Remote work cannot be performed due to environment condition.	System unavailable	Provide a highly portable device with a user friendly execution environment that can be used on any available PC.
Remote PC is destroyed or stolen.	System unavailable Forensic data discovery	Provide a highly portable device with a user friendly execution environment that can be used on any available PC.
The remote work PC's execution environment contains applications not required for remote work that if used may contain exploitable vulnerabilities.	Cyber-attack	Provide an execution environment with only the necessary secured functionality for the specific remote work activities.
No or poor security procedures provided to the remote worker resulting in insecure information handling and processing.	Data loss Cyber-attack	Provide an execution environment with only the necessary secured functionality for the specific remote work activities.
Remote workers have no or insufficient information security skills/experience resulting in insecure information handling and processing.	Data loss Cyber-attack	Provide an execution environment with only the necessary secured functionality for the specific remote work activities.
Misbehaviour or deviant behaviour of remote worker causes breach of confidentiality.	Data loss Cyber-attack	Provide an execution environment with only the necessary secured functionality for the specific remote work activities.
Misbehaviour or deviant behaviour of remote worker impact integrity and availability of PC's execution environment.	Data loss Cyber-attack	Provide an execution environment with only the necessary secured functionality for the specific remote work activities.
Disk sanitisation is not performed prior to PC disposal.	Forensic data discovery	Limit the execution environments access to the internal storage device(s) of the host PC.
Unauthorised user gains access to lost or stolen PC and is able to retrieve sensitive data remnants.	Forensic data discovery	Limit the execution environments access to the internal storage device(s) of the host PC.
Remote worker uses PC for non-work activities and inadvertently affects the integrity and/or availability of the PC's execution environment.	Data loss Cyber-attack System unavailable	Limit the execution environments access to the internal storage device(s) of the host PC.



**Figure 4.2 Secure PESE Conceptual Design Model**

*Each module represents policy enforcement functionality; arrows represent one or two-way communication between modules.*





**Figure 4.3 Secure PESE Operational Model**

*The eleven actions below describe the operation from plugging the secure PESE into a PC, to the user interacting with the executing secure PEE and reading/writing data:*

- 1. Plug-in secure PESE.**
- 2. If secure PEE to be loaded via pre-boot then power on PC – assumes first boot device is secure PESE.**
- 3. Upload and execute the Authentication Application.**
- 4. Authentication Application requests, user authentication credentials and passes to secure PESE.**
- 5. Secure PESE performs authentication.**
- 6. If successful authentication then user access rights are selected.**
- 7. If the user has the required access rights then the appropriate secure PEE is loaded into the PC and executed.**
- 8. As the secure PEE executes any configuration changes are written to the persistent data partition.**
- 9. The secure PEE writes page files/virtual memory to the temporary data partition.**
- 10. User processes data with secure PEE applications as required based upon access rights to data.**
- 11. All data sent to the secure PESE is encrypted with all data sent from the secure PESE decrypted.**

## Developing the Commercial Grade Secure PESE – Design Cycle 3

### 4.2.3 Background

A commercial grade secure PESE provides strong security suitable for commercial work where the protection of sensitive data is important. It satisfies the secure PESE concept and implements the functional requirements identified in Chapter 2.

The experimental secure PESEs described in Chapter 2 (Paper 4) contained secure PEEs based upon either virtualisation or a live CD bootable operating system. As described above, virtualisation was discounted from the research. Input from stakeholders combined with the researcher's own experience identified that in certain remote work situations it is not always possible to connect a secure PESE, power cycle the PC and boot a secure PEE. Therefore, an up-loadable set of hardened applications (that can be used on an available PC executing the Windows operating system) would allow remote work to be performed where booting is not possible. A secure PESE may therefore contain a secure PEE(s) consisting of:

1. A hardened bootable operating system with a small set of hardened applications; or
2. A set of up-loadable applications that execute on a PC running the Windows operating system; or
3. Both 1 and 2.

The commercial grade secure PESE described in this section was constructed from three research artifacts: Fireguard - a hardened browser that executes upon the Windows operating system (i.e. an up-loadable application-based secure PEE); the Mobile Execution Environment (MEE) - a hardened bootable operating system (i.e. a bootable secure PEE); and the Mini-SDV secure storage device that implements the SDV technology. The design and development of Fireguard and the MEE are described in Papers 6 and 7 respectively, whilst the design of the Mini-SDV draws upon the descriptions of SDV products in Papers 2, 4, 6 and 7. The development of the commercial grade secure PESE contributes to answering all three of the doctoral research questions.



## 4.2.4 A Secure Up-loadable Hardened Application

### 4.2.4.1 Preamble

After receiving input from stakeholders<sup>29</sup> a requirement was identified for a secure PEE consisting of a set of applications that are up-loadable from a secure PESE and executed under the control of the host PC Windows operating system. This set of applications could consist of specialist bespoke secure applications required to perform certain work activities (e.g. an emergency response command and control application for a deployed emergency response team) and/or generic applications (e.g. a browser or remote terminal client used for all forms of remote work). All applications in the set would be expected to be either developed specifically as a secure application or an existing application hardened to remove known vulnerabilities. Paper 6 describes the development of Fireguard, a hardened version of the Mozilla Firefox browser designed to manage the risk of cyber-attack, data loss and forensic data discovery. Fireguard is designed to be up-loaded from a secure PESE to a host PC executing Windows to allow access to web servers.

Fireguard is an artifact developed using the DSRM (Peffer et al., 2007). Although the paper neither mentions nor discusses the use of the DSRM, the structure of the paper follows the methodology's activities, i.e. the problem is outlined and the motivation for Fireguard is considered, specific objectives (portrayed as requirements in the paper) for the artifact are enumerated, the design and implementation is presented, the approach used to demonstrate (test) the artifact is discussed and the artifact is evaluated in the paper conclusion.

The paper makes a distinction between a remote worker and mobile worker as it was conceived that Fireguard would be utilised by mobile workers on PCs outside the control of an organisation's ICT governance. The paper makes a contribution to answering the following two doctoral research questions:

*How can a useable and maintainable hardened operating system and/or a small set of hardened applications be developed?*

---

<sup>29</sup> The stakeholders were customers of the researcher's employer who required secure applications to use in a known environment, but where the host PCs were not owned by the stakeholders and therefore could not be fully trusted.

*How can a useable and maintainable execution environment be configured to store all temporary data on a secure PESE partition?*

The construction of Fireguard shows how Mozilla Firefox can be hardened and configured to store temporary data on the secure PESE yet still remain a useable and functional browser.

The term secure PEE (to refer to set of up-loadable secure hardened applications) is not used in the paper, instead Fireguard is referred to as a trusted application. It was decided that as the paper describes a single application (rather than a set) a secure PESE would be presented as a secure storage device packaged with a trusted application to enable secure remote and mobile working.

#### **4.2.4.2 Prior Research and Knowledge**

Prior research identified in 2009/2010 (when the Fireguard artifact was constructed) included work to develop a secure browser (Grier et al., 2008), a comprehensive handbook on browser security (Zalewski, 2009) and development of hardened browsers (Gühring, 2006; Ronchi and Zakhidoc, 2009). This prior research, in particular the browser security handbook (Zalewski, 2009) assisted in identifying and understanding browser vulnerabilities. No prior research was identified that specifically considered the development of a secure hardened general purpose browser artifact that utilised the security mechanisms of a secure storage device to provide a secure PESE capability.

The Fireguard artifact packaged with the Mini-SDV (MiniSDV, 2009) provides a secure PESE that satisfies the research objectives (and hence the concept attributes) and the (relevant) functional requirements, and contributes to answering the doctoral research questions. The following knowledge was consumed during the design and development of Fireguard:

- Descriptive knowledge on browser vulnerabilities. A range of publications are cited in the paper that identify known vulnerabilities and attack scenarios (as of 2010) in browsers. Whilst not always directly applicable to Firefox this knowledge was used to reduce the vulnerabilities in the resultant Fireguard artifact.
- Prescriptive knowledge in the form of the secure PESE design and operational models (Figures 4.2 and 4.3), and a small subset of Fireguard requirements derived from both

the research objectives and the functional requirements identified in Chapter 2. Although still functional, the Fireguard derived requirements are at a (lower) developmental level so as to direct the implementation of the hardened browser.

- Prescriptive knowledge consisting of the Firefox design and source code, and Mini-SDV documentation. Like most freeware, Firefox lacks comprehensive and maintained design documentation. To supplement the documentation available from Mozilla the researcher and co-author found it necessary to read the configuration files, source code and search web sites for postings made by interested parties (which are cited in the paper).
- Informal justificatory knowledge in the form of the co-author and the researcher's software development and security hardening experience providing the inspiration to find workable solutions to Firefox vulnerabilities.
- Formal justificatory knowledge on secure browser design (Grier et al., 2008; Zalewski, 2009; Gühring, 2006; Ronchi and Zakhidoc, 2009) provided ideas and inspiration.

#### 4.2.4.3 Paper 6

**Paper 6** - Griffiths, D & James, P. (2010) **Fireguard – A Secure Browser with Reduced Forensic Footprint**, *The Journal of Network Forensics*, Volume 2, Issue 2, pp 1-24.

#### Abstract

*Fireguard is a secure portable browser designed to reduce both data leakage from browser data remnants and cyber-attacks from malicious code exploiting vulnerabilities in browser plug-ins, extensions and software updates. A browser can leave data remnants on a host PC hard disk drive, often unbeknown to a user, in the form of cookies, histories, saved passwords, cached web pages and downloaded objects. Forensic analysis, using freely available computer forensic tools, may reveal sensitive and confidential information. A browser's capability to increase its features through plug-ins and extensions and perform patch management or upgrade to a new release via a software update provides an opportunity for an attacker to embed malicious software and subsequently launch a cyber-attack.*

*Fireguard has been implemented using both Mozilla Firefox and the storage and protection capabilities of the Mini-SDV, a secure Portable Execution and Storage Environment (PESE). In this paper the design and development of Fireguard is discussed. The requirement for a secure PESE and the functionality of the Mini-SDV is presented. An overview is given of the motivation for the development of Fireguard. The reasons Firefox was selected and the Firefox structure and security vulnerabilities are summarised. The implementation approach adopted is discussed and the results*

*of an analysis of the Firefox implementation are presented. The Mini-SDV configuration for Fireguard and an outline of the concept of operation is given. The changes made to Firefox to implement Fireguard as a browser that reduces the opportunity for data leakage and cyber-attack, and minimises its forensic footprint are discussed. The paper concludes by considering the strengths and limitations of the Fireguard implementation.*

## **Keywords**

Secure Browser, Secure Portable Execution and Storage Environments, Mozilla Firefox, Mini-SDV, Computer Forensics, Anti-forensics.

## **Introduction**

Organisations that enforce best practice ICT security, and with a workforce consisting of remote and mobile<sup>30</sup> workers, often restrict the use of portable storage media and the use of remote connections to corporate servers due to security concerns relating to data leakage and cyber-attacks. The risk of data leakage (NTI, 2004) from loss of portable media and/or the recovery of sensitive data remnants from hidden storage areas on PCs, and cyber-attack from malicious software (malware) introduced onto a corporate system from the Internet and/or portable media (Moscaritolo, 2008) are amongst some of the key security issues confronting organisations (Lemos, 2010; McAfee 2010; Ponemon, 2009). Secure portable storage media (TrueCrypt-Foundation, 2010; IronKey, 2010) has enabled organisations to define policies for the secure transport of sensitive corporate data by remote and mobile workers. However, these products do not provide strong protection for trusted applications which can be uploaded and executed on a host PC to enable secure remote and mobile computing.

## **Secure Portable Execution and Storage Environment (Secure PESE)**

A secure PESE provides both a solution for the transportation of sensitive data and the provision of a trusted application to enable secure remote and mobile computing. The secure PESE is recognised as a way to prevent data loss and cyber-attack by organisations who process sensitive data, yet want to achieve cost savings and productivity gains from allowing remote and mobile working. A secure PESE trusted application can be defined as an environment that limits data leakage, cannot be compromised by malware and leaves no forensic footprint on the host PC upon which it executes, i.e. no evidence exists on the host PC hard disk drive (HDD) that the trusted application has executed on the PC. A secure PESE trusted application can be a bootable environment that is uploaded from the secure PESE when the PC is powered-on, or an application that can be uploaded to a host PC with its own booted and executing operating system (OS).

A functionally strong secure PESE (James, 2008) will include a USB or an eSATA interface, different modes of operation to suit different operational environments, strong pre-boot and post-

---

<sup>30</sup> In this paper a remote or mobile worker is considered to be an individual who works in a location where the physical and logical security controls do not necessarily satisfy the organisational IT security policy defined for the corporate IT environment.

boot authentication, hardware based encryption of the device's storage medium, a choice of trusted applications (e.g. bootable or executable on the host PC OS), protection mechanisms for the trusted applications to protect against malware, and a storage area for temporary data (to prevent data remnants residing in hidden storage areas).

### **Mini Silicon Data Vault (Mini-SDV)**

The Secure Systems Mini-SDV (Secure Systems, 2010) is a USB attachable secure PESE with the following functionality:

- Two modes of operation; Fully Trusted and Assured.
  - Fully Trusted mode allows an authenticated user to 'boot' a trusted application from a Mini-SDV connected to a host PC. The trusted application only uses the CPU and RAM of the host PC and leaves no forensic footprint.
  - Assured mode allows an authenticated user to execute a trusted application (the Fireguard browser) from a Mini-SDV connected to a host PC that is executing the Windows OS. Fireguard leaves a minimal forensic footprint.
- Hardware based encryption of the storage medium.
- Partitioning of storage medium. Partitions can be defined as Read-Only, Read-Write and No-Access.
- User/role profiles with differentiated access rights.
- Storage area for temporary data created by the trusted application.

The Mini-SDV functionality binds together to enable the device to be configured to provide a strong and secure PESE for remote and mobile computing. The Mini-SDV functionality is configured to counter the remote and mobile computing risks of cyber-attack and data leakage by:

- Securing a trusted application in a Read-Only partition that prevents malware becoming embedded into the application and hence the exploitation of the malware to launch a cyber-attack.
- Full encryption of the integral storage medium to protect the confidentiality of data and prevent data leakage if the Mini-SDV is lost or stolen; and
- Storing temporary copies of data created by the trusted application in an allocated storage area on the Mini-SDV which prevents data leakage through the recovery of the data remnants.

The Mini-SDV hardware and firmware provide the platform to counter cyber-attack and data leakage, but trusted applications are also required that utilise the Mini-SDV's functionality to provide a fully functional secure PESE. The Mini-SDV trusted applications available in Fully Trusted mode consist of a minimal functionality OS, secure browser and secure Windows terminal client. Fully Trusted mode is used when no trust can be asserted about the host PC. This mode of operation is

not considered further in this paper. In Assured mode a secure browser, Fireguard, is available. The Assured mode should be used when either a level of trust can be asserted about a PC or if an untrusted PC is used to perform limited Internet transactions through a secure browser. In this paper the development of the Assured mode secure browser Fireguard is considered. In particular the paper discusses how an open source browser was modified to:

1. Allow it to be installed on to, and be protected by, a Mini-SDV.
2. Disable features that could allow malware to become embedded in the browser.
3. Prevent any temporary, configuration or reusable data generated by the browser being written to the host PC's HDD.

As a result of the above three modifications a secure browser with reduced forensic footprint was produced that limits the opportunities for both cyber-attack and data leakage.

## **Fireguard Project – Motivation and Approach**

### **Motivation and Requirements**

The motivation for the development of Fireguard was driven by the requirement for a remote or mobile worker to be able to use an available PC and connect to the Internet with the assurance the session would not be compromised. For instance, if a remote or mobile worker accessed a web site, through a browser available on a public access PC, there is a possibility that the browser may have been compromised through malware caused by infected plug-ins, extensions<sup>31</sup>, Javascript<sup>32</sup> or software updates. This browser will also save a range of temporary, configuration and reusable data that can remain on the PC in hidden directories after the browser session has completed and may be recovered, using basic forensic tools, to reveal information about the user.

Two important design objectives were that the user could neither add functionality to the browser through plug-ins, extensions and software updates (to ensure the user does not inadvertently introduce malware into the browser), nor allow the user to accidentally cause sensitive data to remain on the host PC. Fireguard was therefore developed to provide a secure browser that could neither be compromised by malware exploiting browser vulnerabilities, nor allow data remnants to be recovered that could reveal details about the user.

It was decided that Fireguard should be a browser that executes securely on the Microsoft Windows range of OS'. The following functional requirements were identified for the Fireguard project:

1. To prevent malware and hence cyber-attack it must be executable from a Mini-SDV Read-Only partition.

---

<sup>31</sup> Plug-ins and extensions are sometimes used to describe the same capability. Both browser plug-ins and extensions are additional software components created after the release of the browser that increases the browser's functionality. A plug-in can be defined (and differentiated from an extension) as implementing a narrow specific feature. Whereas an extension implements a new capability or modifies an existing capability. Once integrated, extensions form part of the browser's functionality; an implemented extension can provide the basis for new plug-ins to be added.

<sup>32</sup> Javascript is a language used to add interactive features to webpages and can be executed within the browser.

2. To prevent malware and hence cyber-attack it must prevent plug-ins, extensions and software updates.
3. To prevent forensic discovery it must not update the Windows registry<sup>33</sup>.
4. To prevent data leakage and forensic discovery it must store configuration data and temporary files in a nominated storage area on the Mini-SDV.
5. To prevent data leakage and forensic discovery it must not write to the Host PC HDD, with any created temporary data being written to the Mini-SDV and deleted when the browser terminates.

Implementing these five functional requirements will protect the integrity of the browser's source code, prevent malware becoming embedded into the source code, prevent vulnerabilities in the browser being exploited, prevent data leakage and as far as is practical leave no forensic footprint.

### **Browser Selection**

An investigation was performed into popular browsers to determine if any of these browsers could be configured to achieve the five requirements (defined above) without having to make changes to the source code and user interface, i.e. achieving the five requirements through a browser's user configuration capability. None of the available proprietary browsers provide a configuration capability to achieve the desired outcomes. Whilst configuration settings could enable a number of the requirements to be satisfied, it was determined that changes to a browser's source code and user interface code were necessary to implement all of the five requirements. However, for proprietary browsers access to the source code is not possible, for example Microsoft Internet Explorer (IE) is Microsoft proprietary software and access to its source code is not available. Also a key user requirement for the secure browser was that it should be configurable to have a look and feel similar to the IE browser, because IE is the most popular browser available with a 62% market share (NetMarketShare, 2010).

The development was therefore restricted to open source browsers; Mozilla Firefox and Google Chrome were both considered. It was decided to use Mozilla Firefox because it could be configured to have a similar look and feel to IE, had a number of built-in security features and at the time of the development planning, in late 2009, Google Chrome had been formally released only ten months earlier. At the time development work commenced in January 2010, Mozilla had released its stable Firefox version 3.6.

An obvious version of Firefox to consider was the Firefox Portable Edition as it has been designed to be uploaded from portable storage media and execute on a host PC. However, Firefox Portable Edition modifies the Windows OS registry, which is a necessary aspect of its implementation. Firefox Portable Edition was therefore not selected because registry modifications leave evidence of browser use. It was decided that Fireguard would be based upon Mozilla Firefox version 3.6

---

<sup>33</sup> The Windows registry is a database of configuration settings and global constants/variables for both the OS itself and applications that execute on the OS.



(Mozilla 2010). The Mozilla Foundation licence requires that once the source code has changed the resultant product is no longer a Mozilla product and therefore this resultant product cannot use the Firefox name and logo. The name Fireguard was selected as a 'security spin' on the name Firefox.

## An Overview of the Firefox Implementation

### Firefox – Structure and Vulnerabilities

The first version of Firefox, version 1.0, was released in November 2004. By 2010, Firefox was rated as the second most popular browser in the world with a 24% market share (NetMarketShare, 2010). Firefox is written predominately in C++ with user interfaces written in a mixture of XUL<sup>34</sup> (MDN, 2010a), XBL<sup>35</sup> (MDN 2010b), Javascript (MDN 2010c) and CSS<sup>36</sup> (PHPforms, 2010).

Firefox utilises an installer to create the Firefox folder structure, establish the required registry entries and install the Firefox binary, configuration files and other associated files and data onto the PC HDD. Firefox is a functionally rich browser that allows a user to individually tailor the browser through the addition of extensions and plug-ins which can be added after the browser has been installed.

Firefox provides a good level of built-in security capabilities including:

- Phishing<sup>37</sup> detection – the browser maintains and constantly updates a list of known fraudulent web sites, warning the user when the site is about to be accessed.
- Malware detection – the browser warns of web sites known to contain malware. Also Firefox can integrate with anti-virus software installed on the PC to scan any downloads performed using the browser.
- Master Password – the browser allows passwords required by specific web sites to be retained so subsequent rapid logons are achieved. To protect these passwords from other users a master password can be set.
- Site identity: Allows the user to confirm a web site is genuine and confirm the level of security enforced.

Whilst these built-in security capabilities provide good security for general browsing, Firefox is still susceptible to cyber-attack and data leakage through the following vulnerabilities:

- *Extensions:* Firefox extensions are programs typically written in Javascript. An example extension is the language translation capability ImTranslator (Smart Link Corporation, 2010).

---

<sup>34</sup> XUL (XML User Interface Language) is a markup language; it is an application of XML that defines various user interfaces elements.

<sup>35</sup> XBL (XML Binding Language) is a markup language; it is used to define the behaviour of XML elements.

<sup>36</sup> CSS (Cascading Style Sheets) is a language used to describe the formatting of a document written in a markup language.

<sup>37</sup> Phishing is the fraudulent act of masquerading as a genuine entity with the intent to acquire sensitive information from a user.



Most Firefox extensions do not originate from the Mozilla organisation and have been used (accidentally or intentionally) on occasions to introduce security vulnerabilities into the browser (Barth, A., A. Felt, et al., 2010; Help Net Security, 2009).

- *Plug-ins:* Firefox plug-ins are dynamic link libraries (DLLs) written to conform to an application programming interface (API) provided by Mozilla. An example of a plug-in in common use is the Adobe Flash player (Adobe 2010). Browser plug-ins have been identified as a source of security vulnerabilities (Symantec, 2010).
- *Software updates:* Firefox automatically fixes functionality problems and security vulnerabilities, and when a new version of the browser is released, the user is given the option to upgrade. Recently some research has highlighted how the software update process may be vulnerable to the introduction of malware (Cnet News, 2009).
- *Javascript:* Javascript has been the source of browser vulnerabilities; clickjacking<sup>38</sup> is a good example of a malware technique that utilises Javascript (Zalewski M, 2009). Firefox provides the option to disable Javascript so that a web page using Javascript will not interact with the browser. However, disabling Javascript in Firefox can significantly reduce the range of web sites that can be accessed.
- *Media Autoplay:* Firefox allows detected audio and video media files on a web site to automatically play. Media files can be a source of malware; also the media player launched to play the media file may be compromised. (Symantec 2010)
- *Java Applets:* A number of web pages use Java applets<sup>39</sup>, particularly for interactive content. Firefox allows Java applets to execute if both the Java Runtime Environment has been installed on the host PC and (more recently) if a special browser plug-in has also been installed. Java applets can be a source of malware. (Reynaud-Plantey, 2005)
- *Downloaded Objects:* When an object is downloaded from a web site Firefox will place it in a default location if a specific storage folder is not nominated by the user. If the default storage location is used and the user does not remove the object at the end of the browser session then the object may be accessed at a later date (by an unauthorised individual) and reveal sensitive information about the user.
- *Residual data:* Firefox, like all browsers, retain cookies<sup>40</sup>, site visit histories, saved passwords, form data<sup>41</sup> and cached web pages. This residual data, although hidden, is vulnerable to recovery by a knowledgeable user to reveal details about the user and his use of the browser, thus the browser can facilitate data leakage.

---

<sup>38</sup> Clickjacking is a malicious software technique used to trick a user into performing undesired actions by clicking on a concealed link. For example the technique is used to show a set of dummy buttons which overlay another page; the user thinks he is clicking the visible buttons but is actually performing actions on the hidden page.

<sup>39</sup> Java applets are small applications that perform a specific task; they provide web applications with interactive features that cannot be provided by HTML. Applets execute within the context of the browser or a plug-in.

<sup>40</sup> Cookies are information placed on the browser's PC by the web site and that the browser provides to the web site upon subsequent visits. A cookie can contain a variety of details.

<sup>41</sup> Form data is the data entered into a web page which can be saved in Firefox and re-used automatically at a later date to pre-complete a form.

Fireguard was developed to prevent the exploitation of the above vulnerabilities.

## Firefox Functionality Configuration Files

Firefox utilises multiple configuration files to configure functionality, most of which are located in sub-folders in the folder in which Firefox is installed. Firefox configuration files are generally Javascript files with the extension “.js” and contain calls to the function “pref” that is used to set preferences, e.g. `pref("browser.cache.offline.capacity", 512000)`; sets the maximum capacity of the Firefox offline cache.

In addition to the configuration files in the Firefox installation folder, there is a file “*prefs.js*” located in the user profile. This is created from user preferences set either through the Tools >Options and other Firefox menus or by using the `about:config`<sup>42</sup> uniform resource locator (url) in the Firefox address bar. The settings in “*prefs.js*” may override settings from the configuration files in the Firefox installation folder unless a preference has been locked.

It is possible to direct Firefox to always load a custom preferences file by creating a *loadcustom.js* script that points to another custom script file. This custom script file is located in the Firefox installation folder and by convention is named *firefox.cfg*. Preferences may be set in the *firefox.cfg* file using the function “*lockPref*” (mozillaZine (2010), e.g. `lockPref("privacy.sanitize.sanitizeOnShutdown", true)`; clears all data accumulated during a browser session including histories, form and search data, cookies and active login data. The “*lockpref*” function can be used on a range of preferences to prevent specific settings from being modified.

*Lockpref*, like “*pref*”, sets a preference value but prevents the user from altering the setting via menus or the `about:config` feature, and any corresponding settings in the *prefs.js* file are ignored. The user will see a greyed-out menu option or locked setting in the `about:config` display. The “*lockPref*” method of preference locking is best used in an environment where Read-Only access may be enforced to the program folder to prevent a user from editing the custom preference file, e.g. the Linux OS that supports Read-Only directories or a device like the Mini-SDV which enforces Read-Only partitions.

## Firefox User Interface Configuration Files

Most of the Firefox user interface is implemented in the XUL. XUL uses XML, XBL, Javascript and CSS to specify the interface elements. The XUL files are concatenated into jar<sup>43</sup> archives and stored in the “*chrome*” folder of the Firefox program directory. The XUL files may be extracted and modified to change the interface without having to recompile the Firefox source code.

---

<sup>42</sup> The `about:config` url allows the Firefox configuration settings to be viewed and modified within the browser itself.

<sup>43</sup> A jar archive combines a number of files into a single compressed archive file.

## Analysis of Firefox Implementation

### Analysis of Firefox Source Code

Firefox has been written to be a multi OS application and includes “conditional defines” for the code to be built under Microsoft Windows, Apple OS X and Linux/Unix running X-Windows. Therefore understanding how Firefox interacts, utilises and depends upon an OS is important. An analysis was performed on how Firefox interacted with the Windows OS file system and registry; in particular an understanding was required on the Firefox dependencies for constant special item ID list (CSIDL<sup>44</sup>). The Firefox input and output functions were analysed for Microsoft Windows use of CSIDL and environment variables to identify the locations where Firefox read and wrote Firefox user profiles<sup>45</sup> and temporary files. An analysis was performed of the use of the CSIDL locations in the source code to gain an understanding on how to change the functionality that read and wrote browser generated data. The objective of the analysis was to identify all Windows OS file system and registry dependencies in the source code so that the appropriate code could be changed to ensure no writes occurred to the file system or registry. The analysis identified the following:

- Multiple entries in the registry are created during installation including but not limited to:
  - Identifying installation directory, execution path and location for shared extensions.
  - Specifying additional file types that can be processed by the browser.
  - Identifying the uninstall activities.
- The Firefox installer creates an application folder in the location pointed to by CSIDL\_PROGRAM\_FILES.
- A shortcut to the *firefox.exe* is created in the location pointed to by CSIDL\_DESKTOP.
- On the first execution of Firefox (i.e. the *firefox.exe* program) by a user, a Mozilla folder and a Firefox user profile are created in the location pointed to by CSIDL\_APPDATA .
- Plug-ins provide their own installer and will create their own install folders and registry entries.
- At the start of browser execution a search is performed for available plug-ins.
- Firefox software updates are written to the program directory (CSIDL\_PROGRAM\_FILES) to replace the updated executables.
- Firefox stores cache data in a location pointed to by CSIDL\_LOCAL\_APPDATA.
- Firefox stores temporary data in the location pointed to by the system environment variable TEMP, which is generally mapped to a location in CSIDL\_LOCAL\_APPDATA.
- When a Firefox extension is installed, it is copied to the Firefox profile or if the extension is to be shared with other user instantiations of the browser then it is installed in a custom folder (this

---

<sup>44</sup> A CSIDL is a Windows feature that provides a unique way to identify special folders and variables, but which may not have the same name or location on any given system.

<sup>45</sup> A user profile is a set of files containing user browser information including bookmarks, site visit histories, user preferences and passwords.

type of installation creates a registry entry so that Firefox knows where to locate and load the extension).

- Firefox allows the user to select an image from the Internet to save and set as the desktop background image, which involves a write to the registry and the copying of the image to the Mozilla/Firefox directory in the user's home directory.

The analysis provided a good understanding of the Firefox implementation and how changes could be made to develop a secure browser.

## Implementation Approach

Implementation of the five requirements identified above required both changes to the Firefox configuration settings/files and modification to the Firefox source code. Many of the required changes could be made without changing the Firefox source code and therefore where possible, changes were performed by using configuration settings. However, it was necessary to make changes to both the browser's C++ source code and the XUL user interface. All work was performed on a development platform with the following implementation approach used:

*Configuration Settings:* The *Fireguard.cfg* file was created which contained a series of "*lockPref*" functions to disable functionality. A *loadcustom.js* script was also created that pointed to the *Fireguard.cfg* file to enable Fireguard to disable the required functionality upon start-up. The *Fireguard.cfg* file was co-located with the Fireguard executable in the Fireguard installation folder on a Mini-SDV Read-Only partition to prevent user changes.

*C++ Source Code Changes:* A C++ development and test environment was established to enable source code changes to be made, compiled and tested.

*XUL User Interface Changes:* The user interface files were extracted from jar archives, changes were made and placed back into the jar archives; compilation of the changed XUL files was not necessary.

In a number of cases the source code was changed to completely remove a feature and also the respective feature user configuration settings were disabled for both completeness and to ensure the user could not try to enable a feature that had been removed. The use of the "*lockpref*" function to disable functionality defined in the *Fireguard.cfg* file and the integrity protection mechanism of the Mini-SDV Read-Only partition enabled many of the data leakage and forensic recovery vulnerabilities to be addressed; supported by only minimal source code changes.

Once all the development and testing was complete, the Fireguard binary, configuration files and other associated files and data held on the development platform were imaged onto the Mini-SDV.

## Implementing Fireguard

### Key Design Decision

Although a source of vulnerability it was decided not to disable Javascript in Fireguard for the following reasons:

- Javascript is used extensively by web sites. Disabling the capability in Fireguard would restrict the sites that could be accessed, resulting in both a browser usability issue and a limitation during testing.
- The primary objective of Fireguard is as a secure portable browser to enable remote and mobile workers to access (predominately) trusted corporate systems from untrusted PCs. The use of Javascript by trusted corporate web sites would not be expected to compromise Fireguard.

Preventing a user adding functionality and protecting Fireguard's integrity through the use of a Mini-SDV Read-Only partition capability will prevent hostile Javascript becoming embedded in the browser.

It is acknowledged that retaining Javascript presents a risk to exploit browser vulnerabilities and hence launch a cyber-attack, but for the first release of Fireguard it was considered to be a risk that could be managed.

### Mini-SDV Configuration

The Fireguard executable is installed in a Read-Only partition on the Mini-SDV. Fireguard was imaged from the development platform; the Firefox installer was not used. Once all of the source code and configuration changes (discussed below) were applied and tested on the development platform, the executable and configuration files were imaged on to a Mini-SDV Read-Only partition (labelled the **Fireguard Browser Partition**). The **Fireguard Browser Partition** can be formatted as either a FAT32<sup>46</sup> or NTFS<sup>47</sup> file system.

A temporary Read-Write data partition, the **Fireguard Data Partition**, is created on the Mini-SDV to store the user profile and configuration parameters created by Fireguard. A separate Read-Write partition, the **User Data Partition**, is created for user data to be transported and/or processed. The **User Data Partition** provides the large secure PESE mass storage capability.

---

<sup>46</sup> FAT32 (File Allocation Table) is a relatively simple file system format used for storage devices. Since the introduction of NTFS, FAT is now predominately used on portable devices rather than PC HDDs.

<sup>47</sup> NTFS (New Technology File System) is a Microsoft proprietary file system format.

Figure 1 presents a conceptual model of the Mini-SDV configuration for Assured mode and can be summarised as:

- **Fireguard Browser Partition:** A Read-Only partition that protects and contains the Fireguard executable and configuration files.
- **Fireguard Data Partition:** A Read-Write partition that allows certain browser configuration parameters to be retained and any temporary data necessary for the correct browser operation. The Firefox

user profile folder structure is contained in this partition.

- **User Data Partition:** A Read-Write partition for user data.

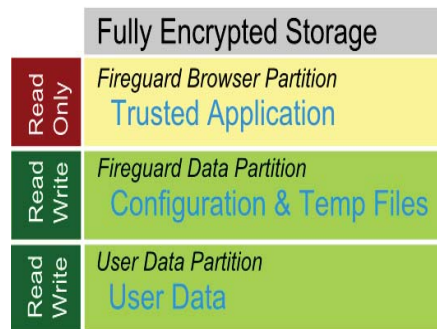


Figure 1: Conceptual Model of Mini-SDV Configuration

The Fireguard Data Partition contains a file structure that mirrors the Firefox file structure created for each user, i.e. a folder structure mimicking a Windows user home directory was created. This folder structure would contain the user profile and temporary data created during browser execution. Figure 2 presents an image of the Fireguard Data Partition hierarchical file structure. By mimicking the user directory structure the level of source code changes could be contained.

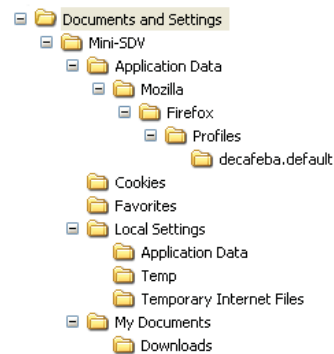


Figure 2: Structure of Fireguard Data Partition

## Concept of Operation

In Assured mode, when the Mini-SDV is plugged into a USB port on a host PC executing the Windows OS an AutoRun<sup>48</sup> is detected and an authentication application is executed. Upon successful authentication Fireguard can be selected; upon selection it is uploaded into the host PC to enable secure remote access to the Internet. Access to data stored on the Mini-SDV is also available.

*Removing file system and registry dependencies:* All CSIDL pointers to PC file system locations were changed to point to locations on the Mini-SDV **Fireguard Data Partition**. A folder structure mimicking a Windows user home directory was created (see Figure 2). When the modified code was executed from the **Fireguard Browser Partition**, it populated the empty Mozilla Firefox user profile folder on the Mini-SDV **Fireguard Data Partition** with its required configuration content.

<sup>48</sup> AutoRun is a Windows OS feature that defines a set of actions to be performed when a drive/volume/partition is mounted; these actions are specified in the file autorun.inf held in the root partition of the drive.

More specifically, once successfully authenticated the Mini-SDV authentication application accesses a custom MBR<sup>49</sup> on the Mini-SDV to retrieve the partition table.

The Windows OS assigns a drive letter for each Mini-SDV partition (defined in the partition table) containing FAT32 or NTFS file systems. Firefox determines its execution path from the registry value CSIDL\_PROGRAM\_FILE and the location of the Firefox user profile from the registry value CSIDL\_APPDATA. However, as Fireguard does not use the registry values, because the Fireguard binary/configuration/interface files and user profile are stored in the **Fireguard Browser Partition** and the **Fireguard Data Partition** respectively, Fireguard needs to determine the drive letters for these partitions. An important design aspect of Fireguard is the ability to determine the drive letters allocated to the **Fireguard Browser Partition** and the **Fireguard Data Partition**.

Upon logging-off the Mini-SDV, no further access to the device is possible without re-authentication.

### Important Source Code Changes

Two key source code changes were made; removing dependencies on the PC file system and registry, and drive letter determination for the **Fireguard Browser Partition** and **Fireguard Data Partition** to ensure drive letter conflict does not occur.

*Drive letter determination:* As Fireguard does not use CSIDL values the following approach was implemented (by making source code changes) to determine the drive letters allocated to each partition:

- **Fireguard Browser Partition:** When the Windows OS creates a process to execute an application it creates a parameter block which includes the application's full file path name, including the allocated drive letter. Therefore when Fireguard is executed it can determine the drive letter for the **Fireguard Browser Partition** by querying its process parameter block.
- **Fireguard Data Partition:** The Mini-SDV is configured so that its first partition is always the **Fireguard Browser Partition**, the second partition is the **Fireguard Data Partition** and the third partition the **User Data Partition**. Logically, it could be assumed that the **Fireguard Data Partition** is allocated the next drive letter after the drive letter allocated to the **Fireguard Browser Partition**. However, this may not be the case as Windows does not necessarily allocate drive letters sequentially. For example if a specific driver letter (e.g. H) is already allocated to a network share and if the drive letters C to F are currently allocated, then when the Mini-SDV is plugged into the PC the **Fireguard Browser Partition** will be allocated drive letter G. However, as drive letter H is already allocated another drive letter (e.g. I) will have been allocated to the **Fireguard Data Partition** and Fireguard needs to identify this drive letter to retrieve the user profile. To determine the drive letter allocated to the **Fireguard Data Partition**

---

<sup>49</sup> MBR (Master Boot Record) is the first sector of a partitioned data storage device that contains the partition table and may bootstrap an operating system if resident on the device.



requires Fireguard to query the Windows OS structure containing partition table details (extracted from the Mini-SDV MBR) and the respective drive letter allocated to each partition.

- **User Data Partition:** Like the **Fireguard Data Partition** the drive letter for the **User Data Partition** is determined by querying the Windows OS structure containing details on each physical partition and the respective allocated drive letter.

A number of more minor source code changes were also made which are identified below together with all the configuration changes made to protect against data leakage, cyber-attack and to minimise the browser's forensic footprint.

## Preventing Data Leakage

Changing the Firefox source code to ensure Fireguard writes browser created temporary data to the Mini-SDV **Fireguard Data Partition** is a key aspect in preventing data leakage. In addition, to specifically prevent data leakage, through the recovery of browser created data, the following features were changed, disabled or removed using "lockpref" functions placed in the *Fireguard.cfg* file to prevent saving:

- *Browser data in the form of history, cookies, saved passwords and form data:* This browser data is saved in the user profile on the Mini-SDV and therefore could be considered to be protected. However, this browser data becomes available to web sites when Fireguard is used to browse the sites. Therefore Fireguard has been configured to always delete history, cookies, saved passwords and form data when the browser exits for the following reasons:
  - Cookies can be used by web sites (hostile or otherwise) to track a user's Internet usage and behaviour, enabling an activity log to be formed.
  - Although passwords and form data are now securely retained on the Mini-SDV, such data can be automatically used by web sites to logon or pre-complete web forms. It was therefore considered good practice to ensure passwords and form data were not retained at the end of a browser session.
  - As the Fireguard history is stored on the Mini-SDV it presents neither a privacy threat nor vulnerability and so could be retained. However, as the Firefox private browsing mode does not save history records it was considered good practice for Fireguard to also not retain history records.
- *Web page cache:* The cache holds copies of recently retrieved content in the user profile for re-use if the same page is accessed again. The cache is retained between browser invocations and is therefore vulnerable to access by a hostile web site when the site is accessed by Fireguard (Soni, R. and B. Adhikari, 2009). In Fireguard cache retention is disabled.
- *Third-party cookies*<sup>50</sup>: Like browser cookies, third party cookies allow a user's Internet usage and behaviour to be tracked. By placing a third party cookie in the user profile an increased

---

<sup>50</sup> Third party cookies are cookies set by one web site that can be read by another web site, i.e. a cookie is placed in the user profile of a web site the user may never have browsed. They are used by advertisers to track website visits.



'picture' can be formed of the user's site visits and interests (Jackson, C., A. Bortz, et al., 2006). Allowing a web site to write third party cookies to the user profile is prevented within Fireguard.

- *Importing another browser's settings*<sup>51</sup>: Firefox allows a user's settings and browser data from an alternative browser to be transferred, including the alternative browser's history, saved passwords, cookies and form data. As this data is not retained in Fireguard for the reasons given above Fireguard prevents the option of importing (into the user profile) an alternative browser's settings.
- *Default download area*: When a user wishes to download and save an object from the Internet and the user does not specify a location, the object is stored in a default folder, usually the *Desktop* or *My Documents* folders. The feature was changed so that no default storage location is possible. The user is forced to save the downloaded object in a specific folder.

The above configuration changes (specified in the *Fireguard.cfg* file) coupled with both the protection of *Fireguard.cfg* in the **Fireguard Data Partition** and source code changes to remove file system and registry dependencies ensure Fireguard does not write data to the host PC HDD and reduces the possibility of data leakage.

### Preventing Cyber-attack

The following features were disabled to prevent the exploitation of browser vulnerability and the introduction of malware from which a cyber-attack could be launched:

- *Plug-ins*: To disable plug-ins both source code changes and "*lockpref*" settings in *Fireguard.cfg* were necessary. The source code function that attempts to determine the location of plug-in files was modified to immediately return an error code to indicate plug-ins are not available. A capability that allows the OS to be scanned for available downloaded plug-ins was disabled using a "*lockpref*" setting.
- *Extensions*: Like plug-ins both source code changes and "*lockpref*" entries in *Fireguard.cfg* were made to prevent the installation and use of extensions. There are 2 types of extensions; user only extensions which are held in the user profile and shared extensions held in a common folder with a registry entry identifying the path name of the common folder. *Lockpref* entries in the *Fireguard.cfg* file were used to prevent the loading of extensions. For completeness the source code was also changed to prevent the registry entry being created that contains the pathname of the folder containing shared extensions.
- *Software Updates*: There are four types of software updates; software patches for the browser, installation of new browser releases, updates for plug-ins and updates to the search engine list maintained by the browser. As the Fireguard binary, configuration and interface code are contained in the Read-Only **Fireguard Browser Partition** software updates are not possible. Also plug-ins are completely disabled so plug-in updates would not be possible. For

---

<sup>51</sup> Firefox allows the browser data and settings established by a user in another browser (e.g. Microsoft IE) to be transferred into Firefox. Transferable data includes bookmarks, histories and saved passwords.

completeness “*lockpref*” entries in *Fireguard.cfg* were set to ensure automatic updates for browser patches, new releases and the search engine list were not attempted.

- *Java*: To stop Java applets executing it is necessary to use a “*lockpref*” setting to prevent the browser detecting if the Java Runtime Environment was installed on the host PC. Also as plug-ins have been disabled it was not possible for the required Java plug-in to be installed to enable applets to execute.
- *Media Autoplay*: To prevent the automatic playing of audio and video files on a web site a “*lockpref*” entry in *Fireguard.cfg* was set.

With the exception of Javascript, disabling the above features prevents currently known sources of browser malware attacks. The placement of the *Fireguard.cfg* file in the Mini-SDV **Fireguard Browser Partition** (a Read-Only Partition) ensures the integrity of the file is preserved.

### Minimising the Forensic Footprint

Many of the techniques used to prevent data leakage and cyber-attack also reduce the forensic evidence of browser use. To reduce the forensic footprint for a browser involves ensuring that after the browser has executed there is no residual data on the PC HDD nor have any changes been made to the PC system configuration. A reduced forensic footprint was achieved in Fireguard by:

- *Ensuring Fireguard did not change the Windows registry settings*:
  - As discussed above, the Firefox installer makes a number of changes to the registry during installation. The installer was not used in the Fireguard development and all dependencies on registry values were changed in the source code. To avoid the use of the installer Fireguard was compiled and built on a development platform and then imaged onto a Mini-SDV Read-Only partition.
  - Plug-ins, extensions and software updates all make changes to the registry, particularly during installation. These three capabilities were disabled in Fireguard as described above.
- *Ensuring Fireguard left no residual data*: The removal/deletion of cookies, histories, saved passwords, form data and cached web pages from the **Fireguard Data Partition** after browser use, as described above, minimised forensic evidence. Any additional retained browser generated data was retained in the **Fireguard Data Partition** thereby leaving no forensic footprint.
- *Not performing a default browser check*: Upon executing Firefox the user is given the option to make Firefox the user’s default browser. If selected as the default browser a registry entry is made. A “*lockpref*” entry in *Fireguard.cfg* was set to disable the default browser option in Fireguard.
- Firefox allows the Windows OS desktop background image to be selected and set by the user, a feature that changes both the registry and performs a write of the image to the PC file system.

In Fireguard setting the desktop background image is prevented by a source code change to a compile time option which has the effect of disabling the capability.

The above changes virtually eliminated the forensic evidence of browser use. However, the Windows OS creates potentially forensically recoverable evidence of Fireguard use through details contained in the OS swap space/page management system (pagefile.sys). Also if Fireguard is used to download an object the OS creates a registry most recently used (MRU) list entry to indicate an object has been saved.

