1-1-2011

Looking to iPhone backup files for evidence extraction

Clinton Carpene
Edith Cowan University

DOI: 10.4225/75/57b2b9e540ce9
9th Australian Digital Forensics Conference, Edith Cowan University, Perth Western Australia, 5th -7th December 2011

LOOKING TO iPHONE BACKUP FILES FOR EVIDENCE EXTRACTION

Clinton Carpene
School of Computer and Security Science, Edith Cowan University
Perth, Western Australia
ccarpene@our.ecu.edu.au

Abstract

iPhone logical backup files can provide forensic examiners with almost the entire contents of its host phone up until the point that the backup took place. This paper serves to provide an overview of the information attainable via the analysis of an iPhone backup, making references to the applicability of such analysis in the digital forensics field.

The paper introduces the backup directories for various common operating systems, and exposes the contents. Information about the property lists (plist files) containing information about the backed-up device and its contents are detailed, along with the mbdb/mbdx database files, and finally the extension-less backup files, is provided. Tools such as the iphonebackupbrowser, iPhone/iPod Backup Extractor and Oxygen Forensic Suite are discussed for their suitability with extracting iPhone backup data. Finally, a taxonomy of potential information of forensic interest is included, highlighting common filenames; the contained information; and their purpose in an investigation.

Keywords

iOS, iPhone, Forensics, Mobile, Smartphone, logical backup.

INTRODUCTION

The Apple iPhone is one of the most popular smartphones available today. With over 108 million iPhones reportedly sold as of March 2011 (Dediu, 2011), it is becoming crucial that these devices are considered during a forensic investigation. The capabilities of a smartphone such as the iPhone lend itself to housing potential evidence. Equipped with between 8 and 32GB of storage capacity, and with built in camera, email, social networking, SMS, calling capabilities and more, the iPhone is a potential gold mine for digital evidence. Unfortunately at present, no method of creating a forensically sound, raw image of the iPhone device exists without the phone being jailbroken (a term meaning the phone has been modified to install homebrew or custom/unlicensed applications) or altered in some fashion. This means that unless the phone is jailbroken, the integrity of the device will need to be compromised in order to extract evidence, or alternatively, logical evidence will need to be examined. Additionally, whilst the iPhone may be a desired article for examiners, the device might not always be present at a crime scene.

Another way to extract evidence from the phone, however, is to turn to the logical backups stored by the complementary application, iTunes. iTunes is a computer based (PC or Mac) application that interfaces with the iPhone, and is required upon initial setup to register the phone and transfer music to the device. iTunes, however, also maintains incremental backups of an iPhone, so that it can be restored in the event of a system failure, or upon receiving a new phone. Incredibly, these backup files are, by default, stored as unencrypted files in a set directory on the host computer, and the information stored within the files is simply staggering. Consequently, such investigations can even proceed in the absence of the suspect's device.

This paper overviews some of the tools currently available for analysis of iPhone backup data. The document also details the locations, contents, filetypes and common information uncovered through investigating an iPhone logical backup directory. Evidence in this report has been extracted from the author's iPhone 3GS mobile device (version 4.3), which was backed up using iTunes (version 10.2.2), however the procedure is relevant for all iPhone models, and iTunes version 9.2 upwards (slight variations exist in iTunes version 9.1 and below that will affect the structure of files listed in this document (rene.devichi, 2010b)). Some of the filenames, attributes and contents have been edited to protect the privacy of the author. Additionally, it is important to note that this document is a paper on the possible evidence recoverable from the iPhone backups, and not an actual forensic analysis of such a device. As a result most forensic procedures have been omitted from the paper. In legitimate situations, it is important to consider all standard forensic practices and local, state and federal laws and regulations regarding data acquisitions and analysis.

EVIDENCE EXTRACTION

BACKUP DIRECTORIES

iTunes stores all backup files in a directory on a host computer system. Depending on the operating system the directory varies. **Error! Reference source not found.** displays a list of common operating systems, and the expected, default location for iTunes backup files.

Table 3 – Common iPhone backup directories (AccessData, 2010)

OS	***Directory***	***Notes***
Windows 7/Vista	***<Systemroot>****:\Users****<Username>****\AppData\Roaming\Apple Computer\MobileSync\Backup*	*<SystemRoot> refers to the Drive letter of the System Drive (typically 'C') and <Username> refers to the user's home directory*
Windows XP	**<Systemroot>**:\Documents and Settings**<Username>**\Application Data\Apple Computer\MobileSync\Backup\	*<SystemRoot> refers to the Drive letter of the System Drive (typically 'C') and <Username> refers to the user's home directory*
Mac OSX	~/Library/Application Support/MobileSync/Backup/	*~/ refers to the users home directory.*

In these directories iTunes creates a subdirectory with the name of the device's Unique Device Identifier (UDID). The unique identifier is a string that is, according to Apple, guaranteed to be unique for each device, and takes the form of a 40 digit hash value of various device hardware identifiers (Apple Inc., 2010). The value of the UDID is stored in various parts of the iPhone backup file, such as the Info.plist, and Manifest.plist files, and is obtainable through iTunes when an iPhone device is connected. Stored within the backup folder is a plethora of files that will be discussed in the following section.

BACKUP FILETYPES

Property list (plist)

The plist file is Apple's proprietary Property List file format. The document uses XML to define data fields and attributes. Plist files are typically used to contain metadata, or properties and attributes pertaining to a device, application or files.

