3-12-2007

Can SDV technology be utilised in a smartphone to prevent forensic analysis?

Peter James
Secure Systems Limited

DOI: 10.4225/75/57ad64ac7ff37
5th Australian Digital Forensics Conference, Edith Cowan University, Perth Western Australia, December 3rd 2007.

Can SDV Technology be Utilised in a Smartphone to Prevent Forensic Analysis?

Peter James[1]
Secure Systems
pjames@securesystems.com.au

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

[1] Peter James is registered on a Professional Doctorate programme at the School of Computer & Information Science at Edith Cowan University. Peter is the CEO of Secure Systems Ltd.

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.

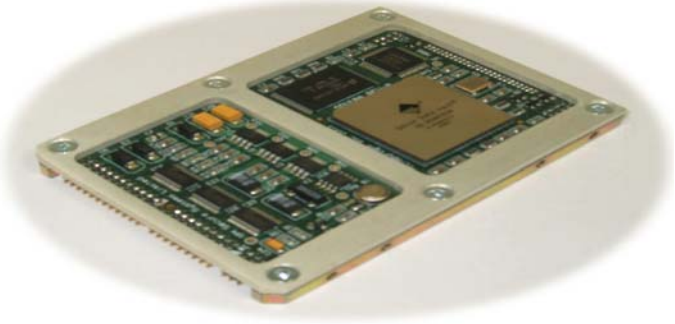

Figure 1 – Picture of Laptop SDV

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[2]. 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

[2] 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.

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.

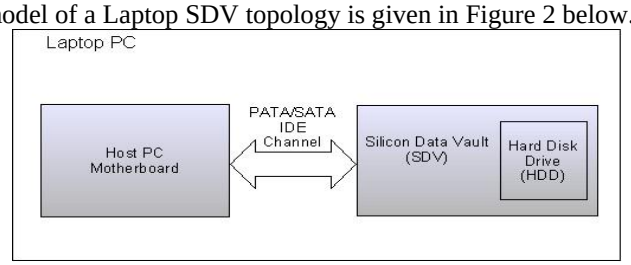

Figure 2 – Conceptual Model of Laptop SDV Topology

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.
- 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[3]

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.

[3] Image obtained from Palm Inc web site May 2007

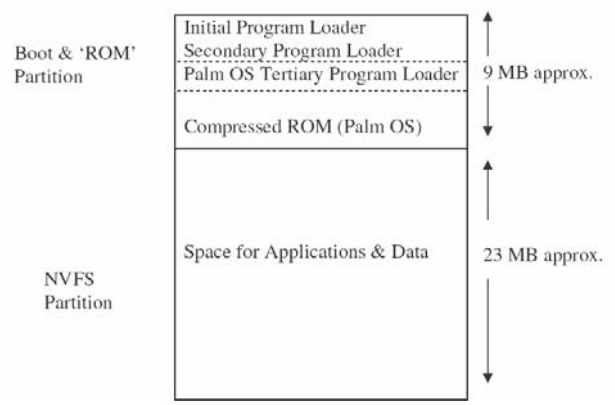

Figure 4 – Non-volatile NAND Flash Memory Map

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.

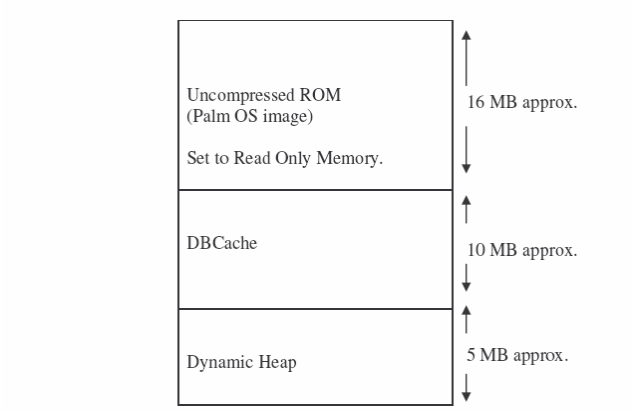

Figure 5 – Volatile SDRAM Memory Map

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[4]. 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

[4] 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

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 applications instruction or when space is required (typically once the DBCache exceeds 9MB). When application's 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.