### **User Interface Changes**

The removal or disabling of Firefox features to create (the secure browser) Fireguard necessitated the modification of the Firefox user interface to remove menu options for features that were no longer available. Without the interface changes removed/disabled features would appear as “greyed-out” options on the browser drop-down menus. The following changes were made to the user interface by making changes to the XUL files held in jar archives in the “*chrome*” folder installed in the ***Fireguard Browser Partition***:

- Removal of the “Add-ons<sup>52</sup>” menu as both plug-ins and extensions are disabled.
- Removal of the Update Tab as software updates are disabled.
- Removal of the Privacy Tab as all history and cookies are cleared when the browser exits.
- Removal of the Password pane in the Security options as all passwords are cleared when the browser exits.

### **Retention of Built-in Security Capabilities**

The Firefox phishing and malware detection capabilities are retained in Fireguard. The phishing and malware database is stored per-user in the respective user profile. For Fireguard, this means the database will be stored in the user profile in the ***Fireguard Data Partition***. The Fireguard phishing and malware protection is either on or off. As phishing and malware identifiers are like virus signatures the list of sites infected by phishing attacks and malware is highly dynamic. The browser checks for infected web sites each time the browser is started; the browser may also check periodically during a session for additional updates.

The site identity security capability is also retained. However, the master password capability is removed as Fireguard does not retain any password details and therefore a master password to protect other passwords is not necessary. The browser also had all default bookmarks removed and the home page set to a blank page to provide a clean setup.

---

<sup>52</sup> Mozilla uses the term Add-ons to generically describe both plug-ins and extensions.

It was decided to retain the bookmark feature so that favourite web sites could be accessed without entering the URL each time access to the site is required. Bookmarks provide no security or privacy vulnerability.

## Fireguard Testing

A comprehensive test plan was created that covered:

- Functionality testing of the core features and user interface of Fireguard. The testing included error trapping to confirm the browser could recover from errors.
- Vulnerability testing to ensure the browser could not be re-configured to include functionality (e.g. plug-ins, extensions, etc) that could make it vulnerable to malware. The testing also included exception, logical and operational testing.
- Stress testing Fireguard under heavy workloads and over prolonged periods.
- Sociability testing of Fireguard in an environment with other executing applications.
- Compatibility testing of the browser across the range of Windows OS' including versions with different service packs.

The above comprehensive testing confirmed that Fireguard operated correctly as a portable secure browser with vulnerable functionality removed.

The final area of testing to be performed was to confirm that Fireguard had a minimal forensic footprint. Digital forensic techniques can be categorised as dead forensic<sup>53</sup> and live forensic<sup>54</sup> techniques. Dead forensic analysis of the PCC HDD to determine if Fireguard had written data to the HDD would not provide a meaningful result as both the OS and other executing applications will also write to the HDD. Therefore the following live forensic tests were performed to verify if evidence remained on a PC HDD after a browser session:

- Monitoring file creation: The "Process Explorer" program (Windows Sysinternals, 2010) was used to monitor the executing Fireguard. This utility lists all resources, including file handles<sup>55</sup> that a running process is using. The list of file handles was monitored during a set of browser functionality tests and no unexpected files were read or written on the host PC HDD. An inspection was also made of locations that the standard Firefox browser would use and no files were found.
- Registry: An export of the registry to text was performed before and after a series of browser functionality tests. The resulting exported text files were compared using the program "Beyond Compare" (Scooter Software, 2010). This program highlights differences between two versions

---

<sup>53</sup> Dead forensic analysis is performed on data at rest, i.e. data acquired from a PC HDD after the PC has been powered down.

<sup>54</sup> Live forensic analysis is performed on a powered PC usually under the control of the PC's operating system, i.e. data is acquired from the PC whilst the PC is still executing - the data may be acquired from the PC's memory, HDD, communication interfaces or internal intelligent devices (e.g. graphics processor).

<sup>55</sup> A file handle is a descriptor indicating the status of an open file.

of the same file. Careful examination of the two versions revealed only changes to most recently used values (MRU). The MRU values were changed by the OS when Fireguard was used to download an object from the Internet.

## Conclusion

### Limitations

Whilst Fireguard provides a secure portable browser, resistant against cyber-attack and data loss, with a minimal forensic footprint it does have the following limitations:

- *Javascript*: By allowing Javascript to remain enabled there exists a mechanism by which malware can be introduced into the browser.
- *Clickjacking Attacks*: Clickjacking attacks occur because standard HTML features coupled with Javascript allow malware to exploit user actions often unbeknown to the user. Nothing has been designed into Fireguard to detect or prevent clickjacking.
- *Monitor Attacks*<sup>56</sup>: The removal of features like plug-ins, extensions, software updates and Java reduce the possibility of malware attacks launched from within the browser, however it does not prevent monitoring attacks from either software legitimately installed on the PC or from rootkits that may have evaded any anti-virus and anti-spyware installed on the host PC. Preventing monitor attacks is considered outside the scope of the Fireguard project.
- *Immediately Executable Malware*: Fireguard does not prevent malware that is covertly presented to the user and is executed immediately (without requiring the user to reboot the PC) by deceiving the user to invoke its execution, e.g. the user opens an email attachment (received through webmail accessed by Fireguard) that is actually malware disguised as an attachment of interest to the user; the malware is able to execute upon the user opening the attachment and compromise Fireguard. Disabled features like plug-ins, extensions and Java may minimise or eliminate the effect of the malware. Despite such malware possibly being able to compromise Fireguard the malware cannot become embedded in the browser's binary because of the Mini-SDV Read-Only partition protection, therefore at worst the compromise occurs for one browser session.
- *Page file/swap space data*: Windows implements virtual memory management based upon a page swapping system. All executing processes have data held in virtual memory. When a process terminates, data may remain in swapped out pages. It is possible that sensitive data viewed or processed by the browser may be retained in virtual memory and recovered by an unauthorised user (Ruff, N., 2008).
- *Hardware Dependent*: Fireguard is dependent on the functionality provided by the Mini-SDV, in particular the partition Read-Only access control feature to protect the integrity of the Fireguard binary. Fireguard is constructed using both amended Firefox software and the Mini-SDV

---

<sup>56</sup> A monitor attack occurs when malware is able to spawn a number of processes that monitor one another to enable resurrection of a hostile malware process if one or more of the original hostile processes are discovered, thus enabling the hostile activity to continue.

capabilities to provide a secure browser; therefore porting the functionality to another secure PESE may not be desirable or even possible. To achieve the equivalent protection against cyber-attack, data leakage and reduced forensic footprint through the use of an alternative secure PESE hardware device would require the device to support multiple partitions with differentiated access rights that work seamlessly with the range of Windows OS'.

- *Distributing a new Fireguard Patch or Release:* Although the integrity of the Fireguard binary and configuration files are protected by a Mini-SDV Read-Only partition this integrity protection mechanism prevents the automated distribution of patches and releases. Currently, the only method of updating the Fireguard binary is for the binary to be re-imaged onto the Mini-SDV Fireguard Browser Partition by an administrator with the appropriate permissions. However, although patch management cannot be automated it can be argued that the application of security patches for Fireguard is not as imperative as it is for standard Firefox as both the Mini-SDV Read-Only partition provides protection against malware including zero day attacks<sup>57</sup> and the disabling of features like extensions, Java, etc reduces the available vulnerabilities which could be exploited for an attack.
- *Source Code Changes:* Satisfying the requirements for a secure browser could not be achieved through changes to configuration files alone, changes had to be made to source code. If Fireguard is to be upgraded to a newer release of Firefox it will be necessary to perform a review of the source code to determine if the code structure and layout has changed and identify if the code changes performed to create Fireguard can still be applied without new analysis.
- *Lacks Full Firefox Functionality:* To produce Fireguard, functionally rich features like plug-ins and extensions have been removed. Lack of this functionality obviously limits how Fireguard can be used. However, the objective of the Fireguard project was to produce a secure portable browser that can be used by remote and mobile workers to access corporate systems, corporate webmail and specific corporate web based applications, not to provide a highly sophisticated browsing tool.

## Strengths

In summary a secure highly portable browser has been developed based on limited changes to an open source browser. Fireguard provides strong protection against cyber-attack, data loss and leaves virtually no evidence on a host PC that it has executed. Fireguard is considered to be a secure tool for remote and mobile workers who need secure Internet access from a PC for which no level of trust can be assumed.

---

<sup>57</sup> A zero-day attack is malware that tries to exploit software vulnerabilities before neither the software vendor is aware and able to provide a patch nor an antivirus product vendor has identified the malware signature to enable its detection.

The strengths of Fireguard are demonstrated in Table 1 by showing conformance to the five requirements established for the project.

Requirement	Conformance
To prevent malware and hence cyber-attack it must be executable from a Mini-SDV Read-Only partition.	The Fireguard binary was imaged and installed in a Mini-SDV Read-Only partition from which it successfully executed.
To prevent malware and hence cyber-attack it must prevent plug-ins, extensions and software updates.	Plug-ins, extensions and software updates were disabled. Also the media autoplay and Java features that can be susceptible to malware were disabled.
To prevent forensic discovery it must not update the Windows registry.	Source code and configuration changes were made so that no Windows registry changes occurred.
To prevent data leakage and forensic discovery it must store configuration data and temporary files in a nominated storage area on the Mini-SDV.	All retained browser data is stored in the <b>Fireguard Data Partition</b> on the Mini-SDV.
To prevent data leakage and forensic discovery it must not write to the Host PC HDD, with any created temporary data deleted when the browser terminates.	Fireguard configuration settings ensure browser data is not retained. Testing of Fireguard confirmed the browser did not write data to the host PC HDD.

Table 1: Fireguard Conformance to Requirements

The Fireguard project has produced a secure PESE trusted application which satisfies the requirements specified for it.

### Further work

Future work could include:

- *Javascript:* Fireguard has been successfully developed and tested. An obvious change could be to disable Javascript with the predominate use of the browser to access corporate web applications that do not utilise Javascript. Alternatively Javascript can remain enabled and the Firefox extension 'NoScript' could be integrated into Fireguard and used to allow Javascript to execute only from certain allowed sites.
- *Developing a capability to detect clickjacking sites:* Research has been conducted into mechanisms to detect and prevent clickjacking (Balduzzi, 2010; Rydstedt, G., E. Bursztein, 2010). A capability within Fireguard that can detect and prevent clickjacking would enhance security. Also if the 'NoScript' extension is integrated into Fireguard clickjacking attacks can be reduced.
- *Transitioning to Firefox version 4:* The changes made to Firefox to create Fireguard have been fully documented for Firefox version 3.6, therefore (in theory) transitioning the source code changes and configuration settings to version 4 should not be difficult. However, open source projects often restructure source code and functionality so a future activity would be to identify how rapidly a Fireguard release based on Firefox version 4 could be achieved.
- *Developing an automated patch management and release tool:* A Mini-SDV capability to enable remote and mobile workers to securely and automatically perform patch management and/or upgrade to a new release of Fireguard will be important if the Mini-SDV with Fireguard is used in large scale and dispersed deployments.



Although the identified future work would enhance the usability and security of Fireguard the project has achieved its objectives and produced a secure PESE trusted application in the form of a secure portable browser suitable for deployment to remote and mobile workers to enable secure Internet access from an available and potentially untrusted PC.

## References

- Adobe (2010). "Flash Player." Retrieved September, 2010, from <http://www.adobe.com/products/flashplayer/>.
- Balduzzi, M., M. Egele, et al. (2010). A solution for the automated detection of clickjacking attacks. ASIACCS '10 Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security, ACM.
- Barth, A., A. Felt, et al. (2010). Protecting browsers from extension vulnerabilities, Citeseer.
- Cnet News (2009). "Using software updates to spread malware." Retrieved October, 2010, from [http://news.cnet.com/8301-27080\\_3-10301485-245.html](http://news.cnet.com/8301-27080_3-10301485-245.html).
- Help Net Security (2009). "Zero-day vulnerabilities in Firefox extensions discovered." Retrieved October, 2010, from <http://www.net-security.org/secworld.php?id=8527>.
- IronKey (2010). "IronKey Technology." Retrieved September, 2010, from <https://www.ironkey.com/hardware-encryption>.
- Jackson, C., A. Bortz, et al. (2006). Protecting browser state from web privacy attacks. WWW '06 Proceedings of the 15th international conference on World Wide Web ACM.
- James, P. (2008). Secure Portable Execution Environments: A Review of Available Technologies. 6th Australian Information Security Conference, Perth.
- Lemos, R. (2010). "Could USB Flash Drives Be Your Enterprise's Weakest Link?". Retrieved September, 2010, from <http://www.darkreading.com/vulnerability-management/167901026/security/storage-security/227200081/index.html>.
- McAfee (2010). McAfee Threats Report: Second Quarter 2010.
- MDN (2010). "Introduction to XBL." Retrieved September, 2010, from [https://developer.mozilla.org/en/XUL\\_Tutorial/Introduction\\_to\\_XBL](https://developer.mozilla.org/en/XUL_Tutorial/Introduction_to_XBL).
- MDN (2010). "JavaScript Guide." Retrieved September, 2010, from <https://developer.mozilla.org/en/JavaScript/Guide>.
- MDN (2010). "XUL Tutorial." Retrieved September, 2010, from [https://developer.mozilla.org/En/XUL\\_Tutorial](https://developer.mozilla.org/En/XUL_Tutorial).
- Moscaritolo, A. (2008) US military bans USB thumb drives. SC Magazine
- Mozilla (2010). "Mozilla Firefox."
- mozillaZine (2010). "Locking preferences." Retrieved October, 2010, from [http://kb.mozillazine.org/Locking\\_preferences](http://kb.mozillazine.org/Locking_preferences).
- NetMarketShare (2010 ). "Top Browser Share Trend." Retrieved September, 2010, from <http://www.netmarketshare.com/browser-market-share.aspx?qprid=1>.
- NTI (2004). "Classified Data Identification & Data Elimination Guidelines." Retrieved August 2010, 2010, from <http://www.forensics-intl.com/riskscan.html>.



- PHPforms (2010). "Introduction to CSS." Retrieved September, 2010, from [http://phpforms.net/tutorial/tutorial.html#cat\\_180](http://phpforms.net/tutorial/tutorial.html#cat_180).
- Ponemon, L. (2009). Trends in Insider Compliance with Data Security Policies - Employees Evade and Ignore Security Policies, Ponemon Institute.
- Reynaud-Plantey, D. (2005). "New threats of Java viruses." Journal in Computer Virology 1(1-2): 11.
- Ruff, N. (2008). "Windows memory forensics." Journal in Computer Virology 4(2): 83-100.
- Rydstedt, G., E. Bursztein, et al. (2010). Busting frame busting: A study of clickjacking vulnerabilities on popular sites. W2SP 2010: Web 2.0 Security and Privacy 2010, Oakland, California.
- Scooter Software (2010). "Beyond Compare." Retrieved October 2010, from <http://www.scootersoftware.com/moreinfo.php>.
- Secure Systems (2010). "Mini Silicon Data Vault." Retrieved October, 2010, from [http://www.securesystems.com.au/index.php?option=com\\_content&view=article&id=6&Itemid=11](http://www.securesystems.com.au/index.php?option=com_content&view=article&id=6&Itemid=11).
- Smart Link Corporation (2010). "ImTranlator." Retrieved September, 2010, from <https://addons.mozilla.org/en-US/firefox/addon/2257/>.
- Soni, R. and B. Adhikari (2009). Browser Security, Carleton University: 46.
- Symantec (2010). Symantec Global Internet Security Threat Report Trends for 2009. Volume XV.
- TrueCrypt-Foundation (2010). "TrueCrypt Users Guide." Retrieved September 2010, from <http://www.truecrypt.org/docs/>.
- Windows SysInternals (2010). "Process Explorer." Retrieved September 2010, from <http://technet.microsoft.com/en-us/sysinternals/bb896653.aspx>
- Zalewski M (2009). Browser Security Handbook, part 2, Google Inc.

#### **4.2.4.4 Synopsis**

**Outcomes and Contribution to Knowledge:** The design science research presented in the paper resulted in a fully tested artifact that satisfied its requirements and formed a secure PEE application. Fireguard's knowledge contribution was a hardened browser utilising the Mini-SDV's capabilities that combine to prevent cyber-attack, data loss and forensic data discovery. An additional outcome is a detailed description on how to harden a browser yet let it remain functional and usable.

Whilst Fireguard does provide protection against cyber-attack, data loss and forensic data discovery, as it executes under the control of the host PC operating system it does share some of the malware vulnerabilities (as will all hardened applications forming a secure PEE) of a virtual machine based secure PEE. For instance, like a virtual machine (as

discussed in Paper 5) Fireguard is susceptible to keyboard loggers and screen shot capturers that maybe embedded in the host PC operating system. To counter this malware the operational approach proposed in the paper is to use the 'Fully-Trusted' mode of the secure PESE whenever there is a concern for the trust worthiness of the host PC, but of course using this mode relies upon the PC not being locked from booting an external device. Fireguard was constructed to provide a secure application to be executed in a known but untrusted environment.

The paper discusses the testing (demonstration) and provides an evaluation of the development (in the paper conclusion). Chapter 5 draws upon the paper's demonstration results with the Fireguard evaluation considered as part of the doctoral research evaluation in Chapter 6.

***Contemporary relevance, linkage with other papers and future direction:*** Browser security is an area of on-going research (Jang et al., 2012; Akhawe and Felt, 2013; Bielova, 2013). The paper contributed to the growing knowledge base on browser security and has itself been cited in the following publication:

Marrington, A., Baggili, I., Al Ismail, T., and Al Kaf, A. (2012), Portable web browser forensics: A forensic examination of the privacy benefits of portable web browsers, In Computer Systems and Industrial Informatics (ICCSII), IEEE International Conference on pp1-6.

The paper is linked to Papers 7, 9 and 10. Fireguard forms part of: the secure PEE described in Paper 7; the secure PESE in Paper 9; and the secure PEE in Paper 10. The paper suggested future work which was successfully implemented with the exception of the patch and release management tool.

The research questions required the research to determine if the artifact was still useable and maintainable after being: hardened; changed to prevent the storage of temporary data on a host PC disk drive; and packaged with a Mini-SDV to form a secure PESE. The paper shows that a useable browser was produced that prevents data remnants but was not readily maintainable in the remote work location, i.e. the Fireguard design does not enable patch management or the update to a new release to be performed remotely. The development of a patch and release management tool (for use by remote workers) was proposed as future work; however, work on the tool was not progressed with the

rationale given at the conclusion of design cycle 3 and in the research evaluation (chapter 6). Following the completion of Fireguard the research moved to focus upon the development of a hardened bootable operating system based secure PEE.

#### **4.2.5 A Secure Hardened Bootable Operating System**

##### **4.2.5.1 Preamble**

A secure PESE with a bootable operating system based secure PEE provides assurance that it is not susceptible to malware resident on the host PC; it provides the most secure form of secure PESE. Paper 7 discusses the development of the Mobile Execution Environment (MEE), a hardened bootable operating system based secure PEE containing a small set of hardened applications (including Fireguard). The design and development of the MEE commenced in 2010 and was completed in 2011. The paper was not finalised until the results of a trial were obtained. A trial by a group of teleworkers could not be arranged and conducted until early 2012 with Paper 7 eventually being published<sup>58</sup> in 2014. The paper is structured using the DSRM activity names (Peppers, 2007) and follows a publication schema for DSR (Gregor & Hevner, 2013). Like Paper 3 in Chapter 2, the paper addresses the telework category of remote working to position the research towards a wider audience following the Australia Government's initiative in 2011 to grow telework in Australia.

The MEE uses the live CD version of the Ubuntu Linux distribution (Ubuntu, 2012) and is designed to satisfy a refinement of the research objectives and functional requirements. The MEE makes a contribution to answering the following two doctoral research questions:

*How can a useable and maintainable hardened operating system and/or a small set of hardened applications be developed?*

*How can a useable and maintainable execution environment be configured to store all temporary data on a secure PESE partition?*

The construction of the MEE shows how a Ubuntu live CD distribution can be hardened and configured to store temporary data on an SDV secure storage device yet still remain a

---

<sup>58</sup> The paper was published in the Journal of Information Management and Computer Security which requires a paper's Abstract is structured and presented under seven sub-headings.

useable and ‘fit for purpose’ operating system. One of the MEE requirements was to also produce a version that could be up-loaded from a USB thumb drive, i.e. build an artifact similar in concept to the LPS (LPS, 2008) and the Becrypt TC (TrustedClient, 2009) products (refer to Chapter 2 for details). The paper considers the advantages and limitations of the MEE on a USB thumb drive. In Paper 4 it was highlighted that a disadvantage of a bootable secure PEE was the usability issue of making the secure PESE the first boot device for the host PC; however, by 2011 most PC/laptops either made the USB port the first boot device or provide a user friendly interface upon power up to select the boot device.

#### 4.2.5.2 Prior Research and Knowledge

Improving the security of, or the development of new secure operating systems (Wood, 2013), has been the subject of on-going research since the development of the early multi-user, multi-processing systems (Dijkstra, 1968). A key issue for developers is the trade-off between security and useability (Gutmann and Grigg, 2005) and therefore one of the following approaches is adopted:

- ***Develop a secure operating system:*** The optimum approach is to ensure security is a design consideration at the commencement of the development, such an approach can result in a highly secure but limited functionality operating system (seL4, 2014) with usability constraints.
- ***Enhance an existing operating system:*** In this approach an existing operating system is augmented with security features which can be configured on or off. A good example is Security Enhanced Linux also known as SELinux (Loscocco and Smalley, 2001). When the security features are configured off the operating system and user applications operate like the non-enhanced version, however when configured on applications may require changes to operate correctly and usability of both the operating system and applications can be impacted (Schreuders, et al., 2011).
- ***Hardening an existing operating system:*** Typically no specific security mechanisms are added nor operating system code changed in this approach. Instead known vulnerabilities are eliminated through any or all of the following techniques: disabling functionality; application white listing<sup>59</sup>; allocation of privilege only where necessary;

---

<sup>59</sup> Application white listing is a technique where access is only allowed to the required functionality.

regular software patching; changing/fixing configuration settings; and removing applications included in the operating system distribution but not required for the end use (DSD, 2012). Hardening is an approach often adopted for web servers (Tracy et al., 2007) and single purpose systems/servers (Scarfone et al., 2008). The impact upon usability will depend upon the degree of hardening and application removal/white listing performed, and the intended level of user interaction with the system. For instance, a network gateway based upon a hardened operating system is unlikely to directly interact with a user and therefore usability in this instance is not a major concern.

- ***Remove all but the essential applications from an existing operating system:*** In this approach only the applications required for the intended use of the system are retained with all other applications removed (wherever possible) from the operating system distribution. No other hardening techniques are used. The LPS product (LPS, 2008) is an example that has adopted this approach.

The 'hardening an existing operating system' approach was adopted for the MEE with the following knowledge consumed during its design and development:

- Prescriptive knowledge in the form of the secure PESE design and operational models (Figures 4.2 and 4.3), and the Ubuntu design and implementation. To supplement the Ubuntu documentation the researcher and co-author found it necessary to read the Ubuntu configuration files, source code and search web sites for postings on the Ubuntu design and operation made by interested parties (these web sites are cited in the paper).
- Prescriptive knowledge obtained from a review of the features and functionality of the LPS (LPS, 2008) and Becrypt TC (TrustedClient, 2009) products.
- Prescriptive knowledge in the form of a set of security and functional objectives derived predominately from the doctoral research objectives and the secure PESE functional requirements identified in Chapter 2. The objectives derived for the MEE are at a developmental level so as to direct the implementation.

- Descriptive knowledge obtained from stakeholders on the commercial use of Linux and a requirement for a version of the MEE on a USB thumb drive that contributed to the development and operational objectives.
- Informal justificatory knowledge consisting of the co-author and researcher's knowledge of Linux design and implementation.
- Formal justificatory knowledge in established and published operating system and application hardening techniques (Tracy et al., 2007; Scarfone et al., 2008; Tanenbaum, 2009).

#### 4.2.5.3 Paper 7

**Paper 7** - James, P. and Griffiths, D. (2014), **A Secure Portable Execution Environment to Support Teleworking**, *The Journal of Information Management and Computer Security*, Volume 22, Issue 3, pp 309-330.

#### Abstract

**Purpose** - Teleworking is an established work practice yet often the information security controls in the teleworking location are weaker than those in a corporate office. Security concerns also prevent organisations allowing personnel to telework. This paper presents the design, development and trialling of the Mobile Execution Environment (MEE), a secure portable execution environment designed to support secure teleworking.

**Design/methodology/approach** - The design science research methodology was applied to develop the MEE and this paper is structured using the process elements of the methodology.

**Findings** – In this paper the problem addressed and the design objectives are defined. The design and implementation are discussed and the testing and trialling approach adopted to demonstrate the MEE is summarised. An evaluation of the demonstration results against the design objectives is presented.

**Research limitations/implications** – The MEE is part of an on-going research project using open source software; the structure and functionality of the software can limit or influence the direction of the research.

**Practical implications** – The MEE provides a secure portable execution environment suitable for transaction-oriented work performed remotely, e.g. teleworkers performing customer support work.

**Social implications** – Contribute to encouraging the implementation of teleworking.

**Originality/value** – The MEE builds upon the concept of a portable executable OS that uploads onto a PC through an external port. The MEE extends this concept by providing a hardened secure

*computing environment that is uploaded from a secure storage device or a standard thumb drive (USB flash drive).*

## **Keywords**

Secure teleworking, design science, hardening, secure portable execution environments.

**Paper type** Research paper.

## **Introduction**

Teleworking (Lister and Harnish, 2011) has a number of advantages for both employee (e.g. reduced travelling time and travel costs, and possibly the opportunity to have flexibility for when the work is to be performed) and the employer (e.g. reduced office space and attracting talented personnel that may not otherwise be able to work in a traditional office environment). High speed broadband has enabled teleworking to become a standard work practice (DBCDE, 2011) yet organisations may not always fully consider the information security risks. Conversely, when the risks are considered information security can be cited as an obstacle to allowing personnel to telework.

The Mobile Execution Environment (MEE) is an example of a secure portable execution environment, an OS and set of applications built as a solution to support a particular work activity. The MEE is one of the outcomes from a research project established to develop secure portable execution and storage environments (secure PESE) to support secure remote and mobile working.

The design science methodology (Hevner et al, 2004) was applied to the research described in this paper. Design science is a developmental research methodology that can be used effectively in information systems research to produce innovative artifacts (information systems' processes, techniques or tools); in this paper the MEE is the artifact. Design science has become an established information systems research methodology with both a strong academic community advocating its use (DESRIST, 2013) and a conference dedicated to the methodology now in its eighth year (DESRIST-2013, 2013). The style of design science used in the research project is based upon the design science research methodology (DSRM) (Peppers et al, 2007); a popular methodology with over 90 citations identified in the Association of Computing Machinery digital library (ACM-DL, 2013). Figure 1 diagrammatically presents the DSRM six process elements (activities) together with a synopsis of the respective element and the application of the element to the MEE research:

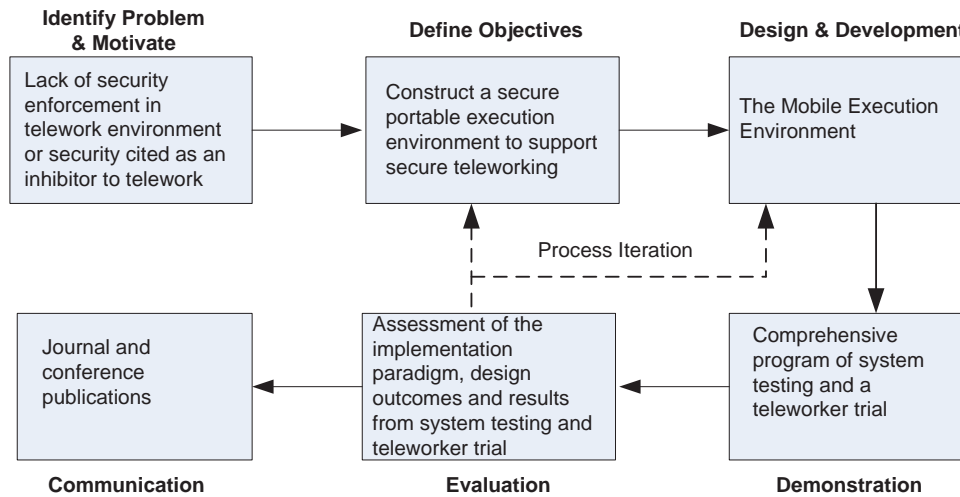


Figure 1. DSRM Process for MEE Project

- **Problem identification and motivation** requires the encapsulation and presentation of the research problem for which an artifact is required and an explanation of why there is a need to find a solution to the problem. For the MEE, the problem considered was how to provide a secure execution environment that could both improve security for existing teleworkers and enable teleworking to be offered to personnel who are denied the opportunity due to the sensitivity of the information they process.
- **Defining objectives for the artifact** allows the researcher to specify the desired outcome(s) of the research. For the MEE a set of developmental, functional, operational and security objectives were defined.
- **The design and development** process element involves the specification, design and implementation of an artifact (the MEE) that satisfies the defined objectives. Following a requirements gathering and analysis process the MEE was designed and implemented using an iterative build-confirm approach.
- **The demonstration of the artifact** requires the test or verification of its capabilities or attributes. The MEE was demonstrated through a comprehensive testing program and then through a trial performed with a commercial organisation's home based customer support staff.
- **Evaluating the artifact** involves assessing the observations/measurements collected during the demonstration to determine the effectiveness of the artifact with respect to addressing the defined problem and satisfying the objectives. The MEE implementation paradigm was evaluated and for each of the objectives defined for the MEE the design outcomes, test results and trial observations were evaluated.
- **The communication** process element of the DSRM requires the publication of the research and the respective outcomes to relevant audiences. The communication of the MEE is achieved through this and other papers (James & Griffiths, 2012).

This paper has been structured using the DSRM process elements to present the MEE research.



## Identification of Problem and Motivation

Information security can be an obstacle to allowing staff to telework (AccessEconomics, 2011). Also some organisations that do allow staff to telework can be unaware of the full set of information security risks (Clear, 2007). The three main information security risks in a teleworking environment (Deloitte, 2011) can be summarised as:

1. Breach of data confidentiality when data is transmitted over the Internet.
2. Compromise of system and/or data integrity in the teleworking computing environment.
3. Breach of data confidentiality in the teleworking computing environment.

Preventing a breach in data confidentiality over the Internet can be mitigated through a range of available cryptographic protocols and end point authentication. The type and strength of the cryptographic primitive and the respective key management and authentication schemes used will depend upon the level of security required to protect the data to be transmitted. The cryptographic and end point authentication countermeasures used to prevent a breach in data confidentiality on the Internet are outside the scope of this paper.

Both a compromise of system/data integrity and a breach of data confidentiality (in the teleworking computing environment) can be mitigated by a range of physical and logical security controls. However, the security controls present in the teleworking location may be less sophisticated than those present in a corporate environment. Recommendations for best practice telework security controls (NIST, 2007) include secure storage cabinets for documents and equipment (e.g. data storage devices) when telework is not being performed, and PC OS security controls comprising VPN, anti-malware software, personal firewall, data encryption, passwords, file and partition access controls and least privilege. Yet organisations implementing best practice telework security controls can find it difficult to enforce these controls as both the teleworker and telework PC and equipment reside outside the organisation's sphere of ICT management. In addition, the standard data processing actions of an OS and software applications can contribute to the unintended and often unknown storage of sensitive corporate data remnants on a teleworker's PC HDD.

The motivation for the MEE emerged from a need to develop a computing environment that improved information security for existing teleworkers and also provided a secure telework solution that removed a barrier to implementing a telework policy. A further motivator was to provide a restricted secure desktop environment that would allow a teleworker familiar with Microsoft Windows to quickly become productive without the need to be trained. The MEE was specifically designed to counter the risk of compromise of system/data integrity and the risk of breach of data confidentiality in the teleworking environment. The MEE was designed to provide a secure execution environment for existing teleworkers and facilitate the introduction of telework where security concerns had previously prevented the work practice.

## Design Objectives

### Assumed Environment

The assumed environment and method of use for the MEE is considered to be:

- A non-public work environment, e.g. a teleworker's home.
- A teleworker would have appropriate IT skills.
- The MEE is required to execute on any available PC that may have previously been compromised.
- A teleworker would not be deliberately hostile (i.e. the teleworker would not knowingly introduce malicious software (malware) to the MEE nor deliberately lose/mislay removal storage media nor attempt to attack either the MEE or a host server).
- The teleworker could unintentionally introduce malware from either external media or the Internet.
- The teleworker will be inquisitive about the MEE's capabilities and may use the MEE in an unintended manner.
- The MEE is used primarily as a remote access platform.
- Sensitive data may be processed in the telework environment.

Design objectives were defined that considered the assumed environment and addressed the aforementioned problems.

### Objectives

Developmental, operational, functional and security objectives were defined for the MEE. The developmental objective directs the artifact's design and development. The operational objective defines the intended use and deployment capabilities. The functional objective defines the intended capability to be implemented. The security objectives define how the risks of compromise of system/data integrity and breach of data confidentiality are to be countered. The MEE developmental, operational and functional objectives were formulated through requirements gathering from potential users and a literature review on how to make teleworking more secure (Clear, 2007; NIST, 2007; NIST 2009). The security objectives were derived from a risk assessment previously performed to identify a set of attributes for a teleworking security model (James, 2011):

- **Developmental** - Implement using freely available open software to exploit existing innovations and enable rapid development.
- **Operational** - Be up-loadable on to any PC from either a secure PESE device or a standard thumb drive (also commonly called a USB flash drive).

- **Functional** - Provide an MEE core system consisting of a hardened execution environment that supports remote access functionality and limits the opportunity for malicious activities by removing known vulnerabilities.
- **Security** - Preserve the integrity of the MEE and data through the use of mechanisms that prevent unauthorised or unintended changes, and prevent the introduction of malware.
- **Security** - Preserve the confidentiality of user data through the use of mechanisms to prevent both unauthorised access and the retrieval of any temporary data and data remnants.

## Design and Development

### Overview of MEE Design

The MEE is a USB bootable secure portable execution environment designed to provide an easy to use and secure computing platform for teleworking. The MEE consists of a core system which is built as a software image and provides a remote access capability that can be augmented with specific telework software prior to distribution; it is designed to prevent a teleworker installing additional applications. The MEE is primarily designed to be a remote access client to a corporate server, although nothing prevents the MEE working as a standalone system (with Internet access as required) when the necessary telework applications have been built into the image.

The MEE design process commenced through a requirements analysis to specify a set of requirements that would satisfy the design objectives. The requirements were then used to direct the design and development work to be performed. For each of the five design objectives a summary of the objective's requirements is given and the rationale for the selected design and implementation is discussed.

### Implement using freely available open software to exploit existing innovations and enable rapid development

#### *Requirements Summary*

The requirements analysis identified that the selected open software should execute on the widest variety of PCs, look and feel like Microsoft Windows, have acceptable licensing conditions, be able to satisfy the MEE objective primarily through configuration settings rather than by source code changes, have a strong technical support community and support a wide range of software from which to select the applications to support telework.

#### *Selection of Ubuntu*

Achieving a Microsoft Windows look and feel could obviously be achieved if the Windows OS had been selected. Although Windows is the most commonly used OS (OS Market Share, 2012) it was eliminated as the basis of the MEE due to licensing restrictions, lack of openness and being proprietary software which would limit the ability to make changes.

A number of Linux and BSD Unix variants were considered and evaluated. The Linux distribution Ubuntu was selected because:

- It is one of the more popular Linux desktop distributions with a large range of application software (BestLinux, 2011).
- Certain releases are commercially supported for five years.
- It can be installed onto external media and uploaded to execute on a PC.
- It has appropriate licensing conditions.
- It has a desktop environment that is configurable to give a Windows-like look and feel.

#### *Selection of Distribution*

The Ubuntu desktop distribution is available as a live DVD (Ubuntu, 2012) or as a system that can be installed onto either a PC's internal or external disk. As a live DVD Ubuntu is packaged as a read-only OS bootable software image that can be uploaded from external media. The core OS on the live DVD image loads into the PC's memory with applications loaded and executed as required. It was decided to use the live DVD version as the development platform for the MEE as the live DVD was designed to:

- Load and execute from portable external media including USB attachable flash memory and disk technology.
- Boot from a wide range of PCs.
- Require no access to the PC's HDD (only the PC's processor and memory are used).

#### *Rapid Development and Customisation*

To achieve the rapid development goal it was decided to customise the Ubuntu live DVD distribution by disabling/removing unnecessary functionality and known vulnerabilities using OS and command configuration options. Being open source Linux live DVD distributions allow customisation, providing a wide range of configuration options to enable an execution environment to be developed to address a particular need. A number of forums and publications are available that provide advice on customising Linux and Ubuntu live DVDs (LiveCDCustomization, 2012; Negus, 2006).

An alternative approach would be to remove and change source code and then recompile and rebuild the system. Such an approach would have taken a considerable amount of effort over a prolonged period. Sophisticated customizations can be achieved through changing system configuration settings without having to make source code changes which would lead to the riskier development route of recompiling, rebuilding and the extensive testing of the code changes.

## **Be up-loadable on to any PC from either a secure PESE device or a standard thumb drive**

### *Requirements Summary*

The requirements analysis identified that the selected open software should be capable of booting from a range of portable USB storage devices and execute on a wide range of PCs without specific device drivers being installed.

### *USB Storage Device*

The MEE was developed as the secure portable execution environment for a secure PESE. A secure PESE (James, 2008) is a solution to enable secure remote working, as it provides both a secure execution environment and secure storage, only requiring the use of a PC's processor and memory to execute. A secure PESE provides mechanisms to protect both the integrity of the execution environment (e.g. the MEE) and the confidentiality of stored data. Thus a secure PESE can minimise the risk to information loss if the teleworker's PC is breached or compromised.

The Silicon Data Vault (SDV) provides a hardware platform to develop a secure PESE. The MEE was designed to utilise the SDV (Secure Systems, 2012) to provide a secure PESE through the use of the following SDV features:

- A bootable mode to load and boot the MEE.
- Hardware based encryption of an integral storage medium.
- Strong authentication before access to both the MEE and integral storage is allowed.
- Partitioning of the storage medium with Read-Only, Read-Write and No-Access permissions.
- Storage area for temporary data created by the installed secure portable execution environment.

The MEE was also designed to be booted from a standard USB thumb drive. The thumb drive obviously lacks the security features of the SDV, but does provide a telework computing environment with an appropriate level of security if no local storage of data is required and if the MEE is used only as a remote access client.

### *Portability*

The Ubuntu live DVD is packaged with a comprehensive set of device drivers (including commodity drivers if specific device drivers are not available) enabling the MEE to be a highly portable system.

## **Provide an MEE core system consisting of a hardened execution environment that supports remote access functionality and limits the opportunity for malicious activities by removing known vulnerabilities**

### *Requirements Summary*

Strategies to harden a computing platform to prevent malicious activities include application whitelisting (i.e. providing access to only the functionality required), regular software patching and allocation of privilege only where necessary (DSD, 2012). Identified hardening requirements for the

MEE were generally aligned with these strategies and included the need for the selected open software to be configured to only allow access to commands necessary to support telework, to remove known vulnerabilities from retained commands and to prevent a user from both installing software and performing privileged/administrator actions.

Instead of patching a deployed MEE it was decided that the software patches/updates requirement could be achieved via building a new MEE software image and distributing the image to a teleworker on a USB device. A further requirement for the hardened environment was to limit external attacks and also protect against a teleworker with good IT skills accidentally or deliberately subverting the MEE.

#### *Ubuntu Live DVD – Architecture and Operation*

The following high level overview identifies aspects of the Ubuntu live DVD architecture and operation that are relevant to support the description and understanding of how the Ubuntu distribution was hardened to produce the MEE. The overview also supports the description of how the security objectives were satisfied.

The Ubuntu OS architecture consists of a monolithic kernel and a collection of libraries and commands (i.e. Ubuntu utilities and applications). The libraries and commands access kernel services through a system call interface. The set of libraries and commands to be included in a distribution are packaged, installed and configured through a package manager. Commands are executed within a process structure with process management ensuring a process is allocated resources and scheduled according to its respective priority. Ubuntu implements a paging memory management scheme with a swap partition on a disk drive that allows memory pages to be released by swapping out the content of pages to the swap partition when a process needs more memory. As the Ubuntu live DVD version may be executed from read-only media it is designed to execute without a swap partition with the least used pages released when the memory is full and a new page is required; although a swap partition can be configured if read-write media is used.

To execute an Ubuntu live DVD OS image from a USB storage device requires the PC embedded firmware to be configured to boot from a USB device when it is plugged into the PC and the PC is powered on. The PC embedded firmware executes the USB device's master boot record resulting in a boot loader commencing the process of loading the Ubuntu live DVD kernel. Although a live DVD is designed to execute from portable read-only media it still requires access to a writeable file system. Provision of a writeable file system is achieved through the use of a RAM disk, a technique to configure part of the PC's memory to appear and behave like a writeable file system on a disk drive. The RAM disk writeable file system is formed from the following three separate entities held on the USB device:

1. A root file system known as the initial RAM disk, containing libraries, commands and data necessary to allow the OS to load and execute.

2. A “squashfs” read-only compressed file system image (Squashfs, 2012), containing the Ubuntu libraries and commands.
3. An optional writeable file system known as the persistent partition, containing data (including any temporary data) changed/created by a user/process. If a persistent partition does not exist then no data changed or created during an Ubuntu live DVD session is retained.

The process of loading the Ubuntu live DVD kernel and constructing the RAM disk based writeable file system consists of the following steps:

#### Loading the Kernel

1. A boot loader loads the kernel and then the kernel establishes the initial RAM disk.
2. The initial RAM contains a number of scripts that execute a range of libraries and commands including creating the default user and establishing the user security and access control policy – a capability known as the Ubuntu PolicyKit allows the system security controls to be defined.
3. Once the executing kernel has determined the PC hardware configuration the appropriate kernel loadable modules containing device and system drivers are loaded.
4. The kernel will continue to load and execute libraries and commands until X-Windows, the underlying Linux/Ubuntu GUI starts and the user is presented with the desktop environment.

#### Constructing the Writeable File System

5. The “squashfs” file system is structured and indexed to allow the loading of libraries and commands as required/selected, i.e. when a process selects a file only the required branch of the read-only file system containing the required file is decompressed and loaded.
6. To enable the presentation of a single file system the “unionfs” module (Unionfs, 2012) is used by the kernel to merge the initial RAM disk with the required decompressed branch(es) of the read-only file system to present a single RAM disk writeable file system.
7. If the (optional) persistent partition contains data (created from an earlier session) the “unionfs” module is used to update the RAM disk writeable file system with relevant data from the persistent partition. If data in the persistent partition contains changes to libraries, commands and data loaded from the “squashfs” read-only compressed file system then the changed data item from the persistent partition takes precedence and overwrites the corresponding item in the RAM disk.
8. As data is changed or created in the RAM disk writeable file system the respective data is also automatically written to the persistent partition on the USB device through a “copy on write” mechanism (Quigley et al, 2006) in the “unionfs” module.

If the Ubuntu live DVD software image is installed on read-only media then it is able to execute without a swap partition, however when the same image is installed on a USB storage device a swap partition can be configured and used on the device.

Ubuntu has two user modes; an all privileged user (super user or su) to perform system administrator actions and a non-privileged user. When the Ubuntu live DVD is started it creates the non-privileged user. The desktop environment allows the user to execute commands, however if the user wants to execute a privileged command then the user has to explicitly change to the privileged mode. The PolicyKit (a set of commands, library and database) establishes the access rights to privileged commands for an unprivileged user as part of the boot process. A command can be executed outside the desktop environment through either a terminal capability that is part of the desktop environment or the virtual console capability.

### *Hardening*

The hardening of Ubuntu to produce the MEE consisted of the following customisation activities:

- *Remove/Disable Unnecessary Commands:* The MEE core system is required to provide a minimal functionality platform upon which an organisation's teleworking application suite can be added; no other commands are required unless needed to support the core system.
- *Removing Vulnerabilities:* By removing known vulnerabilities from commands retained in the MEE core system the attack surface is reduced. Also as unnecessary commands have been removed there is significantly less functionality (that may have contained vulnerabilities) for an attacker to exploit.
- *Disable Privilege:* Disabling the ability to perform privileged actions prevents accidental or deliberate damage by a user/process and also limits the possibility of exploitation of privilege by any introduced malware.
- *Disable Command Line Execution:* The MEE is designed for specific teleworking tasks, the intent is only to provide a user with the application suite required for telework and therefore access to the command line is unnecessary.
- *Prevent Access to PC's HDD:* To prevent the storage of any data on a PC's internal HDD by either the user or the actions of the MEE, access to the HDD is prevented.

### *Removing Unnecessary Commands*

The Ubuntu package management system allows for the selection of packages that install the libraries, commands and supporting data onto a live CD "squashfs" read-only file system image. It was decided that the MEE core system functionality should only consist of:

- A small set of commands to enable remote access to a server – i.e. a browser (Firefox) and a standard remote terminal client.
- Administration commands to configure a network connection, printer and PC peripheral settings.
- Access to data storage partitions on the USB device (holding the MEE) and/or other attachable devices.



- A desktop environment (Gnome) from which to access the included commands and storage partitions.

Although anti-malware, personal firewall, VPN and virtual machine client software were considered key applications for the MEE they are typically server/host technology specific and/or organisation specific and therefore were not included in the MEE core system. It is expected that an organisation using the MEE would build their required security and remote access applications into an MEE software image as part of configuring the system for telework.

Un-installing a large number of packages is possible but not straight forward. Undocumented dependencies exist between packages and therefore a number of repetitive builds and tests were required before the minimum set of packages required was identified. Due to certain dependencies some packages had to be retained resulting in unnecessary commands being included in the MEE core system. To prevent the included but unnecessary commands being executed the execution permissions on the command executable files were removed before building a “squashfs” read-only file system.