Table 4- Property List files in backup directory

File	***Description/Purpose***
Info.plist	*The information property list contains details about the iPhone device, such as name, model, firmware version, and identifiers. (Hook & Gaffaney, 2009)*
Manifest.plist	*This file contains a list on applications from the iPhone device. The Manifest.plist file's role has recently changed (as of iTunes 9.2), where it used to*

	perform the role of the previously non-existent Manifest.mbdb and Manifest.mbdx files. (Hook & Gaffaney, 2009; viaForensics, 2009)
Status.plist	*Status.plist contains information pertaining to the device's backup history. (viaForensics, 2009)*

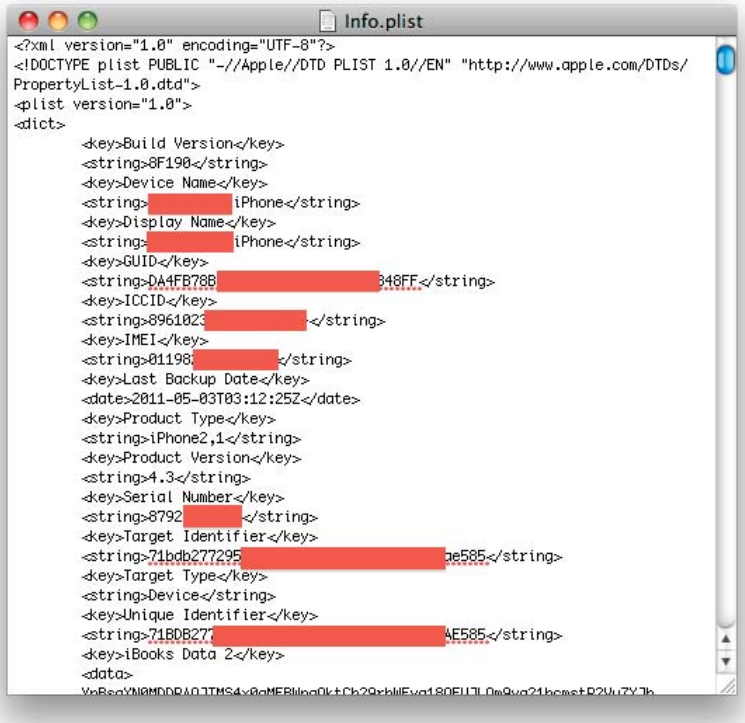

Figure 8 - Info.plist displayed in a primitive text editor

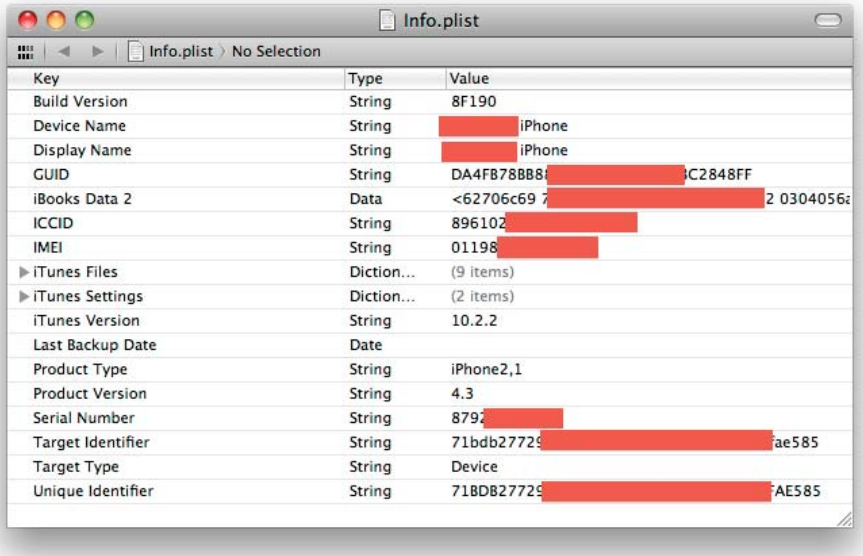

Figure 9 - Info.plist displayed in Apple's Xcode plist viewer.

Mbdb and Mbdx

The mbdb and mbdx files in the backup directory are database files containing the records of files that need to be backed up or restored to the iPhone device. According to programmer rene.devichi (2010b) the mbdb and mbdx files take over the role of the Manifest.plist file from previous iterations of iTunes (pre-iTunes version 9.2).

Table 5 - Mbdb and Mbdx files in the Backup directory

File	***File Header***	***Description/Purpose***
Manifest.mbdx	The first 4 bytes of the file are "6D 62 64 78" which translates to the ASCII "mbdx", denoting an mbdx file.	This file is an index file describing the data that needs to be backed up or restored to the iPhone device. *It contains; the Key of the file (discussed in the "*No file extension.*); and whether the file is a symbolic link, file or directory. (rene.devichi, 2010b)*
Manifest.mbdb	The first 4 bytes of the file are "6D 62 64 62" which translates to the ASCII "mbdb", denoting an mbdb file.	*This file is the database that stores information about the data to be backed up or restored to an iPhone device. Absolute file directory, Timestamp, hash values, file size, and User/Group IDs are stored in this database. (rene.devichi, 2010b)*

No file extension.

There are a number of files in the Backup/ directory that exist without a file extension. Whilst these files exist with no extension, it can be difficult at a glance to determine what the file actually is. *These files are, in fact, a number of different files and formats, including images, videos, voice recordings, sqlite databases, text documents, and other miscellaneous files, with their extension removed. The extension-less files contain all of the actual documents and files that have been backed up from the suspect phone.* A simple UNIX or Linux *file* command can be used to discern the property type of each file in the directory, as depicted in *Figure **10***.

