A Simple Cost-Effective Framework for iPhone Forensic Analysis

Mohammad Iftekhar Husain[1], Ibrahim Baggili[2], and Ramalingam Sridhar[1]

[1] Department of Computer Science and Engineering, University at Buffalo,
The State University of New York, Buffalo, NY 14260
[2] College of Information Technology, Zayed University, UAE
{imhusain,rsridhar}@buffalo.edu, ibrahim.baggili@zayed.ac.ae

Abstract. Apple iPhone has made significant impact on the society both as a handheld computing device and as a cellular phone. Due to the unique hardware system as well as storage structure, iPhone has already attracted the forensic community in digital investigation of the device. Currently available commercial products and methodologies for iPhone forensics are somewhat expensive, complex and often require additional hardware for analysis. Some products are not robust and often fail to extract optimal evidence without modifying the iPhone firmware which makes the analysis questionable in legal platforms. In this paper, we present a simple and inexpensive framework (iFF) for iPhone forensic analysis. Through experimental results using real device, we have shown the effectiveness of this framework in extracting digital evidence from an iPhone.

Keywords: iPhone, Forensics, Smartphone, Jailbreaking, iTunes.

1 Introduction

The Apple iPhone is among the most popular smart phones on the market, since its release in July 2007. According to a recent report on market share of mobile devices by Gartner [1], Apple's share of worldwide smart phone sales grew from 5.3 percent in the first quarter of 2008 to 10.8 percent in the first quarter of 2009. In terms of unit sales, iPhone jumped from 1.7 million in the first quarter of 2008 to 3.9 million during the same period in 2009. Though many smart phones have functionalities similar to iPhone, user interface and prevalence of numerous applications make them popular among many. The iPhone 3rd Generation Cellular Communication device, widely known as iPhone 3G was released in July, 2008 which has featured GPS service and faster Internet connection. Considering the mobility and functional similarity to standard computing devices, experts predict that iPhone can soon become a handy choice for cyber criminals. So, it is important for forensic community to focus on developing sound forensic methods for iPhone, forecasting the potential use of it in cyber crimes.

There are efforts from both commercial and individual forensic experts on iPhone forensics. Commercial products include Aceso by Radio Tactics [2], UFED from Cellebrite [3], Device Seizure by Paraben [4], .XRY by Micro Systemation [5] and

I. Baggili (Ed.): ICDF2C 2010, LNICST 53, pp. 27–37, 2011.

CellDEK by LogiCube [6]. However, these products can be expensive (up to multiple thousand dollars), requires additional hardware and functionality is limited only to the built-in features provided. Also, some approaches alter the firmware of iPhone to access the storage area using a method widely known as "jailbreaking" which is copyright infringement and illegal [7]. It also violates the Association of Chief Police Officers (ACPO) guideline for computer forensics and electronic evidence [8], which clearly states that "No action taken by law enforcement agencies or their agents should change data held on a computer or storage media which may be subsequently be relied upon in court."

In this paper, we propose a forensic framework for iPhone which simple to perform and free from the requirement of additional devices. In addition, this approach does not alter the iPhone firmware which keeps the digital evidence acceptable in legal venues. Using an iPhone device, we show the effectiveness of the framework in retrieving various digital artifacts. Additionally, we show the soundness of the evidence through comparisons to existing approaches using forensic standards. A preliminary version of this framework was tested on iPhone instant messaging forensics in [9].

2 Literature Review

The Apple iPhone OS is an optimized version of Mac OS X. There are two partitions on the iPhone storage device. The first partition is the system partition (approx. 300 MB). This partition includes the operating system and the default applications. The remaining space is partitioned as the user data (or media) partition. This space is where all music, contact, SMS as well as other user data are stored. When an iPhone is connected to a computer, it communicates with it using Apple File Communication protocol and creates a backup folder of user and device configuration data on it. Forensic acquisition of iPhone data can take different approaches such as acquiring the backup folder to analyze available data or obtain a physical image of the storage device.