#### *Removing Vulnerabilities*

With only remote access commands and administration commands to configure a network, a printer and host PC mouse and sound, the MEE provides a minimal attack surface. However, both the browser (FireFox) and desktop environment (Gnome) that form part of the MEE core system are functionality rich and could be exploited.

Firefox was modified to generate a secure browser based upon Fireguard (Griffiths and James, 2010), a browser with vulnerable functionality disabled. The MEE version of Firefox had the following known potentially vulnerable capabilities disabled:

- Installation of browser add-ons.
- Automatic software updates.
- Java applets executing.
- Automatic playing of audio and video files.
- Javascript execution.
- Retention of residual data, i.e. cookies, site visit histories, saved passwords, form data and cached web pages.

The resultant reduced functionality Firefox browser is designed to provide sufficient functionality to support secure remote access activities. The goal was to limit its use as a general purpose browser and therefore the removal of a range of standard browser features satisfied this goal, in particular disabling Javascript execution limited access to many web sites.

Removing access to some commands (packaged separately to Gnome) that form part of the standard Gnome desktop environment was achieved by the package removal process described

above. For other non-required commands included specifically in the Gnome package, access was prevented by removing the execution permissions of the respective command executables before building the MEE software image. Removal of the menu entries in the Gnome GUI for non-required commands was achieved using the Gnome configuration tool; the tool was also used to change settings to prevent the user from modifying the Gnome configuration and to stop the autorun of executable code from removable media. The resultant Gnome configuration produced a simple GUI with access to only the commands included in the MEE core system. As Figure 2 depicts, the desktop environment was able to be configured to present a clean user interface similar to Microsoft Windows.

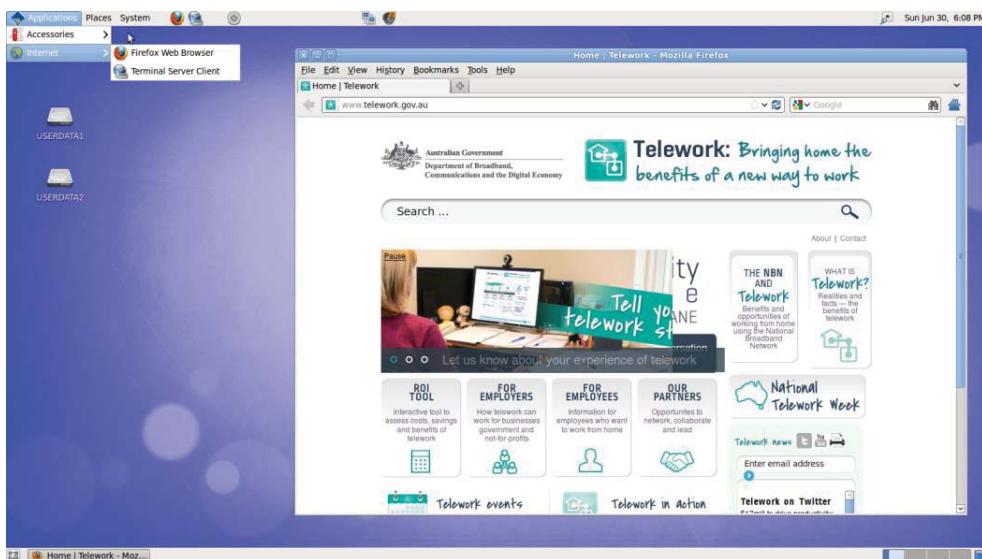


Figure 2. MEE Desktop

### *Removing Privilege*

Removing unnecessary privilege was achieved by deleting and removing specific settings in the PolicyKit database. The database was changed so that a non-privileged user is only allowed to set up and manage a network connection, set the time and time zone, set up and manage a printer and mount attachable storage devices. All other privileged actions including the installation of software are prevented.

### *Removing Command Line Execution*

The Gnome configuration tool was used to disable access to the terminal command line provided by the desktop environment. Disabling the virtual console facility involved changing the appropriate setting in the X-Windows graphics server configuration.

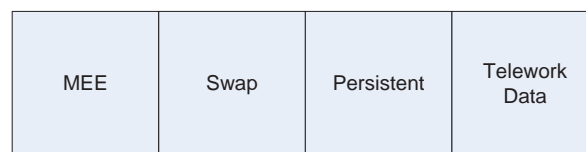
### *Prevent Access to PC's HDD*

Access to the PC internal HDD is determined by the PolicyKit. Applying the setting to prevent the mounting of the PC's internal HDD prevents the HDD being accessed.

The aforementioned set of hardening activities collectively limits the exploitable vulnerabilities whilst providing sufficient generic functionality to deliver a secure teleworking computing platform.

#### *Implementation Paradigm*

The MEE is built as a software image that is copied into its own partition on the USB device; on an SDV the MEE is protected by read-only access permissions that only an administrator can override. Other supporting partitions can be created on the USB device. Figure 3 presents a conceptual model of the storage configuration of a USB read/write storage device consisting of the MEE partition, an optional swap partition, a persistent partition for changed and created data and an optional telework partition for telework specific data.



*Figure 3. Model of MEE Storage Configuration*

Although conceptual this model is typically the configuration used for an SDV. For a standard thumb drive the swap and telework partitions would normally be omitted as the thumb drive lacks mechanisms to protect data confidentiality. It is necessary for the thumb drive to have a persistent partition, but as discussed below this partition can potentially cause a data confidentiality issue.

**Preserve the integrity of the MEE and stored system and user data through the use of mechanisms that prevent unauthorised or unintended changes, and prevent the introduction of malware**

#### *Requirements Summary*

The following risks were identified in a telework information security risk assessment (James, 2011):

- **Risk: Unintentional introduction of malware that could be used to launch a cyber-attack.**
- **Risk: Corruption of the teleworker's computing environment which could cause a denial of service.**

The integrity requirements were therefore to stop malware and system corruption.

#### *Preventing Malware*

In a teleworking environment the accidental introduction of malware can be through:

- Uncontrolled/unmanaged access to the Internet.
- Execution of an application (command) e.g. using an email package and opening a malicious attachment on a received email.

- Allowing the “autorun” of applications from a removable storage device that is plugged into the telework PC.

A PC used for telework may also be used by individuals other than the teleworker, particularly if the PC is owned by the teleworker as it may be used/shared with family. It is possible that the PC's other users could unintentionally (or even possibly deliberately) introduce malware through browsing malicious web sites, executing applications from the PC HDD and/or attaching removable media containing malware.

The MEE is completely separate to the OS and applications on the PC HDD. It does not matter if the PC is used by other individuals who introduce malware as the teleworker does not use the OS and applications installed on the PC HDD. The MEE is able to reduce the possibility of malware being introduced by providing a separate hardened execution environment that is uploaded onto the PC by the teleworker when telework is to be performed. The MEE is designed to prevent any access to the PC HDD and therefore any previously introduced malware on the PC's HDD is unable to execute. It is conceivable that malware may be present in the PC's memory immediately prior to a teleworker loading the MEE. However, as the MEE should only be loaded from a 'cold boot' it is not possible for any malware previously resident in the PC's memory to be present.

It is also possible that the teleworker may browse web sites and unintentionally introduce malware. However, the possibility of malware being introduced has been reduced as the MEE version of Firefox has had known potentially vulnerable capabilities disabled.

With unnecessary commands removed and known potentially vulnerable capabilities in retained applications disabled the MEE provides limited opportunity for malware to be introduced through a command. It would also be expected that any organisation specific teleworking commands would be hardened before forming part of the MEE software image.

The MEE prevents the 'autorun' of applications from an attached USB storage device.

If the MEE is installed on the SDV further protection against malware is provided by the SDV's read-only partition. In the unlikely event of the teleworker inadvertently introducing malware whilst using the MEE the read-only partition in which the MEE is installed prevents the malware becoming stored in the partition. The MEE's underlying Ubuntu live DVD technology is designed to execute from read-only media and therefore the MEE is ideally suited to execute from an SDV read-only partition.

The hardening, removal of known vulnerabilities coupled with the read-only partition capability of the SDV provides a capability to counter the threat of cyber-attack through the unintentional introduction of malware. The teleworking organisation can have assurance that when a PC is powered on and the MEE is uploaded then a secure computing environment is available for telework that is not readily susceptible to malware.

### *Preventing Denial of Service Attacks*

Corruption or destruction of the teleworker's computing environment leading to a denial of service can occur through:

- A malware attack causing deletion and/or alteration of the MEE's executable and data files.
- Accidental or deliberate deletion or alteration by the teleworker of the MEE's executable and data files.
- Direct targeted attack of the MEE's executable and data files by a hostile individual (an unauthorised individual).

The mechanisms and approach adopted to prevent malware attacks have been enumerated above.

The MEE is designed to prevent a teleworker accidentally altering the MEE's executable and data files whilst the MEE is executing through the combination of a minimal functionality hardened desktop, no privilege, removal of unnecessary commands, no command line access and removal of command execution permissions should a teleworker be able to gain access and execute the available commands.

When the MEE is not executing but the thumb drive is plugged into a PC and accessed it is possible for the MEE software image to be accidentally (by a teleworker) or deliberately (by an unauthorised individual) deleted/corrupted simply by overwriting the MEE image with other data. A secure PESE is therefore required to prevent the accidental or deliberate deletion/corruption of the MEE's software image. The SDV's authentication mechanism prevents a hostile individual gaining access to the device and the SDV's read-only partition protects the MEE image and therefore prevents accidental deletion/corruption.

**Preserve the confidentiality of user data through the use of mechanisms to prevent both unauthorised access and the retrieval of any temporary data and data remnants**

### *Requirements Summary*

The following risks were identified in a telework information security risk assessment (James, 2011):

- **Risk: Data loss if the teleworker's PC/laptop is lost or stolen.**
- **Risk: Data loss if non-secured portable storage media is lost/stolen.**
- **Risk: Data leakage if sensitive data remnants reside on a PC/laptop.**

The confidentiality requirements were to stop both data loss and leakage.

### *Preventing Breach of Confidentiality through Data Loss*

The MEE is an uploadable secure portable execution environment that only utilises a PC's processor and memory. The MEE prevents access to, and storage on the PC's HDD, therefore the loss or theft of the PC will not result in a breach of data confidentiality.

The teleworker may need to store sensitive data on removable storage media:

- When data is being processed locally rather than via remote access.
- As a local backup medium.
- If locally generated and processed data is to be transported from/to the telework location to/from another location, e.g. a corporate office.

The telework data partition is provided on the USB device for local processing, backup and transportation. However, a telework data partition should not be created for the thumb drive configuration as the thumb drive has no mechanisms to protect access to the data if the thumb drive is lost/stolen. An SDV should be used for local processing, backup and data transportation as it provides both authentication to prevent unauthorised access and data encryption to protect the confidentiality of data. If the SDV is lost or stolen the authentication will deny access and if the device was dismantled to gain access to its integral storage the encryption capability ensures stored data cannot be read. If data is stored on a device separate to the USB device hosting the MEE then the separate storage device should be a self-encrypting.

#### *Preventing Breach of Confidentiality through Data Remnants*

All PC OS' and many applications make temporary copies of the data being processed as part of their standard operation, often unbeknown to the PC user. This temporary data may remain on the PC HDD if the OS and software applications are not configured to perform a comprehensive "tidy up" at the end of execution. If encryption is not used then at a later date it may be possible for an unauthorised individual to use data acquisition tools to recover sensitive data remnants.

The MEE makes temporary copies of data. The MEE prevents access to the PC HDD and therefore it is not possible for data remnants to be saved (by the MEE or a command) on to the PC HDD. The MEE will save temporary data in both the swap partition and persistent partition resulting in data remnants in these partitions.

As identified above a thumb drive does not provide any security mechanisms to protect the confidentiality of data if the device is lost or stolen. A swap partition is not required for the correct operation of the MEE software image on a thumb drive; however a persistent partition is necessary for correct operation. The persistent partition presents a potential vulnerability as sensitive data remnants may be recoverable from the persistent partition if the thumb drive is lost/stolen.

The SDV provides a solution to prevent the recovery of data remnants as the swap and persistent partitions are protected by authentication and encryption.

## **Demonstration**

A comprehensive program of testing and an end-user trial were used to demonstrate the capabilities of the MEE. The testing was conducted over a period of two months which included

regression testing. The trial took place in a commercial organisation that already had a large team of teleworkers. The organisation wanted to consider alternative telework computing platforms that would improve information security, reliability and productivity.

*Building the MEE*

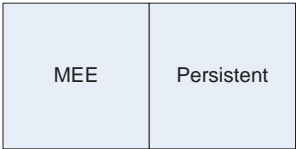
To ensure compatible versions of libraries and commands were used the MEE software image was built on a PC executing the same version of Ubuntu. The MEE image built for system testing was based upon the core system with no additional telework commands added. The user trial did not commence until all system and regression testing had completed and an MEE release was available that was considered product quality. The MEE image built for the trial also included the teleworking software used by the organisation.

The MEE images were placed onto SDVs and standard thumb drives with both devices used for testing and trialling. All possible storage configurations were subjected to system testing but for the trial the storage configurations presented in figures 4 and 5 were utilised.



*Figure 4. SDV Storage Configuration for Trial*

For the trial the SDV was configured with a swap partition and two telework partitions in addition to the standard read-only MEE partition and persistent partition whilst the thumb drive had neither swap nor telework data partitions due to the aforementioned data confidentiality vulnerabilities.



*Figure 5. Thumb Drive Storage Configuration for Trial*

*Testing*

The MEE was subjected to extensive product style testing by a team of independent testers. A test plan, test specification and test procedures were prepared and a test management tool (TestLink, 2012) used to control the test process and record the test results. In addition to confirming the correct operation of functionality test procedures were prepared to test exception handling, vulnerability identification, usability and stress testing. Regression testing was performed for new releases of the MEE that addressed problems found in earlier releases. The test results were captured in the test management tool with issues identified (i.e. bugs and proposed changes) recorded and managed in an issue management tool (MantisBT, 2012).

### *Trial*

A trial of the MEE was arranged with a company that already had a proven track record over a number of years of allowing a sizable number of its customer support staff to telework. These teleworkers provided technical support for internet and telecommunication services and were part of a customer support division (the other customer support staff were office based). The teleworkers were experienced and skilled IT support personnel. To perform support work each teleworker was furnished (by the company) with a customer support application suite consisting of a VPN, virtual machine client and softphone all installed on a laptop running Microsoft Windows. All data storage and processing was performed within a virtual machine executing on a remote server so no commercial data was stored on the laptop. The laptop had a virtual machine client executing over a VPN to the remote server based virtual machine.

The company agreed to allow a group of its teleworkers to participate in the trial and each teleworker was issued with either an SDV or thumb drive (with the MEE and customer support application suite installed). The teleworker used either a personal PC or continued to use the company furnished laptop as the platform to execute the MEE and connect to the remote server to perform customer support work. The trial was conducted over a one week period. A plan was prepared in conjunction with the company that defined how observations would be recorded, how to handle any problems with the MEE during the trial and how to measure the effectiveness of the MEE as a possible replacement solution.

## **Evaluation**

### *Evaluation of Implementation Paradigm*

As a standard thumb drive has no security mechanisms only the MEE's hardened capabilities provide security. For an organisation that implements remote access based teleworking where there is no local data storage the MEE and thumb drive configuration can provide a low cost solution with an appropriate level of security. A full risk assessment would however be necessary before deploying such a solution.

The SDV's read-only partition provides integrity protection for the MEE software image and the device's authentication and encryption mechanisms provide data confidentiality. The MEE and SDV configuration provides a secure solution where system/data integrity and confidentiality is important.



## Evaluation of Demonstration Results

The evaluation of the demonstration results is summarised in tables I and II below and is categorised for each of the five objectives into design outcomes, test results and trial observations:

Design Objective	Evaluation		
	Design Outcome	Test Result	Trial Observations
<b><i>Implement using freely available open software to exploit existing innovations and enable rapid development.</i></b>	<p>A rapid development was achieved - the first release was available within a few months.</p> <p>The Ubuntu live DVD innovations, structure &amp; customisation features provided a reliable and tested platform that enabled the MEE to be built without source code changes.</p>	Using a commercially supported Linux for the MEE provided assurance that system testing could commence from a proven platform. The large user community and support websites assisted in the rapid resolution of issues.	<p>The teleworkers found the MEE desktop interface easy and familiar to use with no training required.</p> <p>A problem that initially impacted the trial was that the organisation's softphone was not available as an Ubuntu command. The MEE software image was rebuilt with a Windows emulator (CrossOver, 2012) to execute the softphone.</p>
<b><i>Be up-loadable on to any PC from either a secure PESE device or a standard thumb drive.</i></b>	Using the Ubuntu live DVD proved to be a good platform to develop a MEE software image that could boot from a USB device onto a large range of PCs.	The MEE was tested on a comprehensive range of x86 PCs and booted and executed without issue; the MEE was not able to boot from some very old PCs that had embedded firmware that did not support a USB boot capability.	The teleworkers used a range of PC/laptops in the trial, The teleworkers reported no issues in loading and executing the MEE. A high level of satisfaction was reported on the MEE's performance and convenience of use.
<b><i>Provide an MEE core system consisting of a hardened execution environment that supports remote access functionality and limits the opportunity for malicious activities by removing known vulnerabilities.</i></b>	<p>The Ubuntu live DVD provided the openness and flexibility to achieve the required hardening. Incompleteness and inconsistencies in the customisation guidelines and documentation did result in a number of iterations of different configuration settings before the desired and optimal solution was achieved.</p> <p>Lack of a software patching capability was probably a security weakness (see details below).</p>	Functionality, stress, exception handling, and useability testing identified a number of issues. The issue resolutions were not overly challenging to identify and apply.	The teleworkers reported no impact to their productivity from the hardened and limited MEE functionality. No failures in the MEE caused loss of work time. The organisation had stated one reason to consider a move away from its laptop solution was periodic laptop outage (usually due to the teleworker changing laptop settings or adding applications) which had caused loss of work time.

Table I. Evaluation of Developmental, Operational and Functional Objectives

*Software patching:* Full consideration (during the design) was given to the lack of an MEE software patching capability and hence the inability for a user to perform software patching (at the telework location) to address a security vulnerability. Software patching is recognised as a hardening and cyber intrusion mitigation strategy (DSD, 2012). Whilst the copy-on-write capability would allow an executable file to be updated for an Ubuntu live DVD image (residing on a writeable storage device configured with a persistent partition), the MEE hardening prevents a teleworker updating/changing an MEE executable.

It was decided that a limited functionality hardened software image that could only include specific telework functionality would provide a sufficiently secure platform. If a security vulnerability was identified a new release of the MEE software image would be issued either on a new (low cost) thumb drive or by an administrator updating the teleworker's (higher cost) SDV. The main disadvantage with a new release approach is the time and logistics required to ensure the release is distributed to all users, especially when multiple releases are issued over a short time period.

Design Objective	Evaluation		
	Design Outcome	Test Result	Trial Observation
<b><i>Preserve the integrity of the MEE and data through the use of mechanisms that prevent unauthorised or unintended changes and prevent the introduction of malware.</i></b>	The thumb drive MEE did not provide a solution that addressed all integrity and confidentiality requirements but the SDV and MEE combination fully satisfied the security objectives.	The vulnerability testing confirmed the strength of the hardening, although a potential vulnerability was identified - if a command with a GUI was placed on the USB device the GUI command could be executed (see details below).	The teleworkers had good IT skills as they provided customer support for IT and telecommunication services, although no teleworker had in depth programming and/or Linux administration experience. As part of the trial the teleworkers were told they could attempt to make changes to (attack) the MEE. No teleworker managed to damage or corrupt the MEE in a manner not already known.
<b><i>Preserve the confidentiality of user data through the use of mechanisms to prevent both unauthorised access and the retrieval of any temporary data and data remnants.</i></b>	The potentially vulnerable persistent partition limits the deployment options for the MEE and thumb drive configuration.		Analysis of the USB devices after processing found only limited intelligible data remnants.

Table II. Evaluation of Security Objectives

*GUI command vulnerability:* If a teleworker placed a GUI command (malicious or otherwise) on to a writeable partition of the MEE's USB device (or on another mountable storage medium) then it was possible to execute the command. As such a (introduced) GUI command would not be able to exploit privileged functionality and with substantial functionality removed the impact of the vulnerability would have limited effect.

### *Evaluation Synopsis*

The evaluation has shown that successful research outcomes were achieved with few potential vulnerabilities identified. The trial demonstrated how a secure portable execution environment could be used for transaction-oriented customer support work. In addition to acknowledging the security features the trial organisation commented that it valued the MEE because it:

- Provided a separate pluggable execution environment that provided a secure thin client for remote access.
- Reduced IT costs – a teleworker can use his/her own PC and no PC sanitation is required before disposal as no sensitive data remnants reside on the PC.
- Would be likely to improve productivity as the restricted hardened environment could not be changed and it only provided the functionality necessary to perform telework.

The organisation considered the MEE had the potential to provide a more secure, reliable and productive computing environment for its teleworkers than its current solution.

### **Conclusion**

The selection of design science as the research methodology proved to be appropriate. Design science enabled structured prototyping to be achieved, supporting the build-test approach adopted. The methodology particularly suited the researchers due to backgrounds in software product development and the use of commercial system design methodologies.

Whilst the lack of a patching capability can be justified for a hardened limited functionality MEE the premise does not hold for non-hardened telework commands which are built into the software image. Increasingly it is vulnerabilities in application software that is exploited by malware (Symantec, 2013). Therefore future work on the MEE development is to include a software patch capability that can be integrated within the MEE's design objectives and implementation paradigm.

The demonstration and evaluation of the MEE provided evidence that the MEE is a solution to address the two problems defined for the research. The design outcomes and test results showed that a secure teleworking platform had been constructed. The trial provided evidence that more closely addressed the problem of improving security for existing teleworkers. The MEE software image on a SDV addresses the two problems and fully satisfies the five design objectives. The MEE software image on a thumb drive did not satisfy the security objectives but could provide a level of security suitable for telework where the MEE is used as a remote client with no local storage of sensitive data; the trial organisation found the low cost thumb drive satisfied its requirements. Both solutions need the organisation defined anti-malware, personal firewall and VPN software built into the MEE software image before deployment.

## References

- AccessEconomics (2010), "Impacts of Teleworking under the NBN", Prepared for the Department of Broadband, Communications and the Digital Economy, Access Economics Pty Limited, July 2010, available at:  
[http://www.dbcde.gov.au/data/assets/pdf\\_file/0018/130158/ImpactsofteleworkingundertheNBN.pdf](http://www.dbcde.gov.au/data/assets/pdf_file/0018/130158/ImpactsofteleworkingundertheNBN.pdf)
- ACM-DL (2013), Citations for A Design Science Research Methodology for Information Systems Research, Association of Computing Machinery Digital Library, September 2013, available at:  
<http://dl.acm.org/citation.cfm?id=1481765.1481768&coll=DL&dl=GUIDE&CFID=363715687&CFTOKEN=95589558>
- BestLinux (2011), "The Linux distro of 2011!", available at: <http://www.tuxradar.com/content/best-distro-2011>.
- Clear, F. (2007), "SMEs, electronically-mediated working and data security: cause for concern?", International Journal of Business Science and Applied Management, Vol. 2, Issue 2, 2007.
- DBCDE (2011), "Telework Forum: Bringing home the benefits of telework using the NBN, A record of the Telework Forum held 3rd August 2011", A partnership between the Department of Broadband, Communications and the Digital Economy and the Australian Information Industry Association, August 2011, available at: <http://www.nbn.gov.au/get-involved/upcoming-events/nsw/telework-forum-bringing-home-the-benefits-of-teleworking-using-the-nbn>.
- Deloitte (2011) "Next Generation Telework: A Literature Review", Report by Deloitte Access Economics for the Department of Broadband, Communications and the Digital Economy, July 2011, available at: [http://www.nbn.gov.au/files/2012/02/Next\\_Generation\\_Telework-A\\_Literature\\_Review-July\\_20111.pdf](http://www.nbn.gov.au/files/2012/02/Next_Generation_Telework-A_Literature_Review-July_20111.pdf).
- DESIST (2013) "Design Science Research in Information Systems and Technology", Web Site , accessed September 2013, available at: [www.desist.org](http://www.desist.org)
- DESIST-2013 (2013) "Eighth International Conference on Design Science Research in Information Systems and Technology", Aalto University, Helsinki, 2013, September 2013, available at: [www.desist](http://www.desist.org).
- DSD (2012), "Strategies to Mitigate Targeted Cyber Intrusions", Cyber Security Operations Centre, Defence Signal Directorate, October 2012, available at:  
[http://www.dsd.gov.au/publications/Top\\_35\\_Mitigations\\_2012.pdf](http://www.dsd.gov.au/publications/Top_35_Mitigations_2012.pdf)
- Griffiths, D. and James, P. (2010) "Fireguard – A Secure Browser with Reduced Forensic Footprint", Journal of Network Forensics, Vol. 2, Issue 2, Summer 2010.
- Hevner, A.R., March, S.T., Park, J. and Ram, S. (2004), "Design Science in Information Systems Research", MIS Quarterly, Vol.28, No 1, pp 75-105, March 2004.
- James, P., (2008), "Secure Portable Execution Environments: A Review of Available Technologies", 6th Australian Information Security Management Conference, December 2008, Edith Cowan University, Perth.
- James, P., (2011), "Are Existing Security Models Suitable for Teleworking?", 9th Australian Information Security Conference, December 2011, Edith Cowan University, Perth.
- James, P., and Griffiths, D. (2012), "The Mobile Execution Environment: A Secure and Non-Intrusive Approach to Implement a Bring Your Own Device Policy for Laptops", 10th Australian Information Security Management Conference, December 2012, Edith Cowan University, Perth.

- Lister, K. and Harnish, T., (2011), "The Shifting Nature of Work in the UK: Bottom Line Benefits of Telework", Telework Research Network, February 2011, available at: <http://www.teleworkresearchnetwork.com/whitepapers>.
- LiveCDCustomization (2012), "Ubuntu Documentation, Community Help Wiki", available at: <https://help.ubuntu.com/community/LiveCDCustomization>
- MantisBT (2012), "Mantis Bug Tracker Administration Guide", November 2012, available at: [http://www.mantisbt.org/docs/master-1.2.x/en/administration\\_guide.pdf](http://www.mantisbt.org/docs/master-1.2.x/en/administration_guide.pdf).
- Negus, C., (2006), "Live Linux CDs : Building and Customizing Bootables", Negus Live Linux Series, Prentice Hall 2006, ISBN 0-13-243274-9.
- NIST (2007) "User's Guide to Securing External Devices for Telework and Remote Access", National Institute of Standards and Technology Special Publication 800-114, U.S. Department of Commerce, November 2007, available at: <http://csrc.nist.gov/publications/nistpubs/800-114/SP800-114.pdf>.
- NIST (2009), "Guide to Enterprise Telework and Remote Access Security", National Institute of Standards and Technology Special Publication 800-114 Revision 1, U.S. Department of Commerce, June 2009, available at: <http://www.distributedworkplace.com/DW/Government/Government%202009/NIST%20Guide%20to%20Enterprise%20Telework%20and%20Remote%20Access%20Security.pdf>.
- OS Market Share (2012), "Desktop Operating System Marketshare", NetMarketShare, available at: <http://www.netmarketshare.com/operating-system-market-share.aspx?qprid=10&qpcustommd=0>
- Peffer, K., Tuunanen, T., Rothenberger, M.A. and Chatterjee, S. (2007), "A Design Science Research Methodology for Information Systems Research", Journal of Management Information Systems, Vol 24, No 3, pp 45-77, Winter 2007-8.
- Quigley, D., Sipek, J., Wright, P. and Zadok, E., (2006), "Unionfs: User- and Community-Oriented Development of a Unification File System", Proceedings of the Linux Symposium, Vol. 2, pp. 349-362, July 19th–22nd, 2006, Ottawa, Ontario, Canada.
- Secure Systems (2012) "Mini Silicon Data Vault", 2012, Retrieved July 2012 from URL: <http://www.securesystems.com.au/secure-systems-mini-sdv.html>.
- Squashfs (2012), "What is Squashfs", The Linux Documentation Project, November 2012, available at: <http://www.tldp.org/HOWTO/SquashFS-HOWTO/whatis.html>
- Symantec (2013), "Internet Security Threat Report 2013, 2012 Trends", volume 18, April 2013, Symantec Corporation, available at: [http://www.symantec.com/content/en/us/enterprise/other\\_resources/b-istr\\_main\\_report\\_v18\\_2012\\_21291018.en-us.pdf](http://www.symantec.com/content/en/us/enterprise/other_resources/b-istr_main_report_v18_2012_21291018.en-us.pdf)
- TestLink (2012), "User Manual, TestLink version 1.9", Manual Version 1.20 available at: [http://teamst.org/\\_tldoc/1.9/testlink\\_user\\_manual.pdf](http://teamst.org/_tldoc/1.9/testlink_user_manual.pdf).
- Ubuntu (2012), "Ubuntu Documentation – LiveCD", Canonical Limited, available at: <https://help.ubuntu.com/community/LiveCD>.
- Unionfs (2012), "Unionfs: A Stackable Unification File System", available at: <http://www.fsl.cs.sunysb.edu/project-unionfs.html>

#### 4.2.5.4 Synopsis

**Outcomes and Contribution to Knowledge:** What differentiates the MEE from products like the LPS and the Becrypt TC and hence makes it a design knowledge contribution is that the MEE can provide a hardened small functionality secure PEE. Although the LPS and Becrypt TC are small functionality secure PEEs they both lack the hardened properties of the MEE like removal of privilege, no access to command line processing, hardened applications (e.g. browser) and hardened desktop environment. When the MEE is combined with the Mini-SDV secure storage device to form a secure PESE it provides an innovative capability. An additional outcome is that the paper describes how to harden a Linux distribution. The knowledge contribution is both prescriptive and descriptive. The prescriptive knowledge is the design know how developed in hardening Ubuntu to create the MEE. The descriptive knowledge is the trial observations and results which are discussed further in Chapter 5. The evaluation discussion presented in the paper is considered as part of the overall the secure PESE demonstration results in Chapter 5 and in the research evaluation in Chapter 6.

**Contemporary relevance, linkage with other papers and future direction:** The paper makes a relevant contribution to the growing area of secure portable computing devices for remote working (Osterman, 2012; Ironkey 2014). The paper also provides further input to the knowledge base of operating system hardening. The paper is linked to papers 9 and 10 as the MEE forms a capability for the high assurance secure PESE presented in Paper 9 and the MEE is proposed as a low cost approach to support a secure capability where an organisation implements a bring your own laptop policy in Paper 10.

The demonstration results and their evaluation showed that the MEE is a useable hardened operating system with hardened applications. Usability was neither impacted by the hardening nor by the storage of temporary data in a secure PESE partition. However, like Fireguard, the MEE design neither supports patching nor upgrading to a new release. Therefore the research only partially addressed the two doctoral research questions. The paper proposed that a patching capability is developed which was not progressed; the issues with implementing such a capability are discussed at the conclusion of design cycle 3 and in the research evaluation (Chapter 6).



## **4.2.6 The Mini SDV – A Platform for a Commercial Grade Secure PESE**

### **4.2.6.1 Overview**

The Mini SDV is a secure portable storage product that is an improvement upon the Pocket SDV. Stakeholder feedback (predominately Secure Systems' customers) and advances made by competing products in the market, coupled with certain secure PESE functional requirements had highlighted the need to make the following improvements to the SDV technology:

1. Reduce the form factor to a more convenient size.
2. Remove the dependency for software to be installed onto a PC for post-boot authentication.
3. Provide an anti-tamper capability.

Utilising the existing SDV technology coupled with the implementation of the above enhancements a new secure portable storage platform was developed and commercialised as the Mini SDV. The Mini SDV when packaged with the MEE and/or a secure PEE consisting of a set of applications forms a commercial grade secure PESE. The Mini SDV makes a contribution to answering the following doctoral research question: *How can anti-tamper mechanisms be implemented into a small form factor and highly portable device?*

### **4.2.6.2 Description in Published Papers**

The features and functionality of the SDV technology and the products constructed from the technology are described in Papers 1, 2, 4, 6 and 7 and also summarised in Chapter 1. Papers 2 and 4 describe the Pocket SDV. As the Mini-SDV is an improvement upon the Pocket SDV the descriptions given in Papers 2 and 4 provide a good introduction to the underlying technology and support the more succinct description of the Mini-SDV given in Paper 6. Paper 7 provides an abstract overview of an SDV device, the actual SDV used and discussed in Paper 7 was a Mini SDV. As good descriptions of the SDV technology are presented in the aforementioned papers, with the conceptual design and operational models (Figures 4.2 and 4.3) providing high level narratives, it is not considered necessary to provide a specific description of the Mini SDV in this section. Only the improved features of the Mini SDV not described in detail in the papers are discussed below.

The Mini SDV when configured as a secure PESE was positioned as a ‘Secure Laptop in Your Pocket’ for remote workers. The rationale for this positioning is discussed in the Chapter 5.

#### 4.2.6.3 Knowledge Consumed

The following knowledge was used in the design of the Mini-SDV:

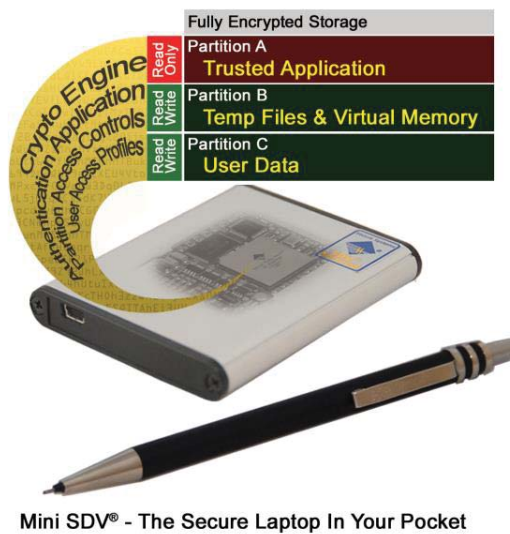
- Prescriptive knowledge in the form the existing SDV technology and the Pocket SDV design.
- Prescriptive knowledge obtained from a review of the features and functionality of the Ironkey (Osterman, 2012; Ironkey 2014), globull (globull, 2010) and the Stealth MXP (Smith, 2008) products.
- Prescriptive knowledge in the form of the concept, secure PESE functional requirements and the three conceptual models (Figures 4.1, 4.2 and 4.3).
- Descriptive knowledge from stakeholders and market monitoring providing the inspiration for the improved design.
- Formal justificatory knowledge on security engineering design (Anderson, 2008), cryptographic engineering design (Schneier, 2007; Ferguson et al., 2011) and anti-tamper design (Skorobogatov, 2012).

#### 4.2.6.4 New Improved Features

**Reduced form factor:** The requirement was to produce a design that was convenient to transport. A design constraint was the 1.8 inch solid state drive (SSD) (Lawton, 2006; SNIA, 2010) providing the storage medium. The dimensions of this SSD were: height 5mm, width 54mm and length 78mm. The SDV technology was implemented onto a printed circuit board (PCB) with similar dimensions to the SSD so that the PCB was positioned on top of the SSD and could be packaged within an enclosure with a height 10mm, width 68mm and length 86mm. The form factor of Mini SDV was very similar to that of the Bull globull. Figure 4.1 presents an image of the Mini SDV positioned against a pen as a comparative indicator of its size. The image depicts a Mini SDV configured as a secure PESE with a secure read-only partition holding a trusted application (i.e. a secure PEE), a



partition for any temporary data created by the trusted application/secure PEE and a partition for user data.



**Figure 4.4 Mini SDV Configured as a Secure PESE**

**Up-loadable Authentication Application:** The Pocket SDV required the installation of an authentication application on the host PC for post-boot authentication; refer to papers 2 and 4 for details on the post-boot authentication process. As highlighted in paper 4, the need to install a portable authentication application (PAA) onto a PC limits the SDV's portability. Products like the Ironkey (Ironkey 2014) and globull (globull, 2010) were designed with an up-loadable authentication application that required no prior installation on a PC.

Key design requirements were the ability to up-load and execute a PAA without the host PC operating system requiring administrator privilege. When a standard portable disk is connected to a PC its operating system examines the disk's partition table and allocates a partition label for each partition on the portable disk; this process is known as partition enumeration (Komski, 2010). However, when an SDV secure storage device is connected to a PC a PAA is required to execute instead of the PC operating system examining the SDV's partition table. Therefore upon successful authentication the PAA examines the SDV's partition table and allocates a label to each accessible partition. The Pocket SDV enabled this action to occur through the pre-installed PAA which executed as a privileged service. The PAA service required privilege because the Pocket SDV used a commercial off the shelf USB interface controller which had no programmable capability. After successful

authentication the privileged service would obtain partition information by simulating the connection of a portable disk for the first time by issuing USB disconnect and reconnect commands. The Mini-SDV was designed with a programmable USB interface (Cypress, 2009) that allowed programmable logic to be coded that would interface with a new uploadable PAA and enable disconnect and reconnect commands to be performed without privilege. The use of the programmable USB integrated circuit (Cypress, 2009) resulted in a highly portable Mini SDV.

**Anti-tamper Capability:** A tamper detection and response mechanism was built into the second version of the Mini SDV. The mechanism is not described in Papers 6 and 7 where the first version of the Mini SDV (without the anti-tamper capability) is used in the described research.

The Mini SDV uses a Common Criteria (CC, 2014) certified secure microcontroller (Wong, 2007) which provides an anti-tamper capability enabling a tamper detection and response mechanism (Keller, 2010) to be developed. A tamper detection and response mechanism is typically associated with a high grade device. In a high grade device such a mechanism should provide multiple layers of tamper protection including an active tamper protection mesh (Skorobogatov, 2012). As a commercial grade device there is no specific requirement (CC, 2014; ASD 2014) for the Mini SDV to have an anti-tamper capability. However, as the secure PESE functional requirements specify an anti-tamper capability to counter threats in the remote work environment, a tamper detection and response mechanism formed part of the Mini SDV design.

The anti-tamper capability consisted of an active tamper mechanism with direct connection to the microcontroller which monitors constantly for tampering. If an attempt is made to dismantle the Mini SDV a tamper event will occur resulting in the deletion of the encryption key. As the complete contents of the Mini SDV are encrypted the deletion of the key renders the device inoperable.

**Secure PESE Modes of Operation:** When configured as a secure PESE the Mini SDV can provide two modes of operation, Fully Trusted (also known as System) mode and Assured (also known as Guarded) mode. Paper 6 discusses how Fireguard, a trusted application (i.e. an application forming all or part of a secure PEE) is used with the Mini SDV in

Assured mode. Although not directly referenced the mode of operation in Paper 7 when the MEE is used with an SDV (i.e. the Mini SDV) is the Fully Trusted mode. The dual modes of operation provide an innovative approach to facilitating a secure PESE as it provides an execution environment to suit the level of risk present in a remote work location.

#### **4.2.6.5 Knowledge Contribution**

Although incorporating innovative design the improved features in the Mini SDV are, in the main routine design work. However, the ability to configuration the Mini SDV's into a secure PESE with two modes of operation can be claimed as a design knowledge contribution. When the Mini SDV was designed (in 2009/2010) no other secure portable computing capability, that could be classified as a 'secure PESE like' provided two modes of operation, i.e. the Bull globull and MXI Security Stealth MXP were both post-boot only devices and the USA DoD LPS and the Becrypt TC were both pre-boot only devices.

In accordance with the demonstration process defined in the research design (Chapter 3) the Mini SDV was subjected to an extensive program of testing and was certified by the Australian Signal Directorate (ASD). The Mini SDV was used in the telework trial discussed in Paper 7. The Mini SDV design addressed the doctoral research question as it was shown that a tamper detection and response mechanism could be constructed into a small form factor and highly portable device.

#### **4.2.7 Combining the Mini SDV, MEE and Fireguard as a Commercial Grade Secure PESE**

The three artifacts designed in cycle 3 were configured into a commercial grade secure PESE that utilised the pre-boot and post-boot authentication to allow the remote worker to select a secure PEE suitable to a particular environment. As such a remote worker could transition to different environments yet always be able to access data. For instance:

1. A teleworker can use the Fully Trusted mode at home and the Assured mode to access data in a corporate office.
2. A mobile worker can use Fireguard in Assured mode from an airport lounge PC (that is figured to prevent booting from an external device) to securely browse the Internet
3. A deployed worker can use Assured mode in a known deployed environment or when working on PCs provided by an ally organisation, and use Fully Trusted mode when

deployed to a hastily established environment where there may be limited assurance on the trustworthiness of the available PCs.

The commercial grade secure PESE did not address the maintainability component of the research questions. Maintaining software through the application of patches and/or updating to a new release is achieved through either:

- an automatic update capability that allows the software vendor to issue updates over the Internet that are downloaded and applied; or
- an organisation taking control of the update process and issuing the updates usual after the updates has been tested and assured.

The MEE and Fireguard were specifically designed to prevent automatic updates because malware could possibly be introduced (Cnet News, 2009) or unintended changes to the hardened configuration may occur, i.e. a vendor issued update may cause the hardening to change or be weakened. The use of a Read-Only partition further compounds the ability to automatically update a secure PESE. The approach adopted is for a secure PESE to be updated by an organisation in a controlled and trusted environment. Chapter 6 evaluates maintainability issue and provides rationale for the approach adopted.

### **4.3 Developing the High Grade Secure PESE – Design Cycle 4**

#### **4.3.1 Background**

A high grade device provides a high level of assurance for the protection of data (DSD, 2010). Like the commercial grade, the high grade secure PESE satisfies the research objectives (and hence the concept), the functional requirements and the three conceptual models (Figures 4.1, 4.2 and 4.3). However, the level of rigour and the strength and depth of the implemented security algorithms and mechanisms distinguish it as a high grade device.

The motivation to develop a high grade secure PESE emerged from discussions with Australian Defence organisations, in particular the Defence Science and Technology Organisation (DSTO)<sup>60</sup> (DSTO, 2014) and the Capability Development Group (CDG)<sup>61</sup> (CDG,

---

<sup>60</sup> DSTO is the Defence agency responsible for applying science and technology to safeguard Australia and its national interests.

2014). The discussions identified a requirement for a secure PESE to support Network Centric Operations (NCO) for remote workers, in particular deployed workers at the remote nodes of the network.

NCO (Fewell and Hazen, 2003) has become a mandatory managerial and technological transition program for the modern sophisticated military force (CIOG, 2010). Preventing cyber and information security attacks is an essential aspect of NCO due to the highly classified information that maybe processed. However, the security enforced at the remote nodes of the network may not be able to achieve the same level of security as a central node possibly due to the mobile nature of the node or possibly because of the hasty establishment of the node during deployment. Weak remote node security may either prevent the dissemination of highly classified information or alternatively if highly classified data is received at the node its confidentiality could be at risk. A high grade secure PESE at a remote node can enforce the required level of security for the processing of highly classified data. The development of a high grade research artifact was supported with funding from the DSTO managed Capability and Technology Demonstrator (CTD) Program<sup>62</sup>. Paper 8 discusses how a secure PESE can support NCO and positions the design research for the high grade secure PESE presented in Paper 9.

#### **4.3.2 Using Secure PESEs in Network Centric Organisations**

##### **4.3.2.1 Preamble**

A truly net-centric organisation implements NCO (Folks and Richard, 2011) allowing remote network nodes to receive, process and send information to other nodes in the network to achieve information superiority over a competitor. A remote node in a net-centric organisation can be fixed, deployed or mobile and can consist of one or more individuals, i.e. an implementation of remote working. The motivation for Paper 8 was the identification of limited cyber and information security controls enforced at some remote nodes due to either the hasty nature of their establishment or from being highly mobile. Whilst a secure PESE can improve security at a remote node the paper also considers how a secure PESE can support other aspects of NCO.

---

<sup>61</sup> CDG is the Defence agency responsible for developing business cases for the modernisation of the Australian Defence Force.

<sup>62</sup> The CTD program provides funding for the development of innovation technology to address a defence problem.

In Paper 8 a secure PESE is called a secure portable application device (secure PAD). It was decided to use the name secure PAD as it was envisaged that the device would be used to perform a set of specific functions using one or more (probably specialist) applications. Consequently the paper also uses the term 'trusted application' instead of secure PEE. A trusted application can be a set of applications that execute directly on the host PC's operating system or a bootable operating system (with integrated applications). The paper also makes a distinction between a remote worker and mobile worker due to the high level of mobility of remote nodes in net-centric organisations.

#### **4.3.2.2 Prior Research and Knowledge**

Paper 8 considers a theory for NCO (Fewell and Hazen, 2003) and discusses how a secure PESE can support the theory. The research presented in the paper consumed the following knowledge:

- Descriptive knowledge in the form of a range of NCO and Network Centric Warfare (NCW) classifications, principles and patterns (Folks and Richard, 2011; NCW, 2009; Alberts et al., 1999) which were utilised to provide context.
- Descriptive knowledge sourced from Stakeholders i.e. the aforementioned Defence agencies.
- Prescriptive knowledge consisting of the design capabilities and implemented functionality of a secure PESE sourced from the research documented so far in this thesis and its published papers, including the conceptual design (Figure 4.2) and operational (Figure 4.3) models.
- Formal justificatory knowledge in the form of a theory for NCO/NCW (Fewell and Hazen, 2003).

#### 4.3.2.3 Paper 8

**Paper 8** - James, P. (2009) **Use of a Secure Portable Application Device as a Component of Network Centric Operations**, The Journal of Information Warfare, Volume 8, Issue 3, pp 39-46.