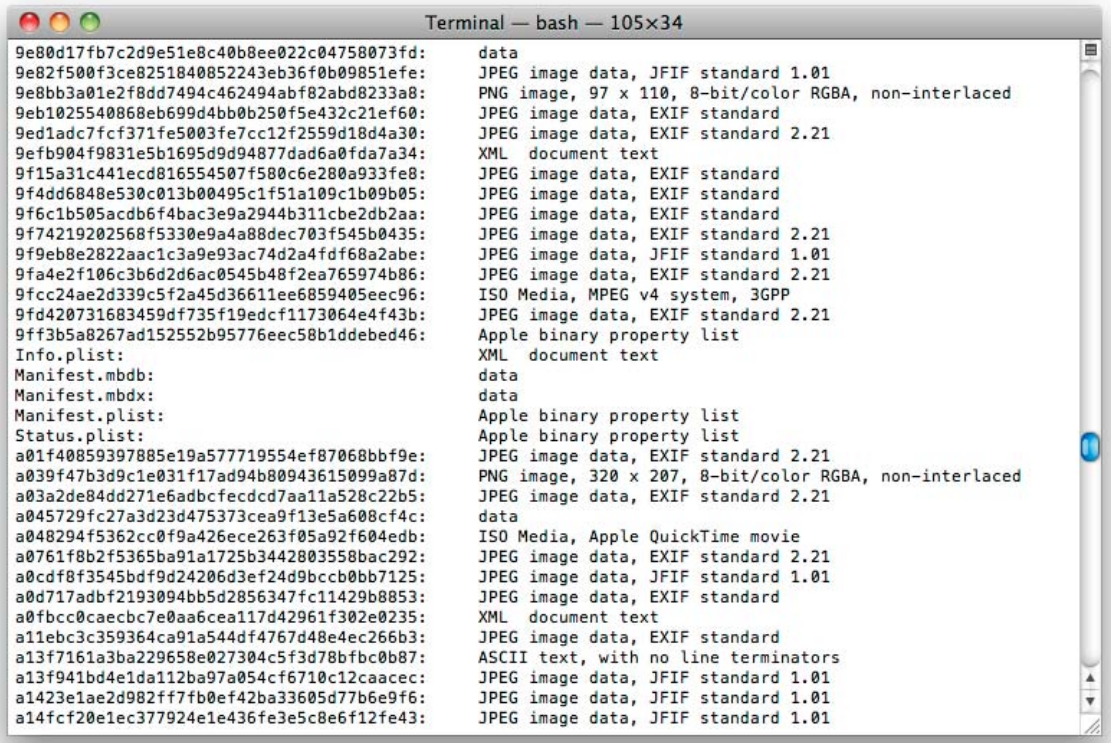

Figure 10 - UNIX file command displaying file types in the Backup directory

These files are characterised by a 40-character string, which is also the file's "key". In order to attain the file's key; its domain and its location on the filesystem is hashed using the SHA1 algorithm (Crosby, 2010). The hash 40-character hash value is then used as the file's filename and key across the backup process. The key is used to identify the file within the relevant plist and mbdb files.

As an example; the SMS database file (sms.db) is a member of the Home Domain and is located at Library/SMS/sms.db. The key for this file would be attained by taking a hash value of the following string; HomeDomain-Library/SMS/sms.db

This results in a hash value of 3D0D7E5FB2CE288813306E4D4636395E047A3D28, which is a valid filename in the Backup directory as evidenced below.

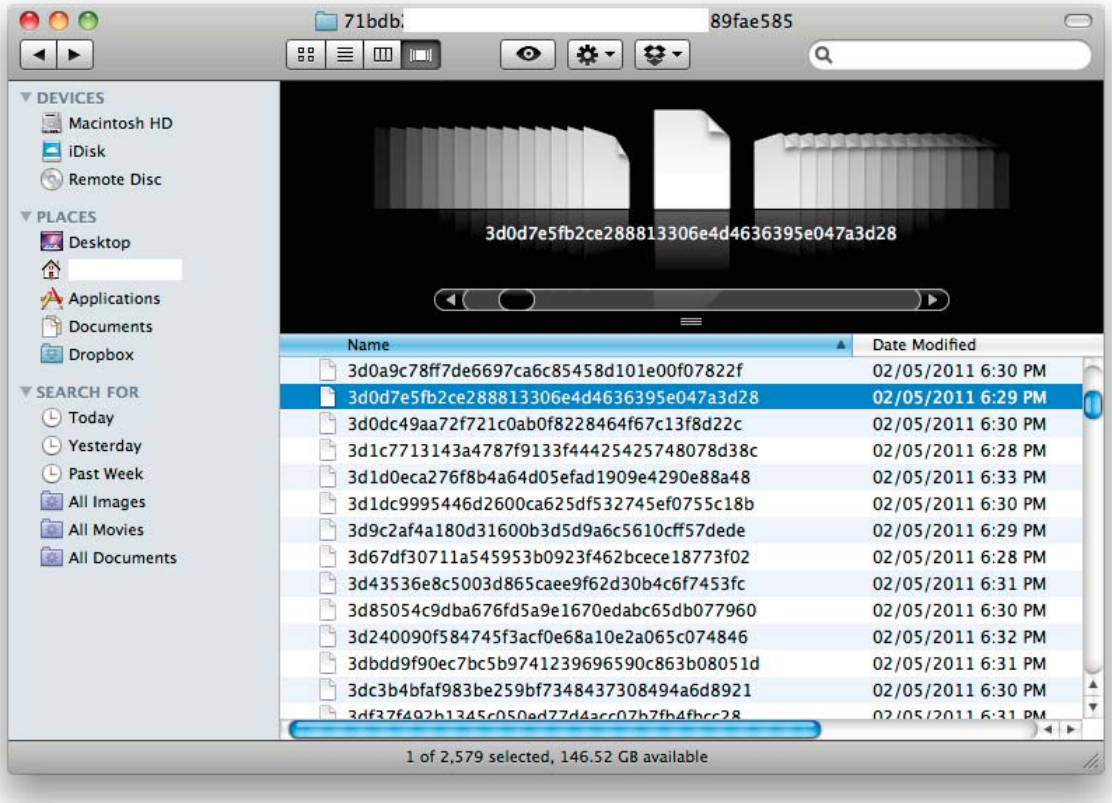

Figure 11 - Backup directory highlighting presence of sms.db.

TOOLS FOR EVIDENCE EXTRACTION

In order to analyse the iPhone backup data, an investigator should utilise software for assistance. Whilst any hex or text editor can be used to view the iPhone backup files, and attempt to piece together information about the device, dedicated software can allow the information to be presented in an easily understandable format. For this report, I will discuss the software that I found the most helpful when analysing the iPhone backups, however many additional options for interpreting the data do exist and may be more beneficial, depending on the scenario.

Iphonebackupbrowser

The "iphonebackupbrowser" (2010a) is an open source application, developed by Google Code developer rene.devichi, that is used to represent the information in the iPhone backup directory in a meaningful fashion to assist with data analysis (rene.devichi, 2010a).

Once launched, the application searches the backup directory (by default uses the iTunes default backup location, but can be manually defined) for the Info.plist file. Once found, the file is loaded into the application, along with the accompanying Manifest.mbdb, Manifest.mbdx, Manifest.plist and Status.plist files. Together, the information from these files is displayed as a list, by application name, as depicted in *Figure **12***. The relevant files associated to each application are listed in a sub window, along with the file's metadata, including the key that can be used to identify and execute the item in the backup directory. All of this information can be exported to a comma separated value (CSV) document using the "List" option.