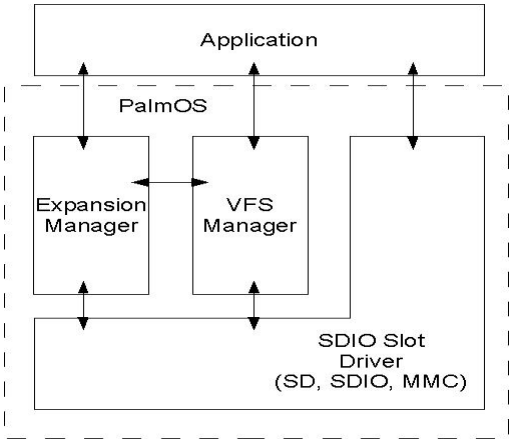

Figure 6 – Model of PalmOS libraries required to support SDIO card

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[5] 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

[5] No detailed documentation could be located on hardware design and schematics of the Treo 650.

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 it 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 "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[6]

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[7]

[6] Image obtained from SD Worldwide web site May 2007
[7] Image obtained from Palm Inc web site May 2007

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.
- 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.

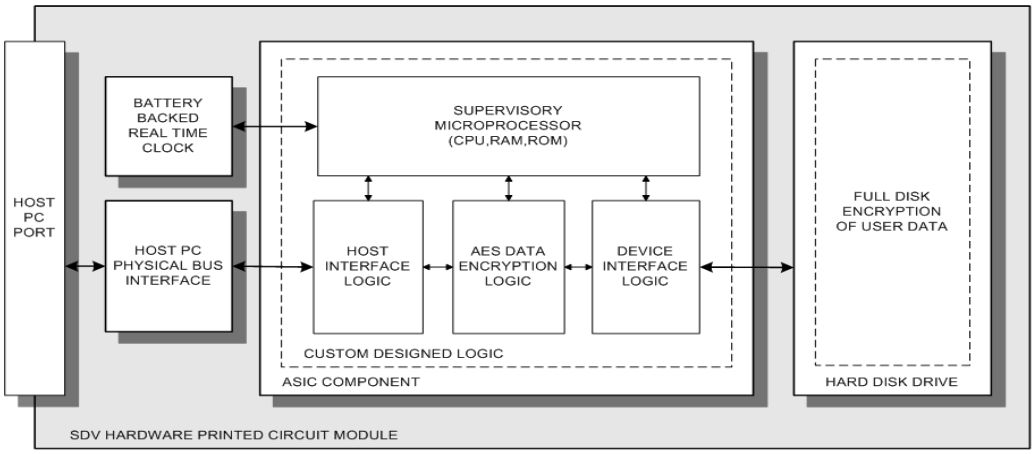

Figure 9 – Model of Key Components and Interfaces in SDV

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 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.

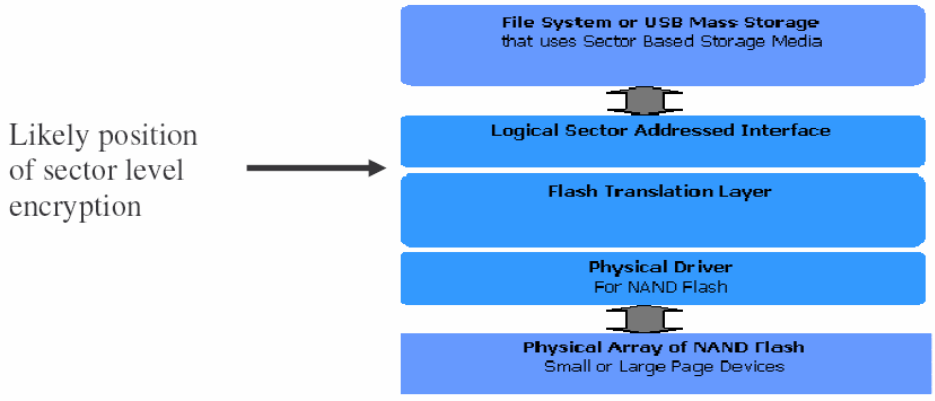

Figure 10 – Position of FTL in Flashing Memory Addressing Scheme

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 like 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 for 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_Inovation_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.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

COPYRIGHT