#### Abstract

*Network Centric Operations (NCO) allows an organisation to structure its people, processes and technology to gather and process information to ensure the right information gets to the right person at the right time in and the right form. NCO enables an organisation to achieve information superiority, and hence gain a competitive advantage.*

*A secure Portable Application Device (PAD) provides a safe and secure method of loading a trusted application into a host PC and then execute the application. A secure PAD is characterised as a USB storage device containing a trusted application which is stored in a protected partition to ensure the application is separated and protected and its integrity is preserved. The device will typically provide strong authentication to only allow an authorised user to load the trusted application into the host PC and full storage encryption to protect the confidentiality of the contents of the device. A secure PAD can be issued by an organisation to an individual to perform a secure transaction on an available host PC. The secure PAD will reduce the risk to the organisation that neither malicious software (resident on the host PC) will infect the secure PAD, nor sensitive data remnants (resulting from the transaction) will remain on the host PC hard disk drive after the secure PAD has been removed.*

*This paper considers how secure PADs can be utilised in a network centric organisation. The characteristics of NCO are identified and the functionality and capabilities of secure PADs are described. Using the characteristics of NCO and a specific use scenario an analysis is performed to determine whether a secure PAD is a suitable tool for a network centric organisation.*

#### Keywords

Information Security, Network Centric Operations, Secure Portable Application Device.

#### Introduction

A network centric (net-centric) organisation is characterised as a decentralised and distributed structure where data is collected and processed to generate accurate, timely and accessible information to enable decisions to be made that deliver a competitive advantage. Network centric operations (NCO) can be considered to be the organisational infrastructure consisting of the net-centric organisation's people, processes and technology. The objective of NCO is to provide the infrastructure in terms of both the soft and hard techniques, tools and technology to achieve information superiority and hence deliver the competitive advantage.

In this paper a secure Portable Application Device (PAD) is considered to be a portable light weight Universal Serial Bus (USB) storage device containing a trusted application and space available for storage. When the secure PAD is plugged into a host PC the trusted application is uploaded and used to perform remote processing or network transactions. The objective of the secure PAD is to provide a high degree of confidence that neither the device and trusted application nor the data processing/transaction will be compromised. The device is configured to allow all temporary data generated by the trusted application (including any virtual memory) to be written to the device rather than the host PC's hard disk drive (HDD). A secure PAD will also provide space to store data, strong authentication to prevent unauthorised access, device encryption to protect the confidentiality of the stored content and partitioning with differentiated access rights to separate and protect the contents of the device.

It is likely that some resources (workers) of a net-centric organisation will be mobile and may need or want to travel with the minimum equipment. These resources will act as information sensors in the network and may gather intelligence data and generate and process information to be distributed across the organisation. The modus operandi for such resources may include using PCs owned by another organisation (most likely a strategic partner or ally) or using a known PC that is shared amongst a number of users (for example a home PC or 'satellite office/position' PC). A secure PAD provides a useful tool for a remote or mobile resource as it can be plugged into any available host PC and used to perform secure transactions with assurance that malicious software will not infect the trusted application nor any temporary data generated by the trusted application will remain on the host PC HDD.

This paper will consider whether a secure PAD can provide a useful piece of the NCO infrastructure for remote and mobile members of a net-centric organisation. An overview of the characteristics of NCO is given. In particular, the characteristics of improved speed of command, the ability to amass effects, self-synchronisation, shared situational awareness, effects based operations, reach back, highly interoperable systems and information security are described. The functionality and capabilities of the type of secure PAD considered in this paper are explained. A hypothetical secure PAD configuration is presented together with a hypothetical use scenario for a net-centric organisation. For each NCO characteristic an analysis is given on how a secure PAD can support the characteristic. The paper concludes with an assessment of the suitability of a secure PAD as a useful component in a net-centric organisation's NCO infrastructure.

## **Characteristics of NCO**

The intent of this paper is to be non-specific with respect to the net-centric organisation considered. However, as much of the reference material utilised in the paper has been generated by organisations/individuals with a defence/military background there may be greater relevance to the application of NCO within a defence/military organisation.



The now numerous texts on NCO (Alberts et al 1999; Arquilla and Ronfeldt 2001; Hutchinson and Warren 2001; Fewell and Hazen 2003) describe the various characteristics of NCO in similar but subtly different ways, sometimes using different names for the same characteristic. This paper will draw upon the characteristics for NCO defined in *Network Centric Warfare – Its Nature and Modelling* (Fewell and Hazen 2003) which are summarised below:

**Speed of Command** is concerned with utilising the available information and infrastructure to allow rapid decision making. NCO facilitates getting the right information to the right user at the right time in the right form to enable accurate decisions to be made quickly.

**Ability to Mass Effects** is the capability of an organisation to mobilise the appropriate resources to address a problem. NCO allows for greater efficiency and effectiveness in the application of resources. Through the delivery of timely, accurate and relevant information, resources can be rapidly deployed to resolve a problem/situation and then redeployed ready for future mobilisation.

**Self-Synchronisation** is the ability of organisational resources to co-ordinate a response without necessarily receiving authorisation from the hierarchy. The decentralised and distributed nature of a net-centric organisation coupled with the ability to get timely, accurate and relevant information to a set of resources at exactly the same time can enable the mobilisation and self-management of the resources to address a problem/situation.

**Shared Situation Awareness** provides a common picture to dispersed resources. The distribution of information through the NCO infrastructure can ensure deployed resources receive the same information. Such information delivery enables a shared awareness of problems/situations among dispersed resources.

**Effects Based Operations** is concerned with manipulating the competitor/enemy to achieve a desired effect. NCO can be used to deliver misleading or incorrect information to a competitor/enemy with the intent of achieving a desired outcome. The combination of improved situational awareness, faster decision making and ability to amass effects delivered by NCO will facilitate effects based operations by allowing the net-centric organisation to dominate the 'infosphere' and thus mislead the competitor/enemy into a making decision or taking an action that achieves the desired effect.

**Reachback** is the ability to access resources not available locally. The integrated network infrastructure used to achieve NCO allows many of the resources required to solve a problem to be back office functions. Front office/front line resources can obtain and generate the information that can be analysed by the back office functions. NCO allows a front line resource to reachback and obtain information and/or solutions to problems.

**Interoperable Systems** allow seamless communication across different networked systems. NCO defines the interfaces and protocols to allow the connection of systems belonging to strategic partners and allies and achieve seamless communication.

**Information Security** ensures that only authorised users are given access to data, the data can not be changed and the data is available when required. NCO infrastructure (including policy, procedures, network and computer security and system redundancy) will enforce information security to ensure only need to know access is granted to information, that information cannot be accidentally or deliberately changed and that the required information is available when required.

It is not claimed in this paper that these six characteristics represent a complete definition for a net-centric organisation, but they are considered to provide a good basis to gauge the usefulness of a secure PAD as the characteristics summarise key operational and functional capabilities. The characteristics can however be considered to have a military focus, which may at first make them appear less relevant to a non-military net-centric organisation, but when each characteristic is applied objectively and in context its relevance can be identified.

## Functionality and Capabilities of Secure PADs

A secure PAD is designed to provide functionality to counter the following three key threats:

1. Malicious software (residing on a host PC) capturing user credentials and data.
2. Sensitive data remnants (resulting from the trusted application storing temporary information) residing on a host PC's HDD following the completion of data processing; and
3. As a result of loss or theft, unauthorised access is gained to (sensitive) data held on an unsecured USB storage device.

These threats represent three of the challenges that remote and mobile workers encounter when using PCs that reside outside the net-centric organisation's security layer. Disallowing remote and mobile computing because of these threats would be counterproductive as it may limit the net-centric organisation's ability to deliver, receive and process information by nodes and sensors at the boundary of the network. The secure PAD provides a platform to enable secure remote and mobile computing in an environment where PCs could be compromised.

Possible configurations and functionality of a secure PAD that counter the aforementioned threats are described in detail in *Secure Portable Execution Environments: A Review of Available Technologies* (James, 2008a). Drawing upon the ideas presented in the paper the type of secure PAD considered in this paper has the following functionality:

**Crypto Engine:** The storage space on the secure PAD will be fully encrypted using on-the-fly hardware encryption to preserve the confidentiality of the trusted application and data residing on the USB device.

**Partitioning:** Partitioning allows the separation and isolation of applications and data to be achieved.

**User Profiles with Differentiated Access Rights:** The secure PAD will allow the definition of access profiles which will define different access rights for different users to the partitions on the secure PAD. Access rights can be Read-Write, Read-Only and No-Access.

**Authentication:** The device will prevent access to its contents and the trusted application cannot be loaded until successful authentication.

**Trusted Application:** The application loaded from the secure PAD into the host PC and used to perform remote processing or secure transactions. The trusted application can take a number of forms including bootable operating system (OS), secure browser, secure bespoke network client or a virtual machine containing a guest OS. The most secure type of trusted application is the bootable OS as the other types of OS are susceptible to malicious software (James, 2008a, James, 2008b). The application is considered to be trusted because it has been acknowledged as secure by the supplier and users. To be considered trusted the OS may have a reduced set of hardened functionality and/or been subjected to independent rigorous evaluation and testing.

This set of functionality binds together to provide a highly secure platform to protect the integrity and confidentiality of both a secure execution environment (the trusted application) and any user data.

A conceptual model of the configuration/architecture of a four partition secure PAD considered in this paper is shown diagrammatically in Figure 1.

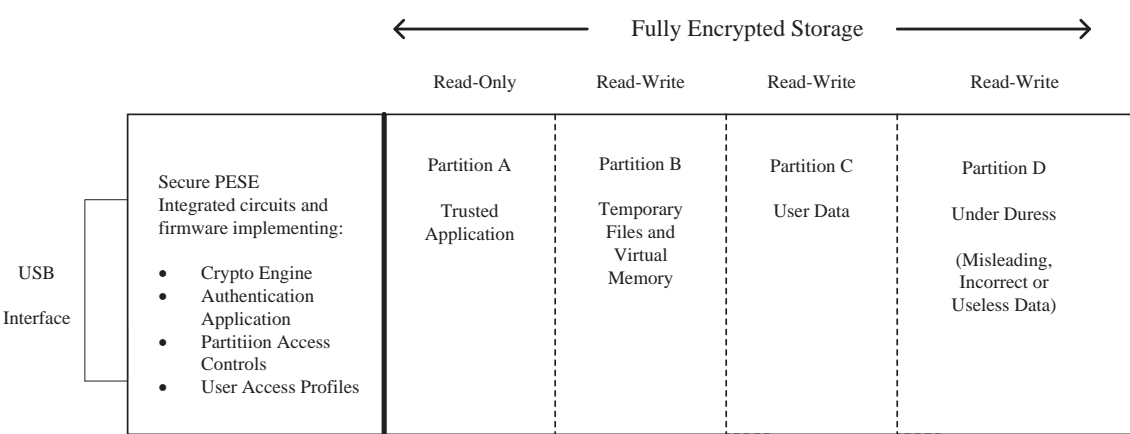


Figure 1: Secure PAD Architecture

The secure PAD presented in Figure 1 has four partitions. The trusted application (a bootable OS) is stored in Partition A; to protect the trusted application’s integrity the partition will be set to Read-Only access for all users. The trusted application will require space to store temporary files and swap space/virtual memory. The usual default approach for an OS loaded from a USB device is to save temporary data and create the swap space on the host PC hard disk drive (HDD). Partition B will have Read-Write access and will be used for temporary files and swap space created by the trusted application. Partition C with Read-Write access will be used to separate and protect any

user data. Finally a fourth partition (Partition D) will contain misleading or incorrect data that can be revealed to an enemy/competitor if the user is under pressure to provide access to the secure PAD.

In this paper the trusted application will be an OS enabling the secure PAD to operate as follows. The secure PAD is plugged into an available USB port on the host PC and if necessary the PC is power-cycled to enable the secure PAD to be the boot device. Upon commencing the boot process an authentication application (AA) is uploaded and executed. Once successfully authenticated the trusted application in the form of an OS is booted. Through the executing OS the user has access to utilities/tools and storage access. The OS is configured to leave no data remnants on the host PC's HDD and completely prevent access to the host PC's HDD, therefore any malicious software resident on the host PC's HDD cannot compromise the trusted application.

In Table 1 an example set of access profiles are defined for the following two hypothetical secure PAD users:

- *Std User* is the standard access profile to be used by a user. The profile has Read-Only access to the trusted application and Read-Write access to the rest of the device.
- *UD User* is a profile a user can use when under duress to reveal the contents of a secure PAD. The profile has Read-Write access to misleading or useless data and No Access to the rest of the device.

User	Partition A	Partition B	Partition C	Partition D
Std User	Read-Only	Read-Write	Read-Write	Read-Write
UD User	No Access	No Access	No-Access	Read-Write

Table 1: Access Profiles

These two access profiles will be used below to show how the secure PAD can support effects based operations. In summary a secure PAD has the following security features to counter the aforementioned threats:

- A trusted application protected by a Read-Only partition that executes within its own trusted environment and that cannot be exploited by malicious software resident on the host PC HDD.
- The trusted application writes all temporary information to a specific partition on the secure PAD, thereby preventing data remnants remaining on the host PC HDD.
- The secure PAD provides strong authentication and full encryption that will prevent the recovery of any data from a lost or found device.

These security features allow secure remote and mobile computing to be performed on a PC that may have been compromised.

## The Use of a Secure PAD in a Net-Centric Organisation

The following scenario is presented to provide the context for the analysis of the suitability of a secure PAD:

*The net-centric organisation has both remote and mobile resources. The remote workers use PCs (e.g. home PCs) outside the organisation's control which are considered to be untrusted. The mobile workers often want to travel light but need to perform secure data processing and securely store data. Mobile workers may use the PCs of strategic partners, home PCs or if absolutely necessary public access PCs (e.g. PCs located in hotels or airport lounges) – all of these PCs are considered to be untrusted.*

*Both remote and mobile workers will use secure PADs to connect to the net-centric organisations servers using the Internet from an available untrusted PC. When the remote and mobile workers are in a 'corporate' office (i.e. a trusted location within the net-centric organisation) they will access the content of the secure PAD from a trusted PC.*

Based upon the above secure PAD configuration and scenario, the contribution a secure PAD can make to NCO is considered in the context of each NCO characteristic.

**Speed of Command:** Dispersed remote and mobile workers are able to securely access net-centric systems using the secure PAD to receive and send data from any available PC, thereby contributing to the information available to make rapid decisions.

**Ability to Mass Effects:** The secure PAD allows remote and mobile workers to be securely 'online' as required (wherever they are located), enabling the mobilisation of remote and mobile resources to address a situation.

**Self-Synchronisation:** The secure PAD enables timely, accurate and relevant information to be delivered securely to distributed workers who can be securely 'online' as required. The ability to securely deliver up to date information to remote and mobile (distributed) resources allows self-synchronisation to be possible.

**Shared Situation Awareness:** The ability to provide a common picture of current corporate operations is an important aspect of NCO. The secure PAD provides the capability for remote and mobile workers to securely receive a common picture of corporate operations and thus maintain a shared situation awareness.

**Effects Based Operations:** The secure PAD includes a partition containing misleading or useless information and the UD User profile to facilitate effects based operations. In a military context if a resource is captured and is under pressure to reveal the contents of a secure PAD to the enemy, the individual can authenticate to the device as the UD User to reveal misleading information to the enemy. Such information may result in the enemy taking action to deliver a favourable outcome for

the net-centric organisation. Similarly in a commercial context a secure PAD could be left at a competitor location (e.g. car park) with the authentication credentials for the UD User profile conveniently left with the device. The intention would be for the competitor find the secure PAD and gain access to information that it may then use to its disadvantage.

**Reachback:** The secure PAD supports the reachback characteristic by providing a trusted application for remote/mobile front office/front line resources to use to send and receive data to/from sophisticated back office capabilities for processing.

**Interoperable Systems:** The secure PAD contributes to the interoperability of NCO. The device's trusted application provides a standard operating environment (SOE). Remote and mobile resources are able to work anywhere and access corporate servers using the SOE (trusted application) thus contributing to the interoperability of the net-centric organisation.

**Information Security:** The primary objective of the secure PAD is to provide a portable extension of corporate security. A combination of strong authentication, encryption, differentiated access rights, user access profiles and the data and network security provided by the trusted application ensure information security is enforced. Much of the focus of NCO security focuses upon enforcing information security at the network and communication levels; however the secure PAD contributes to the provision of security at the content level (Canolin and Kiviharju, 2007).

For the characteristics that summarise an operational capability (i.e. speed of command, ability to mass effects, self-synchronisation, shared situational awareness and reachback) the secure PAD provides essentially the same service that is, the ability to capture, deliver, receive and process information securely (on a PC that may have been compromised) whilst travelling or operating remotely. It can also be claimed that the secure PAD enables the characteristics that relate to functional capabilities (i.e. interoperability and information security) to enhance the existing net-centric organisation's functionality. For a concept like effects based operations the secure PAD can provide a tool to deceive the enemy/competitor.

## Conclusion

The primary contribution a secure PAD can afford a net-centric organisation is through the information security capabilities it delivers. A combination of the device's internal security mechanisms coupled with the way the device can be used to load a trusted execution environment (into any available host PC) enables a secure processing and storage capability to be available to remote and mobile resources. A secure PAD also furnishes user convenience (the device is a light and transportable), useability (the device is simple to use) and if lost or stolen ensures information cannot be revealed.

The secure PAD addresses many of the concerns associated with transporting sensitive data and allowing remote and mobile resources to connect to a 'corporate' network. By severely limiting the opportunity of malicious software to exploit the PC/laptop of a remote or mobile worker the secure

PAD can help prevent data and identity theft and the launch of cyber-attacks from PC/laptops that may otherwise be vulnerable. The secure PAD can allow remote and mobile resources to become an effective part of NCO.

The secure PAD, if used to its full potential can be a powerful component of the NCO infrastructure and allow remote and mobile resources to make a full contribution to the generation of information within a net-centric organisation.

## References

- Alberts D.S, Garstka J.J. and Stein F.P. (1999) Network Centric Warfare – Developing and Leveraging Information Superiority, CCRP Publication Series, Washington.
- Arquilla J. and Ronfeldt D. (Eds) (2001) Networks and Netwars, Santa Monica: Rand, <http://www.rand.org>
- Candolin C and Kiviharju M. (2007) A Roadmap Towards Content Based Information Security, 6th European Conference on Information Warfare and Security, Defence College of Management and Technology, Shirvenham, UK, 2-3 July 2007.
- Fewell M.P. and Hazen M.G. (2003) Network-Centric Warfare – Its Nature and Modelling, DSTO-RR-0262, Defence Science and Technology Organisation, Adelaide.
- Hutchinson W. and Warren M. (2001) Information Warfare: Corporate Attack and Defence in the Digital Age, Butterworth-Heinemann, Oxford.
- James P. (2008) Secure Portable Execution Environments: A Review of Available Technologies, 6th Australian Information Security Conference, Perth.
- James P. (2008) Preventing the Acquisition of Data from Virtual Machine based Secure Portable Execution Environments, 6th Australian Digital Forensics Conference, Perth.

### 4.3.2.4 Synopsis

**Outcomes and Contribution to Knowledge:** Using a theory for NCO it was shown that a secure PESE could support a number of the theory characteristics to provide a secure capability within remote network nodes. The research led to a greater understanding of how a secure PESE could be used in deployed and mobile environments and how the capabilities of the device could be used to support a function of network centric operations like effects based operations; previously an unconsidered function of a secure PESE. The paper broadens the sphere of research by providing an academic analysis for a particular stakeholder issue. The research contributed to the definition of a business case for funding (CTD, 2010) to develop a high assurance secure PESE.

Although the paper states a bootable operating system is the preferred basis for a trusted application (i.e. a secure PEE) the paper also proposes the use of an up-loadable (to a host



PC) set of applications forming the trusted application (secure PEE). The implication in the paper is that the set of applications would be hardened and therefore little or no data remnants would exist on the host PC's disk drive after the hardened application's execution.

**Contemporary relevance, linkage with other papers and future direction:** Network centric operations allows innovative organisations to utilise ICT within a decentralised and distributed structure, enabling decisions to be made that achieve competitive advantage through the delivery of accurate, timely and accessible information. The contemporary relevance of this paper is that it considers how a secure PESE could be used to add further innovation to an organisation. The paper is directly linked to paper 9 as it positions the future direction of the doctoral research towards the construction of a secure PESE artifact.

### **4.3.3 The SDV-HA – A Platform for a High Grade Secure PESE**

#### **4.3.3.1 Preamble**

The SDV-HA is a high assurance secure storage device that utilises the SDV technology and, when configured as a secure PESE, satisfies both the concept and functional requirements. Paper 9 describes the design and functionality of the SDV-HA and how it can make a contribution to supporting successful outcomes in NCW. Paper 9 progresses the ideas presented in Paper 8 through the actual implementation of a secure PESE (or secure PAD as it is termed in the paper). The SDV-HA addresses the doctoral research question: *How can anti-tamper mechanisms be implemented into a small form factor and highly portable device?*

Paper 9 was presented at the Military Communication and Information Systems Conference<sup>63</sup> (MilCIS) in 2011. The paper discusses how the SDV-HA could be used within the Australian Defence Force as development was supported by funding from the Defence CTD program (CTD, 2010), and the presentation was given at a military conference.

As per Paper 8 a secure PESE is called a secure PAD as the device was configured to perform a set of specific functions using one or more (probably specialist) applications.

---

<sup>63</sup> <http://www.milcis.com.au/milcis>



However, unlike Paper 8 the term trusted application is not used. The paper describes four modes of operation: System mode which is the equivalent of the Mini-SDV Fully-Trusted mode, Guarded mode which is the equivalent of the Mini-SDV Assured mode, Under-duress mode which allows the device to be used to present non-valuable data, and Storage-only mode where the device is configured primarily as a secure data storage device.

The MilCIS conference requires that a paper is formatted into two columns per page with references cited by number. The two column format has been re-formatted to reflect the style used by all the other papers presented in this thesis; however, the numbered referencing style has been retained.

#### **4.3.3.2 Prior Research and Knowledge**

The SDV-HA was designed to meet all the secure PESE functional requirements including the challenging anti-tamper requirement. A high assurance product requires an anti-tamper design (Skorobogatov, 2012) that provides complete protection of all of the device's internal circuitry and logic (DSD, 2010). An anti-tamper mesh (Paul et al., 2008; Pham et al, 2013; Bindrup et al., 2014) can provide such protection. Identified prior research and knowledge into anti-tamper design and mechanisms was in the form of patents (Pham et al, 2013; Bindrup et al., 2014) and research performed at the University of Cambridge (Drimer et al., 2008; Paul et al., 2008; Paul, 2013; Skorobogatov, 2013). There may exist innovative anti-tamper designs used in high assurance products that have not been published because once a physical design is known it may be possible to subvert it. Existing active tamper mesh designs were neither considered suitable for a small form factor device nor sufficiently secure to meet the high assurance requirements and therefore a new innovative design was required.

Other challenging design considerations included providing the multiple modes of operation, a requirement emerging from stakeholder discussions, a key management system that satisfied the high assurance requirements and integrating all the functionality into a small form factor package. The following knowledge was consumed in the design:

- Descriptive knowledge from the investigation into NCO (described Paper 8) and from observations and suggestions from defence stakeholders.
- Prescriptive knowledge in the form of the secure PESE concept, the functional requirements and the three conceptual models (Figures 4.1, 4.2 and 4.3).
- Prescriptive knowledge in the form the SDV technology, the selected microcontroller capabilities, and the designs of the Mini-SDV, Fireguard and the MEE.
- Formal justificatory knowledge from the ASD high assurance development standard (DSD, 2010), existing security engineering (Anderson, 2008), cryptographic engineering (Schneier, 2007; Ferguson et al., 2011) and anti-tamper design (Paul et al, 2008; Skorobogatov, 2012).
- Informal justificatory knowledge in the form of advice from the certification authority during the design process.

#### 4.3.3.3 Paper 9

**Paper 9** - James, P. (2011) **A Secure Portable Application Device to Support Network Centric Warfare**, Military Communication and Information System Conference (MilCIS) 2011, Canberra, available from: <http://www.milcis.com.au/milcis2011pdf/2.8a-paper2.pdf>

#### Abstract

*Information security is a critical aspect of a network centric architecture. The Australian Defence Force's transition to a network centric organisation has included strong consideration to system and network security, in particular cross domain solutions. One facet of a network centric organisation where it is difficult to enforce information security is for remote personnel. Information security at the deployed or remote network node often relies upon a mixture of hastily established physical, procedural and logical security mechanisms that may not always provide the same level of assurance as the information security afforded in established and permanent Defence environments.*

*A secure Portable Application Device (secure PAD) provides both portable storage and an execution environment that can allow deployed or remote personnel to upload, from the secure PAD, a trusted standard operating environment to perform secure computing within a network centric organisation. The secure PAD typically includes mechanisms to mitigate data loss, forensic discovery, under-duress threats and cyber, brute-force and physical attacks, whilst enforcing system integrity.*

*This paper considers how the Silicon Data Vault (SDV), a secure PAD, provides a suitable secure portable execution and storage environment to support network centric warfare (NCW). The SDV has been designed to satisfy the Australian Defence Signal Directorate's (DSD) high assurance requirements to protect highly classified data, yet allow the device to be handled as a lower classified device by remote personnel when the device is powered off. The motivation for the development of the SDV and its concept of operation are presented. The device's DSD compliant security functionality is enumerated and how the security functions combat logical and physical security attacks is discussed. The paper concludes by considering how some of the novel features of the SDV support NCW theory.*

## **Keywords**

Secure portable execution and storage environments; network centric warfare; persistent cyber-attack, data loss; forensic discovery; anti-tamper.

## **Introduction**

Defence's intention to implement Network Centric Warfare (NCW) capabilities has been well communicated and documented [1]. As Defence progresses towards its target of achieving a networked Australian Defence Force (ADF) the information technology used to both connect and implement network nodes (e.g. sensors, command and control systems, weapon systems and business systems) must conform to the Defence information security policy requirements [2], [3]. The Single Information Environment (SIE) [4] is a critical piece of NCW infrastructure and will provide an information service architecture to enable Defence's NCW goal of information superiority to be achieved. The SIE will provide a rich services oriented architecture to ensure the right information gets to the right person at the right time and in the right form and hence achieve information superiority.

The SIE will implement one (security policy compliant) network connecting all security levels and therefore the security architecture will need to support a cross domain security solution [5]. Whilst Defence can be assured of satisfying Defence's information security within and across the network for fixed locations and 'on-board systems', it can prove more challenging to achieve the same level of information security assurance (and comply with Defence security policy) for deployed and remote personnel. This group of personnel may reside in environments that lack the physical security of fixed Defence locations (e.g. personnel may be located in hastily established deployed positions), yet this group of transient personnel may be acting as vital sensors gathering intelligence and/or requiring access to classified information to make informed decisions. The protection of the information gathered and accessed is as critical as the security afforded to all other information within the network. The secure Portable Application Device (secure PAD) is a technology that can be used to enforce information security at the remote and deployed network node and enhance the capability of the remote or deployed personnel.

A secure PAD is a portable execution and storage device that combines secure storage with a standard operating environment (SOE) and applications that are located in a protected area on the

device. A secure PAD typically provides a cohesive set of functions to combat persistent cyber-attack<sup>64</sup>, data loss and forensic discovery, and to preserve system integrity. A secure PAD's functionality can also contribute to achieving an organisation's NCW objectives.

The Silicon Data Vault (SDV) is a secure PAD designed to satisfy the Defence Signals Directorate's (DSD) high assurance product standards [6]; Secure Systems worked closely with members of DSD's high assurance team throughout the whole product development. Secure Systems Limited SDV technology was utilised and extended to produce a secure PAD. The SDV is designed to store highly classified data yet allow the device to be handled as a lower classified or unclassified device when it is neither powered nor authenticated. The SDV also provides an execution capability; the device contains a SOE, held in a secure location, which can be uploaded onto a host Personal Computer (PC) to allow secure remote computing<sup>65</sup>. Novel features within the SDV can also be configured to support NCW objectives.

In this paper a concept of operation is presented to enable the reader to visualise the environment(s) in which the SDV could be utilised. The DSD requirements for a high assurance device like the SDV are summarised and the device's functionality is described. An analysis is given of the SDV's ability to prevent data loss, forensic discovery, under-duress threats and cyber, physical and brute force attacks. Fewell and Hazen [7] define eight characteristics of NCW which can be used as a theory to quantify a technology's network centrality (net-centrality). James [8] defined a model for a secure PAD and then qualified the model's net-centrality using the eight characteristics of NCW. Using a subset of the eight characteristics of NCW this paper concludes by considering how certain novel features of the SDV support NCW theory.

## Concept of Operation

In a net-centric ADF, remote personnel will generate and require access to classified information [1]. These personnel may be deployed, working at coalition/ally locations or working away from a fixed secure ADF location. At any of these locations ADF personnel may need to connect to the SIE to access or input classified information. These personnel may also need to travel between locations and transport classified information. The SDV has been designed to provide information security when employed in a number of different operational use scenarios, including scenarios that could provide opportunities to introduce new work practices that are currently not permitted. These operational use scenarios and their respective security issues include:

1. *Deployment:* The information security afforded in deployed environments may not always meet the assurance level commensurate with the classification of data being handled; particularly at the early stage of establishing the deployed environment.

---

<sup>64</sup> A secure PAD is not designed to mitigate all forms of cyber-attack. A secure PAD can mitigate cyber-attacks launched through embedded malicious software, known in this paper as persistent cyber-attacks.

<sup>65</sup> Remote computing is defined as any data processing performed away from the secure corporate environment and in a location that may not have the physical and logical security present within a secure corporate site.

2. *Travelling:* ADF personnel may need to carry classified information. These personnel may also require network access through a PC that resides outside of the ADF information security boundary, and therefore the PC is assumed to have limited or no level of trust, e.g. use of a coalition/ally PC or a defence contractor PC. Such personnel may also become pressured to allow access to portable storage medium. If under-duress it may be opportune to seamlessly provide access to misleading information, whilst ensuring classified information is not revealed.
3. *On-board Systems:* Net-centric nodes like vehicle-mounted task units [1] will have on-board systems that may send and receive classified information. Should the vehicle need to be abandoned, the commander requires assurance that any classified data cannot be retrieved and exploited by the enemy and the speed and ease of secure erasure is important.
4. *Telework:* Whilst remote access to the Defence Restricted Network (DRN) is permitted, information security issues prohibit work practices like teleworking as it is difficult to give any level of assurance to a home based PC.

Collectively these scenarios present existing and new approaches to requiring access to, and protection of classified data whilst conducting business. The SDV is a device that will provide a safe and secure environment that can be used to achieve information security in the above four operational scenarios through the following capabilities:

1. The device provides a protected SOE and applications to support secure network access. The protected SOE can be uploaded onto a host PC where it only utilises the PC's processor and memory, ensuring the PC's internal hard disk drive (HDD) is not accessed and thereby preventing any sensitive data being saved to the PC's HDD through software actions.
2. The security mechanisms of the SDV reduce the opportunity of any malicious software (malware) resident on the host PC initiating a persistent cyber-attack.
3. The device's tamper response, access controls and cryptographic capabilities separate and protect stored classified data.

The SDV implements the above capabilities as a device designed to achieve the DSD high assurance criteria required to protect highly classified data. The device is designed to meet the functional and assurance requirements to protect highly classified data yet allow the device to be handled as a lower classified or unclassified device when it is neither powered nor authenticated. Policy restrictions will apply to prohibit the connection of the device to a completely untrusted PC, e.g. a public access internet PC. However, it will be shown in this paper how the capabilities of the SDV (when employed in the aforementioned operational use scenarios) can combat the threats of data loss, forensic discovery, under-duress and cyber, brute-force and physical attacks. It will also be shown how the device supports a theory of NCW.

## Optima SDV High Assurance Functionality and Features

### DSD High Assurance Requirements

DSD has published a set of criteria [6] to be satisfied by a product aiming to achieve high assurance certification. The criteria specify both assurance and certain functionality requirements. The assurance criteria address design, development and integrity checks for hardware, software and firmware. The functionality criteria address cryptographic, anti-tamper and authenticity functionality. The SDV has been designed to satisfy the criteria.

### Form Factor

The SDV is a 135mm (l) x 81mm (w) x 20mm (h) size device, similar to the form factor of a 2.5" portable HDD; Fig. 1 presents an image of the SDV. The device connects to a host PC through either a powered external serial advanced technology attachment (eSATAp) interface or a universal serial bus (USB) interface; the device also has a host USB interface for connection to an authentication token. The SDV has a large capacity integral storage capability.

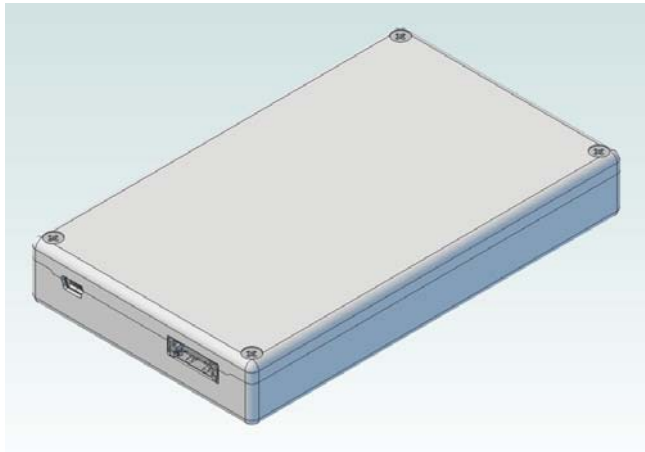


Figure 1 Conceptual Model of SDV

### Hardware Architecture

The SDV utilises a distributed integrated circuit (IC) architecture. A secure microcontroller with cryptographic and tamper response capabilities provides the control. A field programmable gate array (FPGA) implements a bulk data encryption engine, and eSATA and USB interface ICs enable the device to connect to a PC. All security critical communication between the ICs is encrypted. All FPGA programmable logic, firmware and software are stored in encrypted form, being decrypted as required during operation. The architecture also includes redundancy for security critical functionality to ensure data is not compromised, i.e. multiple failures would have to occur before data was compromised.

### Tamper Response

The SDV has multiple layers of tamper detection [6]. The device is able to detect the physical removal and electrical disconnection of the opening of the enclosure, internal access and tampering

with components. The device also monitors environmental and operational conditions with a tamper event triggered if a condition falls below or exceeds a defined value. A tamper detection event will cause the encryption keys to be destroyed, rendering the device inoperable.

A typical tamper detection design involves the use of a one-time mesh structure which once broken would need replacing for the device to be reused. A one-time mesh is both difficult (from an engineering perspective) and expensive to implement and replace. An innovative tamper detection and response design (which DSD requested become a classified design) has been incorporated into the SDV. The SDV tamper detection and response design adopts an approach that has a low implementation cost and allows for product lifecycle support and maintenance without replacement.

### **Encryption and Key Management**

The National Security Authority (NSA) Suite B [9] set of cryptographic primitives is implemented. The FPGA bulk encryption engine utilises the Advanced Encryption Standard (AES) 256-bit primitive [10] to encrypt all stored data, whilst code signing is implemented in Elliptic Curve Cryptography (ECC) 384-bit [11] and authentication utilises the Secure Hash Algorithm (SHA) 256-bit [12]. The key management design (which DSD requested become a classified design) is based upon the concept of a shared secret [13]. The shared secret technique divides an encryption key into multiple pieces that are reconstructable to produce the original key. However if a piece is lost or destroyed the key cannot be reconstructed; a key can only be reproduced from all valid pieces. For example, a secure erase function allows the user to enter a command that deletes the respective key piece and renders the device inoperable.

### **Modes of Operation**

The SDV has four modes of operation; system, guarded, under-duress and storage-only. System mode allows a user to boot a SOE from the device rather than the host PC. In system mode, upon successful authentication, a hardened SOE is loaded into the host PC memory and only the PC's processor and memory are utilised; no data, including temporary SOE data, is written to the host PC's HDD. System mode provides a high level of protection and security. The SOE is located in a secure (Read-Only) partition on the SDV, protecting the SOE from malware attacks and also providing system integrity. System mode allows a user to utilise an available PC and upload a (trusted) SOE to perform secure remote computing. In system mode the SOE can be a Defence accredited SOE or a hardened operating system (OS) with a limited set of applications like the Mobile Execution Environment (MEE) [20].

Guarded mode allows a user to connect the SDV to a PC executing the Windows OS. Upon successful authentication, a guarded application<sup>66</sup> is uploaded from the SDV and executed. Fireguard [14] is an example of a guarded application that provides a secure browser capability, an overview of the Fireguard application is given below.

---

<sup>66</sup> A guarded application is defined as a software application whose integrity is protected, as its functionality is minimised to prevent exploitation by malware and it is configured to leave no forensic footprint on the host PC's HDD.



Under-duress mode enables a user to grant access to an unauthorised user in such a way that no classified or valuable data is revealed, and access to the SOE and guarded application is not possible. However, access to information that is of no value or is misleading is provided.

In system, guarded and under-duress modes access to the data partition(s) is possible based upon the access rights defined for the mode of operation.

Storage-only mode enables the device to be configured as a data storage device. In this mode the device would be connected to a (trusted) PC and, upon successful authentication, access to storage would be through the PC's Windows OS. The SOE and guarded application(s) are not installed in storage-only mode.

### **Authentication**

A two-factor authentication scheme [15] is used by the SDV. The first factor is a complex password and the second factor is a USB token. Both authentication credentials form an important aspect of the key management design.

In system mode authentication occurs at the earliest possible opportunity through the upload of an authentication application upon the powering of the PC. As the authentication application is the only software application executing, a high level of assurance is provided that the authentication credentials will not be compromised or captured. Upon successful authentication the SOE on the SDV is loaded.

In guarded mode an authentication application is uploaded from the SDV to execute under the Windows OS. As password authentication is performed under the control of the host PC OS, guarded mode cannot provide the same level of assurance for the protection of the user password as is possible in system mode. However, the use of a USB token that authenticates directly with the SDV limits any opportunity from the capture of a user password.

### **Access Control and Software and Data Integrity**

The SDV integral storage can be configured to contain a number of partitions with either Read-Write, Read-Only or No-Access permissions. Partition access controls allow for the separation and protection of software and data. For instance, classified data can be separated and protected in a partition that is defined as No-Access when the SDV is used in a less secure environment, and as Read-Write when the environment is secure.

The partition Read-Only and No-Access permissions provide an innovative capability that can be used to protect and hide software [16]. It is the Read-Only partition that is used to protect the system mode SOE and guarded mode application(s); the partition provides software and data integrity as the partition control prevents the software from becoming corrupted. The No-Access permission provides protection for software and data when the device is used in different operating



modes, e.g. when operating in guarded mode, the system mode SOE can be protected by providing No-Access which actually hides the SOE from the user.

Fig. 2 presents a conceptual model of a possible configuration of the SDV integral storage. The example used consists of the integral storage configured with six partitions with the following access permissions:

- **SOE:** This partition will have Read-Only access in system mode and No-Access in all other operating modes.
- **Guarded Application:** This partition will have Read-Only access in guarded mode and No-Access in all other operating modes.
- **Temporary space:** A partition used by the SOE and guarded application to write temporary data. This partition will have Read-Write access in system and guarded modes and No-Access in all other modes.
- **Data:** This partition will have Read-Write access in all modes except under-duress mode where No-Access will be possible.
- **Classified Data:** A partition that has Read-Write access in system and storage-only modes but No-Access in all other modes.
- **Under-duress:** A partition holding non-valuable or misleading data that has Read-Write access in under-duress mode but No-Access in all other modes.

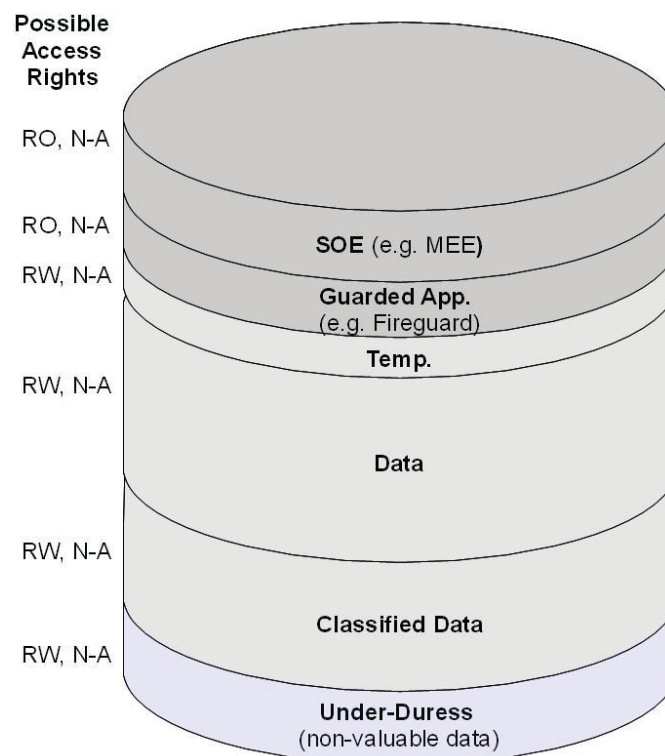


Figure 2 Possible SDV Configuration

Fig. 3 presents the example SDV in system mode. As system mode is the most secure mode, access to classified data is allowed. There is no need to have access to the guarded application and under-duress data in system mode.

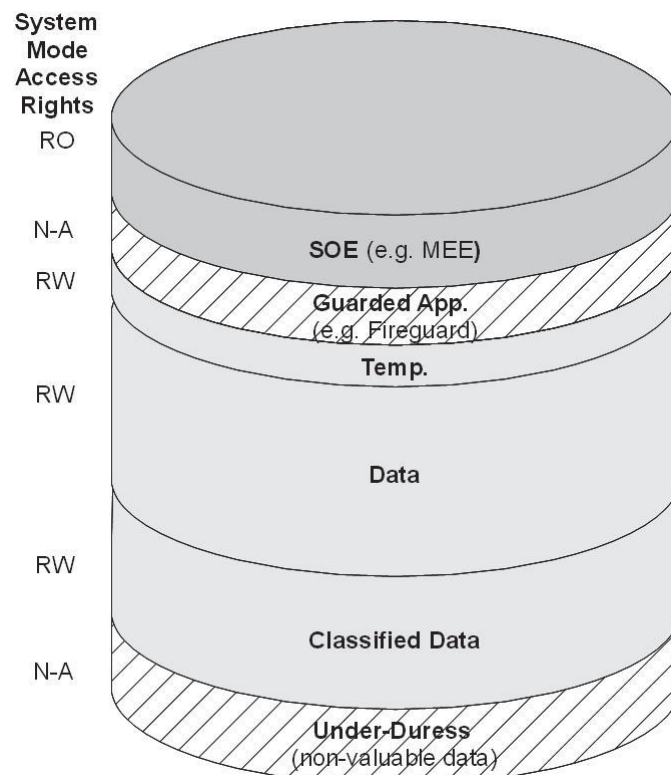


Figure 3 System Mode Access

Fig. 4 presents the example SDV in guarded mode. As guarded mode allows access to the device via a PC running the Windows OS, access to classified data is prevented in this scenario as the PC may not have the required level of classification. There is no need to have access to the SOE and under-duress data in guarded mode.

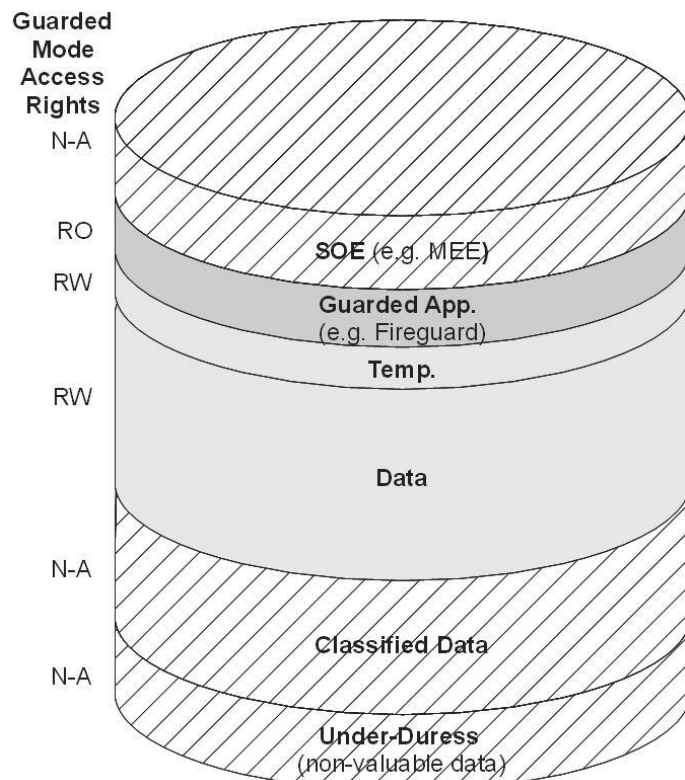
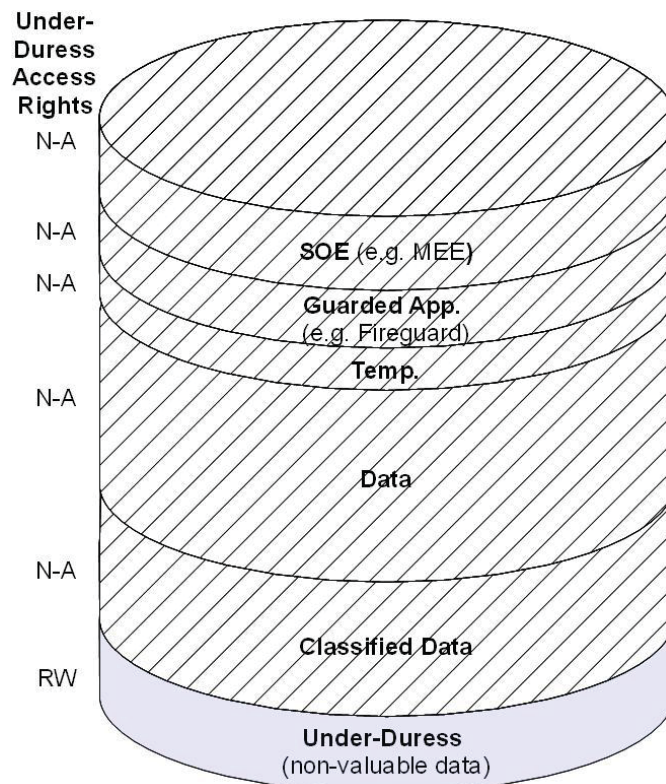


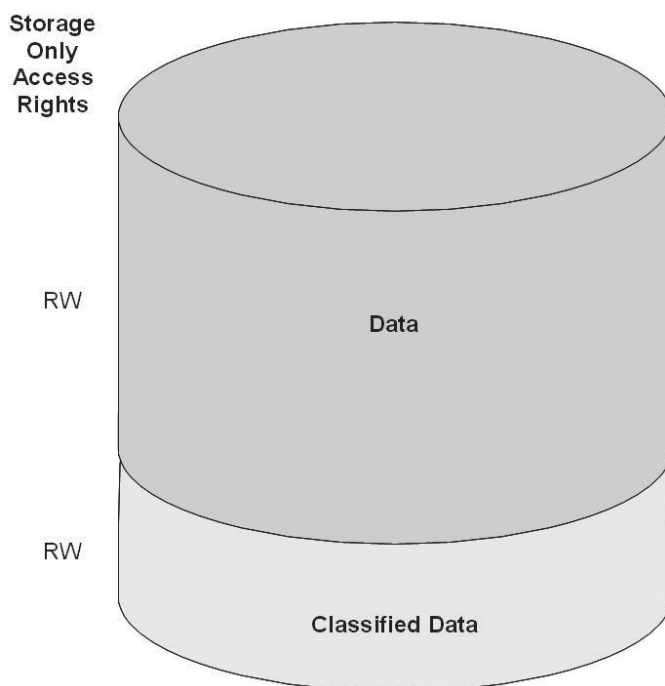
Figure 4 Guarded Mode Access

Fig. 5 models the SDV in under-duress mode. Access to all software and data is prevented except for the non-valuable or misleading data.