At this point it should be noted that none of the information in the iPhone backup directory was altered through the use of the iphonebackupbrowser. The analysis was completely non-invasive. This was verified by taking a hash value of all the files in the directory (outputted to a log file), opening the application and browsing the data, and then repeating the hash values, and comparing the results using a Unix *diff* command. None of hashes differed and the files were seen to be identical.

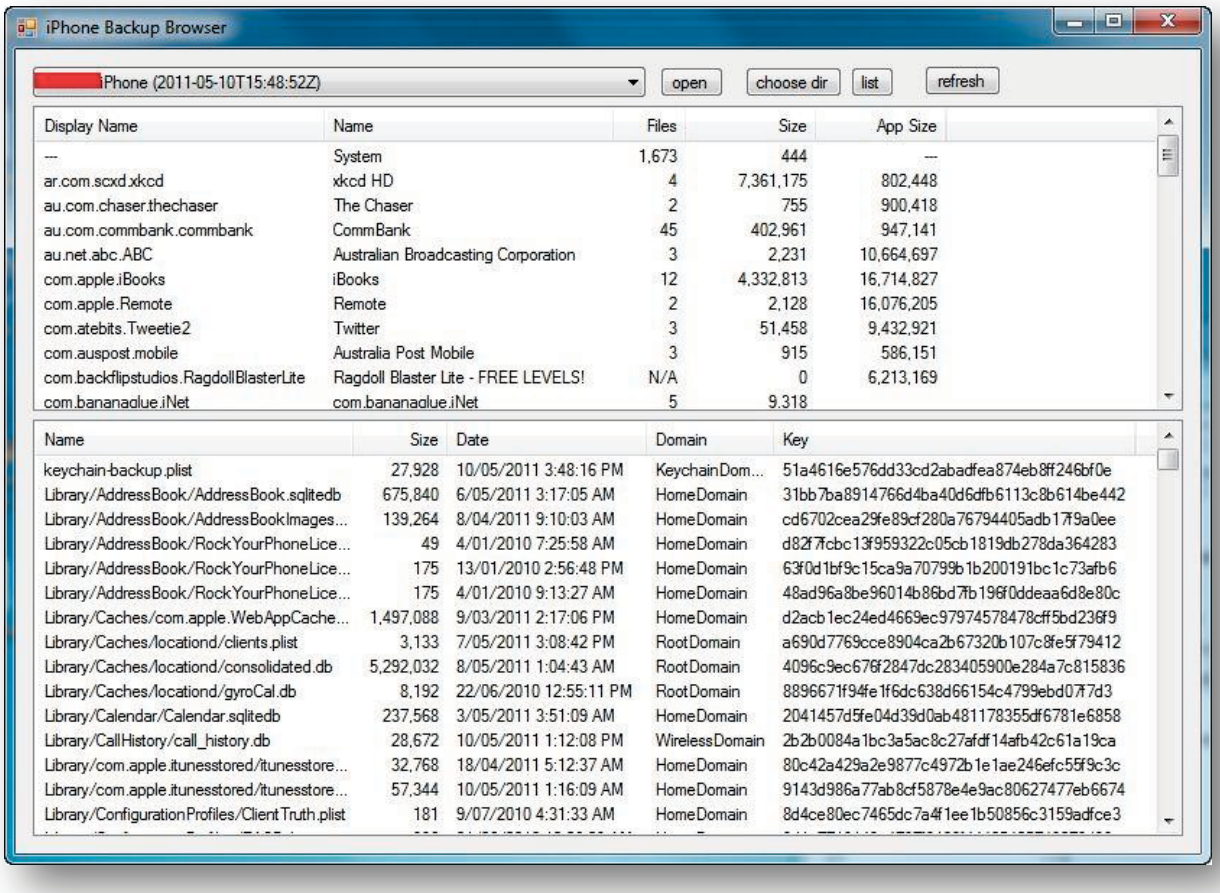

Figure 12 – "iphonebackupbrowser" (rene.devichi, 2010a) GUI with backup data loaded.

Oxygen Forensic Suite 2011:

The "Oxygen Forensic Suite 2011" is a software suite created by Oxygen Software Company (2011) that is used to acquire and analyse mobile device data for forensic investigations. The commercial, proprietary software includes the ability to acquire or use an existing logical backup of an iPhone device, and then extract and display the data. This software suite can aid significantly in streamlining the process of evidence extraction and analysis from the iPhone, even constructing a timeline of events from the device, however it is an expensive program to purchase.