Commercially available iPhone forensic products such as Aceso, UFED, Device Seizure, .XRY and CellDEK have some common drawbacks. Some products require additional hardware to perform the forensic analysis such as Aceso, UFED and CellDEK . Prices of most products vary from one to fifteen thousand USD according to our survey [10]. In addition, none of these solutions guarantees a complete recovery of device data.

Individual effort such as Zdziarski [11] approaches this problem through a bit-by-bit copy of the physical data in the iPhone. However, this approach modifies a read-only system partition which may eventually make the evidence questionable at legal venues. Forensic experts [12] extensively reviewed this approach and commented "I feel certain that without 15+ years of highly technical experience, I would have likely failed or would have certainly taken much longer to succeed (and perhaps sacrifice the iPhone data a few times along the way)." Efforts that use "jailbreaking" modify the user data partition of the iPhone opening it for legal challenges according to ACPO guideline.

3 Proposed Method

We propose a simple and cost-effective forensic framework for iPhone. Our framework contains all three phases of forensic data acquisition, data analysis and data reporting. Figure 1 depicts the overall structure of proposed iPhone forensic framework (iFF).

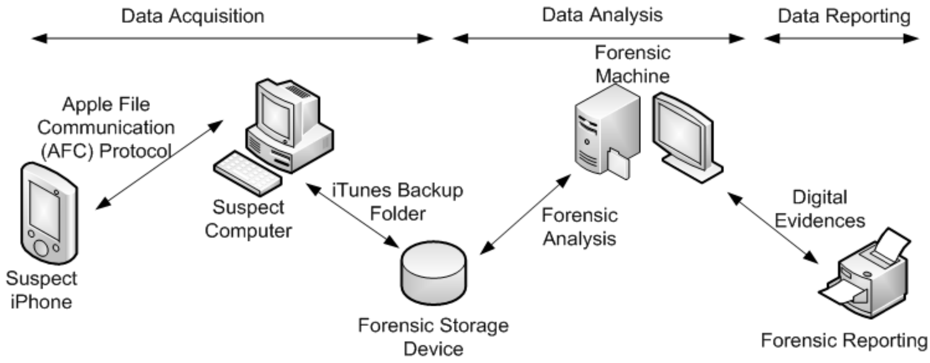


Fig. 1. Framework for iPhone Forensic Analysis

3.1 Data Acquisition

In our framework, forensic data acquisition from the suspect iPhone is based on acquiring the iPhone backup data from a machine on which the iPhone synchronized iTunes software exists. Alternatively, a forensic investigator can force backup an iPhone to a forensic examination machine using iTunes. On a Windows machine, the iTunes software saves logical copies of files on iPhone at: C:/Users/UserName/ AppData/Roaming/AppleComputer/MobileSync/Backup. By right-clicking on the device icon, when the iPhone is connected to a computer via iTunes, one can choose the backup option to backup a logical copy of iPhone data. Once the folder is acquired, the forensic investigator can use appropriate data integrity techniques and store a copy of it to a designated forensic storage device for further analysis.

3.2 Data Analysis

Depending on the iPhone firmware version, the iTunes backup folder might contain slightly different contents. In firmware version 2.0 and older, it contains mdbackup files where the actual data and the metadata reside together. The metadata describes where the actual data exists originally on the device. In some 2.0 and all 3.0 versions, the data and metadata are kept in two separate files, .mdinfo file contains only the metadata portion and .mddata contains the actual data. These binary files need to be processed to be human readable lists and databases. In our framework, we perform the data analysis as shown in Figure 2.