*Figure 5 Under-duress Mode Access*

Finally Fig. 6 models the storage-only mode configuration for the device. Both data partitions are available. In storage-only mode the SDV is only configured for storage.



*Figure 6 Storage-only Mode Access*

The SDV includes a feature that allows a user to dynamically change the access rights to a partition [17], e.g. on-the-fly change a partition from Read-Write to No-Access. The ability to change the access permission to a partition can allow a user to change access to data based upon the work being performed, e.g. a remotely based worker processing classified data off-line may require network access but as a precaution will change the access permission to the classified data partition to No-Access before connecting to the network.

### **Code Signing**

Code signing is a technique used to digitally sign software (code) and validate its integrity and authenticity [18]. The SDV implements a certificate chaining protocol [19]. The device has a master root (public) certificate permanently installed that is used to verify the signature of each certified code component of programmable logic, firmware and software. This approach to code signing allows the accrediting authority to appoint a different certificate signer for each code component, adding a further level of security and integrity.

### **System Mode SOE – MEE**

The SOE can be either a Defence accredited SOE (e.g. a hardened version of Windows OS) or the Secure Systems MEE, a hardened and reduced functionality version of Ubuntu. The MEE [20] has been designed to provide a minimum number of applications (e.g. a browser and remote terminal client) and to rapidly load from and save any created temporary data to the SDV. Upon exiting the

MEE no evidence of its execution will exist on the host PC HDD. The MEE is designed for portability and to allow it to load and execute from any PC. Some of the important security features of the MEE are:

- No data (temporary or otherwise) is written to the host PC's HDD, i.e. all swap/page files, temporary files, etc. are written to the SDV.
- The user can't change or install software applications.
- The user has no privileged access to the MEE.

### **Guarded Mode – Fireguard**

Fireguard [14] is a secure browser built to execute on the Windows OS, it is an example of a guarded application designed to reduce data leakage, malware attacks and forensic discovery of its use. Fireguard is based on the Mozilla Firefox browser. A standard browser can leave data remnants in the form of cookies, histories, saved passwords, cached web pages and downloaded files on the host PC's HDD. A standard browser is also vulnerable to malware attacks through plug-ins and extensions [21], Javascript [22], Java Applets [23] and media files [24]. Fireguard ensures no data remnants remain on the host PC's HDD by writing all browser generated data to a partition on the SDV. Fireguard limits the opportunity for malware attack by preventing the installation of plug-ins and extensions, and disabling both Java and the auto-playing of media files.

## **Combating Information Security Attacks and Protecting Data**

The functionality of the SDV has been designed to prevent the threat of cyber-attack, data loss, forensic discovery, brute-force attack, physical attack and the loss of data when under-duress. For each of these threats an overview is given of the threat and how it may be countered by the SDV.

### **Cyber-attack**

Cyber-attacks can take many forms, one of the most common being attacks initiated through embedded malware. The use of embedded malware allows the attack to conduct persistent cyber-attacks. Defence adopts best practice IT security, which as a minimum includes secure networks incorporating firewalls, anti-virus and anti-spyware software and has locked SOEs on desktop PCs, which prevents the installation of software (intentionally or unintentionally in the case of malware) by a PC user. These IT security practices limit the opportunities for malware to become embedded in a PC OS. However, the use of portable computing and storage devices in any of the four operational scenarios could compromise security as these devices will connect to PCs that may not have the required level of assurance. As a result, these portable computing and storage devices could become contaminated with malware and subsequently contribute to spreading the malware, allowing a persistent cyber-attack to be launched.

The SDV limits the opportunity for malware based cyber-attacks by:

- Using access controls that enable the storing of the system mode SOE and guarded application in Read-Only partitions that cannot be written to.
- Implementing code signing to prevent hostile code becoming embedded in the device itself; and
- For the MEE and Fireguard giving the user no privileged access and preventing the installation of software, legitimate or otherwise.

### **Data Loss**

Data loss/leakage can occur in two ways; (1) when an unsecured portable computing and storage device is lost and (2) when sensitive data remnants from a portable computing and storage device become resident on a PC's HDD due to the storage techniques utilised by the PC's OS (e.g. swap/page files and temporary files).

The SDV prevents data loss when a device is lost through the use of authentication, access controls and encryption. The device prevents data remnants residing on the host PC's HDD by writing and storing all temporary created data in a specific partition on the SDV.

### **Forensic Discovery**

Forensic analysis techniques can be used to recover data remnants from PC HDDs that have become resident (on the PC HDD) following the use of a portable computing and storage device to access and process sensitive data (that is stored on the device). The SDV prevents forensic discovery by ensuring the installed SOE and guarded applications write all data (temporary or otherwise) to the appropriate partitions on the device's integral storage device.

### **Brute-Force Attack**

A brute-force attack on the SDV would involve trying to determine the authentication credentials. The device uses strong passwords based on a minimum length and a combination of printable ASCII characters, limited unsuccessful authentication attempts and a second factor cryptographic token to mitigate brute-force attacks.

### **Physical Attack**

Physical attacks include dismantling the SDV to gain access to the integral storage and attempting to embed hostile hardware or code (i.e. programmable logic, firmware or software) into the device to subsequently gain access to classified information. Physical attacks are prevented through multiple levels of tamper response and code signing to allow only certified code to be loaded into the device.

### **Under-duress**

There are two forms of under-duress that the SDV counters. The first is when the SDV is used as part of a vehicle system. If the vehicle is to be abandoned then the commander can enter a simple command that performs a secure erasure, rendering the device inoperable. The second countermeasure allows a user who is under-duress to reveal non-valuable or misleading data to the

hostile individual applying pressure in a way that would convince the hostile individual that the revealed data is genuine. Any classified data on the device would not be revealed.

## Summary

Table I summaries the SDV functionality that provides countermeasures to the identified threats.

*Table I - Functionality to Counter Threats*

SDV Countermeasure	Threats					
	<i>Cyber-attack</i>	<i>Data Loss</i>	<i>Forensic Discovery</i>	<i>Brute-force Attack</i>	<i>Physical Attack</i>	<i>Under-duress</i>
Architecture		X			X	
Tamper Response		X			X	
Encryption & Key Management		X		X	X	X
Authentication		X		X		
Access Controls	X	X				X
Code Signing	X	X			X	
System Mode - MEE	X	X	X			
Guarded Mode - Fireguard	X	X	X			
Under-duress Mode						X

## Novel Support for NCW

### Satisfying a Theory of NCW

Fewell and Hazen [7] identified speed of command, ability to mass effects, self-synchronisation, shared situational awareness, effects based operations, reachback, interoperable systems and information security as a set of NCW characteristics that can be used as metrics to measure a level of net-centricity. This set of characteristics can be considered a theory for NCW that can be used to qualify technology models. James [8] defined a secure PAD model and used the set of NCW characteristics to qualify the model's net-centricity, the NCW characteristics were further categorised into three subsets; operational, functional and manipulative. It was found that for the operational characteristics (i.e. speed of command, ability to mass effects, self-synchronisation, shared situational awareness and reachback) a secure PAD provides a similar capability to other secure portable computing and storage devices, i.e. the ability to capture, deliver, receive and process information whilst travelling or working remotely. However, the secure PAD does provide net-centric capabilities with respect to the functional (i.e. interoperability and information security) and manipulative (i.e. effects based operations) categories that can be used to enhance the NCW capabilities of an organisation. As an implementation of a secure PAD that conforms to the model [8], the SDV implements features that satisfy the functional and manipulative categories of the NCW theory.



## Functional Characteristics

One of the fundamental inhibitors to achieving interoperability is compatibility issues between versions of PC operating systems. Through the system mode SOE and the guarded applications the SDV is able to provide a secure and safe interoperable computing platform for remote workers. The device will provide an execution environment that cannot be changed nor corrupted by the user or malware, and will therefore contribute to the interoperability of the net-centric organisation.

The primary objective of the SDV is to provide information security in remote environments. As shown in Table I, the device provides a rich set of security functionality to combat a range of security threats. The device through its security capabilities can enable remote NCW use scenarios to be achieved within a net-centric organisation that would otherwise not be possible. In Table II the security functionality relevant to enabling the SDV to be used in any of the four remote NCW use scenarios is presented.

Table II - Functionality to Support Use Scenarios

SDV Features	Remote NCW Use Scenario			
	Deployment	Travelling	On-board system	Telework
Architecture	X	X	X	X
Tamper Response	X	X	X	X
Encryption & Key Management	X	X	X	X
Authentication	X	X	X	X
Access Controls	X	X	X	X
Code Signing	X	X		X
System Mode - MEE	X	X		X
Guarded Mode - Fireguard	X	X		X
Under-duress Mode		X		

## Manipulative Characteristics

Effects based operations is concerned with manipulating the enemy to achieve a desired effect. Net-centric systems allow an organisation to dominate the infosphere and therefore manipulate information to mislead the enemy. The SDV under-duress mode provides an opportunity to implement an effects based outcome. Two scenarios to achieving effects based operations through the under-duress mode are potentially possible. The first scenario involves the device becoming available in such a way that it is obtained by the enemy with under-duress mode authentication either loaded or available, possibly leading the enemy to act upon misleading information. The



second scenario, as previously outlined, involves the authorised user entering the under-duress authentication credentials when under pressure to do so, similarly revealing misleading information to seduce the enemy into taking the desired decision.

## **Future work**

One possible area for future investigation could include identifying and quantifying any threats posed by the unified extensible firmware interface (UEFI). The UEFI standard is now becoming the standard firmware in PCs. UEFI is comprehensive and configurable firmware that is replacing the basic input/output system (BIOS) in PCs. UEFI is claimed to be a more secure standard, however the highly configurable nature of the firmware may make it susceptible to embedded malware and thus possibly provide an avenue for attack against the SDV.

## **Conclusion**

The SDV has been shown to be an implementation of a secure PAD. The device provides a tool to implement information security and enable secure NCW to be achieved at remote locations. It has been proposed in this paper that the device can be used in a number of operational use scenarios that could extend the capabilities of a networked ADF and provide opportunities to implement work practices that could enable Defence to become more efficient and effective. It has been shown that some of the novel information security and functional capabilities of the SDV can implement the functional and manipulative characteristics of a theory for NCW. If used to its full potential, the SDV could be used to enable remote personnel to make a full contribution to the generation and processing of information within a networked ADF.

## **References**

- [1] Capability Development Group, "NCW roadmap 2009", Defence Publishing Service, Department of Defence, Australian Government, DPS:FEB005/09, 2009.
- [2] Defence Security Authority, "Electronic defence security manual", Department of Defence, Australian Government, AL5, April 2011.
- [3] Defence Signals Directorate, "Australian Government information and communications technology security manual", Department of Defence, Australian Government, November 2010.
- [4] Chief Information Officer Group, "Single Information Environment (SIE) - Architectural Intent 2010", Defence Publishing Service, Department of Defence, Australian Government, DPS:DEC013-09, 2010.
- [5] T. Howell, "Infrastructure analysis of industry approaches for the development of a high level design for the X-DeBI", Rapid Prototyping Development and Evaluation, Department of Defence, Australian Government, Quicklook 068, QL068-10-1, version 1.0, May 2011.
- [6] Defence Signal Directorate, "Standards for developing high assurance products" Department of Defence, Australian Government, 2008/2366, release 1.0 (version 19), January 2010.
- [7] M.P. Fewell and M.G. Hazen, "Network-centric warfare – its nature and modelling", Maritime Operations Division, Defence Science and Technology Organisation, DSTO-RR-0262, September 2003.

- [8] P. James, "Use of a Secure Portable Application Device as a component of Network Centric Operations", Secure Systems Limited (and Edith Cowan University), Journal of Information Warfare, issue 3, vol. 8, pp. 39-46, December 2009.
- [9] Central Security Service, "NSA Suite B cryptography" National Security Agency, Department of Defence, United States Government, url: [http://www.nsa.gov/ia/programs/suiteb\\_cryptography/](http://www.nsa.gov/ia/programs/suiteb_cryptography/), accessed June 2011.
- [10] Federal Information Processing Standards, "Announcing the Advanced Encryption Standard (AES)", National Institute of Standards and Technology, United States Government, FIPS Pub 197, November 2001.
- [11] Federal Information Processing Standards, "Digital signature standard", National Institute of Standards and Technology, United States Government, FIPS Pub 186-3, June 2009.
- [12] Federal Information Processing Standards, "Secure Hash Standard (DSHS)", National Institute of Standards and Technology, United States Government, FIPS Pub 180-3, October 2008.
- [13] A. Shamir, "How to share a secret", Communications of the ACM, vol. 22, no. 11, pp. 612-613, November 1979.
- [14] D. Griffith, P. James, "Fireguard – A secure browser with reduced forensic footprint", Journal of Network Forensics, vol. 2, issue 2, pp. 1-24, Summer 2010.
- [15] R. Elrod, "Two-factor authentication", East Carolina University, DTEC 6870 project, pp. 3-7, July 2005.
- [16] R. Kabzinski, M. Hearn, "Security system and method for computer operating systems", World Intellectual Property Organization, WO 2004/086228 A1, 7 October 2004.
- [17] M. Wynne, M. Geddes, "Partition access control system and method for controlling partition access", World Intellectual Property Organization, WO 2005/086005 A1, 15 September 2005.
- [18] M. Naedele, T. E. Koch, "Trust and tamper-proof software delivery", pp. 51-57, SESS'06, Proceedings of the 2006 international workshop on software engineering for secure systems, ACM, May 2006, Shanghai.
- [19] J. R. Michener, T. Acar, "Managing system and active content integrity", Computer, vol. 33, issue 7, pp. 108, July 2000, IEEE Computer Society.
- [20] D. Griffiths, P. James, "MEE – A hardened Mobile Execution Environment to protect against cyber-attack, data loss and forensic discovery", unpublished.
- [21] A. Barth, A. P. Felt, P. Saxena, A. Boodman, "Protecting browsers from extension vulnerabilities", Proceedings of the 17th Network and Distributed Systems Security Symposium, San Diego, March 2010.
- [22] M Zalewski, "Browser security handbook, part 2", Google Inc, 2009, url: <http://code.google.com/p/browsersec/wiki/Part2>.
- [23] D. Reynaud-Plantey, "New threats of Java viruses", Journal in Computer Virology, Springer-Verlag, vol. 1, no. 1-2, pp. 32-43, September 2005.
- [24] Symantec, "Symantec internet security threat report trends for 2009", Symantec Corporation, vol. XV, April 2010, url: [http://eval.symantec.com/mktginfo/enterprise/white\\_papers/b-whitepaper\\_exec\\_summary\\_internet\\_security\\_threat\\_report\\_xv\\_04-2010.en-us.pdf](http://eval.symantec.com/mktginfo/enterprise/white_papers/b-whitepaper_exec_summary_internet_security_threat_report_xv_04-2010.en-us.pdf)

#### 4.3.3.4 Synopsis

**Outcomes and Contribution to Knowledge:** The SDV-HA is a portable secure high assurance storage solution that can be configured into a secure PESE that satisfies the concept, functional requirements and three conceptual models. The SDV-HA was commercialised from a research artifact into certified high assurance secure storage device approved to protect highly classified information. The paper discussed how the SDV-HA (configured as a secure PESE) could provide a secure information storage and processing solution for remote nodes in a net-centric organisation implementing NCO; addressing a security issue in deployed and mobile military work environments. Although Fireguard and the MEE were packaged with the SDV-HA to form a high grade secure PESE it should be emphasised that neither Fireguard nor the MEE artifacts can be claimed to be high grade in their own right. These artifacts were packaged with SDV-HA artifact to form a demonstrator high grade secure PESE. It was only the SDV-HA that was developed to the high assurance standard (DSD, 2010). In Chapter 5 the trialling of the high grade secure PESE demonstrator at remote nodes in a net-centric organisation is discussed.

The SDV-HA makes a knowledge contribution through both the construction of the device and through certain novel mechanisms that were developed. The SDV-HA can be claimed to be a knowledge contribution as no other high assurance device is available that contains the same level of functionality within a small form factor. The only comparable product identified is the General Dynamics ProtecD@R (GD, 2014), an anti-tamper secure portable storage solution but with a lower assurance level, less functionality and a larger form factor. The SDV-HA's anti-tamper mechanism and the multiple modes of operation mechanism are knowledge contributions. The anti-tamper mechanism is an exaptation where a technology designed for a different solution is utilised to develop an anti-tamper mesh. Details of the design cannot be disclosed in this thesis as the certification authority classified the anti-tamper design to limit an attacker developing an understanding. The multiple modes of operation are an improvement upon the Mini-SDV design. In particular, the under-duress mode provides a novel capability that enables effects based operations to be supported. Like the anti-tamper mechanism further details on the design of under-duress mode cannot be disclosed in this thesis.

***Contemporary relevance, linkage with other papers and future direction:*** The contemporary relevance of the paper is that it describes a research artifact that was commercialised into secure portable storage solution. The paper represents the final design paper prepared and therefore it is not linked to further papers. The future work proposed is discussed in Chapter 7. The SDV-HA design addressed the research question by confirming a sophisticated active tamper mesh could be implemented and packaged into a portable and small form factor device.

#### **4.4 Summary**

This chapter has presented DSR in the form of models to diagrammatically capture the secure PESE concept and the design of four individual artifacts that were used to create two secure PESEs. The research has substantially addressed the research questions, although maintainability of the secure PEE component is considered further in Chapter 6.

A commercial grade secure PESE was constructed using the Mini SDV, the MEE and Fireguard artifacts, and an overview of its demonstration is given in Papers 6 and 7 with a more inclusive discussion given in Chapter 5. Similarly, a high grade secure PESE was constructed using the SDV-HA, the MEE and Fireguard artifacts with its demonstration discussed in Chapter 5. A high grade secure PESE is a specialist device designed for use scenarios requiring high assurance technology.

This chapter has described design research where the key outcomes were secure PESE instantiations that satisfy the research objectives (and hence concept), functional requirements and the three conceptual models. Secure PESEs were designed and constructed to manage information security risks within the remote work environment.

## **5 Demonstration**

### **5.1 Overview**

This chapter explains how the secure PESEs described in Chapter 4 were tested, trialled, certified and (where appropriate) commercialised. Testing demonstrated that a secure PESE was correctly implemented and satisfied the doctoral research objectives. Trialling demonstrated the secure PESE provided a useful and effective secure solution for remote working. Certification demonstrated the secure portable storage device used for the secure PESE had been independently assured to be secure. Commercialisation demonstrated the secure PESE had the potential to move from research artifact to product. The tenth and final paper forming this PhD with publication is presented in this chapter.

The approach to testing, trialling, certification and commercialisation is presented before discussing the application of each of these demonstration activities to a secure PESE. Papers 7 and 8 (presented in Chapter 4) discuss the demonstration of the Fireguard and MEE artifacts respectively. The demonstration activities and results presented in these papers are referenced rather than repeated. Paper 10 (presented in this chapter) forms part of the commercial grade secure PESE demonstration. The paper considers the use of the MEE as a solution to improve security where no or limited sensitive data is processed for an organisation implementing a bring your device policy (BYOD) for laptops.

### **5.2 Demonstration Process**

A comprehensive approach to demonstrating the four research artifacts and the two secure PESEs (constructed from the artifacts) was adopted. System testing of each artifact was performed before testing the integrated and fully configured secure PESEs. Each secure PESE was trialled to gauge its use and effect for a particular category of remote work. Certification is limited to the Mini SDV and SDV-HA secure storage product. The commercialisation activities forming part of the demonstration are those that confirmed a contribution to knowledge through independent assessment. The results for each demonstration activity are discussed and summarised for each secure PESE.

### 5.2.1 Testing

The research utilised the test process defined in Chapter 3. Ensuring both an individual artifact and an integrated set of artifacts (forming a secure PESE) is exhaustively tested was a complex technical activity. The Fireguard and MEE artifacts were hardened versions of tried and tested open software products whilst the Mini SDV and the SDV-HA were bespoke storage devices constructed from new and existing hardware and (embedded and non-embedded) software. Thus different testing tools, techniques, methodologies and strategies were required to ensure all software and hardware was tested. Thorough testing is stratified starting at unit testing and going through component, integration and finally systems testing (SDLC-V, 2014). Unit, component and integration testing are performed by the developer with systems testing typically performed independently.

It is the system testing of the two secure PESEs that forms part of the demonstration process discussed in this chapter. An important aspect was to ensure neither the researcher nor any of the small development team involved in the Mini SDV and SDV-HA artifacts performed system testing. A team of three independent testers were used that prepared test plans and test procedures to conduct the testing with the test results recorded in a test management system (TestLink, 2012).

The test plan was structured using the doctoral research objectives as the test propositions. As each objective was allocated functional requirements<sup>67</sup> test procedures were prepared that tested each requirement. The allocation of test procedures to objectives enabled the formation of a comprehensive set of test propositions. Test procedures were prepared for the following categories of system test to ensure the security functionality was specifically able to limit the risks of cyber-attack, data loss and forensic data discovery:

**Functional testing** was performed for all features and functionality of the secure PESE. Functional testing involved not only confirming the feature performed as required but also included error trapping to ensure the feature was robust. Functionality testing developed

---

<sup>67</sup> The research objectives were the seven secure PESE concept attributes (plus a further two usability and performance objectives). As each concept attribute was allocated functional requirements in Chapter 2, seven of the objectives had testable statements allocated. The further two objectives were implementation-oriented and were therefore testable statements.

a deeper insight into the design and logic of the product which is essential before applying test procedures for the other categories (below).

**Exception testing** identified the stability of the secure PESE under unexpected logical and operational conditions. Exception testing confirmed the secure PESE either handled an exception and continued to operate securely or moved into a fail-safe state.

**Vulnerability testing** determined the secure PESEs ability to prevent data loss and forensic data discovery and its ability to withstand cyber, brute force and physical attacks through the exploitation of known or perceived vulnerabilities.

**Stress/soak testing** examined the performance of the secure PESE under heavy workloads and over prolonged periods. It was performed by automating functional test procedures so they could be repeatedly performed.

**Usability testing** examined the interaction between end-users and the secure PESE. Usability testing determined if the secure PESE was easy to use and was intuitive, it also tested the correctness, clarity, completeness and consistency of user documentation and online help facilities.

**Compatibility testing** identified the secure PESE's ability to execute with different PCs and operating systems.

**Sociability testing** examined how the secure PESE operated in an environment running with other applications, this testing was particularly relevant for a secure PESE configured with a secure PEE consisting of a set of up-loadable applications. The testing ensured that other applications performed as expected while the secure PESE is connected and the secure PEE applications are executing.

**Environmental testing** can be considered to be a form of functional and exception testing that tested the secure PESE's ability to operate at high and low temperatures and tolerate shocks from dropping and from radiated or conducted electrical energy (i.e. immunity and electromagnetic testing).

System testing was performed through a number of iterations following the identification of issues (i.e. bugs or suggested design changes). All issues identified were captured in an issue management (MantisBT, 2012) system and reviewed to determine resolution. For each system test iteration a regression<sup>68</sup> test plan was prepared and applied.

### **5.2.2 Trial**

To gauge the usefulness and effectiveness of secure PESEs, trials were arranged using actual remote workers from an organisation. Some difficulty was experienced finding an organisation prepared to invest the time and resources as there was concern that the trial may impact business operations. Eventually two organisations agreed to participate but both requested certain details about the trial remain confidential, in particular neither organisation would allow its name to appear in published material.

The trial for the commercial grade secure PESE was conducted by a group of existing teleworkers performing a technical support service for internet and telecommunication services within the organisation. It is understood the organisation allowed the trial to be conducted by teleworkers performing actual technical support work for customers. The organisation gave permission for the results of the trial to be discussed.

The trial for the high grade secure PESE was performed by a net-centric organisation performing NCO. It is understood the trial was performed during a simulated deployed exercise and did not involve sensitive information. The organisation would neither release nor discuss in detail the results of the trial allowing only limited detail on the trial to be discussed in this thesis.

### **5.2.3 Certification**

The Mini SDV and the SDV-HA storage products were certified. Certification was performed by the Australian Signals Directorate (ASD)<sup>69</sup>. The secure PESE configurations were not certified, however as the underlying secure SDV storage platform was assurance is provided in the underlying strength of the SDV security mechanisms (i.e. the cryptographic and key management mechanisms, anti-tamper capability, and the

---

<sup>68</sup> Regression testing is the selective testing of a system using a subset of the test procedures following system enhancement.

<sup>69</sup> The ASD is the Australian certification authority for security products used by the Australian Government. The ASD certifies products that have passed either a Common Criteria or its own high assurance criteria.



authentication and access controls). For portable storage technology it is important the data protection mechanisms are certified to provide a level of assurance when data is transported outside a secure location. The certification of any up-loadable execution environment is often a lesser concern as the processing of data may be performed within secure locations such that the possibility of the introduction of malware is limited. A product like the SDV-HA is typically used to protect highly classified data with an approved, but uncertified up-loadable execution environment because the data is processed at system-high mode<sup>70</sup>.

#### **5.2.4 Commercialisation**

There are many definitions for commercialisation; a Google search for “definition of commercialisation” gives 19,100 results<sup>71</sup>. The researcher considers the following definition is indicative of the approach taken to commercialise the secure PESEs and underlying storage products that were developed:

Commercialisation is the process of transforming innovative ideas or knowledge into a product or service that is suitable for sale. It is driven by the desire to add value and gain a positive return on investment.

The above definition was formed from merging the three respective definitions given in: the “Pathways to Technological Innovation” report prepared for the Australia Parliament (APH, 2006), the Australian Institute for Commercialisation (AIC, 2014) and the Cambridge Business English Dictionary (CBED, 2014). Commercialisation activities include research, licensing, product development, and marketing. The testing, trialling and certification activities described above are all part of the commercialisation process. The other commercialisation activities to be discussed as part of the demonstration process are those that demonstrate an acknowledgement of the knowledge contribution of the secure PESE through independent assessment; such activities include industry awards and the award of commercialisation funding.

---

<sup>70</sup> System high is a security mode that requires the IT system and all its users to be at the same clearance level as the data to be processed.

<sup>71</sup> The Google search was performed in March 2015.

### 5.2.5 Demonstration Configuration

The testing, trialling and certification demonstration activities require different secure PESE configurations. System testing requires that all possible deployable configurations are tested whilst trialling requires just the specific configuration to be trialled and certification requires only the configurations to be certified. For each of these three demonstration activities a diagrammatic model, using the conceptual memory layouts in Figures 4.2 and 4.3 as the basis, is given for the secure PESE configuration. Figure 5.1 represents an example of the model consisting of:

- No Access (NA) storage area where the secure PESEs firmware and software are held.
- Read-Only (RO) partition containing the MEE, i.e. a bootable secure PEE.
- Read-Only (RO) partition containing Fireguard, i.e. a secure PEE consisting of a set of up-loadable applications.
- Read-Write (RW) persistent/data partition for any secure PEE configuration changes made during the execution of the secure PEE.
- Read-Write (RW) partition for temporary data created by a secure PEE, e.g. pagefiles, temporary print files, temporary copy of a file made by a word processor, etc.
- Read-Write (RW) a data storage partition for user data.

N/A	RO	RO	RW	RW	RW
<ul style="list-style-type: none"><li>• Firmware</li><li>• Pre-boot &amp; Post-boot Authentication Applications</li></ul>	Secure PEE (MEE)	Secure PEE (Fireguard)	Persistent (for configuration data)	Temporary Data	User Data

**Figure 5.1 Example Secure PESE Configuration**

### 5.3 Commercial Grade Secure PESE

The commercial grade secure PESE consisted of the Mini SDV, MEE and Fireguard. All of the demonstration activities outlined above were applied with a specific configuration of the secure PESE defined for the respective activity.

5.3.1 Testing

**Test Configuration:** Figure 5.2 models the secure PESE test configuration which had both types of secure PEE installed. The model represents the full configuration used for the testing, however a subset of configurations were used to test each secure PEE artifact separately, these configurations are described in Papers 6 and 7.

N/A	RO	RO	RW	RW	RW
<ul style="list-style-type: none"><li>Firmware</li><li>Pre-boot &amp; Post-boot Authentication Applications</li></ul>	Secure PEE (MEE)	Secure PEE (Fireguard)	Persistent (for configuration data)	Temporary Data	User Data

Figure 5.2 Commercial Grade Secure PESE Test Configuration

**Test Approach:** A test plan comprising the full range of system tests was applied by the independent testers. Paper 6 describes the system testing performed for the secure PESE with Fireguard installed and Paper 7 gives an overview of the secure PESE with the MEE installed. System testing was performed over an extended period as new releases addressing resolved issues were subjected to regression testing.

**Test Results:** Testing highlighted a range of issues with the majority identified in the stress/soak, usability, compatibility and sociability test categories. Fewer issues were identified for the functionality, exception, vulnerability, and environment test categories reflecting the strength of the security design and that the Mini SDV was based upon existing tried and tested technology. Example issues identified during testing and their resolution are discussed in Paper 6 (in the ‘Fireguard Testing and ‘Conclusion’ sections) and in Paper 7 (in the ‘Evaluation’ section).

5.3.2 Trial

**Trial Configuration:** Figure 5.3 models the configuration used for the trial. Only the bootable secure PEE image was installed. The image had the organisation’s technical support applications included, although the majority of the technical support applications used by the teleworkers were client-server based using the hardened MEE’s browser and remote access client to execute the server based application.

N/A	RO	RW	RW	RW
<ul style="list-style-type: none"> <li>Firmware</li> <li>Pre-boot &amp; Post-boot Authentication Applications</li> </ul>	Secure PEE (MEE)	Persistent (for configuration data)	Temporary Data	User Data

**Figure 5.3 Commercial Grade Secure PESE Trial Configuration**

**Trial Approach:** A good description of the trial is provided in Paper 7 (in the ‘Demonstration’ section). It is understood the organisation allocated the secure PESE to five teleworkers to trial over a one week period whilst performing actual technical support work for customers. A further group of teleworkers in the organisation used a standard (non-secure) thumb drive with the MEE image installed (as described in Paper 7) – this configuration is not discussed here; the demonstration results of this configuration support the synopsis discussion for Paper 10.

**Trial Results:** Results of the trial are summarised in Paper 7 (in the ‘Evaluation’ section), with expanded observations given below. The trial demonstrated that a secure PESE provides a solution for telework that was:

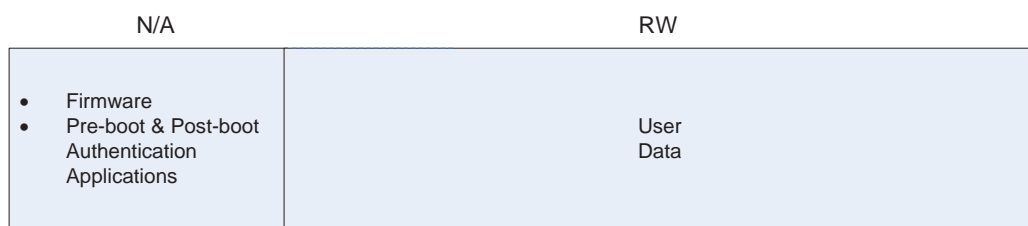
- Easy to use with similar performance to the existing execution environment used by the teleworkers. Minimal or no training was required before use. The trial teleworkers were able to move rapidly from their existing execution environment to the secure PESE.
- Able to provide productivity increases. The organisation expects to accrue productivity increases as a teleworker is provided with an execution environment secured in a protected partition that includes only the required technical support applications, and prevents both new applications from being installed and the teleworker changing the execution environment. The organisation had previously experienced teleworkers unable to perform technical support work due to a PC corrupted through malware or the teleworker’s actions.
- Able to reduce costs. The ability of the secure PESE to operate with any PC allowed for a bring your own PC policy and the ability for a teleworker to move to an alternate PC following a PC failure resulting in no loss of technical support service.

- Secure. A post trial audit identified no malware or other security issues, although the short duration of the trial did limit the opportunity for attack.

Following the trial the organisation decided to implement a secure PESE solution for its telework team.

### 5.3.3 Certification

**Certification Configuration:** Figure 5.4 models the Mini SDV configuration used for evaluation. It has no secure PEE installed and therefore none of the partitions required to support the secure operation of a secure PEE.



**Figure 5.4 Commercial Grade Secure PESE Certification Configuration**

**Certification Approach:** As discussed in Chapter 4 the Mini SDV is an improvement upon the Common Criteria evaluated (CC, 2014) Pocket SDV. ASD certified Pocket SDV as a Common Criteria EAL2 product (PocketSDVCert, 2012) suitable for the protection of sensitive government information, and therefore suitable for commercial use. Using the Pocket SDV as the baseline ASD used its certification maintenance program to evaluate the improved security features of the Mini SDV.

**Certification Result:** Following the successful completion of the evaluation ASD certified the Mini SDV (MiniSDVCert, 2012) to the same level as the Pocket SDV, i.e. certified to protect sensitive government information.

### 5.3.4 Commercialisation

Remote working may involve retaining little or no sensitive information on a teleworker's PC, or it could involve the local storage of considerable amounts of sensitive information and valuable intellectual property. As ICT has allowed the remote location to become an extension of the corporate office it is important that the same level of security provided in the corporate office is also provided in the remote location even though the threat and

risk profile is different. The commercial grade secure PESE was targeted as a computing environment for remote working with the proposition: 'The Mini SDV when configured as a secure portable execution and storage environment provides a "Secure Laptop in Your Pocket" that conceptually extends the secure corporate environment to the remote location (Pearcy, 2006)'.

To test the proposition and validate a knowledge contribution the secure PESE was entered into the Global Security Challenge (GSC), an international competition for small innovative security technology companies that is organised by InnoCentive (InnoCentive, 2014). The competition attracts companies from all over the world and typically has over 150 applications. Despite the secure PESE not being fully complete it was presented as the "Secure Laptop in Your Pocket" at the 2009 competition (GSC, 2009) where it was a finalist. Feedback received indicated that the proposition was innovative and a knowledge contribution to secure portable computing technology.

### **5.3.5 Extending the Use Case of the Bootable Secure PEE**

#### **5.3.5.1 Preamble**

The telework trial included a secure PESE configuration consisting of the MEE image installed onto a standard (non-secure) thumb drive. Whilst this configuration lacked the security features provided by the Mini SDV (i.e. authentication, encryption, access controls and anti-tamper) it did provide a very low cost solution where:

- The teleworker's PC acts primarily as a thin client with all processing performed on a secure corporate server/cloud.
- No data is to be stored in the remote work location.
- An execution environment is required that cannot be changed and is designed to limit the introduction of malware.

The weaknesses of the solution included:

- No mechanism exists to protect the integrity of the MEE, it was possible to delete or corrupt the image.
- No authentication nor access controls to prevent unauthorised access to the MEE.

- No encryption so the MEE and stored data can be read and copied.

The organisation hosting the trial considered the weaknesses of the thumb drive based secure PESE and decided it presented an acceptable risk. As the technical support work performed in the trial processed all data on corporate servers the organisation considered the thumb drive based secure PESE offered a low cost, low risk solution for teleworker's whom furnished their own PC (refer to Paper 7 for trial results). The thumb drive based secure PESE was therefore included in the trial together with the Mini SDV based secure PESE.

Following the trial the researcher and paper co-author decided to consider the thumb drive PESE as a low cost solution to enforce security for a bring your own laptop policy for both office and remote workers. The thumb drive secure PESE with the MEE image is similar to the LPS (LPS, 2008) and Becrypt TC (Becrypt, 2009) products, however the key difference is the MEE and its applications are hardened.

The paper does not document new design science research but instead looks at how to expand the usefulness of an artifact. The paper follows a DSR presentation style, i.e. the problem is identified in terms of risks, a literature review of alternative technologies is presented, objectives/requirements are defined, the design is described, and a demonstration and evaluation is presented in the form of conformance to requirements and the ability of the design to counter the risks. The paper supports Paper 7 in addressing the research question: *How can a useable and maintainable hardened operating system and/or small set of hardened applications be developed?*

#### **5.3.5.2 Prior Knowledge and Research**

The following knowledge was consumed during the investigation:

- Descriptive knowledge acquired from:
  - The security model attributes identified in Paper 3. The attributes were used as the basis to specify security requirements.
  - The trial results described in Paper 7.

- The remote work security threats defined in Chapter 2. The threats were used to determine a high level set of security and business risks.
- Prescriptive knowledge describing the MEE design, presented in Paper 7.

#### 5.3.5.3 Paper 10

**Paper 10** - James, P. and Griffiths, D. (2012) **The Mobile Execution Environment: A Secure and Non-Intrusive Approach to Implement a Bring Your Own Policy For Laptops**, 6th Australian Information Security Management Conference, Perth, pp 82-97.

#### Abstract

*Bring Your Own Device (BYOD) has become an established business practice, however the practice can increase an organisation's information security risks. The implementation of a BYOD policy for laptops must consider how the information security risks can be mitigated or managed. The selection of an appropriate secure laptop software configuration is an important part of the information security risk mitigation/management strategy. This paper considers how a secure laptop software configuration, the Mobile Execution Environment (MEE) can be used to minimise risks when a BYOD policy for laptops is implemented.*

*In this paper the security and business risks associated with the implementation of such a policy are identified and discussed before giving an overview of a range of laptop software configuration options suitable for the implementation of a secure BYOD policy. The design objectives and security requirements of the MEE are enumerated and its key features described. For each identified risk, the MEE features that mitigate/manage the risk are presented. The paper concludes by considering the type of work for which the MEE is most suited and also how the security features of the MEE can be enhanced when the MEE forms part of a secure portable execution and storage environment.*

#### Keywords

BYOD, information security, portable execution environments, hardened operating system.

#### Introduction

The Bring Your Own Device (BYOD) approach has emerged through recognition that employees were using their own (more sophisticated) laptops to perform work, either in the workplace or at home. The recent accelerated adoption of the BYOD paradigm can also be attributed to an organisational need to both introduce human resource (HR) policies that attract and retain talented employees, and contribute to the containment of information technology (IT) infrastructure costs. This paper will consider the implementation of a secure BYOD policy for laptops using a simple, yet secure laptop software configuration called the Mobile Execution Environment (MEE).



The purchase of sophisticated and technologically advanced laptops for home and private use has created a workforce that has high expectations of the capabilities of the laptops furnished by their employer. Accordingly, the requirement to furnish and maintain laptops to a growing percentage of employees who need a laptop, and to refresh the laptop fleet on a regular basis, has increased organisational IT costs. From a HR and financial perspective (typically key organisational business drivers), the introduction of a BYOD policy for laptops has a number of advantages (Paul, 2009; PWC, 2012) as it:

- **Empowers employees:** Selecting your own laptop (and operating system) can instil a sense of loyalty to the organisation and hence contribute to ownership of corporate objectives and strategy.
- **Allows for flexibility:** An employee has the same platform for both professional and private computing potentially enabling work to be more readily performed (as required) outside of the office. For mobile workers and teleworkers a single platform for both professional and private activities can contribute to the concept of a more agile workforce.
- **Delivers infrastructure cost savings:** Often the BYOD policy is implemented through an allowance which the employee can supplement if desired, possibly resulting in the purchase and use of a more superior and powerful laptop than would otherwise be used if the employer furnished the laptop. As the user owns the laptop he/she is more inclined to keep the laptop in a good state of repair resulting in lower costs for an organisation's break/fix support team. Also the "technology savvy" employee is more likely to upgrade to a new more powerful model using their own money more often than the typical three year replacement cycle of most organisations.

These perceived HR and financial advantages are accelerating the transition to BYOD particularly where an organisation is conducting projects using activity based working and/or is allowing staff to telework. There are, however, a number of security and business risks that need to be considered before a BYOD policy for laptops is introduced. The selection of a secure laptop software configuration can mitigate or manage some of these risks. In this paper a secure laptop software configuration is modelled and presented as consisting of the following three components:

1. **Professional computing environment:** The set of software applications provided by the employer to allow an employee to conduct work. The professional computing environment executes within the computing delivery software.
2. **Computing delivery software:** This software facilitates the execution of the professional computing environment and separates it from personal/private software; examples of computing delivery software are a remote desktop client, a browser or a virtual machine. The computing delivery software executes on the platform software.
3. **Platform software:** The software that manages and makes available the capabilities of the laptop hardware; examples of platform software are operating systems and type 1 hypervisors.

An appropriate selection of computing delivery software and platform software can provide the basis for a secure laptop software configuration. In this paper the MEE (a secure laptop software configuration) is presented and discussed as a suitable laptop software configuration to counter a set of identified security and business risks associated with implementing a BYOD policy for laptops. To enable the reader to appreciate the laptop software configurations that can be selected a number of the popular computing delivery and platform software components are outlined. The design of the MEE is discussed and an assessment of how the MEE security features mitigate or manage the identified security and business risks is given. The style and type of work most suited to the use of the MEE is outlined, and finally limitations of the MEE are highlighted and the use of a secure portable execution and storage environment that addresses the limitations is outlined.

## **Security and Business Risks**

A risk assessment should be performed before the implementation of any new IT initiative that involves the processing of sensitive corporate data, particularly if the processing is to occur outside of the boundary enforced by the organisation's security policy. The introduction of a BYOD policy for laptops without full consideration of the information security risks will inevitably result in breaches of confidentiality of sensitive corporate information (Markelj and Bernik, 2012). High speed broadband, coupled with the secure laptop software configurations available, is allowing organisations to implement secure remote access architectures that achieve an appropriate level of information security when employee owned laptops are used to process sensitive corporate data.

The following set of risks has been identified as those that need to be considered before the implementation of a BYOD policy for laptops. Each risk is presented without any consideration given to the laptop software configuration that can be used to mitigate or manage the risk. For each risk its impact on confidentiality, integrity and availability is identified:

***Corporate Data Becomes Resident on the Laptop:*** When an employee owned laptop is used for both professional and private purposes it is possible that the laptop will have corporate data stored on it. This corporate data may be intentionally stored or inadvertently stored due to the actions of the user or the standard processing actions of the laptop software configuration (Jones et al, 2008). As the laptop is employee owned it will be used for private and social activities where it could be vulnerable to loss or theft, possibly resulting in unauthorised access to corporate data. Also a laptop fault may require warranty/service work to be performed by the supplier, possibly resulting in the laptop supplier having unauthorised access to any sensitive corporate data residing on the laptop and allowing for the possibility of the supplier to embed malicious software. This risk could result in a breach of data confidentiality and/or system integrity.

***Personal Laptop Use Affects the Integrity of the Professional Computing Environment and Data:*** An employer furnished laptop is provided to an employee on the basis that it is used predominately for professional purposes, with possibly some private use. However, a BYOD laptop policy will result in the laptop being used for a range of personal activities as the employee owns

the laptop. Such personal activities may result in the installation of software packages and data that could affect the integrity of the professional computing environment and/or corporate data stored on the laptop. Software and data integrity issues including damage to configuration settings and corruption of software and data could also result in the laptop not being available to perform work, impacting productivity. This risk will impact software and data integrity, and could impact laptop availability.

***Laptop Not Available for Work:*** When a laptop is used for private purposes it may be subject to handling that impacts its physical and/or logical capabilities resulting in damage that prevents the laptop being available for professional work. Also an employee owned laptop is unlikely to be locked (i.e. prevented from installing application software) and/or configured to run with minimum user privileges. An unlocked and/or privilege high laptop may therefore allow software to be installed for private use that may conflict with, and/or prevent the operation of, the professional computing environment resulting in the laptop not being available for work. Further, if the employer provides a break/fix support capability yet allows employees to select any laptop they desire then there may be delays with the support team addressing laptop issues due to the team's inability to have the necessary skills and knowledge in all the different laptops and OS' that have been selected. Similarly, allowing an employee to select any laptop/OS may result in a selection that is functionally new to the user causing both a learning curve that impacts productivity and a lack of technical experience to resolve laptop issues when they occur. Finally, as identified in the risk above, if the integrity of the professional computing environment is compromised through private use the laptop may not be available for work. This risk will impact laptop availability.