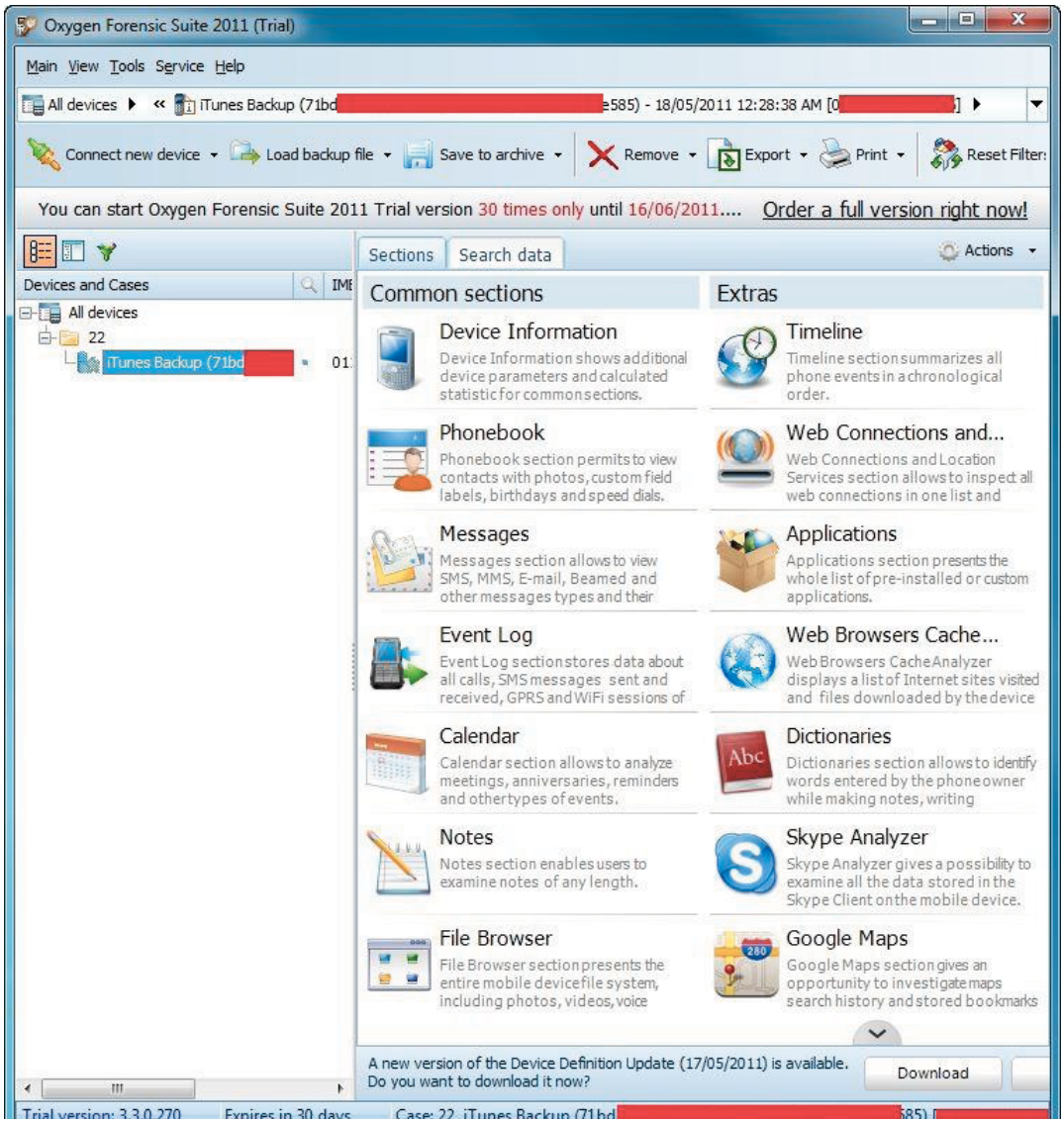

Figure 13 - "Oxygen Forensic Suite 2011" (Oxygen Software Company, 2011) GUI with backup files loaded.

iPhone/iPod Backup Extractor

The "iPhone/iPod Backup Extractor" (Pádraig, n.d.) is a simple Unix based application that can be used to extract data from an iPhone backup. The application works by parsing the information in the Manifest.mbdb file and categorising the data into relevant applications (similar to the "iphonebackupbrowser" application). The user is prompted to select an appropriate backup from a list. The user is then given the option to extract the data from each application into the directory of their choice by using the Extract function. The data that is extracted can then be natively opened using the correct utility for that filetype (e.g. Preview for photos, QuickTime for movie files, etc.)

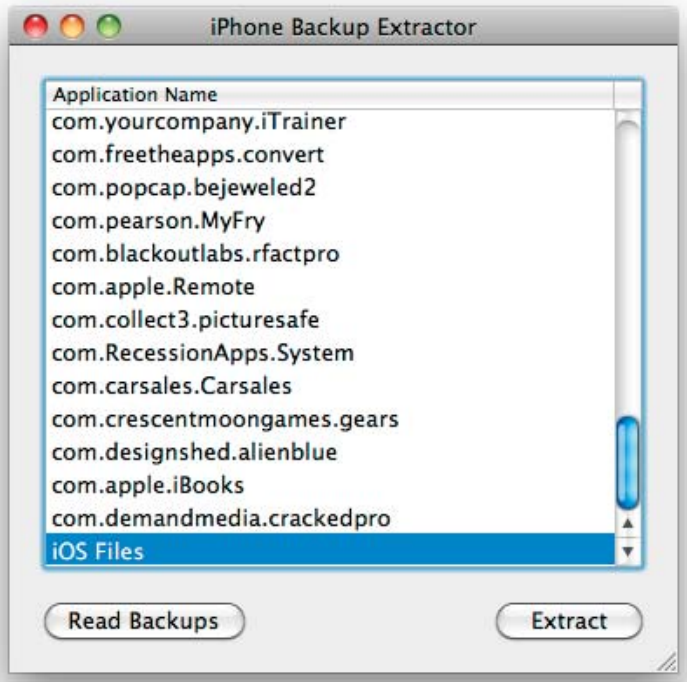

Figure 14 - User interface for "iPhone/iPod Backup Extractor"

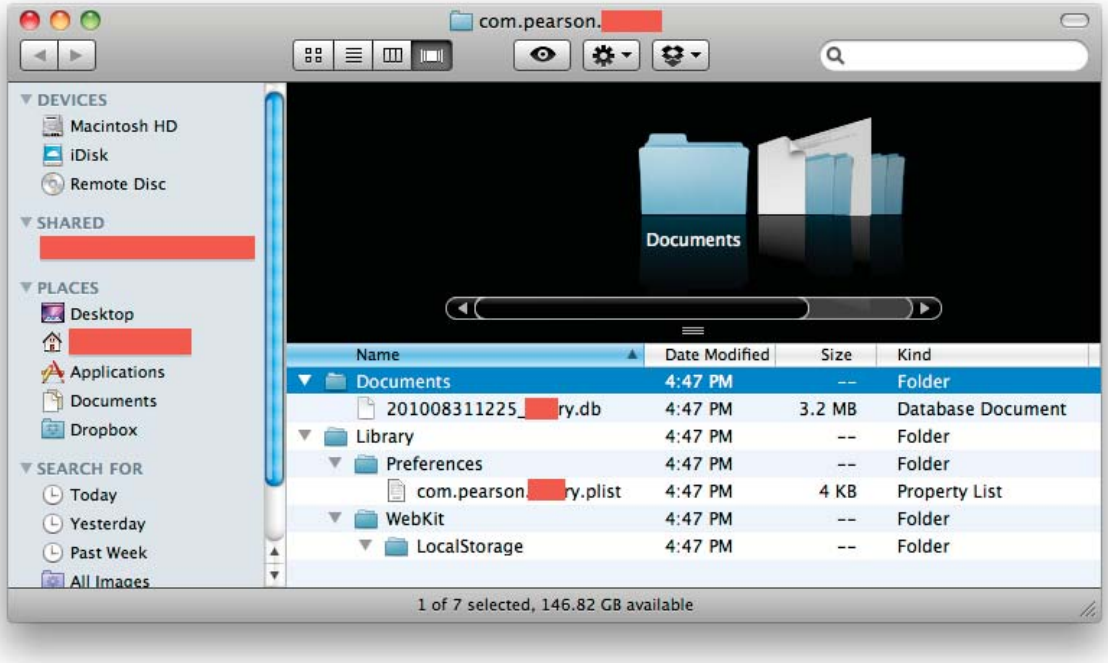

Figure 15 - Sample extracted application data from "iPhone/iPod Backup Extractor"