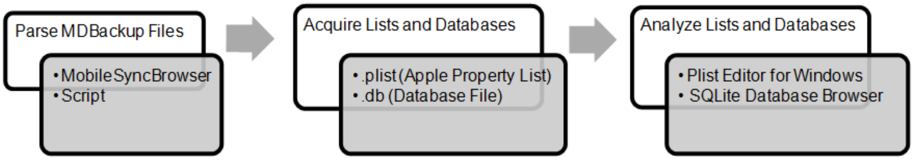

Parse MDBackup Files
• MobileSyncBrowser
• Script
Acquire Lists and Databases
• .plist (Apple Property List)
• .db (Database File)
Analyze Lists and Databases
• Plist Editor for Windows
• SQLite Database Browser

Fig. 2. Data Analysis Framework

We use MobileSyncBrowser (MSB) [13] to parse the binary backup files into lists and databases. To analyze database files, we use SQLite Database Browser [14]. Plist Editor for Windows [15] is used to analyze the Apple Property List files. Although we have used these three softwares in our current experiment, our framework is generic in the sense that a forensic investigator can choose other software as long as it serves the required purpose of data parsing and analysis. For example, one can write a simple script to parse the mdbackup files instead of MSB and choose other softwares to read databases and lists.

3.3 Data Reporting

Depending on the nature of retrieved evidence, the forensic investigator can use suitable reporting format to present it to appropriate authority. This report may be a written report, oral testimony, or some combination of the two. The investigator should adhere to RFC 4998 "Evidence Record Syntax (ERS)" [16] whenever possible.

4 Experimental Results

For this experiment, we used an iPhone third generation with firmware version 3.0 which is not jailbroken. The phone was heavily used including: Email, contacts and calendar, web browsing, phone calls, text messages, multiple Wi-Fi networks, camera images, and songs via iTunes, YouTube movies, Google maps and notes. For privacy reasons, personal information will be redacted as needed throughout the experiment.

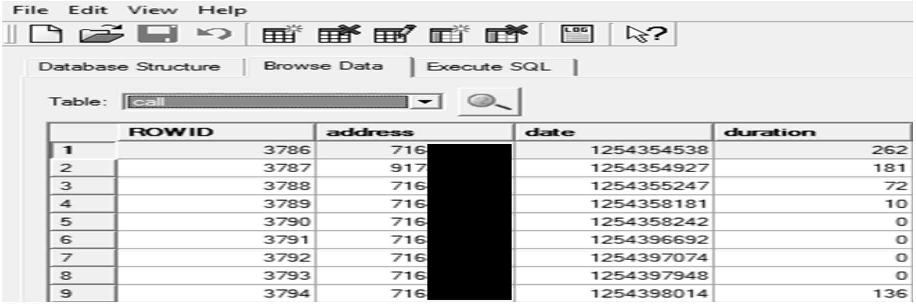
The following is a list of digital evidence that can be found by analyzing the data from iPhone using our proposed framework.

4.1 Voice Communication Related Evidence

The backup of iPhone data contains most of call related information. For example: the *callhistory.db* (figure 3) file under Library folder contains recent call history with timestamp and duration. The *voicemail.db* file contains information regarding the received voicemails including callback number, timestamp and duration. Each voicemail is given a unique identifier. The actual content of the voicemail is saved as identifier.amr narrow band content file. A *.token* file contains credential which is used to retrieve voicemails from the cellular provider.

4.2 Text Communication Related Evidence

Three types of text communication related evidence can be recovered from the backup folder: e-mail, instant messaging (IM) and short text messaging (SMS). Information



	ROWID	address	date	duration
1	3786	716[redacted]	1254354538	262
2	3787	917[redacted]	1254354927	181
3	3788	716[redacted]	1254355247	72
4	3789	716[redacted]	1254358181	10
5	3790	716[redacted]	1254358242	0
6	3791	716[redacted]	1254396692	0
7	3792	716[redacted]	1254397074	0
8	3793	716[redacted]	1254397948	0
9	3794	716[redacted]	1254398014	136

Fig. 3. Voice Communication Related Evidence

on the e-mail accounts set up on built in iPhone application can be found at *com.apple.accountsettings.plist* file (figure 4). It includes the SMTP and IMAP server name, authentication type, user name and e-mail address. A significant amount of information on instant messaging can also be found. For example, Encrypted password and Yahoo! ID can be found in *com.yahoo.messenger.plist* file in preferences folder. This file also contains the time when a particular user last accessed the IM service from the iPhone. Conversations with timestamps are found in yahoo-*accountname.db* file. *yAddressBook_accountname.xml* contains the buddy list. Evidence from the conversation are found at the *session.log.db* file. Similar information can be recovered for other IM applications such as AIM. The file *sms.db* contains the entire short messages including timestamp and contents.