***Malicious Software (Malware) Becomes Resident on the Laptop:*** Malware can be introduced through numerous sources; some of the most popular sources include a browser, email, portable storage media and application software (Symantec, 2012; Sophos 2012). A laptop used for private and professional use could be more susceptible to malware (due to the private use) than a locked and/or low privilege employer supplied laptop. Although anti-virus and anti-malware software is now a standard feature of a laptop purchase, increasingly such software is unable to keep pace with the complexity and growing number of malware/viruses. Malware allows cyber-attacks to be launched. Such attacks can result in the loss of sensitive corporate data. This risk could result in a breach of data confidentiality, impact software and data integrity, and impact laptop availability.

***Private Use Affects Productivity:*** As a user's personal application software suite will be installed on a BYOD laptop there may be the temptation to use the laptop for non-work activities (during a working day), resulting in reduced productivity from the employee. Although this risk is not a direct security risk, this business risk could contribute to the security risks (identified above) occurring.

The computing delivery and platform software options identified below are designed to counter many of the aforementioned risks. This paper considers the MEE as a combined implementation of a professional computing environment, computing delivery and platform software that addresses all the identified security and business risks in a non-intrusive way (i.e. the laptop software

configuration that is provided by the MEE neither intrudes upon nor interferes with the personal/private software).

## Laptop Software Configuration Options

As identified above, in this paper the laptop software configuration is defined as a set of software components which are modelled diagrammatically in Figure 1.

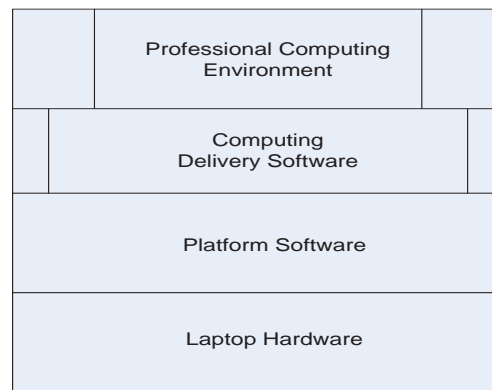


Figure 1: Conceptual Model of Software Components forming the Laptop Software Configuration

The professional computing environment is the set of software applications provided by the employer to enable the employee to conduct work. The applications may consist of commercial off the shelf packages and/or bespoke software.

Computing delivery software is based upon thin client (3Com, 1999), virtualisation (VMware, 2006) or zero client (David, 2002) technology and the most popular examples include:

**Browser:** A browser is designed to allow the retrieval, presentation and creation of information across a network and therefore, depending upon the work to be performed, can provide an ideal computing delivery model to access an application (that forms part of a professional computing environment) located on a remote server.

**Remote Desktop:** The standard terminal server application provided with most operating systems is used to connect to a corporate server to access the professional computing environment. Like the browser all processing is performed remotely and therefore no corporate applications have to be installed on the laptop. Remote desktop has traditionally been a popular delivery model for remote computing.

**Desktop Virtualisation:** Virtualisation can be achieved either through installing a virtual machine (VMware, 2006) containing an image of the 'professional computing environment' on the laptop, or alternatively installing a virtual machine client (Rouse, 2011a) that connects to a corporate server that is hosting the virtual machine containing an image of the 'professional computing environment'.

**Application Hosting or Streaming:** Application hosting software (Citrix, 2011) makes an application (in the professional computing environment) available on the laptop but the application is actually installed in a virtual machine that is either downloaded from a server or executed directly on the server. Conversely, application streaming software (Citrix, 2011; Rouse 2011b) has an application installed on a server which is streamed to the laptop in full or in part as required. In both cases the laptop user is unaware the application is not installed locally.

The above computing delivery software options allow the laptop to act either as a terminal to access a server over a network or as a networked workstation, if the network is the Internet then a secure protocol (e.g. virtual private network or hypertext transfer protocol secure) would be used to protect data transmission. Each computing delivery option has advantages and disadvantages and therefore selection will depend upon a number of different factors including, bandwidth of communication technology, information security, the type of work performed and the employer's software, hardware, communications and technical support infrastructure. To enable any of the above computing delivery options to be implemented one of the following platform software options could be utilised:

- **Laptop Operating System (OS):** The selected computing delivery model can be installed on to the laptop OS.
- **Type 1 Hypervisor:** A type 1 hypervisor (VirtualComputer, 2012) runs directly on the laptop hardware (instead of the laptop OS). The type 1 hypervisor acts as a virtual machine manager, i.e. it allows a number of virtual machine images to execute. A hypervisor as the platform software replaces the laptop OS as the laptop bootable software. The user can then utilise private or professional virtual machine images as required.
- **Dual Boot:** The dual boot option requires the installation of two OS' (each in a separate partition) on the laptop HDD; one for professional use and the other for private use. When the laptop is powered on the user selects which OS to load and execute.
- **Secure Portable Execution Environment (secure PEE):** A secure PEE (James, 2008) is an execution environment (e.g. a highly portable OS and set of applications) contained on an external attachable device (e.g. a USB thumb drive). When the external attachable device is plugged into the laptop and is set as the first boot device it loads the secure PEE. The secure PEE requires no access to the laptop HDD and uses only the processor and memory of the laptop. Depending upon how the secure PEE is packaged it can be considered to be platform software or a combined platform and computing delivery software.

Like the computing delivery software options the selection of the platform software will be dependent upon a number of factors. The MEE is an example of a secure PEE that can be packaged with any of the computing delivery software options (identified above) to provide a secure laptop software configuration that can be uploaded from portable storage media onto a laptop.

The dual boot platform software option is similar to the portable PEE option, the difference being that the computing delivery software and professional computing environment is installed on the laptop HDD together with the private operating system and applications, albeit each professional/private environment in a separate partition. Hence, the dual boot option makes the professional computing environment accessible and therefore potentially vulnerable when the private operating system and applications are executing.

A key difference between the MEE and the other single operating system and type 1 hypervisor platform software options is that only software applications forming part of the MEE can be utilised whilst the MEE is executing, access to private data/applications on the laptop HDD is not possible. The MEE effectively separates private from professional computing whilst the MEE is executing.

## **MEE – Features and Functionality**

The MEE is part of an on-going research project to develop secure portable execution and storage environment (secure PESE) devices (James, 2008) to support secure remote computing. The goal of the MEE is to provide a simple hardened secure PEE that can be executed from a secure PESE device. A detailed description of the design of the MEE is given in James and Griffiths (2012).

An outcome from the research project is that the MEE can be imaged onto a standard USB thumb drive and used as a laptop software configuration to support a secure BYOD policy for laptops. As a laptop software configuration, the MEE consists of the professional computing environment, computing delivery software and platform software that is up loaded from a thumb drive onto a laptop and executed.

## **MEE Project Business and Operational Objectives**

A project development goal was to build the MEE from freely available open software. The following business and operational objectives were defined to direct both the selection of open software and the development of the MEE:

1. Enable the rapid development of the MEE.
2. Be bootable from a USB storage device.
3. Ability to run on the widest variety of laptops without additional driver installation.
4. Support a wide range of application software.
5. Have acceptable licensing conditions.
6. Be easy to use.
7. Provide the basis to separate the professional computing environment from the personal/private software applications.

Following a comprehensive review of a number of open software operating systems the 'live CD' version of Ubuntu (Ubuntu, 2012), the commercially supported Linux distribution was selected.

Table 1 presents the rationale for the selection of Ubuntu by showing conformance to the business and operational objectives of the MEE project.

*Table 1: Rationale for Selection of Ubuntu*

<b>Business/Operational Objective</b>	<b>Ubuntu Based MEE</b>
Enable the rapid development of MEE.	Ubuntu is a commercially supported Linux distribution that is professionally packaged and structured such that the MEE could be constructed to a tight development schedule.
Be bootable from a USB storage device.	The 'live CD' version of Ubuntu can be ported to, and executed from, portable USB attachable storage media.
Ability to run on the widest variety of laptops without additional driver installation.	Ubuntu is considered to be one of the most portable Linux distributions (Lifehacker, 2009).
Support a wide range of application software.	Due to its commercial support Ubuntu has one of the widest ranges of application software (Vaughan-Nichols, 2012).
Acceptable licensing conditions.	The Linux Gnu licence allows a distribution to be configured for commercial use provided the kernel is not changed.
Easy to use.	Ubuntu provides an easy to use and configurable user interface.
Provide the basis to separate the professional computing environment from the personal/private software applications.	As Ubuntu can be tailored, configured and then imaged onto a portable USB storage device it can be booted to provide a professional computing environment that is separate to the personal/private software held on the laptop HDD.

### **MEE Security Requirements**

The MEE was developed as part of a secure PESE device project which was designed to satisfy the following teleworking/remote working security model attributes (James, 2011):

- Protect data transmitted over a network.
- Ensure only authorised access to the teleworker's computing environment is achieved.
- Protect the confidentiality of data processed by the teleworker.
- Prevent temporary data and data remnants (created through data processing) being accessed by an unauthorised user.
- Protect the integrity of the teleworker's computing environment and stored data from malicious software and/or user induced damage.
- Protect the confidentiality and integrity of any software and data stored on a portable storage device.
- Ensure the availability of the teleworker's computing environment.



Using the teleworking security model attributes as a basis the project defined and allocated the following set of security requirements to the MEE:

1. Prevent the storage of corporate data on laptop HDD.
2. Prevent the user from installing software.
3. Prevent the user from performing privileged/administrator actions.
4. Prevent the laptop HDD being accessed.
5. Limit the introduction of malware.

Certain security requirements (e.g. “prevent the user from installing software” and “prevent the laptop HDD being accessed”) can be considered to conflict with the BYOD HR objective of empowering and trusting users. However, satisfying these security requirements in the MEE provides a secure and elegant approach to separating professional and personal computing activities on the one laptop. As outlined in a later section of this paper, the MEE may not be appropriate for all BYOD laptop users but for a particular category of worker the MEE provides a secure solution to implement a BYOD policy for laptops.

### **MEE Design**

An MEE development goal was to construct the MEE with minimal or no software changes, and in particular avoid Ubuntu kernel changes. This development goal was achieved through making configuration changes to the desktop manager, specific application configuration settings and operating system configuration changes; no kernel changes were necessary. The key features of the MEE that enable it to be a secure laptop software configuration are:

**Simple Desktop:** In its most basic form, the MEE has only a browser and remote terminal client available; all other applications have been removed/disabled. Upon booting the MEE the user is presented with either the simple desktop interface or a default application. From the desktop the user can only access applications that are packaged with the MEE; no command line access is possible.

**Single Unprivileged User:** The MEE has a single user. The user has no privileges, other than those required to configure the MEE to print and access a network. The user is unable to switch to a more privileged role.

**No Access to Laptop HDD:** A user of the MEE is neither able to access the laptop HDD nor mount a volume/partition on the HDD.

**Data Storage:** The thumb drive containing the MEE provides a volume/partition for user data storage and also a separate partition to keep application settings and virtual memory (N.B. it is possible to select to not store virtual memory on the thumb drive as continuous writes to the drive can reduce the life of the flash storage). The user data partition is configured as the first partition



on the thumb drive and formatted as a FAT-32 file system, allowing the user to access the data partition when the thumb drive is plugged into a PC running Microsoft Windows.

**Secure Browser:** The MEE browser has been configured to reduce the opportunities for malware to exploit the browser. A detailed description of the design of the secure browser is given in Griffiths and James (2010).

**Utilising the Capabilities of a Secure PESE Device:** The MEE is designed to exploit the security features of the Secure Systems' Mini Silicon Data Vault (Secure Systems, 2012); a secure PESE device. In particular, the MEE is designed to utilise the SDV's secure partitioning, authentication and encryption capabilities (James and Griffiths, 2012). The utilisation of the SDV features by the MEE provides a secure remote computing solution to protect highly sensitive corporate data when it is processed outside the secure corporate environment. When installed on a secure PESE device the MEE writes all temporary data to the device and the MEE is protected by a secure partition, strong authentication and encryption. The MEE as part of a secure PESE solution is considered later in this paper.

Table 2 presents a requirements conformance matrix, mapping the MEE features developed to address the MEE security requirements.

Table 2: Requirements Conformance Matrix

MEE Security Requirement	MEE Feature That Satisfies Requirement.
Prevent the storage of corporate data on laptop HDD.	No access to the laptop HDD is possible.
Prevent the user from installing software.	Simple locked down desktop and single unprivileged user prevents the installation of software.
Prevent the user from performing privileged/administrator actions.	Single unprivileged user.
Prevent the laptop HDD being accessed.	No access to the laptop HDD is possible.
Limit the introduction of malware.	Simple locked down desktop and single unprivileged user and secure browser limits the opportunity for malware to attack or become embedded in the MEE.

## The MEE as a Secure Laptop Software Configuration

The MEE is a laptop software configuration that enables the packaging of the professional computing environment, computing delivery software and platform software into a single separate loadable execution environment. Ubuntu provides the platform software component of the MEE. In its simplest configuration the MEE provides a browser and remote desktop as computing delivery software, although any computing delivery software that can execute on Ubuntu can be installed. The professional computing environment is either installed as part of the MEE or installed on a remote server and accessed remotely through the MEE computing delivery software. The MEE provides a solution to achieve a secure BYOD laptop policy as it mitigates/manages the identified BYOD security and business risks, as follows:

**Corporate Data Becomes Resident on the Laptop:** The MEE provides a completely separate execution environment for professional work and the processing of corporate data. The MEE prevents access to the laptop HDD and therefore it is not possible for sensitive corporate data to become stored on the laptop HDD; only the laptop processor and memory are used by the MEE. The laptop can be used for private and personal activities without concern that corporate data may be resident on the laptop. If there is a need to store corporate data and organisational policy allows for external storage of data then the data can be stored on the user data partition of the MEE thumb drive or alternatively on another external storage device.

**Personal Laptop Use Affects the Integrity of the Professional Computing Environment and Data:** As the MEE provides the professional computing in a separate bootable execution environment, personal use of the laptop will not affect its integrity. The MEE thumb drive will not be plugged into the laptop when the laptop is being used for personal computing. The separation of personal and professional computing activities on a laptop limits the possibility of deliberate or accidental damage to the professional computing environment.

**Laptop Not Available for Work:** Lack of laptop availability will not prevent work being performed as the MEE thumb drive can be booted and used from any available PC or laptop.

**Malware Becomes Resident on the Laptop:** Any malicious software resident on the laptop HDD cannot attack the MEE as the MEE prevents access to the laptop HDD; the MEE is a completely separate system that boots and loads from a thumb drive. The MEE's small set of fixed applications and the lack of a privileged user capability reduce the possibility that any malware introduced during professional work can successfully attack or become embedded in the MEE.

**Private Use Affects Productivity:** The MEE provides a separate execution environment and only includes applications that form the professional computing environment. When the MEE is executing the user is not able to access and use private application software.

## Conclusion

In this paper the MEE has been proposed as a secure laptop software configuration to support the implementation of a secure BYOD policy for laptops. However the MEE should only be considered to be part of a security solution when implementing such a policy. Appropriate security policy and procedures, training and awareness, network auditing and need to know access controls should all be considered as an integral part of a BYOD policy (Valli, 2012).

The use of the MEE as a laptop software configuration for a BYOD policy is particularly suited to teleworkers involved in transaction-oriented processing, e.g. remotely based customer support or back office business functions. Whilst transaction-oriented processing may not require highly qualified knowledge workers the increasingly sophisticated nature of the work is necessitating that policies like BYOD and teleworking are implemented to attract and retain appropriately skilled personnel. Transaction-oriented processing typically involves a continuous stream of activities that

are performed consecutively. Each activity requires the worker to be completely focussed on the activity until its conclusion. Transaction-oriented processing is characterised by a fixed set of repetitive activities and hence requires a professional computing environment consisting of a (small) suite of dedicated software applications. There should be no requirement for a laptop software configuration, used for transaction-oriented processing, to be changed by the worker nor should access to personal software applications be necessary as the worker is likely to be processing a continuous stream of transactions. The MEE therefore provides a suitable laptop software configuration for remotely based transaction-oriented processing as it provides a dedicated professional computing environment that is separated from personal software applications. The provision of the MEE on a thumb drive provides a solution for organisations where highly sensitive data is not processed but where security of data is still an important consideration.

As a secure laptop software configuration the MEE:

1. Prevents sensitive data residing on a laptop HDD.
2. Prevents personal activities affecting the integrity of the professional computing environment.
3. Allows work to be performed if the laptop is not available.
4. Limits the opportunity for any malware to become embedded on the laptop HDD;
5. Limits the occurrence of private activities whilst work is being conducted.

These five capabilities of the MEE provide a secure and non-intrusive solution for the implementation of a BYOD laptop policy. The MEE does, however, have limitations which include:

1. No mechanism to protect the integrity of the MEE on a thumb drive.
2. No authentication mechanism or access controls to prevent unauthorised access and use.
3. No encryption so the MEE and any data stored on the thumb drive can be read and copied.

These limitations are a consequence of installing the MEE on a standard thumb drive. The MEE was developed as part of a secure PESE project. A secure PESE provides security functionality that complements, and integrates with the MEE. When used with a secure PESE, the MEE is protected by a secure partition, access controls, authentication and encryption; these secure PESE security mechanisms address the above limitations. Many of the features of a secure PESE are not necessary unless highly sensitive data is being processed.

## References

- 3Com (1999). The Net Impact of Thin Clients – Technical Brief, 3Com Corporation, September 1999, Retrieved July 2012 from URL: [http://www.pulsewan.com/data101/thin\\_client\\_basics.htm](http://www.pulsewan.com/data101/thin_client_basics.htm).
- David, B. (2002). Thin Client Benefits, Newburn Consulting, Version 1b, March 2002, Retrieved July 2002 from URL: [http://www.thinclient.net/pdf/Thin\\_Client\\_Benefits\\_Paper.pdf](http://www.thinclient.net/pdf/Thin_Client_Benefits_Paper.pdf).
- Griffiths, D and P. James (2010). Fireguard – A Secure Browser with Reduced Forensic Footprint, Journal of Network Forensics, Vol. 2, Issue 2, Summer 2010.

- James, P. (2008). Secure Portable Execution Environments: A Review of Available Technologies, 6th Australian Information Security Conference, December 2008, Edith Cowan University, Perth.
- James, P. (2011). Are Existing Security Models Suitable for Teleworking?, 9th Australian Information Security Conference, December 2011, Edith Cowan University, Perth.
- James, P and D. Griffiths (2012). A Hardened Mobile Execution Environment to Enable Secure Remote Working. Awaiting publication.
- Jones, A., Valli, C., Dardick, G., & Sutherland, I. (2008). The 2007 Analysis of Information Remaining on Disks offered for sale on the second hand market. Journal of Digital Forensics, Security and Law, Vol. 3, Issue 1, 2008.
- Lifehacker, (2009). Five Best Live CDs, Lifehacker, February 2009, Retrieved July 2012 from <http://lifehacker.com/5157811/five-best-live-cds>.
- Markelj, B. and I. Bernik (2012). Mobile Devices and Corporate Data Security, Journal of Education and Information Technologies, Vol. 6, Issue 1, 2012.
- Paul, A. (2009). Bring Your Own Computer Policies, The Generation V, November 2009, URL: <http://www.thegenerationv.com/2009/11/bring-your-own-computer-byoc-policies.html>.
- PwC (2012). Bring your own device: Agility through consistent delivery, Price Waterhouse Coopers LLP 2012, URL: [http://www.pwc.com/en\\_US/us/increasing-it-effectiveness/assets/byod-1-25-2012.pdf](http://www.pwc.com/en_US/us/increasing-it-effectiveness/assets/byod-1-25-2012.pdf).
- Rouse, M. (2011a). Client-based Virtual Machine, SearchVirtualDesktop TechTarget, November 2011, Retrieved July 2012 from <http://searchvirtualdesktop.techtarget.com/definition/Client-Based-Virtual-Machine>.
- Rouse, M. (2011). Application Virtualisation, SearchVirtualDesktop TechTarget, November 2011, Retrieved July 2012 from <http://searchvirtualdesktop.techtarget.com/definition/app-virtualization>.
- Secure Systems, (2012). Mini Silicon Data Vault, 2012, Retrieved July 2012 from URL: <http://www.securesystems.com.au/secure-systems-mini-sdv.html>.
- Symantec, (2012). Internet Security Report 2011 Trends, Symantec Corporation, April 2012, URL: <http://www.symantec.com/threatreport/>
- Sophos, (2012). Sophos Threat Report 2012 – Seeing the Threats Through the Hype, Sophos Ltd. Retrieved July 2012 from URL: <http://www.sophos.com/en-us/security-news-trends/reports/security-threat-report.aspx>.
- Ubuntu, (2012). Ubuntu Documentation - LiveCD, Canonical Limited, Retrieved July 2012 from URL: <https://help.ubuntu.com/community/LiveCD>.
- Valli, C. (2012). Bring Your Own Disaster, Secau – Security Research Centre seminar, Edith Cowan University, Joondalup, WA, 22nd March 2012.
- Vaughan-Nichols, S. (2012). Ubuntu 12.04 vs. Windows 8: Five points of comparison, ZDNet, May 2012, Retrieved July 2012 from URL: <http://www.zdnet.com/blog/open-source/ubuntu-12-04-vs-windows-8-five-points-of-comparison/10900>.
- VMware, (2006). Virtualization Overview, VMware White Paper, VMware Inc. Retrieved July 2012 from URL: <http://www.vmware.com/pdf/virtualization.pdf>
- VirtualComputer, (2012). Type-1 vs. Type-2 Client Hypervisor, Retrieved July 2012 from <http://www.virtualcomputer.com/type-1-vs-type-2-hypervisor>.

#### 5.3.5.4 Synopsis

**Outcomes and Contribution to Knowledge:** The paper presented an extended use case for the thumb drive based secure PESE. The investigation represents an additional outcome from the previously performed design research rather than a new knowledge contribution. The paper does include a simple conceptual model of the software components forming a BYOD laptop solution allowing the MEE to be positioned against alternative solutions. The investigation made the conjecture that the approach taken in the design of the MEE provided a low cost solution for transaction-oriented work where little or no sensitive data was stored in the remote work location. The trial described in Paper 7 supports this conjecture as the teleworkers processed a constant stream of technical support inquiries, i.e. the teleworkers performed transaction-oriented work.

**Contemporary relevance, linkage with other papers and future direction:** Preserving both the confidentiality of information and the integrity of a laptop's execution environment continues to be a relevant security concern for an organisation implementing a BYOD policy. A recent white paper (Seltzer, 2013) discusses how the Imation Ironkey (Ironkey, 2014) packaged with the Microsoft Windows To Go (Win2Go, 2014) portable operating system provides a BYOD solution identical to the proposition proposed in the paper. The paper has been cited in the following publications further confirming its contemporary relevance:

Dawson, P. (2015), Five ways to hack and cheat with bring-your-own-device electronic examinations, British Journal of Educational Technology, doi: 10.1111/bjet.12246.

Mahesh, S., & Hooter, A. (2013), Managing and securing business networks in the smartphone era, 5<sup>th</sup> Annual General Business Conference, Sam Houston State University, Huntsville, Texas, April 19 - 20, 2013.

The paper is the final paper in the set of ten presented in this thesis with publication.

#### 5.3.6 Summary of Results

The demonstration activities show that the commercial grade secure PESE is a successful instantiation of the secure PESE concept and functional requirements. The testing confirmed the functional requirements were implemented to provide a secure remote

work solution that addressed the security issues identified in Chapter 2. The trial demonstrated that the secure PESE could be used successfully for teleworking. The ASD certification provided assurance of the security mechanisms and the additional commercialisation activities provided validation of the knowledge contribution by independent experts. The demonstration results verified the claims made in Chapter 4 with respect to addressing the research questions.

The Mini SDV was commercialised as a secure portable storage product with user documentation provided showing how to configure the Mini SDV as a secure PESE. Both Fireguard and the MEE were left as demonstrators, i.e. they remained research artifacts that were available as prototype solutions. Chapter 6 discusses why Fireguard and the MEE were not commercialised.

### 5.4 High Grade Secure PESE

The high grade secure PESE consisted of the SDV-HA, MEE and Fireguard. Although both Fireguard and the MEE formed part of the configuration neither of these secure PEE artifacts is claimed to be a high assurance component. These two secure PEE artifacts were included to demonstrate capability. It is recognised that a high grade secure PESE would include a secure PEE that had an appropriate level of assurance and/or accredited for use with classified data. All of the demonstration activities outlined above were applied with a specific configuration of the secure PESE defined for the respective activity.

#### 5.4.1 Testing

**Test Configuration:** The test configuration is modelled in Figure 5.5 and consists of a fully configured secure PESE to ensure all features and functionality are tested.

N/A	RO	RO	RW	RW	RW
<ul style="list-style-type: none"> <li>Firmware</li> <li>Pre-boot &amp; Post-boot Authentication Applications</li> </ul>	Secure PEE (MEE)	Secure PEE (Fireguard)	Persistent (for configuration data)	Temporary Data	User Data

**Figure 5.5 High Grade Secure PESE Test Configuration**

**Test Approach:** A test plan was prepared that contained rigorous test procedures that addressed all of the test categories. A particular emphasis was given to vulnerability

testing to demonstrate the strength of the security mechanisms against cyber-attack, data loss, brute-force attack, physical attack and an under-duress scenario. Table 1 in Paper 9 identifies the respective mechanisms tested to prevent these attacks.

**Test Results:** Extensive and rigorous testing highlighted issues across all test categories, but none were insurmountable. Environmental testing, particularly drop testing resulted in enhancements to the anti-tamper design to ensure false tamper events did not occur.

### 5.4.2 Trial

**Trial Configuration:** A fully configured secure PESE (as shown in Figure 5.6) identical to the test configuration was provided for the trial.

N/A	RO	RO	RW	RW	RW
<ul style="list-style-type: none"> <li>Firmware</li> <li>Pre-boot &amp; Post-boot Authentication Applications</li> </ul>	Secure PEE (MEE)	Secure PEE (Fireguard)	Persistent (for configuration data)	Temporary Data	User Data

**Figure 5.6 High Grade Secure PESE Trial Configuration**

**Trial Approach:** The organisation that agreed to participate in a trial did so strictly on the basis that only limited feedback would be provided to the researcher. A further restriction was placed upon what details could be published. The following is the authorised summary of the trial:

- The trial took place during a simulated scenario.
- Both the secure PEEs installed were tested.
- The secure PESE was utilised at remote nodes (both static and mobile) as part of a NCO deployment.

**Trial Results:** The organisation reported that the trial had been successful and provided feedback on a few features where it would like to see changes or enhancements, e.g. user interface changes were proposed. The trial demonstrated the secure PSES in a net-centric environment where simulated NCO was performed. The trial successfully demonstrated the mode of use proposed in Papers 8 and 9.

### 5.4.3 Certification

**Test Configuration:** The SDV-HA as a secure portable storage device was submitted to ASD for a high assurance evaluation (DSD, 2010). The evaluation was performed in parallel with the design and development to ensure any evaluation issues with the design were identified early and could be addressed. The evaluation configuration is modelled in Figure 5.7 and depicts the SDV-HA as a storage device with a Government accredited version of the Microsoft Windows operating system (Windows, 2013) installed. Fireguard and the MEE did not form part of the evaluation as these artifacts were neither designed as high assurance artifacts nor accredited for use with a high assurance storage device.

N/A	RO	RW
<ul style="list-style-type: none"><li>• Firmware</li><li>• Pre-boot &amp; Post-boot Authentication Applications</li></ul>	Windows OS	User Data

**Figure 5.7 High Grade Secure PESE Certification Configuration**

**Test Approach:** The methodology used by ASD to perform the high assurance evaluation is not published. It is understood that all requirements defined in the high assurance standard (DSD, 2010) are subjected to rigorous examination. The evaluation considered the SDV-HA as storage device and also as a bootable storage device. All design documentation, hardware schematics, software and programmable logic source code, and test plans, procedures and test results were provided for the evaluation.

**Test Results:** The SDV-HA successfully passed the ASD high assurance evaluation and was certified as a high grade storage device approved to protect highly classified information (SDV-HA, 2013). As noted in Chapter 4, at the time of writing the researcher is aware of only one other product (GD, 2014) that has the same level of certification as the SDV-HA. However, this product neither has the equivalent capabilities nor the compact form factor of the SDV-HA.

### 5.4.4 Commercialisation

The Australian Department of Defence established the Priority Industry Capability Innovation Program (PIC IP) to provide funding to support the commercialisation of



innovative technologies that conformed to Defence's priority capability requirements (PICIP, 2014). One of the priority capabilities was anti-tamper technologies. The funding was awarded competitively following assessment of the technology by subject matter experts. The assessment considers if the design is innovative and whether the technology/research artifact makes a knowledge contribution. The researcher submitted an application for the inaugural funding round which was successful and awarded in 2012 (PICIP-SSL, 2012). The PIC IP funding award was an acknowledgement that the SDV-HA made a knowledge contribution.

#### **5.4.5 Summary of Results**

The demonstration activities showed that a high grade secure PESE with sophisticated security functionality, including an anti-tamper capability could be developed that satisfied both the secure PESE concept and functional requirements. The testing, trialling, certification and commercialisation verified the SDV-HA product provided an innovative and very secure but still usable and effective device. The demonstration results corroborated the claims made in Chapter 4 with respect to addressing the research questions. The SDV-HA was commercialised as a secure portable storage product with user documentation provided to show how to configure it as a secure PESE.

### **5.5 Summary**

Both a commercial grade and a high grade secure PESE were demonstrated using a defined and comprehensive process. Each secure PESE was shown to be fit for the purpose for which it was constructed. The commercial grade secure PESE was demonstrated as a secure computing environment for teleworkers. The bootable secure PEE, the MEE when imaged onto a standard USB thumb drive was also shown to be a low cost secure PESE where sensitive data is neither processed nor stored in the remote work environment. The high grade secure PESE was shown to provide the functionality and security to provide a high assurance capability for remote nodes in a net-centric organisation. Secure PESEs have been demonstrated to manage information security risks within the remote work environment.

## **6. Evaluation and Discussion**

### **6.1 Approach**

Chapter 6 presents the thesis evaluation and discussion. In this chapter the research findings are considered, an academic explanation of the research is presented and the contribution made to the area of study is discussed. The research findings are measured through evaluating the demonstration results and appraising how the research questions, objectives and problem were satisfied. As described in the Research Design (Chapter 3) a knowledge contribution framework (Gregor and Hevner, 2013) and an anatomy of a design theory (Gregor and Jones, 2007) are used to both present an academic explanation of the research and show how the body of knowledge made a contribution to the area of study.

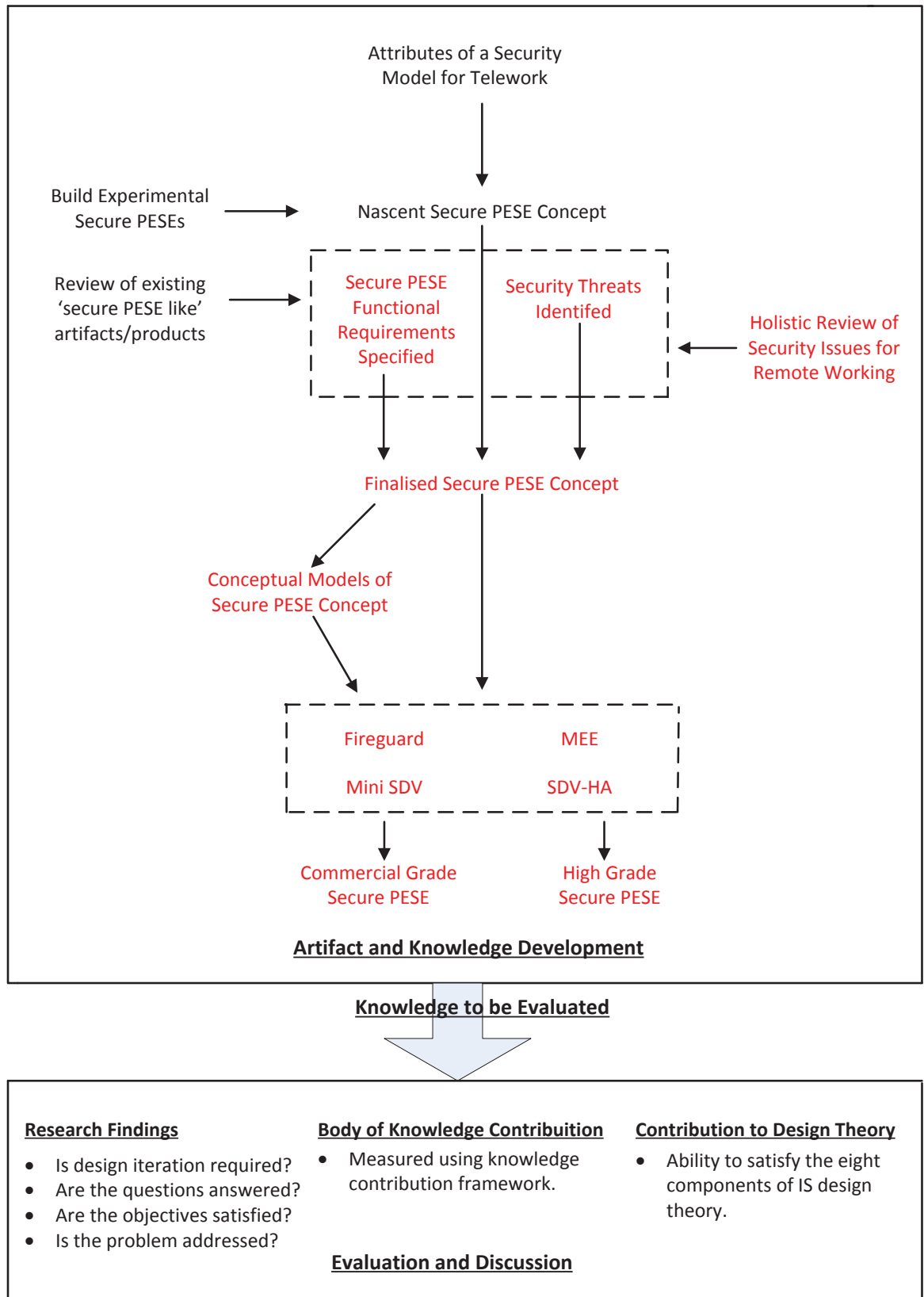
Figure 6.1 diagrammatically models the evolution of the research outcomes by illustrating how an initial secure PESE concept evolved into a formalised concept that directed the design of a set of artifacts. It also summarises diagrammatically the doctoral research described in Chapters 2, 4 and 5 and provides an illustrative synopsis of the research to assist in the presentation of the evaluation and discussion. The evaluated knowledge contributions are highlighted in red. The model also depicts the evaluation of the research findings and the assertion of a contribution to knowledge and design theory.

### **6.2 Evaluation of Research Findings**

The demonstration results were evaluated to determine if iteration was required to refine the research objectives and/or change an artifact's design. The research problem, objectives and questions were specified sequentially in Chapter 2. This evaluation considers each in the inverse order. A further section discusses notable commentary on the research outcomes.

#### **6.2.1 Demonstration Results**

Chapter 5 describes the artifact demonstration process, the application of the process and the demonstration results. A summary of the demonstration outcomes is given in Chapter 5 with an evaluation of the results given below.



**Figure 6.1 - Model Showing the Evolution of the Research Outcomes (especially those representing a knowledge improvement).**

The evaluation determined that the demonstration process adopted (comprising testing, trialling, certification and commercialisation) achieved the desired outcomes. No flaws were identified in the overall process, although further trials of longer duration would have been preferable. The process activities provided the evidence required to demonstrate the research argument that ‘a secure PESE can be used to manage information security risks within the remote work environment’. Specifically the research evaluation highlighted the following:

- **Testing:** Twenty issue reports raised during the testing were not resolved at the end of the research. The issues were mainly suggestions for design improvements and were not operational issues (i.e. not bugs). A lack of available time and resources and, for some proposed changes, the complexity involved underpinned the decision to address certain design issues in a future product release. The ability to capture issues and schedule resolution for a later release supports the ‘artifact mutability component’ of the design theory discussed later in this chapter.
- **Trial:** Whilst the two trials provided good observations, further trials of longer duration may have provided further evidence confirming the usefulness and effectiveness of the secure PESE. In particular, a prolonged trial may have confirmed the ability (or otherwise) of a secure PESE to counter a zero day attack<sup>72</sup>. Evaluation of the user interface recommendations from the high grade secure PESE trial resulted in a design iteration. The changes necessary to implement the user interface recommendations did not result in a major design change as the user interface was implemented in its own self-contained module. Design iteration is discussed later in this chapter.
- **Certification** – No negative observations of the product certification process were identified. As the Mini SDV was based upon the Pocket SDV design, and the SDV-HA design occurred in parallel with its certification, both certifications were completed in a timely manner prior to the finalisation of this thesis.
- **Commercialisation:** The two commercialisation activities chosen to form part of the demonstration process involved independent assessment. The feedback from the

---

<sup>72</sup> A zero day attack exploits vulnerabilities (previously unknown) in a system for which no patch is immediately available.

independent assessors had positive outcomes for the research. Whilst the Mini SDV configured as a secure PESE and positioned as the “Secure Laptop in Your Pocket”, was a finalist in the 2009 Global Security Challenge competition (GSC, 2009) it did not win. Analysis of the competition judge’s comments indicated that the secure PESE was considered an advancement upon secure portable computing devices; however, other entries in the competition provided greater innovation with the winner considered to be an invention. The award of commercialisation funding against strong competition enabled the SDV-HA to be moved from research artifact into a commercially available product. The funding assessors commented upon the innovative anti-tamper and multiple modes of operation of the SDV-HA.

### **6.2.2 Research Questions**

Three research questions were identified. To answer each question, secure PESE artifacts were required that included novel designs. Four novel artifacts (i.e. Fireguard, the MEE, the Mini SDV and the SDV-HA) were designed (using the research questions as design drivers) which were then used to construct two secure PESE artifacts. In Chapter 4 (which describes the artifact design and development) and Chapter 5 (which describes the artifact demonstration), an assessment is given in which a specific artifact design addressed all or part of a research question(s). This section complements the Chapter 4 and 5 assessments by providing a broad analysis of the three research questions following design, development and demonstration of all of the artifacts forming this doctoral research.

*How can a useable and maintainable hardened operating system and/or a small set of hardened applications be developed?* A bootable secure PEE (the MEE) and an application (Fireguard) designed to be part of an up-loadable application-based secure PEE were developed; with published Papers 6 and 7 describing their respective design, demonstration and evaluation. Both the MEE and Fireguard were subjected to extensive hardening and, in the case of the MEE, unnecessary applications (i.e. commands, tools and programs) were removed. Despite the hardening resulting in a constrained and reduced functionality execution environment, the trials showed that the users of both the MEE and Fireguard raised no usability concerns.

Whilst hardening can result in an execution environment that is resistant to a range of attacks, a COTS application is likely to be difficult to harden, as access to configuration files and source code may be prohibited by the vendor. A further limitation is whenever a new application is required to be added to a secure PEE it will require hardening. Assuming it is possible to harden the application it may prove a time consuming activity<sup>73</sup>. An approach worthy of further investigation is to use a 'sandbox jail' (McFearn, 2011) to allow a difficult to harden COTS application to form part of a hardened secure PEE. Through a 'sandbox jail' an unhardened COTS application is executed within a restricted file-system namespace, which protects the secure PEE and remote worker's data against any malware.

The maintainability of a secure PEE, in particular the automatic application of updates (i.e. patches and/or new releases) was identified as an issue in both Papers 6 and 7 as it contradicts the 'hardening ethos'. In particular, the MEE and Fireguard were specifically designed to prevent automatic updates due to concerns that malware could be introduced (Cnet News, 2009) and that a vendor issued patch/release would be incompatible and may 'un-harden' the software. Protecting the secure PEE in a Read-Only partition also impacts maintainability. Whilst a Read-Only partition protects the integrity of the secure PEE, from both accidental user actions and external and internal deliberate attacks, updates cannot be applied unless administrator privilege is used; secure PEEs have a least privilege design.

Papers 6 (Fireguard) and 7 (the MEE) proposed the development of a tool to allow updates to be performed in the remote work location; such a capability is consistent with best practice strategies to mitigate cyber intrusions (DSD, 2012). The secure PESE 'hardening ethos' prevents vendor issued automatic updates and therefore a capability managed by the (remote work) organisation is required; however, the following challenges to providing such a capability were identified:

1. If the tool performed automatic updates then it would have to execute as a privileged daemon process<sup>74</sup> to change the partition access control to Read-Write during an

---

<sup>73</sup> It took approximately 60 work days of effort to harden and extensively system test the Fireguard application.

<sup>74</sup> A daemon process is an application that executes as a background process and not under the control of a specific user.

update. Allowing the partition (where the secure PEE is held) to be written to during operational use could result in accidental or deliberate corruption of the secure PEE, e.g. a malware attack occurs and/or the user accidentally writes to the partition whilst an automatic update is being performed. A failure of the tool during an update could also leave the partition in a Read-Write state.

2. Providing the user with the privilege to make updates could result in accidental or deliberate corruption of the secure PEE. Assigning privilege to a user would breach the least privilege design principle.

The most secure approach to performing updates is within an off-line, trusted and controlled environment, e.g. the secure PEE on a secure PESE could be updated by the IT department/team in a corporate office. The arguments in favour of this approach are:

1. The secure PEE itself is hardened and provides only a small attack surface (as all unnecessary functionality is removed or disabled), thereby reducing the opportunities for attack.
2. The secure PESE Read-Only partition provides protection for the secure PEE against exploitation of a vulnerability until an update can be applied. It is possible introduced malware could exploit a single session as the partition allows writes to occur, but any written malware will disappear upon power-down of the secure PESE.
3. The MEE was designed as a locked down image that could only be updated through complete re-imaging, so it is difficult for malware to become embedded into the image.

This approach has the obvious limitation of not keeping a secure PEE immediately up to date with patches/releases that address new vulnerabilities; and is therefore not implementing best practice strategies to mitigate cyber intrusions (DSD, 2012). The approach is likely to be resistant to application layer malware attacks as hardening limits exploitation; however, it is less likely to prevent operating system kernel attacks, e.g. in the MEE the introduction of a vulnerability like the 'Heartbleed' bug (Cyberoam, 2014), which attacks the kernel's core secure communications functionality, could exploit a (single) remote work session as such core functionality is not hardened. The identification of operating system kernel malware would require an organisation to recall secure PESEs

for updates as soon as possible. Given the lack of an automatic remote work update capability it could be argued that the research question was not satisfactorily addressed. However, the researcher asserts that the Read-Only partition, the hardening and the locked down image provide a sufficiently secure solution until a secure PESE is updated in an off-line, trusted and controlled environment.

*How can a useable and maintainable execution environment be configured to store all temporary data on a secure PESE partition?* The hardening of the MEE and Fireguard included making changes to ensure all temporary created data was written to a specific partition on the secure PESE. The trials identified no usability or performance issues with temporary data being held on the secure PESE.

Whilst it is possible to ensure that a bespoke or open source system/application stores all temporary data on the secure PESE, a COTS system/application may be more challenging. Some COTS vendors publish details on the storage location used for temporary data; however, others do not. To determine if a COTS system/application would operate successfully when temporary data was written to a specific partition, the Microsoft Windows operating system (Windows, 2013) was installed into a secure PESE Read-Only partition on a secure PESE. The Windows configuration parameters pointing to temporary file locations were changed to the allocated temporary data Read-Write partition, e.g. the page swapping system file 'pagefile.sys' was located on the Read-Write partition. Following a degree of trial and error (to ensure all temporary data files were moved out of the Read-Only partition to the Read-Write partition), the Windows operating system was shown to work successfully over a prolonged period. The use of Microsoft Windows also demonstrated how a COTS operating system can be used as a secure PEE, albeit one not conforming to the hardening and reduced functionality requirements.

The research question also sought to determine if a secure PEE, supporting a temporary data partition, was maintainable. The provision of the temporary data partition does not affect the maintainability, but lack of an automatic update capability could. Therefore given the lack of an automatic update facility that can be used in the remote work location it could be argued that the research question was not satisfactorily addressed. However,



the researcher contends that for the reasons given above, there is no need for updates to a secure PEE to be performed in the remote work environment.

*How can anti-tamper mechanisms be implemented into a small form factor and highly portable device?* Anti-tamper mechanisms were designed for both the Mini SDV (the platform for the commercial grade secure PESE) and the SDV-HA (the platform for the high grade secure PESE). Both the Mini SDV and the SDV-HA are compact and highly portable devices.

The Mini SDV's anti-tamper design provided a level of protection commensurate with its intended use of protecting sensitive but not highly classified data. The Mini SDVs anti-tamper mechanism ensures data is inaccessible if attempts are made to dismantle the device.

The SDV-HA implements an anti-tamper design based upon an active tamper mesh. This design is novel as it incorporates such a mesh within a small form factor device. The SDV-HA is able to withstand a range of tamper attacks including physical, electrical, tampering with components, and environmental and operational conditions (if a condition falls below or exceeds a defined value). A tamper detection event will cause the encryption keys to be destroyed, rendering the device inoperable.

The researcher considers the question was fully addressed.

### **6.2.3 Satisfying the Research Objectives**

Nine research objectives were specified; seven were directly derived (without change) from the attributes defined for the secure PESE concept and the other two objectives were stakeholder inspired. Each objective is evaluated to determine if it is satisfied by the research.

*Prevent unauthorised access to a secure execution environment and any stored data:* Both secure PESEs implemented authentication and access controls to prevent authorised to the installed secure PEE(s) and stored data. The high grade secure PESE implemented two-factor authentication which afforded a high level of assurance in protecting the secure PEE and stored data.