Mbdb Parser Script

The untitled mbdb parsing python script created by programmer galloglass (2010) can also be used to achieve a full dump of the otherwise incomprehensible mbdb file. The script outputs the information in the file into a Unix like *ls* format.

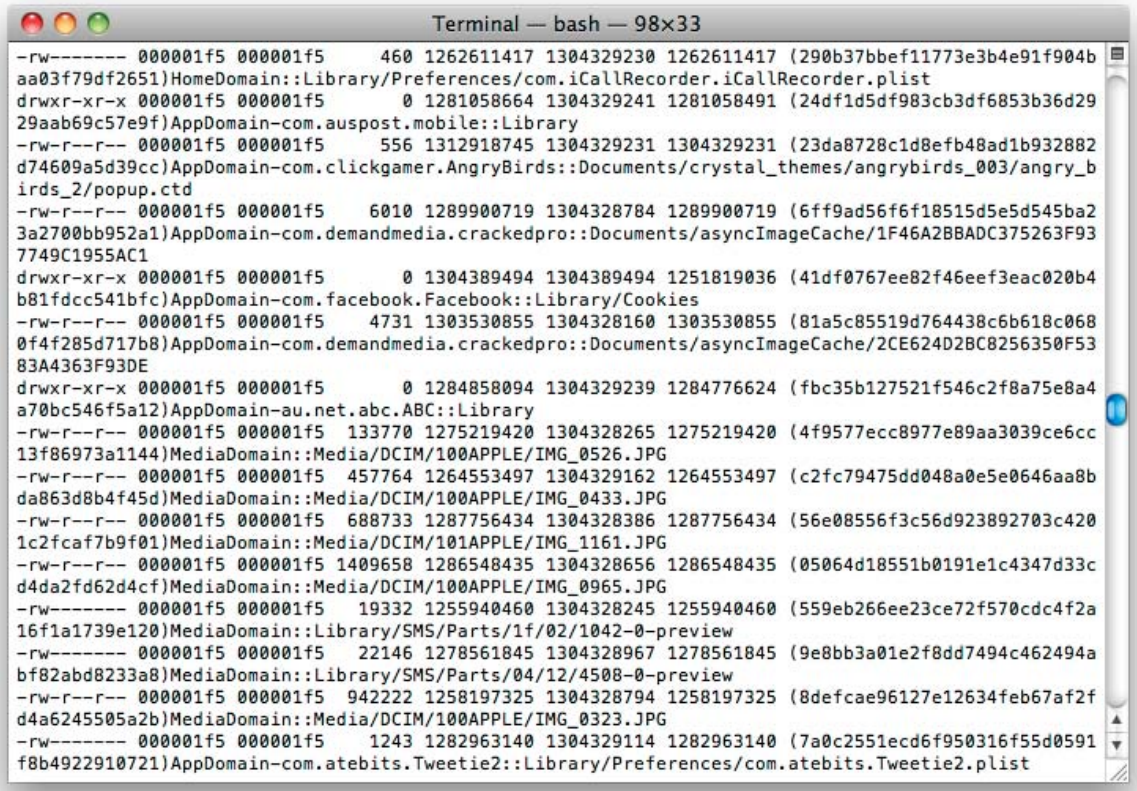

Figure 16 - Sample output of mbdb parsing python script

Plist Viewer:

When analysing iPhone backup files, a decent property list file viewer is important to have, as much of the device's system and application metadata is stored in various property list files across the filesystem. Fortunately there are many decent Plist viewers and Editors available that are suitable for the task. It is worth noting that the plist files exist in plain text (with the exception of binary property lists), so any text editor will be able to display the files, however it may be easier to interpret the data using a more sophisticated solution.

Apple's development kit "Xcode" (Apple Inc., 2011c) includes a plist viewer/editor that can be used to open the plist files found in the iPhone backup. "Xcode's" plist viewer is useful as it automatically aligns the data based upon the XML tags used for ease of interpretation, as displayed in Figure 9. Unfortunately this proprietary application is only available for Mac OSX. Whilst it can be purchased through the Mac App Store, it is free for members of the Apple Developer Program (Apple Inc., 2011b).

TAXONOMY OF POTENTIAL EVIDENCE

The following section details some of the potential evidence that can be extracted out of the logical iPhone backups. The filenames, directories and keys listed are applicable to current versions of iTunes, however are subject to unforseen changes in the future and therefore should only be used as a guide. The files without keys listed are dynamic, and are unlikely to have the same key on another device.

Device Information

The iPhone backups contain an array of information that can be used to identify the device that was backed up, and tie the device to that backup file. The following table lists the data attainable, its format, location, a description, and how it can be useful as evidence.