4.3 Network Related Evidence

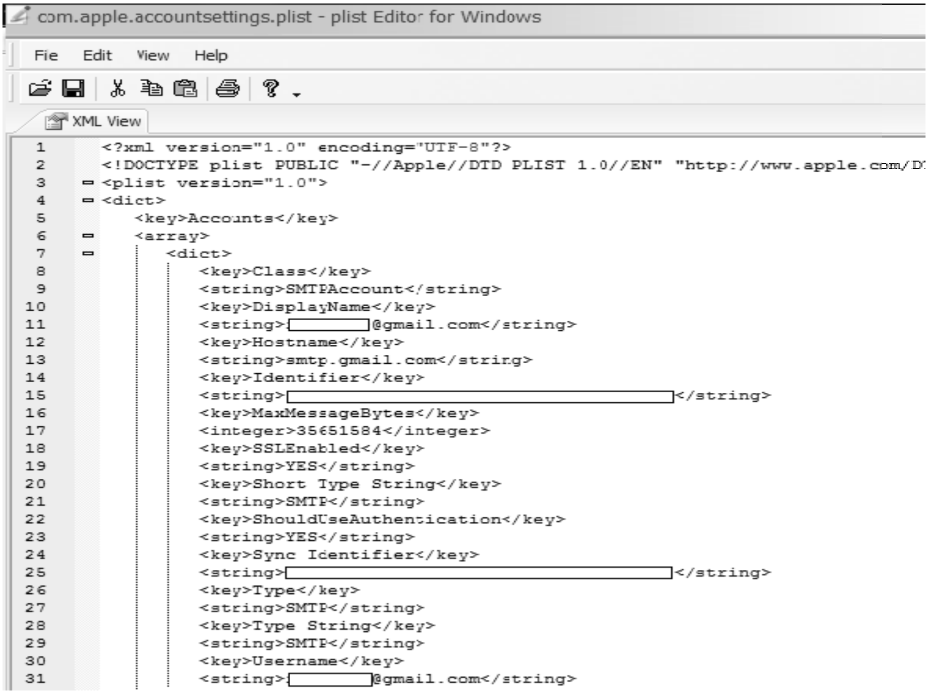
Logical acquisition data from an iPhone contains both cellular and Wi-Fi network related information. For example, *com.apple.wifi.plist* file contains the name of all the networks accessed including their SSID, type of encryption, user name, date and time the network was last accessed (figure 5). Historical network TCP/IP assignments with timestamps can be found at *com.apple.network.identification.plist*. The file CSIDATA contains GSM network options and settings.

4.4 Audio-Visual Evidence

The DCIM folder contains images captured by the iPhone as .jpeg files. Each image contains the date and time when the image was originally captured as well as other image properties such as resolution, bit depth, and color scheme. Some iPhone screenshots are also saved in this folder as *.png* files. The Recordings folder contains recorded voice memos as *.m4a* (Mpeg 4 Audio) files. A separate database file *recordings.db* contains the list of all recorded files with duration time.