*Preserve the integrity and availability of the execution environment:* The integrity (and as a consequence the availability) of a secure PEE was protected by a Read-Only partition which prevented accidental or deliberate corruption. Other measures implemented to preserve the integrity and availability of the secure PEE included:

- Hardening to reduce the opportunity to compromise the secure PEE.
- Incorporation of anti-malware and firewall applications as part of the limited application set.
- In the unlikely event the secure PEE image is corrupted, it can be readily replaced with a new image (in an off-line, trusted and controlled environment).

*Preserve the confidentiality and integrity of any stored data:* All data written to a secure PESE is encrypted to preserve the confidentiality and integrity of stored data. The MEE implements the strongest currently available wireless network protocols to protect the confidentiality of data on a wireless local area network. Anti-tamper mechanisms prevent an attacker subverting a secure PESE, e.g. if an attacker attempted to embed a hostile mechanism that allows the encryption mechanism to be bypassed, a tamper event will occur rendering the secure PESE inoperable.

*Preserve the confidentiality and integrity of any data sent to/from the remote location:* The MEE provides a secure browser and secure remote access client that implemented network encryption thus enabling the secure PESE to be a secure network client.

*Provide a highly portable device with a user friendly execution environment that can be used on any available PC:* Both the commercial grade and high grade secure PESEs were designed to be portable with small convenient form factors. Compatibility testing confirmed the secure PESEs executed with a wide range of x86 PCs. The MEE and Fireguard had user friendly interfaces confirmed through independent usability testing and the user trials.

*Provide an execution environment with only the necessary secured functionality for the specific remote work activities:* Both the MEE and Fireguard were hardened with all unnecessary functionality disabled or removed.

*Limit the execution environment's access to the internal storage device(s) of the host PC:* The MEE prevents a user accessing the internal disk drive of the host PC. Fireguard requires the user to specify the storage location for any downloaded files thus ensuring the default browser location is not used. Any temporary data created by a secure PEE is stored in a Read-Write partition on the secure PESE.

*The execution environment shall have a look and feel similar to Microsoft Windows:* The MEE desktop configuration had a look and feel similar to Microsoft Windows; refer to Paper 7 for a graphic image of the MEE's desktop.

*The execution environment shall have similar performance to a PC operating system executing from an internal disk drive:* Testing and trialling of both the MEE and Fireguard showed performance to be comparable with the operating system and applications executing from the host PC's internal disk drive.

The researcher considers all of the objectives were satisfactorily addressed by the research.

#### **6.2.4 Research Problem**

**Background:** Following the identification of an initial secure PESE concept and the review of the security issues in the remote work environment (described in Chapter 2), a set of secure PESE functional requirements was specified and the attributes of the concept were finalised. An artifact that satisfied the concept attributes would be considered to be a fully compliant secure PESE. The functional requirements were allocated to the concept attributes to identify the functionality to be implemented to produce a fully compliant secure PESE. Existing 'secure PESE like' artifacts and products were assessed against the functional requirements and it was identified no existing artifact/product implemented the full set of requirements, hence a gap was identified and the following research problem was defined:

**A requirement exists to develop an enhanced secure PESE that limits the exploitation of vulnerabilities by hardening the execution environment, providing a tamper detection and response capability and ensuring no data remnants are recoverable from the host PC.**

***Satisfying the Problem:*** As described in Chapters 4 and 5, the problem was addressed through the construction of two secure PESEs. The research has demonstrated that the constructed secure PESEs addressed the problem through the following outcomes:

- The secure PEEs (i.e. the MEE and Fireguard) were hardened, reduced functionality artifacts.
- The secure portable storage devices utilised the SDV technology which was enhanced to include anti-tamper mechanisms.
- The two secure PESEs were configured with a storage area for temporary data and the MEE and Fireguard were configured so that any temporary data created by either artifact did not get stored on the host PC's internal disk drive.

The demonstration showed that through testing, trialling, certification and commercialisation the secure PESEs were usable, secure, innovative knowledge contributions. The researcher considers the problem was satisfactorily addressed.

#### **6.2.5 Further Observations on Research Outcomes**

The following additional evaluation observations provide further clarification on the research and its relevance.

***Iteration during the Research:*** Design iteration during the research can occur following external observations (e.g. feedback from participants in a trial) or internal observations (e.g. comments made by colleagues, PhD supervisor or paper co-author). The structured peer review process (forming part of the development environment described in chapter 3) was a key driver of internal iteration. The review process was used with colleagues and paper co-authors at defined research milestones, as it enabled design flaws and optimisations to be identified at an earlier stage, saving time and resources. Internal iteration also contributed to the development of the secure PESE concept as it allowed definition and refinement of the concept's attributes through the many design reviews.

The DSRM (Peffers et al., 2007) methodology allows for iteration during the evaluation and communication activities (as modelled in Figure 3.1) which accommodated any external observations made about the research outcomes. Evaluation of the (external) received observations from the high grade secure PESE trial resulted in iteration of the

user interface design (as outlined earlier in this section). Fortunately, there was no other design iteration<sup>75</sup> required which was probably attributable to the on-going internal iteration performed during the design and demonstration activities.

**Leaving the MEE and Fireguard as Research Artifacts:** Neither the MEE nor Fireguard were certified or commercialised for the following reasons:

- **Secure PEEs were constructed as demonstrators:** The intent of the secure PEE artifacts was to demonstrate research ideas; commercialisation of these artifacts was not a primary objective. The secure PESE concept allows a range of technologies to be assembled to form a (fully or partially compliant) secure PESE and therefore alternative systems/applications can be used as secure PEEs. It was decided to leave the MEE and Fireguard as research artifacts that could be made available (as required) as proof of concept demonstrators.
- **Flexibility to use other secure PEEs:** As discussed above, Microsoft Windows was used as a secure PEE to demonstrate how a COTS operating system could be configured to ensure all created temporary data is stored in a partition on the secure PESE. The use of Microsoft Windows demonstrated a partially compliant secure PESE; as it was not a hardened secure PEE it was obviously not fully compliant. This demonstration showed that alternative systems could be used to partially satisfy the secure PESE concept.
- **Adding applications to a secure PEE:** The MEE (including Fireguard) does provide a thin client remote access capability (as discussed in Papers 7 and 10) that can be used without change where remote work applications execute on a server. Using the MEE as a thin client configuration can be a low cost secure solution. It is argued in this thesis that a secure PEE can be specifically configured and packaged with hardened applications for a particular remote work scenario. Such a scenario may require one of more general purpose or specialist applications to be packaged with the MEE. Therefore, commercialising the MEE (and Fireguard) as a secure thin client remote access solution would only address one particular requirement. It was decided to offer the MEE and Fireguard to interested parties (e.g. customers of the Mini SDV and the SDV-HA) as demonstrators that could, if required, be tailored to their needs.

---

<sup>75</sup> Design iteration following evaluation of research findings can result in delayed research outcomes as it may lead to artifact re-design.

- ***Use of open source software:*** To enable the exploitation of existing innovations and facilitate rapid development, the secure PEEs used open source software. Such software is not always suited to the security certification criteria due to the lack of design documentation (as highlighted in Papers 6 and 7). It is feasible that the MEE and Fireguard could be certified for commercial grade use, but it is unlikely that they could be certified as part of a high grade secure PESE, e.g. for use in a cross domain solution<sup>76</sup>.
- ***Certifying and re-certifying a secure PEE:*** Even if the MEE and/or Fireguard were commercialised and could be certified the costs involved could be prohibitive. Secure PEE certification would only be for the configuration submitted for certification (CC, 2014; DSD, 2010). Applying patches or the upgrade to a new release, or the addition of applications would require costly re-certification. It is the researcher's experience<sup>77</sup> that the certification cost and then the re-certification to account for changes has prevented many organisations from certifying execution environments.

***Market Trends for Secure Portable Computing Devices for Remote Working:*** When the research commenced (in 2007) the portable computing devices used for remote working were mainly laptops and to a much lesser extent smartphones. During the period of the research there has been a prolific increase in portable computing devices particularly for mobile working (Dade, 2013). It is predominately tablets and smartphones that have seen the greatest growth (tti, 2014; iOS, 2014; Android, 2014), although devices that conform (fully or partially) to the secure PESE concept have also continued to emerge.

The introduction by Microsoft of Windows To Go (Win2Go, 2014) has facilitated the development of products that could partially satisfy the secure PESE concept. Windows To Go is a feature introduced in Windows 8 (Windows, 2013) that allows Windows 8 to be portable between different x86 PCs as it boots and runs from a USB drive. Windows To Go is the first truly portable Microsoft operating system. This portable operating system supports the same features as Windows 8 Enterprise with a few differences including not allowing access to the internal disk drive of the host PC. Microsoft has certified several

---

<sup>76</sup> A cross domain system provides assurance that a system that can access different security domains does not allow the highly classified data in one domain to be transferred to another domain with a low classification.

<sup>77</sup> The researcher established the first ASD accredited Common Criteria assessment security facility in Australia.

products that combine secure portable storage devices with Windows To Go to provide 'secure PESE like' solutions (Seltzer, 2013; Spyryus, 2014; Ironkey, 2014). A Windows To Go image was installed on to a SDV-HA in a Read-Only partition, i.e. the SDV-HA was configured in System mode (refer to Paper 10 for details on System mode) using Windows To Go as the secure PEE. The page swapping file was installed on the Read-Write partition. This partially compliant secure PESE was tested using a small subset of the tests from a secure PESE test plan and was found to operate without issues.

### **6.3 Contribution to Design Knowledge and the Area of Study**

The knowledge produced in each design cycle (summarised in the synopsis of an artifact's description) collectively forms the doctoral research body of knowledge. For each of the four design cycles the knowledge produced is enumerated below. A knowledge contribution framework (Gregor and Hevner, 2013) is used to measure the contribution that the body of knowledge has made to the area of study.

#### **6.3.1 Summary of Knowledge Contributions**

Each knowledge contribution within each cycle is categorised as prescriptive or descriptive knowledge, with prescriptive knowledge further categorised as a construct, model, method or instantiation. The prescriptive knowledge is further categorised to aid the identification of knowledge used to assert a design theory for secure PESEs.

The presentation of outcomes using design cycles was an approach identified towards the end of the research. The approach broadly reflected how the research was performed and also provided a structure for compartmentalising design activities that at the time they were performed appeared to be disparate, e.g. the early design investigations described in Papers 1, 2, 3 and 4. These investigations provided a knowledge contribution and direction for the research and were presented logically as part of design cycle 1. Similarly the research into virtual machine vulnerabilities (described in Paper 5) formed part of design cycle 2 as it contributed to confirming the design approach for secure PEEs. The secure PESE concept, which was not presented in a published paper as it was evolving as the research progressed and was only recognised as an important outcome in the latter years, was able to be presented in design cycle 1 where it was initially formulated.

#### **6.3.1.1 Design Cycle 1 - Establishing the Research Problem and Objectives**

The knowledge outcomes from this cycle were:

- A conceptual design of a security module for a smartphone (prescriptive knowledge – method).
- A toolkit for a secure portable disk drive that prevents data remnants (prescriptive knowledge – instantiation).
- The identification of the attributes for a teleworking security model (prescriptive knowledge – construct).
- A set of experimental secure PESEs (prescriptive knowledge – instantiation).
- A comprehensive description of the remote work security issues and the respective threats (descriptive knowledge).
- Functional requirements for a secure PESE (prescriptive knowledge – method).
- The secure PESE concept and its attributes (prescriptive knowledge – construct and model).

#### **6.3.1.2 Design Cycle 2 – Baselining the Design**

The knowledge outcomes from this cycle were:

- A comprehensive description of the virtualisation security vulnerabilities and possible countermeasures (descriptive knowledge).
- Conceptual design models of the secure PESE concept (prescriptive knowledge – model).

#### **6.3.1.3 Design Cycle 3 – Commercial Grade Secure PESE**

The knowledge outcomes from this cycle were:

- Fireguard - a secure browser artifact that is an application within a secure PEE based up-loadable application set (prescriptive knowledge – instantiation).
- The Mobile Execution Environment (MEE) - a secure PEE artifact (prescriptive knowledge – instantiation).
- The Mini-SDV - a secure storage artifact (prescriptive knowledge – instantiation).



- A commercial grade secure PESE (prescriptive knowledge – instantiation) constructed from the three aforementioned artifacts produced in this cycle.
- The empirical data obtained from the trial of the commercial grade secure PESE by teleworkers for transaction-oriented work (descriptive knowledge).

#### **6.3.1.4 Design Cycle 4 - High Grade Secure PESE**

The knowledge outcomes from this cycle were:

- An understanding of how a secure PESE could support a theory for Network Centric Warfare (descriptive knowledge).
- The SDV-HA – a high grade portable storage device artifact (prescriptive knowledge – instantiation), including novel anti-tamper and multi-modes of operation mechanisms.
- A high grade secure PESE (prescriptive knowledge – instantiation) constructed from the two secure PEE artifacts produced in design cycle 3 and the SDV-HA.
- The empirical data obtained from the trial of the high grade secure PESEs for deployed working with an organisation supporting NCO (descriptive knowledge).

### **6.3.2 Classification of Knowledge Contribution**

#### **6.3.2.1 Knowledge Considered**

As shown above the four design cycles delivered a comprehensive body of knowledge in the research area of secure portable computing devices for remote working. The key knowledge contributions (from the aforementioned design cycles) that are inputs into the knowledge contribution framework (Gregor and Hevner, 2013) are summarised below:

- A holistic review of the security issues in the remote work environment and the identification of the respective information security threats.
- A set of secure PESE functional requirements.
- A secure PESE concept.
- Conceptual design models of the secure PESE concept consisting of a threat model, design model and operational model.
- Secure PEE artifacts – the MEE and Fireguard.
- Secure portable storage artifacts – the Mini SDV and the SDV-HA.

- Commercial grade secure PESE artifact.
- High grade secure PESE artifact.

It is the evolution of these key knowledge contributions and their respective relationships that are highlighted in red and shown diagrammatically in Figure 6.1.

### 6.3.2.2 Classifying the Knowledge Contribution

The Gregor and Hevner knowledge contribution framework (described and modelled in Figure 3.4) classifies new knowledge as one of the following:

- Improvement – the development of new solutions for known problems.
- Exaptation – the application of an existing solution to a new problem.
- Invention – the development of a new solution for a new problem.

The researcher considers the doctoral research to be an improvement to the area of study. Information security is a known problem for remote working and the researcher asserts that secure PESEs have been constructed that are an improvement over existing solutions. To be considered an improvement, the contribution of the body of knowledge should result in a more efficient and effective solution than currently available. The improved solution must clearly demonstrate genuine advances upon previous solutions with the new knowledge grounded in design theories (justificatory knowledge). The research should also explain how and why the new solution differs from current solutions.

### 6.3.2.3 Evidence of Improvement

The following rationale supports the claim that the knowledge contribution is an improvement:

***More efficient and effective solution than previously available:*** The two constructed secure PESE artifacts are new improved solutions to enable secure remote working. As the literature analysis showed, information security in the remote work environment is a known problem (Sturgeon, 1996; Gosler, 2000; GSA 2002; Whiteman and Dick, 2006) that has existed since the commencement of telework (Clear, 2007; Joice, 2007). Whilst the Internet has facilitated an increase in remote working it has also introduced cyber-attacks (Gibson et al., 2002; Pearcey 2006; Bates, 2010) that significantly increased the

information security risks (Harris, 2010; Deloitte, 2011). The research has shown that a secure PESE can provide a secure solution for remote working. 'Secure PESE like' devices have been used for remote working prior to this doctoral research (as the literature review discussed). However, no existing 'secure PESE like' devices satisfied the full set of secure PESE functional requirements specified following the review of security issues in the remote work environment. The constructed commercial grade and high grade secure PESEs addressed all of the functional requirements providing a more secure solution to allow efficient and effective remote working.

***A genuine advance over previous solutions:*** The doctoral research contributed to the advancement of secure portable computing devices for use in remote working through:

- A well-defined concept, represented by a set of informal constructs and a model that enables analysis of remote work security.
- A documented holistic review of the security issues in the remote work environment that led to the identification of information security threats and the specification of a set of secure PESE functional requirements.
- The production of conceptual models for the secure PESE concept.
- The implementation of the functional requirements for the secure PESE to produce a set of artifacts that were configured into two secure PESE artifacts.

The research is considered to be a genuine advance to the area of study as it has formalised a method for secure remote working and provided new solutions with enhanced security. The secure PESE concept provides a methodology for either analysing the suitability of 'secure PESE like' devices for remote work or for constructing a partially or fully compliant solution. The security review and the identified threats provide a comprehensive definition that complements the less detailed research previously performed (Sturgeon, 1996; Clear, 2007; Pyoria, 2011; Ampomah et al., 2013). The conceptual models provide reference and design resources to support implementation. The constructed artifacts include novel features not available in previous artifacts/products.

***Knowledge grounded in theory (justificatory knowledge):*** The underlying theories and knowledge grounding the research were sourced from established and recognised

publications in conceptual design, information security threat and risk assessment, security systems design, cryptographic systems engineering and operating systems design. The key publications used and referenced as justificatory knowledge in this thesis are listed below. To justify each publication's status as a source of theory/knowledge the respective number of citations identified in Google Scholar<sup>78</sup> as of January 2015 is given; whilst Google Scholar lacks the academic reputation of bibliographic databases like Scopus<sup>79</sup> and Web of Science<sup>80</sup> it captures citations across a broader range of publications (including books, patents and presentations) that are not captured in other databases, and therefore provides a rigorous source to support justificatory knowledge claims:

- Security Engineering (Anderson, 2008) – 2,024 citations.
- Specification and Design of Embedded Systems (Gajski et al., 1994) – 794 citations.
- Information Systems Security Design Methods (Baskerville, 1993) – 438 citations.
- Applied Cryptography (Schneier, 2007) – 12,815 citations.
- Modern Operating Systems (Tanenbaum, 2009) – 3,429 citations.
- Cryptographic Engineering (Ferguson et al., 2011) – 167 citations.
- The Security Risk Assessment Book (Landoll, 2005) – 116 citations.

Other specialist theories and justificatory knowledge were used and are cited where appropriate in a designed artifact's consumed knowledge section.

***How and why the new solutions differ from current solutions:*** The two secure PESEs differ to other previous solutions through the implementation of hardened secure PEEs, multiple modes of operation and active anti-tamper mechanisms.

### 6.3.3 Summary

The researcher considers he has used the knowledge contribution framework to assert that the constructs, models, method and instantiations forming the secure PESE body of knowledge have made a contribution to the area of study.

---

<sup>78</sup> Google Scholar is a search engine for accessing scholarly literature.

<sup>79</sup> Scopus is a bibliographic database owned and maintained by Elsevier.

<sup>80</sup> Web of Science is a bibliographic database owned and maintained by Thomson Reuters.

## 6.4 Contribution to Design Theory

It is asserted that the doctoral research makes a contribution to design theory. The Anatomy of a IS Design Theory (AISDT), proposed by Gregor and Jones (Gregory and Jones, 2007), is used to assert a design theory for secure PESEs and their use to improve remote work information security. The AISDT has eight components and, to claim a contribution to design theory, it is necessary that the research addresses each of the eight components. As an artifact can be intangible (i.e. a novel construct, model or method) and/or tangible (i.e. an instantiation incorporating novel design) the AISDT allows a theory to be claimed for the underlying principles of an artifact and/or the act of implementation. The asserted secure PESE design theory is underlined by both intangible and tangible knowledge, i.e. the doctoral research has produced construct, models, methods and instantiations.

Chapter 3 gives an overview of the eight components of the AISDT. The application of the AISDT to the secure PESE body of knowledge adopts the following structure: for each of the eight components a more detailed description (than given in Chapter 3) of the theory requirement is specified and the secure PESE knowledge that supports a secure PESE theory is described.

### 6.4.1 Purpose and Scope

**Theory requirement:** This component of the AISDT defines the boundaries of the asserted design theory. The component requires a definition of what the research aimed to achieve. The definition acts as a set of meta-requirements (i.e. a set of generalised requirements that allow not just one but a class of artifacts to be constructed).

**Secure PESE Theory:** The secure PESE concept attributes provide the purpose and scope for the design theory and define a generalised concept that allows a range of different but conforming secure PESEs to be constructed. The concept attributes were assigned as the research objectives (augmented by a further two stakeholder inspired objectives) and therefore provided the direction and boundaries for the research.

## 6.4.2 Constructs

**Theory requirement:** The AISDT constructs are the basic entities of a theory. A construct may be physical or abstract. The constructs afford a vocabulary or notation to the design theory.

**Secure PESE Theory:** The constructs in the secure PESE design theory can be expressed as both abstract and physical. The abstract constructs are defined within the secure PESE concept and are represented by the terms: “preventing unauthorised access”, “preserve confidentiality”, “preserve integrity”, “availability”, “portability”, and “execution environment”. The physical constructs are defined in the two secure PESE instantiations and are represented by the terms: “secure encrypted storage”, “secure partitioning”, “authentication”, “access controls” and “secure PEE”.

## 6.4.3 Principles of Form and Function

**Theory requirement:** This component refers to the structure, organisation and functioning principles of an artifact and captures the properties and features of the artifact. The component can be addressed through an abstract model or architecture that shapes the design (the principles of form), or through an algorithm or a specification that instructs the development (the principles of function). Prescriptive knowledge in the form of models and methods satisfy this theory requirement.

**Secure PESE Theory:** The three conceptual models defined in Chapter 4 describe the principles of form. The threat model (Figure 4.1) encapsulates the security properties, the conceptual design model (Figure 4.2) presents the structure and organisation, and the operational model (Figure 4.3) depicts the concept of operation. The secure PESE functional requirements describe the principles of function as they provide the instructions for development.

## 6.4.4 Artifact Mutability

**Theory requirement:** Artifacts can be subject to on-going design change and improvement. The artifact mutability component defines the degree of artifact change encompassed and supported in the theory. Artifact mutability can be expressed in the

form of proposed future work to address limitations or subjecting the artifact to a continuous improvement cycle.

**Secure PESE Theory:** Artifact mutability is supported by the secure PESE concept and the research test environment which assist continuous improvement, specifically:

- A stated goal of the secure PESE concept is to enable its constructs and model to facilitate a mutable approach to develop or qualify fully or partially compliant secure PESEs.
- The research demonstration results, in particular the test results, identified several design improvements that were captured in an issue management system to enable continuous design improvement. Whilst some proposed improvements were implemented others were left for future releases.

#### 6.4.5 Testable Propositions

**Theory requirement:** The AISDT defines testable propositions as the truth statements about the design theory. For a formal mathematically specified model (Lindsay, 1998) the test propositions will be mathematical proofs, for a method (e.g. algorithm) the test propositions will be an implementation and for an instantiation the test propositions are the repeatable set of test procedures.

**Secure PESE Theory:** As the secure PESE is represented by hypothesis (research argument), model, method and instantiations, the test propositions are stratified. At the top of the test hierarchy is the research argument which provides a single test proposition. The mid-level test propositions are the secure PESE concept attributes which provide a testable model of security properties. In Chapter 2 the secure PESE functional requirements were allocated to each attribute and these requirements (a method) are testable statements. The test procedures that test each functional requirement provide the lowest level of test propositions that could be applied to a secure PESE instantiation.

#### 6.4.6 Justificatory Knowledge

**Theory requirement:** This component has been defined as formal justificatory knowledge in Chapter 3 and is used throughout this thesis, so the definition is not repeated here. The

AISDT requires that an assertion of a contribution to design theory has underlying justificatory knowledge/theories to support the asserted theory.

**Secure PESE Theory:** It has been shown in this doctoral research that the knowledge contribution is underpinned by justificatory knowledge. The theories used have been enumerated above and are also identified throughout the thesis where specific theories inform prescriptive knowledge or an artifact's design.

#### **6.4.7 Implementation Principles**

**Theory requirement:** The implementation principles are concerned with how a design is transformed into a solution. This component can be satisfied by a policy document, an implementation specification, a design description or configuration guideline depending upon the type of artifact to be implemented.

**Secure PESE Theory:** The development environment used (refer to Chapter 3) requires documentation to be developed at each design/development phase (refer to Figure 3.2) and therefore comprehensive and detailed documentation encapsulating the full implementation principles for the two constructed secure PESEs. This documentation is proprietary to Secure Systems but does provide supporting evidence that the researcher's employer has comprehensive documentation to satisfy this AISDT component.

The implementation principles are also contained in the published papers describing the designs of Fireguard and the MEE. The Fireguard and MEE descriptions (Papers 6 and 7) provide sufficient design and configuration details to enable the reconstruction of the artifacts.

#### **6.4.8 Expository Instantiation**

**Theory requirement:** The AISDT considers that a design theory should be translatable into an instantiation as expository evidence of its correctness. The AISDT considers a design theory as an abstract expression of ideas, while an instantiation that implements the ideas verifies the theory.

**Secure PESE Theory:** The research effected the design and construction of two secure PEE artifacts and two secure portable storage artifacts. These artifacts enabled the creation of



a commercial grade secure PESE and high grade secure PESE demonstrating expository instantiation.

#### **6.4.9 Summary**

The researcher believes the doctoral research has addressed the eight components of the AISDT and therefore asserts a design theory for secure PESEs has been defined.

### **6.5 DSR Contribution Type**

Gregor and Hevner developed a three level knowledge maturity model to gauge the completeness and progression of a DSR knowledge contribution. (Gregor and Hevner, 2013). The maturity model is described in Chapter 3 with the model presented diagrammatically in Figure 3.5. The model defines a level three contribution (the most abstract) as a well-developed design theory, a level two contribution as a nascent design theory and a level one contribution (the least abstract) as a situated artifact implementation. A DSR research project can produce artifacts on one or more of the three levels.

The doctoral research has produced instantiations (a level one contribution type) and constructs, models and methods (a level two contribution type). The knowledge contribution framework has been used to show a knowledge improvement has been made and the AISDT has been used to assert a contribution to design theory. The researcher therefore believes that the doctoral research is a level 2 contribution type, i.e. a nascent design theory underpinned by a secure PESE concept, design models and two instantiations.

### **6.6 Summary**

This evaluation and discussion chapter has shown that the research satisfied the research questions, objectives and problem. Using a knowledge contribution framework, an anatomy of IS design theory and a knowledge maturity model, the body of knowledge developed is asserted to be an improvement to the area of study and a nascent design theory is claimed. The research has created artifacts in the form of constructs, models, method and instantiations to demonstrate that the research argument has been satisfied.

## 7. Conclusion

This concluding chapter considers if the research had impact and was successful. The originality, significance and limitations of the research are also discussed and the suitability of DSR as the research paradigm is reviewed. Finally suggestions are given for future work.

### 7.1 Impact and Success of Research

The impact and success of research can be measured by reflecting upon its original aspirations which were encompassed in the motivating goals, and research problem and objectives. In Chapter 6 it was shown that the research problem and objectives were successfully addressed. Using each of the following motivating goals (defined in Chapter 1) an appraisal is performed to gauge if the research outcomes provide sufficient evidence to fulfil the original motivation for its conduct.

***Improve Remote Work Security:*** The goal was to contribute to research into remote work information security risk management. In Chapter 6 it was shown that the research made a knowledge contribution. Part of the contribution relevant to this goal was a comprehensive identification of the security issues and the respective information security threats. Countermeasures to these threats, in the form of a set of secure PESE functional requirements, were defined which when fully implemented would improve the capability for managing the information security risks in the remote work environment.

***Increase Security Awareness:*** The approach adopted to achieve this goal was to conduct the doctoral research as a PhD by publication. The preparation and publication of research papers facilitated a venue to increase the awareness of the remote working information security risks. One measure of research impact is the citation of the published papers by other researchers; the thesis (in Chapter 2, 4 and 5) identifies material that cites the published papers.

***Develop New Solution:*** The goal was to develop a secure computing capability to improve information security in remote work locations. The research defined the secure PESE concept and a set of functional requirements which enabled two secure PESEs to be constructed. The secure PESEs contained innovative security features not available in

existing secure portable computing artifacts/products used for remote working. The testing and trialling of the two secure PESEs (discussed in Chapter 5) demonstrated an enhanced secure solution had been constructed for remote working.

***Increase Remote Work:*** The goal sought to increase remote working by providing a body of knowledge and demonstrable artifacts that would be used to both improve existing security and enable an organisation vetoing the practice (due to security concerns) to reconsider. It has been shown in this thesis that the research made a contribution to the area of study and generated secure PESEs that were successfully trialled in remote work scenarios. The teleworking trial (described in Chapter 5) demonstrated how an organisation with an established teleworking team used the commercial grade secure PESE to improve security. The deployed working trial (described in Chapter 5) provided a successful example of the use of the high grade secure PESE by an organisation that had previously vetoed the processing of sensitive information remotely.

***Commercialise Research:*** A motivator for the researcher was to use the academic research to support his employer's product development which was successfully achieved with the commercialisation of the Mini SDV and SDV-HA secure portable storage products.

***Summary:*** The research can be claimed to be successful as it achieved its motivating goals. It has been shown that the research has had an impact as:

- The published papers have been cited in other research.
- The secure PESEs were successfully trialled by organisations which subsequently adopted them for remote working.
- The secure portable storage technology developed during the research has been commercialised.

## **7.2 Originality and Significance of Research**

The originality and significance of the research are exhibited through both the knowledge contribution to the area of study and the assertion of a nascent design theory for secure PESEs.

The broad and structured approach adopted facilitated a body of knowledge contributing an improvement to the area of study by:

1. The comprehensive specification of the remote work security issues, threats and functional requirements for a secure PESE.
2. The definition of a concept for secure PESEs.
3. The design and development of novel artifacts enabling the construction and configuration of two secure PESEs.

For DSR to have advanced design theory, the research must have the following properties: a generalised definition; a vocabulary/notation; a model and/or method; the flexibility to encompass improvements; testable propositions; a theoretical knowledge base; a documented basis to enable implementation; and one or more working instantiations (Gregor and Jones, 2007). It has been shown in Chapter 6 that the doctoral research furnishes these properties and therefore a nascent design theory for secure PESEs for use in remote work has been asserted.

### **7.3 Limitations of Research**

Limitations in the research findings have been discussed as they were identified in the thesis and are also summarised below:

***Hardening:*** The approach adopted in the research was to develop secure PEEs based upon hardening. This approach allowed vulnerabilities and unrequired functionality in open source and/or bespoke systems/applications to be removed to produce a secure and “fit for purpose” capability with only the functionality required. The limitations of this approach are it is difficult to fully harden COTS systems/applications and if a new open source/bespoke application is to be added to a secure PEE it will need to be hardened. These limitations may constrain the use of hardened secure PEEs.

***Lack of certification of secure PEE for high assurance use:*** A high grade secure PESE was constructed to demonstrate how highly sensitive information could be stored and processed in remote work environments where such activities would not normally occur. The limitation of the high grade secure PESE was that it used the two secure PEEs developed for the commercial grade secure PESE. These two secure PEEs were

constructed by hardening open source software and as discussed in Chapter 5 would not meet the criteria to be certified as high assurance software. The trial utilising the high grade secure PESE was a simulated, rather than a live, exercise and therefore it was acceptable to use secure PEE software as an indicative demonstrator. It is recognised that a certified or accredited<sup>81</sup> high assurance secure PEE would be needed for high assurance use scenarios.

***A secure PESE may not suit all remote work scenarios:*** The research objectives directed the design of the secure PEEs to be secure, hardened, functionally restricted, execution environments that prevented a user adding or changing functionality. Whilst the trials identified no useability issues with the secure PEEs it is recognised that these execution environments may not suit all remote workers. It is acknowledged in the research that such secure PEEs are ideally suited to teleworkers performing transaction-oriented work and deployed workers using a fixed set of applications. However, as a separate pluggable device, a secure PESE provides the option of using it with a PC as required for sensitive work and then it can be unplugged to allow the unconstrained PC to be used for non-sensitive work.

***Use of smartphone and tablets:*** Although not a limitation it is relevant to note that when the research commenced tablets and smartphones were not widely used for remote work; however, by the conclusion of the research their use had become prolific (TTI, 2014). As outlined in this thesis secure PESEs were not designed to work with tablets and smartphones. Security mechanisms for tablets and smartphones are often considered an impediment to the user experience (IDC, 2013), and therefore these devices usually do not include or support strong information security technologies (iOS, 2014; Android, 2014). Limited hardening of these devices is possible but security conscious organisations place access limitations on these devices to sensitive corporate data and servers (ASD, 2013). A secure PESE could therefore provide a complementary capability that is used with a PC, when processing of sensitive information or accessing a protected corporate server is required, thus allowing the tablet/smartphone to be used for non-sensitive work.

---

<sup>81</sup> Some organisations use tried and tested COTS systems/applications with hardening applied where possible and then accredit the system/application for security critical operations.

**Threats not considered:** The threats from a hardware-based keyboard logger and from PC firmware (i.e. the Basic Input Output system (BIOS) (Fisher, 2014) or the Unified Extensible Firmware Interface (UEFI) (Anthony, 2011) were considered to be outside of the scope of the research. A hardware keyboard logger could be embedded in the PC or attached to a concealed PC USB port unbeknown to the user and could capture and store/transmit key strokes. The PC's BIOS/UEFI could be infected with malware that subverts a secure PESE.

#### **7.4 Assessment of the Research Design**

DSR proved to be an ideal paradigm as it facilitated research through design and the evaluation of the outcomes purely from a design perspective. The DSRM (Peppers et al., 2007) provided the structure required to conduct the research as its process elements enabled the work to be staged. The DSRM also allowed the implementation, documentation and review policies and practices of the researcher's workplace product development environment to be effectively used.

The work by Gregor, Hevner and Jones (Gregor and Hevner, 2013; Gregor and Jones, 2007) was identified towards the end of the research and provided excellent frameworks to present, position and assert the doctoral research. Their work facilitated the completion of the research design by proposing the use of design cycles to present knowledge developed over time and also defined evaluation criteria that was founded upon knowledge and theory contribution structures to position and assert DSR. The design cycle approach further structured the research by facilitating start and end points for knowledge and artifact development. The knowledge and theory contribution structures underpinned an academic explanation of the research outcomes.

#### **7.5 Future Work**

The following five areas of future work have been identified:

**Validate Design Theory:** The design theory asserted in this thesis has been categorised as 'nascent' with a level 2 knowledge contribution (as determined in Chapter 6 using the knowledge maturity model defined in Chapter 3). The next step required is to move the design theory from 'nascent' to 'well-developed' and become a level 3 knowledge contribution. Therefore future work should include preparing a paper(s) that presents the

design theory for secure PESEs. The intent would be for the publication to attract both comments and possibly the application of the theory to further refine and validate it.

**Longer term trials:** The duration of each trial was approximately one week. Whilst each trial was of sufficient duration for the organisation to assess the capability, an extended trial would have been desirable to further validate the secure PESE's security capabilities, in particular its ability to protect against zero day attacks. Future work should include trials conducted over a three to six week period duration involving continuous use.

**PC Firmware Threat:** As identified above a secure PESE is not designed to counter a threat from the PC's firmware (i.e. the BIOS or the UEFI). UEFI has now replaced the BIOS as the firmware in most PCs. As identified in Paper 9, future work should include identifying the threats and vulnerabilities of the UEFI to a secure PESE and any possible countermeasures.

**Remote Work Security Model:** Paper 3 identified the attributes for a telework security model. These attributes were used as the basis of the initial nascent secure PESE concept. The secure PESE concept and the asserted design theory represent a conceptual body of abstract knowledge that can be used to progress a security model that captures the policy enforcement and security properties for remote work. Future work should consider the development of a remote work security model.

## **7.6 Research Synopsis**

In this chapter the doctoral research has been shown to be original, have significance and impact, and successfully achieve the goals that were set. Limitations were highlighted and four areas for future work were proposed.

This thesis with publication has presented a body of research that has verified the research argument, generated a body of knowledge that has made a contribution to the area of study and asserted a design theory.

## References

- 3Com (1999). The Net Impact of Thin Clients – Technical Brief, 3Com Corporation, September 1999, available from:  
[http://www.pulsewan.com/data101/thin\\_client\\_basics.htm](http://www.pulsewan.com/data101/thin_client_basics.htm).
- Abrams, R. S. (2009), Uncovering the network-centric organization, University of California, Irvine, available from:  
[http://media.proquest.com/media/pq/classic/doc/1866206951/fmt/ai/rep/NPDF?\\_s=nD1iS%2B5mmDKSRvUsQx75d1uYNVM%3D](http://media.proquest.com/media/pq/classic/doc/1866206951/fmt/ai/rep/NPDF?_s=nD1iS%2B5mmDKSRvUsQx75d1uYNVM%3D)
- AccessEconomics (2010), “Impacts of Teleworking under the NBN”, Prepared for the Department of Broadband, Communications and the Digital Economy, Access Economics Pty Limited, July 2010, available at:  
[http://www.dbcde.gov.au/data/assets/pdf\\_file/0018/130158/Impactsofteleworkin%20gundertheNBN.pdf](http://www.dbcde.gov.au/data/assets/pdf_file/0018/130158/Impactsofteleworkin%20gundertheNBN.pdf)
- Access Economics (2010b), Australian Business Expectations for the National Broadband Network, prepared for the Department of Broadband, Communications and the Digital Economy, Access Economics, November 2010.
- ACMA (2013), Report 3 – Smartphones and Tablets Takeup and Use in Australia, Australian Communications and Media Authority, January 2013, available from:  
<http://www.acma.gov.au/~media/Communications%20Analysis/Comms%20Report%202011%2012/PDF/Report%203%20Smartphones%20and%20tablets%20Summary%20report%20Comms%20report%2020112012%20series.pdf>
- ACM-DL (2014), Citations for A Design Science Research Methodology for Information Systems Research, Association of Computing Machinery Digital Library, September 2013, available at:  
<http://dl.acm.org/citation.cfm?id=1481765.1481768&coll=DL&dl=GUIDE&CFID=363715687&CFTOKEN=95589558>
- Adams, R. (2013), The Advanced Data Acquisition Model (ADAM): A Process Model for Digital Forensic Practice, PhD Thesis, Murdoch University.
- ADoD (2007), Joint Operations for the 21st Century, Australian Department of Defence, available at: <http://www.defence.gov.au/publications/docs/FJOC.pdf>
- AIC (2014), The Australian Institute of Commercialisation, Commercialisation Defined, available at: [http://www.ausicom.com/about\\_aic.php](http://www.ausicom.com/about_aic.php)
- Akhawe, D., and Felt, A. P. (2013), Alice in Warningland: A Large-Scale Field Study of Browser Security Warning Effectiveness, Usenix Security, available from:  
<http://static.googleusercontent.com/media/research.google.com/en//pubs/archive/41323.pdf>



- Alberts D.S, Garstka J.J. and Stein F.P. (1999) Network Centric Warfare – Developing and Leveraging Information Superiority, CCRP Publication Series, Washington.
- Al-Zarouni, M. (2006), The reality of risks from consented use of USB devices, Proceedings of 4th Australian Information Security Management Conference, Edith Cowan University, Perth, Western Australia.
- Al-Zarouni, M. (2007), Introduction to mobile phone flasher devices and considerations for their use in mobile phone forensics, Proceedings of the 5th Australian Digital Forensics Conference, Edith Cowan University, Perth Western Australia.
- Ampomah, M., De Silva, Y., Li, H., Pahlisa, P., Yang, Q., & Zhang, Q. (2013), Information security strategy and teleworking (in) security, Department of Computing and Information Systems, University of Melbourne, available at: [https://minerva-access.unimelb.edu.au/bitstream/handle/11343/33341/300309\\_2013\\_Ampomah\\_Strategy.pdf?sequence=1](https://minerva-access.unimelb.edu.au/bitstream/handle/11343/33341/300309_2013_Ampomah_Strategy.pdf?sequence=1)
- Anderson, R. (2008), Security Engineering, John Wiley & Sons.
- Android (2014), Android, the world's most popular mobile platform, available from: <http://developer.android.com/about/index.html>
- Anthony, S. (2011), Demystifying UEFI, the long-overdue BIOS replacement, available from: <http://www.extremetech.com/computing/96985-demystifying-uefi-the-long-overdue-bios-replacement>
- Antill, L. (1986), Action research in information systems design, Design Studies, Volume 7, Issue 4, pp 192-198, October 1986.
- Antonopoulos, A. M. (2007), Combining work and play threatens business security, Network World, available from: <http://www.networkworld.com/article/2286767/lan-wan/combining-work-and-play-threatens-business-security.html>
- APH (2006), Pathways to Technological Innovation, The Parliament of the Commonwealth of Australia, available from: [www.aphref.aph.gov.au-house-committee-scin-pathways-report-fullreport.pdf](http://www.aphref.aph.gov.au-house-committee-scin-pathways-report-fullreport.pdf)
- Armbrust, M., Fox, A., Griffith, R., Jospwph, A.D., Katz, R. H., Konwinski, A., Lee, G., Patterson, D. A., Rabkin, A., Stoica, I., Zaharia, M. (2009), Above the Clouds: A Berkeley View of Cloud Computing, Technical Report No. UCB/EECS-2009-28, February 2009, Electrical Engineering and Computer Sciences, University of California at Berkeley.
- ASD (2013), iOS Hardening Configuration Guide for iPod Touch, iPhone and iPad devices, available from: <http://www.asd.gov.au/publications/protect/ios-hardening-guide.htm>

- ASD (2014), Australian Government Information Security Manual Executive Companion 2014, Australian Signals Directorate, Available from:  
[http://www.asd.gov.au/publications/Information\\_Security\\_Manual\\_2014\\_Exec\\_Companion.pdf](http://www.asd.gov.au/publications/Information_Security_Manual_2014_Exec_Companion.pdf)
- ASIAL (2014), Australian Security Industry Association Limited, Standards and codes, available from: <http://www.asial.com.au/resource-centre/standards-and-codes>
- Astani, M., Ready, K., Tessema, M. (2013), BYOD Issues and Strategies in Organizations, Issues in Information Systems, Volume 14, Issue 2, pp.195-201, 2013.
- Avison, D., Fitzgerald, G. (2006), Information Systems Development – Methodologies, Techniques and Tools, Edition 4, McGraw-Hill Education, 2006.
- Baker, W., Goudie, M., Hutton, A. Hylender, C., Niemantsverdriet, J. and Novak, C. (2010), Verizon 2010 data breach investigation report, available from:  
[http://www.verizonbusiness.com/resources/reports/rp\\_2010-data-breach-report\\_en\\_xg.pdf](http://www.verizonbusiness.com/resources/reports/rp_2010-data-breach-report_en_xg.pdf).
- Baruch, Y. (2000). Teleworking: benefits and pitfalls as perceived by professionals and managers, New Technology, Work and Employment, Volume 15, Issue 1, pp34-49.
- Baskerville, R. (1993), Information systems security design methods: implications for information systems development, ACM Computing Surveys, Volume 25, Issue 4, pp375-414.
- Baskerville, R. and Wood-Harper, A. T. (1998), Diversity in information systems action research methods, European Journal of Information Systems, Volume 7, pp 90-107, 1998.
- Bates, B. J. (2010), Secure Telework and Remote Delivery of Dispersed Teams, IAnewsletter, Volume 13, Issue 4, Fall 2010, pp8-11.
- Becrypt (2009), Case Study, Foreign and Commonwealth Office deploy Becrypt Trusted Client to provide secure remote access, available from:  
<https://www.becrypt.com/assets/files/cs/Foreign%20and%20Commonwealth%20Office%20-%20Trusted%20Client.pdf>
- Becrypt (2013), Becrypt Ltd, London, accessed October 2013, available at:  
<http://www.becrypt.com/products/trusted-client>.
- Bielova, N. (2013), Survey on JavaScript security policies and their enforcement mechanisms in a web browser, The Journal of Logic and Algebraic Programming, Volume 82, Issue 8, pp 243-262, available from: <http://www-sop.inria.fr/members/Natalia.Bielova/papers/BIEL-12-Survey.pdf>

- Bell, D.E, and LaPadula, L.J. (1976), Secure Computer Systems: Unified Exposition and Multics Interpretation, ESD-TR-75-306, MTR 2997 Rev. 1, The MITRE Corporation, March 1976.
- BerkeleySecurity (2014), Recommended Resources for System Hardening, University of California, available from: <https://security.berkeley.edu/node/143>
- Bindrup, R., Yamaguchi, J., and Boyd, W. E. (2014), U.S. Patent No. 8,637,985. Washington, DC: U.S. Patent and Trademark Office.
- Bragg R., Phodes-Ousley M., et al (2004), Network Security: The Complete Reference, McGraw-Hill/Osborne, 2004.
- Brewer D., and Nash M. (1989), The Chinese Wall Security Policy, IEEE Symposium on Security and Privacy, 1989, pp.206-216.
- Bull (2014), Bull SAS, Bull Group, Les Clayes sous Bois, France available at: <http://www.bull.com/about-bull/index.php>
- BullDirect (2008), Bull Direct – The Newsletter from Bull, No. 26, May 2008, available from: [http://www.bull.com/bulldirect/N26/BullDirectN26\\_en.pdf](http://www.bull.com/bulldirect/N26/BullDirectN26_en.pdf)
- Burger, T. (2012), The Advantages of Using Virtualization Technology in the Enterprise, Intel Developer Zone, May 2012, available at: <http://software.intel.com/en-us/articles/the-advantages-of-using-virtualization-technology-in-the-enterprise>
- BusinessWire (2008), RingCube and MXI Security™ Partner to Deliver a Virtual Desktop on the Hardware-Encrypted Stealth MXP™ Biometric USB Drive, July 2008, available at: <http://www.businesswire.com/news/home/20080715005160/en/CORRECTING-REPLACING-RingCube-MXI-Security-TM-Partner#.VKiXcSuVKNN>
- CBED (2014), Cambridge Business English Dictionary, Definition of Commercialisation, available from: <http://dictionary.cambridge.org/dictionary/business-english/commercialization>
- CC (2014), Common Criteria, Version 3.1, Release 4, available from: <https://www.commoncriteriaportal.org/cc/>
- CDG (2014), About CDG, available from: <http://www.defence.gov.au/CDG/>
- CertAustralia (2013), Cyber Crime & Security Survey Report 2013, Cert Australia, Australian Government, available at: <https://www.cert.gov.au/newsroom>
- Chan, J., Nepal, S., Moreland, D., Hwang, H., Chen, S., and Zic, J. (2007), User-controlled collaborations in the context of trust extended environments, Enabling

Technologies: Infrastructure for Collaborative Enterprises, 2007, WETICE 2007, 16th IEEE International Workshops on (pp. 389-394), IEEE.