Data Label	Format	Example	Attribute Location	Description	Evidence Purpose
Device Name	Variable length string.	John's iPhone	Info.plist, Manifest.plist	The common name given to the device that was backed up in iTunes.	Can identify the device's owner by their name. Can also link the backup file to the backed up device.
Display Name	Variable length string.	John's iPhone	Info.plist	The common name given to the backup itself in iTunes.	Can identify the device's owner by their name. Can also link the backup file to the backed up device.
GUID	32 character hex string.	DA4FB78BB4456779296DC2498B4568FF	Info.Plist	The globally unique identifier (GUID) "is a unique hexadecimal number that is assigned to an object at the time that the object is created." (Tech-FAQ, n.d.)	Can be used to identify the device that was backed up.
ICCID	Either 20 digits, or 19 digits.	8961123456781087654	Info.Plist, Manifest.plist	The integrated circuit card identifier (ICCID) is assigned to a SIM card at manufacturing, and is supposed to be a unique identifier for the SIM card. The code is usually printed on the SIM as well as	The ICCID can be used to indicate a SIM card's presence in the backed up iPhone.

				being stored in the SIM (ETSI PT12, 1994)	
IMEI	Either 14 or 16 digits.	0112345674999 76	Info.Plist	The international mobile equipment identifier (IMEI) is designed to be a unique identifier for a mobile device. (ETSI PT12, 1994)	IMEI can be used to identify a mobile device has accessed a cellular network. Can also be used to further identify which mobile device was backed up.
Last Backup Date	Timestamp; Yyyy-mm-ddThh:mm:ssZ	2011-05-03T03:12:25Z	Info.Plist	The time that the backup was made.	Can be used to discern the backup age.
Product Type	Variable length string	iPhone2,1	Info.Plist, Manifest.plist	The model of the device that was backed up.	Can be used to identify the model of the device that was backed up in the absence of the physical phone.
Product Version	Variable Length String	4.3.2	Info.plist, Manifest.plist	The firmware version number of the backed up device.	
Serial Number	11 Character String	12345M456NQ	Info.plist	The device's serial number, which is a semi-unique identifier of the hardware itself.	Can be used to identify the device that was backed up. An also be cross referenced with Apple to identify original purchaser of device.
Target Identifier	40 Character Hex String	9D989E8D27D C9E0EC3389F C855F142C3D4 0F0C50	Info.plist	The device's unique identifier, created by hashing various hardware identifiers. Guaranteed by Apple to be unique (Apple Inc., 2010).	This is the key value for the backup directory, and is a guaranteed unique identifier, thus can be used to verify the device.
Unique Identifier	40 Character Hex String	9D989E8D27D C9E0EC3389F C855F142C3D4 0F0C50	Info.plist, Manifest.plist	The device's unique identifier, created by hashing various	This is the key value for the backup directory, and is a guaranteed

				hardware identifiers. Guaranteed by Apple to be unique (Apple Inc., 2010).	unique identifier, thus can be used to verify the device.

iPhone User Data

The iPhone stores a plethora of user data in its backup files. The following tables list common sources of potential evidence that can be analysed in an investigation. It is important to note that many 3rd party applications exist that may provide possible evidence. Consequently this report has focused on the built-in iPhone applications, as well as a select few common applications. In a genuine investigation all applications on the device would need to be analysed.

Location Data

File	Description	Key	Data Contained	Evidence Purpose
Library/Caches/locationd/consolidated.db	As has been recently noted in the media (Dilger, 2011; Whittaker, 2011) as of iOS version 4, the iPhone maintains a comprehensive database of location data, which it obtains through a combination of the built-in GPS module as well as Cellular and WiFi antennae.	4096c9ec676f2847dc283405900e284a7c815836	Timestamps GPS coordinates of Cell phone towers GPS coordinates of WiFi Access Points	Can be used to track a suspects locations at various times.

Photos Library

File	Description	Key	Data Contained	Evidence Purpose
Media/PhotoData/Photos.sqlite	The user's photo library. This is the iPhone's default location for storing photos and videos either taken by the device (using the camera) or saved from other locations (e.g. website, MMS, email, etc.). Note: Photos and videos may exist in other locations on the iPhone.	bedec6d42efe57123676bfa31e98ab68b713195f	Photos Videos Timestamps	Potentially incriminating photos or videos can be discovered.

Media/PhotoData/ PhotosAux.sqlite	If enabled, the Camera App saves location data for each photo or video taken.	0fc8189497f46a2e2 511c846acbbb318d 3a43ec3	Latitude Longitude Timestamps	Can be used to track location of photo's origin.

Safari Data

File	Description	Key	Data Contained	Evidence Purpose
Library/Safari/Bo okmarks.db	Users can use the Safari App to save bookmarks of their preferred websites. These bookmarks are stored in this database.	d1f062e2da26192a6 625d968274bfda8d 07821e4	Bookmark names Bookmark URLs	Track browser activity.
Library/Safari/His tory.plist	User browsing history is recorded in Safari automatically. The data can be manually cleared via the Safari App; otherwise the data is rarely cleared automatically.	1d6740792a2b845f 4c1e6220c43906d7f 0afe8ab	Recently viewed website URLs	Track browser activity.

Email Accounts

File	Description	Key	Data Contained	Evidence Purpose
Library/Preferenc es/com.apple.accou ntsettings.plist	The Mail App is the iPhone's email client. This file contains the email addresses for the configured accounts on the Mail App.	5fd03a33c2a311065 03589573045150c7 40721dd	Email Address	Link user to an online presence or email account.

SMS database

File	Description	Key	Data Contained	Evidence Purpose
HomeDomain-Library/SMS/sms.d b	*This is the database of SMS records used by the Messages App.*	*3D0D7E5FB2CE28 8813306E4D46363 95E047A3D28*	SMS Sent SMS received Sender/Recipient Phone number Timestamps	Can be used to trace correspondence between suspects.

Call history

File	Description	Key	Data Contained	Evidence Purpose
Library/CallHistor y/call_history.db	This database contains the call history information used by the Phone App.	2b2b0084a1bc3a5a c8c27afdf14afb42c 61a19ca	Call data monitoring Most recent 100 calls	Can be used to trace correspondence between suspects.

			Call durations Addresses Timestamps Country code.	

Address Book

File	Description	Key	Data Contained	Evidence Purpose
Library/AddressBook/AddressBook.sqlitedb	Contains the user's address book entries. The address book is populated using the "Contacts" application on the iPhone.	31bb7ba8914766d4ba40d6dfb6113c8b614be442	Contact Names Contacts email address. Phone number. Street address.	Can be used to trace correspondence between suspects.

Application Data

Applications refer to the third party software packages that can be downloaded and installed onto the iPhone via the App Store marketplace, or via iTunes. Currently there are over 350,000 apps available from Apple's App Store (Apple Inc., 2011a), with varying intent and functionality.