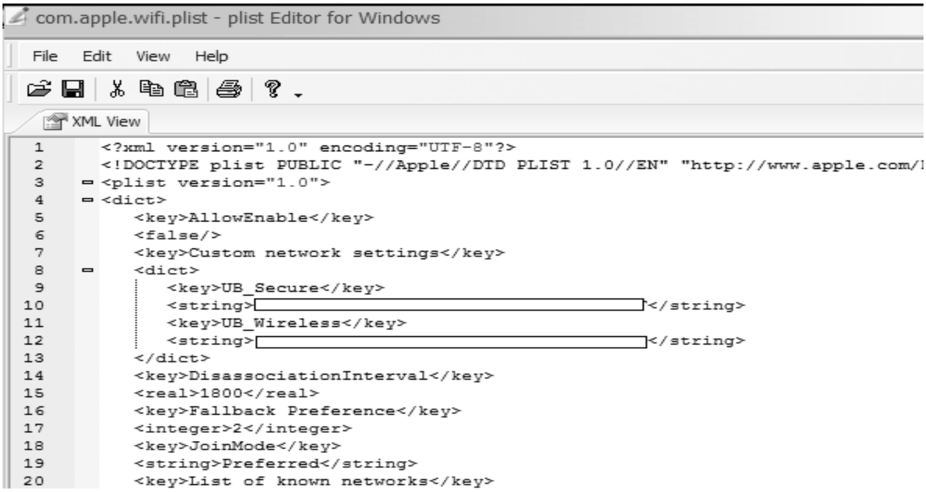
4.5 Location Related Evidence

Because of the default GPS capability, the iPhone backup folder contains a considerable amount of location related information. The *history.plist* file at Map folder



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/D
<plist version="1.0">
<dict>
    <key>Accounts</key>
    <array>
        <dict>
            <key>Class</key>
            <string>SMTPAccount</string>
            <key>DisplayName</key>
            <string>[redacted]@gmail.com</string>
            <key>Hostname</key>
            <string>smtp.gmail.com</string>
            <key>Identifier</key>
            <string>[redacted]</string>
            <key>MaxMessageBytes</key>
            <integer>35651584</integer>
            <key>SSLEnabled</key>
            <string>YES</string>
            <key>Short Type String</key>
            <string>SMTP</string>
            <key>ShouldUseAuthentication</key>
            <string>YES</string>
            <key>Sync Identifier</key>
            <string>[redacted]</string>
            <key>Type</key>
            <string>SMTP</string>
            <key>Type String</key>
            <string>SMTP</string>
            <key>Username</key>
            <string>[redacted]@gmail.com</string>
```

Fig. 4. Text Communication Related Evidence



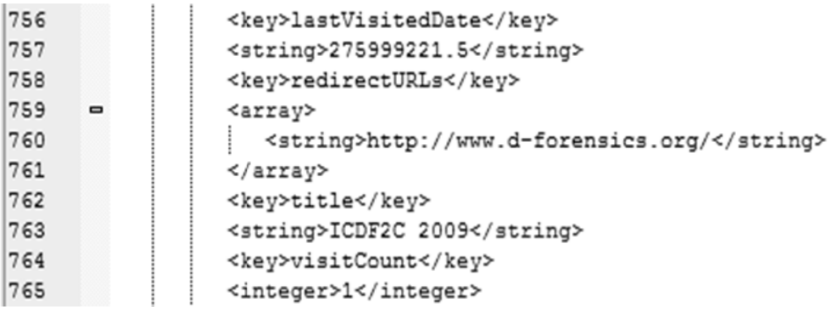
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/I
<plist version="1.0">
<dict>
    <key>AllowEnable</key>
    <false/>
    <key>Custom network settings</key>
    <dict>
        <key>UB_Secure</key>
        <string>[redacted]</string>
        <key>UB_Wireless</key>
        <string>[redacted]</string>
    </dict>
    <key>DisassociationInterval</key>
    <real>1800</real>
    <key>Fallback Preference</key>
    <integer>2</integer>
    <key>JoinMode</key>
    <string>Preferred</string>
    <key>List of known networks</key>
```

Fig. 5. Network Related Evidence

contains the information of places searched by the iPhone user. *com.apple.maps.plist* file contains the information on last searched location and last user location with timestamp.

4.6 Online Activity Related Evidence

Online activity of the user leaves a large amount of traces in the backup data. In the Safari folder, *bookmarks.plist* contains the bookmarks saved by the user. *history.plist* contains the list of all URLs visited by the user with timestamp and visit count (figure 6). *suspendedStates.plist* saves the information of the web pages being accessed before the safari application was suspended. *com.apple.youtube.plist* contains the accessed YouTube video history (figure 7). Information regarding online games can also be found.