Chiasson, M., Germonprez, M., Mathiassen, L. (2008), Pluralist action research: a review of the information systems literature, *Journal of Information Systems*, Volume 19, pp 31-54, 2008.

Ciampa, M. (2007), *CWSP Guide to Wireless Security*, Thompson Course Technology, Thompson Learning Inc., ISBN-10: 1-4188-3637-0, 2007.

CIOG (2010) *Single Information Environment (SIE) - Architectural Intent 2010*, Chief Information Officer Group Defence Publishing Service, Department of Defence, Australian Government, DPS:DEC013-09, 2010.

Cisco (2014), *Cisco 2014 Annual Security Report*, Cisco System Inc., accessed January 2014, available at: <http://www.cisco.com/web/offers/lp/2014-annual-security-report/index.html>

Citrix (2013), *Application Streaming*, Citrix Product Documentation, Citrix Systems Inc., available from: <http://support.citrix.com/proddocs/topic/xenapp6-w2k8/ps-stream-intro-wrapper-for-xenapp-library-v2.html>

Citrix (2014), *Introducing XenAPP*, Citrix Systems Inc., available from: [http://passthrough.fw-notify.net/download/204585/http://www.citrix.com/content/dam/citrix/en\\_us/documents/products-solutions/xenapp-datasheet.pdf](http://passthrough.fw-notify.net/download/204585/http://www.citrix.com/content/dam/citrix/en_us/documents/products-solutions/xenapp-datasheet.pdf)

Clark D. D., and Wilson D. R. (1987), *A Comparison of Commercial and Military Computer Security Policies*, *Proceedings of the 1987 Symposium on Security and Privacy*, IEEE.

Clear, F. (2007), "SMEs, electronically-mediated working and data security: cause for concern?" *International Journal of Business Science and Applied Management*, Vol. 2, Issue 2, 2007.

Cnet News (2009), *Using software updates to spread malware*, Retrieved October, 2010, available from [http://news.cnet.com/8301-27080\\_3-10301485-245.htm](http://news.cnet.com/8301-27080_3-10301485-245.htm)

Colliers (2011), *Activity Based Workplaces Can it work for everyone?*, Colliers International White Paper, Spring 2011, available from: <http://www.colliers.com.au/Find-Research/~media/Files/Corporate/Research/Speciality%20Reports%20and%20Property%20White%20Papers/Activity%20Based%20Workplaces%20white%20paper%20%20Spring%202011.ashx>

Corbett, C. J., Blackburn, J. D., & Van Wassenhove, L. N. (1999), *Partnerships to improve supply chains*, *MIT Sloan Management Review*, Volume 40, Issue 4, Summer 1999, pp71-82.

- Cresswell, J. W. (2009), *Research design: qualitative, quantitative and mixed methods approaches*, 3<sup>rd</sup> Edition, Sage Publications, 2009.
- Crossler, R. E., Johnston, A. C., Lowry, P. B., Hu, Q., Warkentin, M., and Baskerville, R. (2013), Future directions for behavioural information security research. *Computers & Security*, Volume 32, pp90-101.
- CSIRO (2008), *Virtual Machines: An Initial Analysis of Threats and Remedial Actions*, available from: <http://www.csiro.au/Portals/Publications/Research--Reports.aspx>
- CTD (2010), *Secure Portable Application Device to Support Network Centric Warfare*, Defence Science Technology Organisation, available from: [http://www.dsto.defence.gov.au/sites/default/files/basic\\_pages/documents/previ\\_ously\\_funded\\_projects\\_rounds\\_15-6.pdf](http://www.dsto.defence.gov.au/sites/default/files/basic_pages/documents/previ_ously_funded_projects_rounds_15-6.pdf)
- Cyberoam (2014), *Security Advisory - OpenSSL Heartbleed Vulnerability in OpenSSL*, available from: <http://kb.cyberoam.com/default.asp?id=2909&Lang=1>
- Cypress (2009), *USB 2.0 to ATA/ATAPI Bridge Data Sheet, Model CY7C68320*, available from: <http://www.cypress.com/?docID=36507>
- Dade, J. (2013), *New Way of Working at Siemens: internal social responsibility and positioning a firm as employer of choice*, *Journal of European Management & Public Affairs Studies*, Volume 1, Number 1, pp11-16, 2013.
- David, B. (2002), *Thin Client Benefits*, White Paper, Newburn Consulting, Version 1b, March 2002, available from: [http://www.thinclient.net/pdf/Thin\\_Client\\_Benefits\\_Paper.pdf](http://www.thinclient.net/pdf/Thin_Client_Benefits_Paper.pdf)
- Davis, A. (2011). *Telework Productivity and Effectiveness: Factors that Influence Results-Oriented Job Assessments*, University of Oregon, Applied Information Management Program, (Doctoral dissertation, Intel Corporation).
- DBCDE (2011), "Telework Forum: Bringing home the benefits of telework using the NBN, A record of the Telework Forum held 3rd August 2011", A partnership between the Department of Broadband, Communications and the Digital Economy and the Australian Information Industry Association, August 2011, available at: <http://www.nbn.gov.au/get-involved/upcoming-events/nsw/telework-forum-bringing-home-the-benefits-of-teleworking-using-the-nbn>.
- DBCDE (2013), *Advancing Australia as a Digital Economy: An Update To The National Digital Economy Strategy*, Department of Broadband, Communications and the Digital Economy, 2013, available from: [http://www.archive.dbcde.gov.au/\\_\\_data/assets/pdf\\_file/0006/173049/Advancing\\_Australia\\_as\\_a\\_Digital\\_Economy.pdf](http://www.archive.dbcde.gov.au/__data/assets/pdf_file/0006/173049/Advancing_Australia_as_a_Digital_Economy.pdf)

- Deloitte (2011), "Next Generation Telework: A Literature Review", Report by Deloitte Access Economics for the Department of Broadband, Communications and the Digital Economy, July 2011, available at:  
[http://www.nbn.gov.au/files/2012/02/Next\\_Generation\\_Telework-A\\_Literature\\_Review-July\\_20111.pdf](http://www.nbn.gov.au/files/2012/02/Next_Generation_Telework-A_Literature_Review-July_20111.pdf).
- DESRIST (2014), "Design Science Research in Information Systems and Technology", Web Site, accessed September 2014, available at: [www.desrist.org](http://www.desrist.org)
- DESRIST-2014 (2014), "Ninth International Conference on Design Science Research in Information Systems and Technology", Florida International University, Miami, 2014, May 2014, available at: [www.desrist2014.fiu.edu](http://www.desrist2014.fiu.edu)
- Dijkstra, E. W. (1968), The Structure of the "THE" - Multiprogramming System, Communications of the ACM, Volume 11, Issue 5, pp 341 - 346.
- Drimer, S., Murdoch, S. and Anderson, R. (2008), Thinking inside the box: system-level failures of tamper proofing, Computer Laboratory, University of Cambridge, UCAM-CL-TR-711.
- DSD (2010), Standards for Developing High Assurance Products, Cyber and Information Security Division, Defence Signals Directorate, File Reference 2008/2366, Release 1.1.
- DSD (2011), Australian Government Information Security Manual, Defence Signals Directorate, Commonwealth Government of Australia, August 2011, available at: [http://www.asd.gov.au/publications/Information\\_Security\\_Manual\\_2010.pdf](http://www.asd.gov.au/publications/Information_Security_Manual_2010.pdf)
- DSD (2012), Strategies to Mitigate Targeted Cyber Intrusions, Cyber Security Operations Centre, Defence Signals Directorate, available at: [www.dsd.gov.au/publications/Top\\_35\\_Mitigations\\_2012.pdf](http://www.dsd.gov.au/publications/Top_35_Mitigations_2012.pdf).
- DSTO (2014), About DSTO, available from: <http://www.dsto.defence.gov.au/discover-dsto/about-dsto>
- ECU (2012), Postgraduate Research: Thesis with Publication, Policy PL036, Issue 2, August 2012, Edith Cowan University.
- Explorable (2013), Research Designs, available at: <http://explorable.com/research-designs>
- Ferguson, N., Schneier, B., and Kohno, T. (2011), Cryptography Engineering: Design Principles and Practical Applications, John Wiley & Sons.
- Ferrie, P. (2007), Attacks on more virtual machine emulators, *Symantec Technology Exchange*.
- Fewell, M. P., & Hazen, M. G. (2003), Network-Centric Warfare - Its Nature and Modelling (No. DSTO-RR-0262), Defence Science and Technology Organisation

Salisbury (Australia), Systems Sciences Lab, available from:  
<http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA420257>

Fisher, T. (2014), Everything You Need to Know About BIOS, available from:  
<http://pcsupport.about.com/od/termsb/p/bios.htm>

Ford, B. and Cox, R. (2008), Vx32: Lightweight User-level Sandboxing on the x86. In USENIX Annual Technical Conference (pp. 293-306).

Folks, I. I., and Richard, L. (2011), Network Centric Warfare in the Age of Cyberspace Operations, March 2011, U.S. Army War College, Carlisle Barracks, PA, available from: <http://www.dtic.mil/dtic/tr/fulltext/u2/a547453.pdf>.

Furnell, S. (2005), Handheld hazards: The rise of malware on mobile devices, Computer Fraud & Security, 2005 Issue 5, pp4-8, available at:  
<http://www.sciencedirect.com/science/article/pii/S1361372305702104>

Furnell, S. (2006). Securing the home worker, Network Security, Issue 11, 2006, pp6-12, available at:  
<http://www.sciencedirect.com/science/article/pii/S1353485806704512>

Furnell, S. (2009) Mobile Security: A Pocket Guide, IT Governance Publishing, IT Governance Limited, ISBN 9781849280204.

Furnell, S. and Clarke, N. (2012), Power to the people? The evolving recognition of human aspects of security. Computers & Security, Volume 31 Issue 8, pp983-988.

Gajski, D. D., Vahid, F., Narayan, S., and Gong, J. (1994), Specification and Design of Embedded Systems, Englewood Cliffs: PTR Prentice Hall.

Garner, G. and Dick, G. (1997), Telecommuting: A Managerial Perspective, Multiconference on Systemic, Cybernetics and Informatics, Caracas, Venezuela, July, pp 374-381.

Garfinkel, S. L., & Shelat, A. (2003). Remembrance of data passed: A study of disk sanitization practices Security & Privacy, IEEE, Vol. 1, Issue 1, pp17-27, February 2003.

Garfinkel, T., Pfaff, B., Chow, J., Rosenblum, M., & Boneh, D. (2003), Terra: A virtual machine-based platform for trusted computing, ACM SIGOPS Operating Systems Review, Volume 37, Issue 5, pp. 193-206.

GD (2014), General Dynamics ProtecD@T, available from:  
<http://www.gdc4s.com/products/secure-voice-and-data/data-at-rest-encryption>



- Gibson, J. W., Blackwell, C. W., Dominicus, P., & Demerath, N. (2002). Telecommuting in the 21st century: Benefits, issues, and a leadership model which will work. *Journal of Leadership & Organizational Studies*, Volume 8, Issue 4, pp75-86.
- globull (2010), globull brochure, Bull SAS, available at: [http://www.myglobull.com/globull-brochure\\_en.pdf](http://www.myglobull.com/globull-brochure_en.pdf)
- globull (2011), Combining mobility & security, Bull SAS, 2011, available at: <http://www.bull.com/p/register.php?id=127>
- Godlove, T. (2012), Examination of the Factors that Influence Teleworkers' Willingness to Comply with Information Security Guidelines, *Information Security Journal: A Global Perspective* Volume 21, Issue 4, pp216-229.
- Goslar, M. (2000) The New e-Security Frontier, *Informationweek*, July, 2000.
- Goth, G. (2012), Mobile Security Issues Come to the Forefront, *IEEE Internet Computing*, Volume 16, Issue 3, pp7-9, June 2012.
- Gregor, S., and Hevner, A. R. (2013), Positioning and presenting design science research for maximum impact. *MIS Quarterly*, Volume 37, Issue 2, pp337-356.
- Gregor, S., and Jones, D. (2007), The anatomy of a design theory, *Journal of the Association for Information Systems*, Volume 8, Issue 5, pp312-335.
- Grier, C., Tang, S., and King, S. T. (2008), Secure web browsing with the OP web browser, *IEEE Symposium on Security and Privacy*, pp. 402-416.
- GSA (2002), Analysis of Home-Based Telework Technology Barriers, prepared by Booz Allen Hamilton for Office of Governmentwide Policy, General Services Administration, available from: <http://www.gsa.gov/portal/content/102447>
- GSA (2007), Information Technology and Telecommunications Guidelines for Federal Telework and Other Alternative Workplace Arrangement Programs, *FMR Bulletin* 2007-Bf, 2<sup>nd</sup> March 2007, available from: [www.gsa.gov/teleworklibrary](http://www.gsa.gov/teleworklibrary).
- GSC (2009), Secure Systems Finalist in the Global Security Challenge 2009, available from: <https://www.omnicompete.com/files/security/Press%20Release%20-%20Asia%202009.pdf>
- Gühring, P. (2006), Concepts against man-in-the-browser attacks, available from: [www.cacert.at/svn/sourcerer/CACert/SecureClient.pdf](http://www.cacert.at/svn/sourcerer/CACert/SecureClient.pdf)
- Gutmann, P., and Grigg, I. (2005), Security usability, *Security & Privacy, IEEE*, Volume 3, Issue 4, pp 56-58.



- Harris, C., (2010), DoD Finds Teleworking is Possible With a New Tool, ClearanceJobs.com, available at: <http://news.clearancejobs.com/2010/10/28/dod-finds-teleworking-is-possible-with-a-new-tool/>
- Herrmann, D. S. (2002), Using the Common Criteria for IT Security Evaluation, Auerbach Publications, CRC Press.
- Hevner, A. R., March, S.T., Park, J. and Ram, S. (2004), "Design Science in Information Systems Research", MIS Quarterly, Volume 28, Issue 1, pp 75-105, March 2004.
- Hevner, A. R., Chatterjee, S. (2010), Design Research in Information Systems, 2010, New York: Springer.
- Hoogendijk L. (2006), A Risk Analysis Methodology for Secure Teleworking, Master Thesis, School of Informatics, Erasmus University of Rotterdam.
- Hughes, J., and Stytz, M. R. (2003), Advancing Software Security - The Software Protection Initiative, 8th International Command and Control Research and Technology Symposium, National Defense University, June 2003.
- Humphrey, W.S. (2000), Introduction to the Team Software Process, Addison-Wesley Professional.
- Husted, N., Saïdi, H., & Gehani, A. (2011), Smartphone security limitations: conflicting traditions. In Proceedings of the 2011 Workshop on Governance of Technology, Information, and Policies (pp. 5-12). ACM.
- IBM (2008), IBM Internet Security Systems X-Force® 2008 Trend & Risk Report, available from: <https://www-935.ibm.com/services/us/iss/xforce/trendreports/xforce-2008-annual-report.pdf>
- IDC (2013), Android and iOS Combine for 91.1% of the Worldwide Smartphone OS Market in 4Q12 and 87.6% for the Year, International Data Corporation Inc, press release, February 2013, accessed September 2013, available at: <http://www.idc.com/getdoc.jsp?containerId=prUS23946013>.
- Imation (2014), Imation Corp., available at: <https://www.imation.com>
- InnoCentive (2014), Global Security Challenge, available from: <http://www.innocentive.com/files/node/casestudy/case-study-global-security-challenge.pdf>
- Ironkey (2014), Ironkey by Imation, available at: <http://www.ironkey.com>
- iOS (2014), About the iOS Technologies, Apple, available from: <https://developer.apple.com/library/ios/documentation/miscellaneous/conceptual/iphoneostechoverview/Introduction/Introduction.html>

- ISO 31000:2009 (2009), Risk Management - Principles and Guidelines, , International Organization for Standardization, ISO Central Secretariat,1, ch. de la Voie-Creuse, CP 56, CH-1211 Geneva 20, Switzerland.
- ISO/IEC 27001:2013 (2013), Information Technology – Security Techniques - Information Security Management Systems - Requirements, International Organization for Standardization, ISO Central Secretariat,1, ch. de la Voie-Creuse, CP 56, CH-1211 Geneva 20, Switzerland.
- ISO/IEC 27002:2013 (2013), Code of Practice for Information Security Management , International Organization for Standardization, ISO Central Secretariat,1, ch. de la Voie-Creuse, CP 56, CH-1211 Geneva 20, Switzerland.
- ISO/IEC 27005:2011 (2011), Information Security Risk Management, International Organization for Standardization, ISO Central Secretariat,1, ch. de la Voie-Creuse, CP 56, CH-1211 Geneva 20, Switzerland.
- Jackson, W. (2010), Air Force lab finds a simple way to support telework, GCN, available from: <http://gcn.com/articles/2010/10/18/gcn-awards-air-force-lightweight-portable-security.aspx>
- Jalil, M.M. (2013), Practical Guidelines for Conducting Research, The Doner Committee for Enterprise Development, February 2013, Department of Computing and Information Systems, University of Melbourne, available from: [www.Enterprise-Development.org](http://www.Enterprise-Development.org)
- James, P. (1987), Rapid Prototyping of VDM Specifications Using Smalltalk-80, MSc. Dissertation, University of Manchester, 1987.
- James, P. (1991), The Managerial Issues of Teleworking, Diploma Dissertation, Graduate Diploma of Management Studies, Manchester Metropolitan University, Manchester, 1991.
- James, P., & Woodward, A. (2007), Securing VoIP: A Framework to Mitigate or Manage Risks, 5th Australian Information Security Management Conference, Perth, Western Australia.
- James, P. (2008), Preventing the Acquisition of Data from Virtual Machine based Secure Portable Execution Environments, 6th Australian Digital Forensics Conference, Perth, Western Australia, pp 82-97.
- Jang, D., Tatlock, Z., and Lerner, S. (2012), Establishing browser security guarantees through formal shim verification, In Proceedings of the 21st USENIX conference on Security symposium, USENIX Association.
- Jilani, U., Ahimmat, A., Raso, A., Thorpe, D., & Tran, M. (2013), Ready, Steady Telework – Information Security essentials for the teleworker, Department of Computing and

Information Systems, University of Melbourne, available at: [https://minerva-access.unimelb.edu.au/bitstream/handle/11343/33342/300311\\_2013\\_Jilani\\_SETA.pdf](https://minerva-access.unimelb.edu.au/bitstream/handle/11343/33342/300311_2013_Jilani_SETA.pdf)

Joice W., (2007) Implementing Telework: The Technology Issue, The Public Manager, Summer 2007, Volume 36, Issue 2, pp64-68.

Jones, A., and Valli, C. (2011), Building a Digital Forensic Laboratory: Establishing and Managing a Successful Facility, Butterworth-Heinemann.

Jones, A., Valli, C., Dardick, G., and Sutherland, I. (2008), The 2007 Analysis of Information Remaining on Disks offered for sale on the second hand market, Journal of Digital Forensics, Security and Law, Volume 3, Issue 1, 2008.

Jonsson, E. (2006), Towards an integrated conceptual model of security and dependability, The 1<sup>st</sup> International Conference on Availability, Reliability and Security, ARES 2006, IEEE, pp1-8.

Keller, J. (2010), Anti-tamper technologies seek to keep military systems data in the right hands, Military & Aerospace Electronics, available at: <http://www.militaryaerospace.com/articles/2010/04/anti-tamper-technologies-seek-to-keep-critical-military-systems-data-in-the-right-hands.html>

Kinsman, F. (1987), The Telecommuters, Wiley Books.

Khokhar, R. (2006), Smartphones – a call for better safety on the move, Network Security, 2006 Issue 4, pp6-7, available from: <http://www.sciencedirect.com/science/article/pii/S1353485806703543>

Kowalski, K. B. and Swanson, J. A. (2005), Critical success factors in developing teleworking programs, Benchmarking: An International Journal, Volume 12, Issue 3, pp236-249.

Komski, P. (2010), Referencing Hard Drive Partitions, The PaulSki Pages, available from: <http://paulski.com/zpages.php?id=2102>

Kwan, P. C. and Durfee, G. (2007), Practical uses of virtual machines for protection of sensitive user data, Information Security Practice and Experience (pp. 145-161). Springer Berlin Heidelberg.

Kyrnin, J. (2014), What is HTTPS - Why Secure a Web Site, About Tech, available from: <http://webdesign.about.com/od/ecommerce/a/aa070407.htm>

Landoll, D. J. (2005), The Security Risk Assessment Handbook: A complete guide for performing security risk assessments, Auerbach Publications, Taylor & Francis Group.

Lawton, G. (2006), Improved flash memory grows in popularity, Computer, Volume 39, Issue 1, pp 16-18.

- Lindsay, P. (1998), Specification and validation of a network security policy model, Proof in VDM: Case Studies (pp. 65-93), Springer London.
- Liska, A. (2003), The practice of network security: Deployment strategies for production environments, Prentice Hall Professional.
- Lister K., Harnish T. (2011), The Shifting Nature of Work in the UK - Bottom Line Benefits of Telework, Telework Research Network, February 2011.
- LiveCD 2013, The LiveCD List, accessed October 2013, available at:  
<http://www.livcdlist.com>
- Loscocco, P. and Smalley, S. (2001), Integrating flexible support for security policies into the Linux operating system, In Proceedings of the USENIX Annual Technical Conference, available from: <ftp://130.251.61.4/pub/person/ChiolaG/sic00-01/slinux-200104121417.pdf>
- LPS (2008), Lightweight Portable Security, Software Protection Initiative, available at:  
<http://www.spi.dod.mil/lipose.htm>
- Luse, A., Mennecke, B. E., Triplett, J. L., Karstens, N., & Jacobson, D. (2011), A Design Methodology and Implementation for Corporate Network Security Visualization: A Modular-Based Approach, AIS Transactions on Human-Computer Interaction, Volume 3, Issue 2, pp104.
- MantisBT (2012), "Mantis Bug Tracker Administration Guide", November 2012, available at: [http://www.mantisbt.org/docs/master-1.2.x/en/administration\\_guide.pdf](http://www.mantisbt.org/docs/master-1.2.x/en/administration_guide.pdf).
- McFearin, L. D. (2011), Chroot Jail, Encyclopedia of Cryptography and Security, Springer US, pp. 206-207.
- McKay, J. and Marshall, P. (2005), A Review of Design Science in Information Systems, 16th Australian Conference on Information Systems, December 2005, Sydney.
- McKay, J. and Marshall, P. (2007), Science, Design and Design Science: Seeking Clarity to Move Design Science Research Forward in Information Systems, 18<sup>th</sup> ACIS, December 2007, Toowoomba.
- McLaren, T. S., Head, M. M., Yuan, Y., & Chan, Y. E. (2011), A multilevel model for measuring fit between a firm's competitive strategies and information systems capabilities, MIS Quarterly, Volume 35, Issue 4, pp909-929.
- Microsoft (2013), Microsoft Security Development Lifecycle, Microsoft Inc, accessed August 2013, available at: <http://www.microsoft.com/security/sdl/default.aspx>
- MiniSDV (2009), Mini SDV, Secure Systems Limited, available from:  
<http://www.securesystems.com.au/index.php/silicon-data-vault-technology>

- MiniSDVCert (2012), Mini SDV Consumer Guide, Australian Signal Directorate, available from:  
[http://www.asd.gov.au/infosec/epl/view\\_document.php?document\\_id=ODk4lyMjMjAzLjU5LjgzLjE2NQ==](http://www.asd.gov.au/infosec/epl/view_document.php?document_id=ODk4lyMjMjAzLjU5LjgzLjE2NQ==)
- Mitchell, B. (2014), VPN - Virtual Private Network, About Tech, available from:  
[http://compnetworking.about.com/od/vpn/g/bldef\\_vpn.htm](http://compnetworking.about.com/od/vpn/g/bldef_vpn.htm)
- MojoPac (2006), "What is MojoPac?" available at: <http://www.mojopac.com/portal/content/what/>
- MTMSpec (2010), TCG Mobile Trusted Module Specification, Version 1.0, Trusted Computing Group, available at:  
[http://www.trustedcomputinggroup.org/resources/mobile\\_phone\\_work\\_group\\_mobile\\_trusted\\_module\\_specification](http://www.trustedcomputinggroup.org/resources/mobile_phone_work_group_mobile_trusted_module_specification)
- MTMUse (2011), Mobile Trusted module 2.0 Use Cases, Version 1.0, Trusted Computing Group, available at: [http://www.trustedcomputinggroup.org/resources/mobile\\_trusted\\_module\\_20\\_use\\_cases](http://www.trustedcomputinggroup.org/resources/mobile_trusted_module_20_use_cases)
- MXI (2008), MXI Security Inc., available at: [www.mxisecurity.com](http://www.mxisecurity.com)
- Myers, M. D., and Avison, D. (1997), Qualitative research in information systems, Management Information Systems Quarterly, Volume 21, pp241-242.
- NCW (2009), "NCW roadmap 2009", Capability Development Group, Defence Publishing Service, Department of Defence, Australian Government, DPS:FEB005/09, 2009.
- Nepal, S., Zic, J., Hwang, H., and Moreland, D. (2007), Trust extension device: providing mobility and portability of trust in cooperative information systems. In On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS (pp. 253-271), Springer Berlin Heidelberg.
- Nepal, S., Zic, J., Liu, D., and Jang, J. (2011), A mobile and portable trusted computing platform. EURASIP Journal on Wireless Communications and Networking, 2011 Volume 1, pp1-19.
- Ng, B. Y. and Rahim, M. A. (2005), A socio-behavioural study of home computer users' intension to practice security, 9<sup>th</sup> Pacific Asia Conference on Information Systems, Bangkok, Thailand, pp234-247.
- NIST (2007), User's Guide to Securing External Devices for Telework and Remote Access, National Institute of Standards and Technology Special Publication 800-114, U.S. Department of Commerce, November 2007, available at:  
<http://csrc.nist.gov/publications/nistpubs/800-114/SP800-114.pdf>.

NIST (2009), Guide to Enterprise Telework and Remote Access Security, National Institute of Standards and Technology Special Publication 800-114 Revision 1, U.S. Department of Commerce, June 2009, available at:  
<http://www.distributedworkplace.com/DW/Government/Government%202009/NIST%20Guide%20to%20Enterprise%20Telework%20and%20Remote%20Access%20Security.pdf>.

Norton (2007), Mobile Security, available at: <http://au.norton.com/norton-mobile-security>

Nunamaker, J. F., Chen, M., Purdin, T. D. M. (1990), Systems development in information systems research, Journal of Management Information Systems, Volume 7, Issue 3, pp 89-106, 1990.

OMB (2011), Implementing the Telework Enhancement Act of 2010: Security Guidelines, M-11-27, Executive Office of the President, Office of Management and Budget, 15<sup>th</sup> July 2011, available from:  
<http://www.whitehouse.gov/sites/default/files/omb/memoranda/2011/m11-27.pdf>

OPM (2011), Guide to Telework in the Federal Government, Office of Personnel Management, April 2011, available from:  
[http://www.telework.gov/guidance\\_and\\_legislation/telework\\_guide/telework\\_guide.pdf](http://www.telework.gov/guidance_and_legislation/telework_guide/telework_guide.pdf)

Optus (2012), Mobile Working - Future of Work, SingTel Optus Pty Limited whitepaper, available from:  
[https://www.optus.com.au/dafiles/OBCA/downloads/other/Mobile\\_Working\\_Whitepaper-Future\\_of\\_Work\\_Series.pdf](https://www.optus.com.au/dafiles/OBCA/downloads/other/Mobile_Working_Whitepaper-Future_of_Work_Series.pdf)

Ormandy, T. (2007), An empirical study into the security exposure to hosts of hostile virtualized environments, available from:  
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.105.6943&rep=rep&type=pdf>

Osterman (2012), In Search of a Better Way to Enable Telework, An Osterman Research White Paper, available from: [www.ostermanresearch.com](http://www.ostermanresearch.com)

Ouyang, C., Wynn, M. T., Fidge, C., Hofstede, A. H., & Kuhr, J. C. (2010), Modelling complex resource requirements in business process management systems, 21<sup>st</sup> ACIS 2010 Proceedings, Brisbane, December 2010.

Overby, S. (2009), Bring your own... laptop, CIO UK Magazine, January 2009, available at:  
<http://www.cio.co.uk/insight/strategy/bring-your-own-laptop/?page=1>

Pagefile (2013), RAM, Virtual memory, pagefile and memory management in Windows, Microsoft Inc, accessed September 2013, available at:  
<http://support.microsoft.com/kb/2160852>

- PalmDev (2007), Palm® Developer Guide, Palm OS Platform Software and Hardware Rev. F April 30, 2007
- Palvia, P., Mao, E., Salam, A., Soliman K. (2003), Management Information System research: Whats there in a methodology, Communications of the Association of Information Systems, Volume 11, Number 16, 2003.
- Paul, P., Moore, S. W., and Tam, S. (2008), Tamper Protection for Security Devices, Symposium on Bio-inspired, Learning and Intelligent Systems for Security, IEEE Computer Society, pp. 92-96.
- Paul, P. (2013), Microelectronic security measures, Technical Report No: 829, Computer Laboratory, University of Cambridge, UCAM-CL-TR-829.
- Peacey, A. (2006), Teleworkers – extending security beyond the office, Network Security, Issue 11, 2006, pp14-16, available at:  
<http://www.sciencedirect.com/science/article/pii/S1353485806704536>
- Peffers, K., Tuunanen, T., Rothenberger, M.A. and Chatterjee, S. (2007), “A Design Science Research Methodology for Information Systems Research”, Journal of Management Information Systems, Vol 24, No 3, pp 45-77, Winter 2007-8.
- Pham, C. V., Chubin, D. E., Clarke, R. A., & Kuan, A. D. (2013), U.S. Patent No. 8,399,781. Washington, DC: U.S. Patent and Trademark Office.
- PICIP (2014), Defence Material Organisation Priority Industry Capability Innovation Program, available from:  
<http://www.defence.gov.au/dmo/DoingBusiness/Industry/IndustrySupportPrograms/PriorityIndustryCapabilityInnovationProgram/>
- PICIP-SSL (2012), PIC IP Award, Secure Portable Anti-tamper Data Storage Solutions, 2012-07-SEC, available from:  
[http://www.defence.gov.au/dmo/Multimedia/PICIP\\_Round1-9-4475.pdf](http://www.defence.gov.au/dmo/Multimedia/PICIP_Round1-9-4475.pdf)
- PocketSDV (2006), Pocket SDV, Secure Systems Limited, accessed September 2013, available at: <http://www.securesystems.com.au/index.php/pocket-silicon-data-vault>
- PocketSDVCert (2012), Pocket SDV, Evaluated Products List, Australian Signal Directorate, available from:  
[http://www.asd.gov.au/infosec/epl/index\\_details.php?product\\_id=MzEwlyMjMjAzLjU5LjgzLjE2NQ==](http://www.asd.gov.au/infosec/epl/index_details.php?product_id=MzEwlyMjMjAzLjU5LjgzLjE2NQ==)
- Pointsec (2007), Pointsec Mobile Technologies Inc, available at: <http://www.pointsec.com>
- Ponemon (2012), 2013 State of the Endpoint, Ponemon Institute, December 2013, available at:



[http://www.ponemon.org/local/upload/file/2013%20State%20of%20Endpoint%20Security%20WP\\_FINAL4.pdf](http://www.ponemon.org/local/upload/file/2013%20State%20of%20Endpoint%20Security%20WP_FINAL4.pdf)

Protective-ICT (2011), Physical security management guidelines, Physical security of ICT equipment, systems and facilities, Attorney-General's Department, Australian Government, available from:

<http://www.protectivesecurity.gov.au/physicalsecurity/Documents/Physical-security-of-ICT-equipment-systems-and-facilities.pdf>

Protective-Management (2011), Physical security management guidelines, Security zones and risk mitigation control measures, Attorney-General's Department, Australian Government, available from:

<http://www.protectivesecurity.gov.au/physicalsecurity/Documents/Security-zones-and-risk-mitigation-control-measures.pdf>

Pyöriä, P. (2003), Knowledge work in distributed environments: issues and illusions, *New Technology, Work and Employment*, Volume 18, Issue 3, pp166-180.

Pyöriä, P. (2011), Managing telework: risks, fears and rules, *Management Research Review*, Volume 34, Issue 4, pp386-399.

QEMU (2014), QEMU open source processor emulator, available from:  
<http://wiki.qemu.org>

Radhakrishnan, M. and Solworth, J. A. (2007), Quarantining untrusted entities: dynamic sandboxing using LEAP, In *Computer Security Applications Conference, 2007, ACSAC 2007. Twenty-Third Annual* (pp. 211-220), IEEE.

Rae, A., Fidge, C., & Wildman, L. (2006), Fault evaluation for security-critical communication devices. *IEEE Computer*, Volume 39, Issue55, pp61-68, 2006.

Riswadkar, A. and Riswadkar, A. V. (2009), Balancing the Risks of Remote Working: Walking the Telecommuting Line, *The John Liner Review*, Volume 23, Issue 2, pp89-94.

Ronchi, C., and Zakhidov, S. (2009), *Hardened Client Platforms for Secure Internet Banking*. ISSE Securing Electronic Business Processes, ISSE, pp. 367-379, Vieweg+Teubner.

Rouse, M. (2011a), Client-based Virtual Machine, *SearchVirtualDesktop TechTarget*, retrieved July 2012, available from:  
<http://searchvirtualdesktop.techtarget.com/definition/Client-Based-Virtual-Machine>.

Rouse, M. (2014), Remote Desktop, available from:  
<http://searchenterprisedesktop.techtarget.com/definition/remote-desktop>



- Russinovich, M. (2006), Junction, Systems Internals, Retrieved May 2007, from <http://www.systeminternals.com>
- SanDisk (2007), "What is a U3 Smart Drive", Retrieved May, 2007, from <http://www.u3.com/smart/default.aspx>
- Scarfone, K., Jansen, W. and Tracy, M. (2008), Guide to General Server Security, Computer Security Division, National Institute of Standards and Technology, Special Publication 800-123, available from: <http://csrc.nist.gov/publications/nistpubs/800-123/SP800-123.pdf>
- Schneier, B. (2007), Applied Cryptography: protocols, algorithms, and source code in C, John Wiley & Sons.
- Schreuders, Z. C., McGill, T. and Payne, C. (2011), Empowering end users to confine their own applications: The results of a usability study comparing SELinux, AppArmor, and FBAC-LSM, ACM Transactions on Information and System Security (TISSEC), Volume 14, Issue 2, Article 19.
- SDIO (2007), SD Specifications Part E1, SDIO Simplified Specification, Version 2.0, 8/2/07, Technical Committee, SD Card Association.
- SDLC-W (2014), Software Development Life Cycle, Waterfall model design, available at: [http://www.tutorialspoint.com/sdlc/sdlc\\_waterfall\\_model.htm](http://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm)
- SDLC-V (2014), Software Development Life Cycle, V-model design, available at: [http://www.tutorialspoint.com/sdlc/sdlc\\_v\\_model.htm](http://www.tutorialspoint.com/sdlc/sdlc_v_model.htm)
- SDV-HA (2013), Silicon Data Vault High Assurance, product details, URL: <http://www.securesystems.com.au/index.php/high-assurance-silicon-data-vault>
- SecureSystems (2013), Company Website, URL: [www.securesystems.com.au](http://www.securesystems.com.au).
- Seebacher, N. (2014), Guess Where Workers Have New Rights to Work from Home?, CMCWire, available from: <http://www.cmswire.com/cms/social-business/guess-where-workers-have-new-rights-to-work-from-home-025740.php#null>
- seL4 (2014), seL4 high assurance Secure Operating System, URL: <http://sel4.systems/About>
- Seltzer, L. (2013), Windows To Go: Empower and Secure the Mobile Workforce, Activate Marketing Services LLC for Imation LLC, available from: <https://www.imation.com/en/ironkey/#brandhome-4>
- Skorobogatov, S. (2012), Physical attacks and tamper resistance, Introduction to Hardware Security and Trust (pp. 143-173), Springer New York.
- Skorobogatov, S. (2013), Tamper resistance and hardware security, Computer Laboratory, University of Cambridge, available at:

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.297.7004&rep=rep1&type=pdf>

Smith, M., (2008), MXI Security Stealth MXP – Product Review, ICT Review Journal, September 2008, available at: <http://ictreview.blogspot.com.au/2008/09/mxi-security-stealth-mxp-product-review.html>

SNIA (2010), Solid State Storage Form Factors, Solid State Storage Initiative, available from: <http://www.snia.org/forums/sssi/knowledge/formfactors#SSD>

Sophos (2008), Sophos Security Threat Report, available from: <http://www.sophos.com/en-us/press-office/press-releases/2008/01/security-report.aspx>

Sophos (2009), Threatsaurus: the a-z of computer and data security threats, Sophos Group, Oxford, UK.

SourceWire (2007), BeCrypt to demonstrate secure access Trusted Client capability with Juniper Networks at InfoSec New York, available at: <http://www.sourcewire.com/news/33638/becrypt-to-demonstrate-secure-access-trusted-client-capability-with-juniper#.VKi-uyuVKNM>

Spyrus (2014), Spyrus Secure Portable Workplace, available from: <http://www.microsoft.com/en-au/windows/enterprise/products-and-technologies/devices/windowstogo.aspx>

SSL-WP-17 (2006), Product Development – Process and Documentation, SSL-WP-0017, Issue 1.2, available upon request for Secure Systems Limited.

Standing, C. (2008), How to Complete a PhD, e-book, available from: <http://completephd.com/Contents>

Sternstein A. (2007), Survey: Unauthorized teleworkers a security risk, Government Executive, National Journal Group, available from: <http://www.govexec.com/technology/2007/06/survey-unauthorized-teleworkers-a-security-risk/24577/>

Symantec (2008), Symantec Global Internet Security Threat Report, available at: [http://eval.symantec.com/mktginfo/enterprise/white\\_papers/b-whitepaper\\_internet\\_security\\_threat\\_report\\_xiii\\_04-2008.en-us.pdf](http://eval.symantec.com/mktginfo/enterprise/white_papers/b-whitepaper_internet_security_threat_report_xiii_04-2008.en-us.pdf)

Sturgeon, A. (1996), Telework: threats, risks and solutions, The Journal of Information Management & Computer Security, Volume 4, Issue 2, pp 27-38, 1996.

Tanenbaum, A. S. (2009), Modern Operating Systems, Prentice Hall, Pearson Education Inc.

- TCG (2013), Trusted Computing Group, accessed November 2013, available at: <http://www.trustedcomputinggroup.org/>
- Telework (2013), Telework – A new way to work, Department of Communications, Australian Government, Web Site, accessed September 2013, available at: <http://www.telework.gov.au>
- TestLink (2012), TestLink Open Source Test Management, available from: <http://testlink.org/>
- TPM (2008), Trusted Computing Group, Trusted Platform Module, available from: [http://www.trustedcomputinggroup.org/?e=category.developerDetail&urlpath=trusted\\_platform\\_module&resource\\_type\\_id=-1&is\\_faq=true#nav\\_jump](http://www.trustedcomputinggroup.org/?e=category.developerDetail&urlpath=trusted_platform_module&resource_type_id=-1&is_faq=true#nav_jump)
- Tracy, M., Jansen, W., Scarfone, K. and Winograd, T. (2007), Guidelines on Securing Public Web Servers, Computer Security Division, National Institute of Standards and Technology, Special Publication 800-44, Version 2, available from: <http://csrc.nist.gov/publications/nistpubs/800-44-ver2/SP800-44v2.pdf>
- Treo (2007), Product description and specification of Palm Treo 650, available at: <http://www.palm.com/au/products/smartphones/treo650/>, accessed May 2007
- Trivedi, K. S., Kim, D. S., Roy, A., and Medhi, D. (2009), Dependability and security models, 7th International Workshop on Design of Reliable Communication Networks, DRCN 2009, IEEE, pp11-20.
- TrustedClient (2009), Becrypt Trusted Client- Secure low cost remote working, available from: <http://www.exacttrak.com/wp-content/uploads/2011/03/TrustedClient.pdf>
- TTI (2014), Computer Market Contracts – Tablets and Smartphones Grow, tti MaketEYE, June 2014, available from: <http://www.ttiinc.com/object/me-bishop-20140603.html>
- Ubuntu (2012), “Ubuntu Documentation – LiveCD”, Canonical Limited, available at: <https://help.ubuntu.com/community/LiveCD>.
- Vaishnavi, V. and Kuechler W. (2014), Design Science Research in Information Systems, accessed May 2014, available at <http://desrist.org/desrist/>
- Venable, J. R. (2006) The Role of Theory and Theorising in Design Science Research, 1st International Conference on Design Science Research in Information Systems and Technology (DESIST), 2006, Claremont, California.
- VMware (2014), VMware Workstation, available from: <http://www.vmware.com/au/products/workstation/features.html>
- Watts-Englert, J., Szymanski, M., Wall, P., Sprague, M. A., Dalal, B. (2012), Back to the Future of Work: Informing corporate renewal. 8th Annual Ethnographic Praxis in

Industry Conference (EPIC), October 14th - 17th 2012, Savannah College of Art & Design, Savannah, Georgia.

Weber, A. (2009), Implementation of a digital workspace, Project Report, Master in Information Technology, Universitat Polytechnica de Catalunya - Facultat de Informatica de Barcelona.

Whiteman, S. A. and Dick, G. N. (2006), Telecommuting – In this virtual world, what is holding it back? e-Networks in an Increasingly Volatile World, 11<sup>th</sup> International Workshop on Telework, Fredericton, New Brunswick, August, 2006, pp204-211.

Windows (2013), Windows Operating System, Microsoft Inc, accessed October 2013, accessed October 2013, available at: <http://windows.microsoft.com/en-us/windows/home>.

Win2Go (2014), Windows 8.1 Enterprise in Your Pocket, available from: <http://www.microsoft.com/en-au/windows/enterprise/products-and-technologies/devices/windowstogo.aspx>

WinPE (2013), Windows Preinstallation Environment, Microsoft Windows, available at: [http://technet.microsoft.com/en-us/library/cc766093\(v=ws.10\).aspx](http://technet.microsoft.com/en-us/library/cc766093(v=ws.10).aspx)

Wong, W. (2007), Secure That Microcontroller, Electronic Design, available at: <http://electronicdesign.com/boards/secure-microcontroller>

Wood, C. A., (2013), Secure Operating Systems, Department of Computer Science, Rochester Institute of Technology, available from: [http://people.cis.ksu.edu/~danielwang/Investigation/System\\_Security/SecureOS.pdf](http://people.cis.ksu.edu/~danielwang/Investigation/System_Security/SecureOS.pdf)

WSJ (2013) Considerations for 'Bring Your Own Computer', The Wall Street Journal, Deloitte Insights, 11<sup>th</sup> February 2013, available from: <http://deloitte.wsj.com/cio/2013/02/11/managing-the-complexity-of-bring-your-own-computer/>

x86 (2013), Definition of x86, Encyclopedia, PCmag.com, Ziff Davis LLC, accessed September 2013, available at: <http://www.pcmag.com/encyclopedia/term/54979/x86>.

Ye, L. R. (2012), Telecommuting: Implementation for Success, International Journal of Business and Social Science Volume 3 Issue 15, August 2012

Zalewski, M. (2009), Google browser security handbook, available from: <https://code.google.com/p/browsersec/wiki/Main>

Zbar, J. (2000), Working home alone? How's the security?, Network World, November 2000, available from: <http://www.highbeam.com/doc/1G1-67582437.html>

## Appendix 1 – Co-Author Statements

To Whom It May Concern,

I Peter James, prepared and structured the following paper using notes and materials prepared by Peter Hannay:

Hannay, P. & James, P. (2007), Pocket SDV with SDGuardian: A Secure and Forensically Safe Portable Execution Environment, *5<sup>th</sup> Australian Digital Forensics Conference*, Perth, pp 154-163.

I Peter James, confirm I prepared over 50% of the paper.



I Peter Hannay as a Co-Author, endorse that this level of contribution by the candidate indicated above is appropriate.



Peter Hannay

6<sup>th</sup> May 2015

To Whom It May Concern,

I Peter James, prepared and structured the following three papers using notes and materials prepared by Don Griffiths:

Griffiths, D & James, P. (2010), Fireguard – A Secure Browser with Reduced Forensic Footprint, *The Journal of Network Forensics*, Volume 2, Issue 2, pp 1-24.

James, P. & Griffiths, D (2012), The Mobile Execution Environment: A Secure and Non-Intrusive Approach to Implement a Bring Your Own Device Policy for Laptops, *10th Australian Information Security Management Conference*, Perth, pp 82-91.

James, P. & Griffiths, D (2014), A Secure Portable Execution Environment to Support Teleworking, *The Journal of Information Management and Computer Security*, Volume 22, Issue 3, pp 309-330.

I Peter James, confirm I prepared over 50% of each paper.



I Don Griffiths as a Co-Author, endorse that this level of contribution by the candidate indicated above is appropriate.



Donald Griffiths

7<sup>th</sup> May 2015