Application List

File	Description	Key	Data Contained	Evidence Purpose
Info.plist	Info.plist contains a list of the applications applicable to the backed up iPhone device. Note: the uniform iPhone applications (such as Messages, Phone, Safari, etc.) do not appear in this list.	N/A	Applications installed on device. Full iTunes applications library.	Can be used to determine which applications are or have been installed on a device. This information can then be used to determine more possible locations for evidence (e.g. data concealing applications, etc.)

Facebook data

Facebook is a popular social networking platform that allows users to interact with each others via notification broadcasts (called statuses), messages or event invites. Given its present popularity, with over 74 million active users of the Facebook iPhone App (WebMediaBrands Inc., 2011), it should be considered for evidentiary purposes.

File	Description	Key	Data Contained	Evidence Purpose
Library/CallHistory/call_history.db	Database of calls made to Facebook friends.	4402f91c8b7ec6cc473400a6e6074286a9c76399	Facebook Call history Call Timestamps	Can be used to trace correspondence between suspects.

Documents/friends.db	Database of cached Facebook friend information	6639cb6a02f32e0203851f25465ffb89ca8ae3fa	Friend names Profile URL Hashed email address Phone numbers	Can be used to trace correspondence between suspects.

Skype

Whilst much of the data contained within the Skype directory of the iPhone has been encrypted to a degree, a large amount of information pertaining to the accounts used on the device, and some of the calls made/received, can be inferred simply by perusing the *Library/Application Support/Skype* directory on the iPhone. When a new account is logged into the mobile Skype application, a new subdirectory is made with the name of the logged in "Username" (usually and email address, but may be abbreviated). Contained within this directory is information pertaining to that particular user's application usage history.

File	Description	Key	Data Contained	Evidence Purpose
Library/Application Support/Skype/<Username>/	Per-user Skype data storage directory.	N/A	Limited Call history Limited Chat History Limited Friend Data	Can be used to trace correspondence between suspects.

CONCLUSION

The iPhone backup files can provide forensic examiners with a wealth of information pertaining to a suspect's iPhone. By looking to iPhone backup files for evidence extraction examiners do not risk compromising the contents of a live device, whilst still maintaining a forensically sound, replicable, method of evidence gathering. Information such as cell phone call and SMS history, email accounts, facebook friends, applications lists and various device identifiers, are all readily available to the examiner. Furthermore, the device itself may not even be required for the investigation to take place. Unfortunately a caveat to the process is that the evidence extracted is not a raw image of the device, and rather a logical set of data, which isn't as desirable, however if available should be considered. Additionally, if the iPhone user has encrypted the backup files using the iTunes option, then there is less of a chance of attaining comprehensible data from the device (although it can still be achieved by breaking the password, or by jailbreaking and acquiring a raw disk image of the device). Overall however, the benefits of examining the iPhone logical backup files outweigh the detractors of the exercise, and like all possible evidence, should be considered for suitability within an investigation.

REFERENCES

AccessData. (2010). Macintosh System Artifacts Retrieved May 17th, 2011, from http://accessdata.com/downloads/media/MAC_Shortcuts.pdf

Apple Inc. (2010). UIDevice Class Reference. *iOS Developer Library* Retrieved May 7th, 2011, from http://developer.apple.com/library/ios/ - documentation/uikit/reference/UIDevice_Class/Reference/UIDevice.html

Apple Inc. (2011a). Apps for iPhone Retrieved May 17th, 2011, from http://www.apple.com/iphone/apps-for-iphone/

Apple Inc. (2011b). Download Xcode 4 Retrieved May 15th, 2011, from http://developer.apple.com/xcode/

Apple Inc. (2011c). Xcode 4 (Version 4.0) [SDK]. California.

Crosby, A. (Producer). (2010, May 10th, 2011). iPhone Forensics, sans iPhone. [Keynote Presentation]

Dediu, H. (2011). Review of Apple's unit numbers released in legal filing prior to earnings Retrieved May 5th, 2011, from http://www.asymco.com/2011/04/19/review-of-apples-unit-numbers-released-in-legal-filing-prior-to-earnings/

Dilger, D. E. (2011). Apple, Pandora, Backflip sued over iPhone data privacy Retrieved May 11th, 2011, from http://www.appleinsider.com/articles/11/05/10/class_action_suit_filed_against_apple_pandora_backflip_over_iphone_data_privacy.html

ETSI PT12. (1994). Specifications of the SIM-ME Interface *GSM 11.11*.

galloglass. (2010). Mbdb Python Script. Retrieved from http://stackoverflow.com/questions/3085153/how-to-parse-the-manifest-mbdb-file-in-an-ios-4-0-itunes-backup/3130860 - 3130860

Hook, A., & Gaffaney, K. (2009). iPhone Forensics-MDBackup Extract. *ViaForensics, Jun*.

Oxygen Software Company. (2011). Oxygen Forensic Suite 2011 (Version 3.3).

Pádraig. (n.d.). iPhone / iPod Touch Backup Extractor (Version 1.2.3). Retrieved from http://supercrazyawesome.com/

rene.devichi. (2010a). iphonebackupbrowser (Version rev 31). Retrieved from http://code.google.com/p/iphonebackupbrowser/

rene.devichi. (2010b, 17th July, 2010). MbdbMbdxFormat Retrieved May 5th, 2011, from http://code.google.com/p/iphonebackupbrowser/wiki/MbdbMbdxFormat

Tech-FAQ. (n.d.). Active Directory Terminology and Concepts Retrieved May 16th, 2011, from http://www.tech-faq.com/active-directory-terminology-and-concepts.html

viaForensics. (2009). Forensic analysis of iPhone backup directory Retrieved May 17th, 2011, from http://viaforensics.com/iphone-forensics/forensic-analysis-iphone-backup-directory.html

WebMediaBrands Inc. (2011). Application Statistics: Facebook for iPhone Statistics Retrieved May 17th, 2011, from http://statistics.allfacebook.com/applications/single/facebook-for-iphone/6628568379/

Whittaker, Z. (2011). Apple responds to Congress: Questions still remain over users' location data Retrieved May 11th, 2011, from http://www.zdnet.com/blog/igeneration/apple-responds-to-congress-questions-still-remain-over-users-location-data/9892