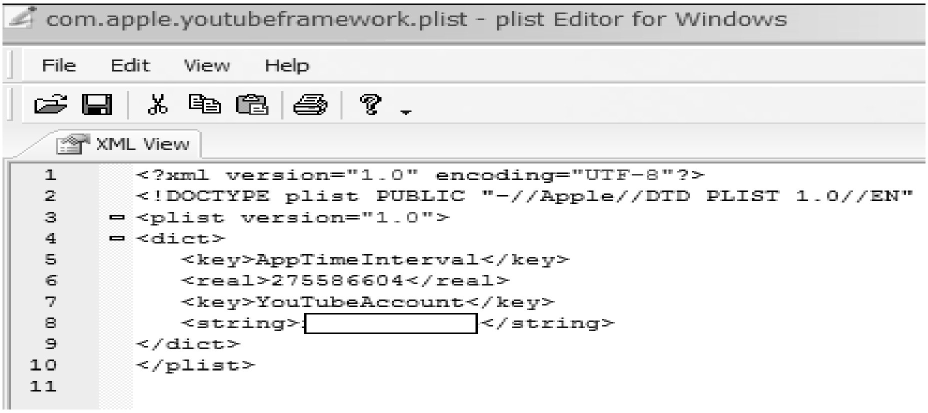


```
<key>lastVisitedDate</key>
<string>275999221.5</string>
<key>redirectURLs</key>
<array>
    <string>http://www.d-forensics.org/</string>
</array>
<key>title</key>
<string>ICDF2C 2009</string>
<key>visitCount</key>
<integer>1</integer>
```

Fig. 6. Online Activity Related Evidence-1

4.7 User Activity Related Evidence

AddressBook.sqlitedb database contains two important tables. *ABPerson* contains the contacts list. *ABRecent* contains the name and e-mail addresses with timestamp to whom the user has replied or sent mails recently using iPhone's inbuilt mail applications. *Calendar.sqlitedb* database also contains some important tables. Events table contains all the registered events including location and time. It also contains alarm information. The *notes.db* file contains the contents of user notes with creation date and time. The Keyboard folder contains a default key logger file called dynamic-text.dat file. Apple iPhone uses this file to provide auto complete feature.



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
<plist version="1.0">
<dict>
    <key>AppTimeInterval</key>
    <real>275586604</real>
    <key>YouTubeAccount</key>
    <string>[redacted]</string>
</dict>
</plist>
```

Fig. 7. Online Activity Related Evidence-2

5 Evaluations of the Digital Evidence

According to RFC 3227 “Guidelines for Evidence Collection and Archiving” [17], the legal considerations of collected evidence should be:

Admissible: Evidence must conform to certain legal rules before it can be put before a court. In our framework, we strictly follow the ACPO guidelines to make evidence sound in legal venues.

Authentic: There has to be a way to tie evidentiary materials to the incident. In the iFF framework, we recover different types of digitals evidence which make it possible to tie it to the incident according to necessity.

Complete: It must tell the whole story and not just a particular perspective. Our proposed framework retrieves a large amount of evidence to give the snapshot of the overall occurrence.

Reliable: Evidence collection method has to be free from doubts about its authenticity and veracity. As our framework follows the clearly stated procedures in Figure 1 and 2, reliability is maintained throughout the evidence collection.

Believable: It must be readily believable and understandable by a court. The digital evidence retrieved by our framework is in human readable format and it can be quickly reproduced in the courtroom for authenticity.

6 Comparisons with Existing Products and Methods

Table 1 compares our framework iFF with the existing commercial iPhone forensics products discussed in the literature review. The second column compares whether the product requires an additional hardware or standalone device. This information is taken from the product information webpage of respective companies. Next column compares, whether these products can retrieve media file (contacts, call log, sms, images etc.) related evidence. All the approaches show good performance in this feature. However, in recovering configuration files (application information, settings, and network related information), performance of Aceso and UFED is not satisfactory.

Table 1. Comparison of iFF with Existing Products and Methods

Method	Additional HW	Media Files	Configuration Files	Price (USD)
Aceso	Required	Yes	Unknown	NA
UFED	Required	Incomplete	No	4K
Device Seizure	Not Required	Yes	Yes	1K
.XRY	Not Required	Incomplete	Incomplete	9K
CellDEK	Required	Incomplete	Yes	15K
iFF (proposed)	Not Required	Yes	Yes	<50

The last column compares prices of these products found at Paraben Forensics product comparison webpage [10]. In our framework, only MobileSyncBrowser charges a very nominal fee of USD20 for parsing the backup files. Plist Editor for Windows and SQLite Database Explorer are freeware. This brings the cost of executing our forensic framework below USD50 which is quite inexpensive compared to other products. Table 2 shows the detailed comparison of extracted media files from commercial products [12] with our approach. Table 3 shows the comparison of configuration files.

Table 2. Comparison of Extracted Media Files

	Aceso	UFED	Device Seizure	.XRY	CellDEK	iFF
Call Logs	Yes	Yes	Yes	Yes	Yes	Yes
SMS	Yes	Yes	Yes	Yes	Yes	Yes
Contact	Yes	Yes	Yes	Yes	Yes	Yes
Calendar	Yes	No	Yes	Yes	Yes	Yes
Notes	Yes	No	Yes	Yes	Yes	Yes
Images	Unknown	Yes	Yes	Yes	Yes	Yes
Audio	Unknown	Yes	Yes	No	No	Yes

Table 3. Comparison of Extracted Configuration Files

	Aceso	UFED	Device Seizure	.XRY	CellDEK	iFF
Web History	Unknown	No	Yes	Yes	Yes	Yes
Bookmarks	Unknown	No	Yes	Yes	Yes	Yes
App Info	Unknown	No	Yes	Yes	Yes	Yes
Passwords	Unknown	No	Yes	No	No	Yes
Lists/XML	Unknown	No	Yes	Yes	Yes	Yes
Phone Info	Unknown	Yes	Yes	Yes	Yes	Yes
Wi-Fi Info	Unknown	No	Yes	No	Yes	Yes
HTML	Unknown	Yes	No	No	Yes	Yes

7 Challenges and Limitations

Despite the proposed method's advantages over other existing forensic methods, there are some challenges and limitations. These challenges are listed and briefly discussed below.

Most of the existing methods cannot bypass the pass-code mechanism and encryption if it is enabled. However, the proposed method inherently captures this fact and overcome this issue by taking the suspect computer into custody as one can still gain access to the iPhone even with the pass-code protection on if iPhone pairing files are captured from an iPhone backup.

Jailbreaking is a term used by iPhone users that wish to install applications that are not authorized by Apple. The jailbreaking process allows users access to the iPhone's file system. It would be critical to investigate if the proposed method is affected by jailbreaking the iPhone.

Understanding the backup structure of the iPhone may not be a onetime task. With the different generation of iPhones hitting the market, and the different firmware versions, the data structure of the backup may change. This in return pushes researchers to continuously investigate the iTunes backup structure.

We notice that with the various iPhone updates, Apple is keen on taking security measures that may impact iPhone forensics. It is critical to understand how current security updates, and future updates as well, may affect the proposed method [18] [19] [20] [21] [22].

The proposed method has the limitation of logical acquisition from the iPhone. Because, the data is logically acquired from the iPhone, all the deleted data may not be available in the backup. There are other methods, like the Zdziarski [11] method which performs a physical acquisition of the iPhone, enabling the forensic examiner to retrieve data that has been deleted. A research study confirms that the physical acquisition method was the method capable of retrieving deleted data from the iPhone [12].

8 Conclusion

In this paper, we present a simple forensic framework for iPhone. The approach is cost efficient and does not include complex tasks difficult to be performed by a forensic investigator. It follows the ACPO guideline and RFC 3227 for evidence collection and archiving to keep the recovered digital evidence sound in legal venues. Comparison of our framework with existing products also showed significant promise. We believe that our approach will be beneficial for both forensic researchers and investigators who want to experience legally sound iPhone forensics in an uncomplicated and cost effective manner.

References

1. Milanesi, C., Gupta, A., Vergne, H., Sato, A., Nguyen, T., Zimmermann, A., Cozza, R.: Garner Technology Business Research Insight. In: Dataquest Insight: Market Share for Mobile Devices, 1Q09, http://www.gartner.com/DisplayDocument?id=984612
2. Radio Tactics Ltd.: Aceso - Mobile forensics wrapped up. In: Radio Tactics | Mobile Phone Forensics, http://www.radio-tactics.com/products/aceso/
3. Cellebrite Forensics: Cellebrite Mobile Data Synchronization UFED Standard Kit. In: Cellebrite Mobile Data Synchronization, http://www.cellebrite.com/UFED-Standard-Kit.html
4. Paraben Corporation: Cell Phone Forensics. In: Paraben Corporation, Cell Phone Forensics Software, http://www.paraben-forensics.com/cell_models.html

5. Micro Systemation: XRY Physical Software. In: XRY the complete mobile forensic solution, http://www.msab.com/products/xry0/overview/page.php
6. Logicube: Logicube CellDEK Cell Phone Data Extraction. In: Logicube.com, hard drive duplication, copying hard drive & computer forensics, http://www.logicubeforensics.com/products/hd_duplication/celldek.asp
7. Lohmann, F.: Apple Says iPhone Jailbreaking is Illegal | Electronic Frontier Foundation. In: Electronice Frontier Foundation, Defending Freedom in the Digital World, http://www.eff.org/deeplinks/2009/02/apple-says-jailbreaking-illegal
8. Association of Chief Police Officers: Good Practice Guide for Computer based Electronic Evidence. In: Association of Chief Police Officers, http://www.dataclinic.co.uk/ACPO%20Guide%20v3.0.pdf (accessed June 2010)
9. Husain, M., Sridhar, R.: iForensics: Forensic Analysis of Instant Messaging on Smart Phones. In: Goel, S. (ed.) ICDF2C 2009. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 31, pp. 9–18. Springer, Heidelberg (2010)
10. Paraben Corporation: Forensic Software Comparison Chart. In: Paraben Corporation, Cell Phone Forensics, http://www.paraben-forensics.com/cell-phone-forensics-comparison.html
11. Zdziarski, J.: iPhone Forensics. O'reilly Media, Sebastopol (2008)
12. Hoog, A., Gaffaney, K.: iPhone Forensics. In: viaForensics, http://viaforensics.com/wpinstall/wp-content/uploads/2009/03/iPhone-Forensics-2009.pdf
13. Vaughn, S.: MobileSyncBrowser | View and Recover Your iPhone Data. In: MobileSyncBrowser | View and Recover Your iPhone Data, http://homepage.mac.com/vaughn/msync/
14. Piacentini, M.: SQLite Database Browser. In: SQLite Database Browser, http://sqlitebrowser.sourceforge.net/
15. VOWSoft Ltd.: Plist Editor For Windows. In: Download iPod software for Windows, http://www.icopybot.com/plistset.exe
16. Gondrom, T., Brandner, R., Pordesch, U.: Electronic Record Syntex. Request For Comments 4998, Open Text Corporation (2007)
17. Brezinski, D., Killalea, T.: Guidelines for Evidence Collection and Archiving. Request For Comments 3227, In-Q-Tel (2002)
18. Apple Inc.: About the security content of the IPhone 1.1.1 Update, http://support.apple.com/kb/HT1571
19. Apple Inc.: About the security content of IPhone v1.1.3 and iPod touch v1.1.3, http://support.apple.com/kb/HT1312
20. Apple Inc.: About the security content of IPhone v2.1, http://support.apple.com/kb/HT3129
21. Apple Inc.: About the security content of IPhone OS 3.0 Software Update, http://support.apple.com/kb/HT3639
22. Apple Inc.: About the security content of IPhone OS 3.1 and IPhone OS 3.1.1 for iPod touch, http://support.apple.com/kb/HT3860
23. Apple Inc.: Apple iPhone. In: Apple-iPhone-Mobile Phone, iPod, and Internet Device, http://www.apple.com/iphone/