KIPS Tr. Comp. and Comm. Sys.
Vol.5, No.7 pp.173~180 pISSN: 2287-5891



http://dx.doi.org/10.3745/KTCCS.2016.5.7.173

# A Study on the Evidence Investigation of Forged/Modulated Time-Stamp at iOS(iPhone, iPad)

Sanghyun Lee[†] · Yunho Lee[††] · Sangjin Lee[†††]

## ABSTRACT

Since smartphones possess a variety of user information, we can derive useful data related to the case from app data analysis in the digital forensic perspective. However, it requires an appropriate forensic measure as smartphone has the property of high mobility and high possibility of data loss, forgery, and modulation. Especially the forged/modulated time-stamp impairs the credibility of digital proof and results in the perplexity during the timeline analysis. This paper provides traces of usage which could investigate whether the time-stamp has been forged/modulated or not within the range of iOS based devices.

Keywords : iOS Forensic, Forged Time-Stamp, Modulated Time-Stamp, iPhone Forensic, Digital Forensic

# iOS(iPhone, iPad)에서의 타임스탬프 위·변조 흔적 조사에 관한 연구

이 상 현[†] · 이 윤 호[††] · 이 상 진[†††]

## 요 약

스마트폰은 다양한 사용자 정보를 저장하고 있으므로, 디지털 포렌식 관점에서 앱 데이터 분석을 통해 사건과 관련된 유용한 정보를 얻을 수 있다. 하지만 스마트폰의 특성상 높은 이동성과 데이터의 손실이나 위·변조 가능성이 크기 때문에 그에 맞는 포렌식 방법이 필요하다. 특히 위·변조된 타임스탬프는 디지털 증거의 신뢰성을 저하하며, 타임라인 분석에 어려움을 가져온다. 이 논문은 iOS 운영체제를 기반으로 하는 디바이스 내에서 타임스탬프 위·변조 여부를 조사할 수 있는 사용 흔적들을 제시한다.

키워드 : iOS 포렌식, 타임스탬프 위조, 타임스탬프 변조, 아이폰 포렌식, 디지털 포렌식

## 1. 서 론

현재 세계적으로 많은 종류의 스마트폰 운영체제가 상용화되고 있으며, 그 중 Android와 iOS의 2014년 시장 점유율은 약 96%였으며, 2013년보다 2.5% 상승한 수치였다[1]. 그리고 국내 스마트폰 가입자 수는 2014년 11월 기준 4천만 명을 돌파한 만큼 1인당 1대의 스마트폰을 보유할 정도로 보급되었다[2].

스마트폰에는 다양한 사용자 정보가 생성되고 저장된다. 따라서 디지털 포렌식 관점에서 스마트폰 콘텐츠 분석을 통해 사건과 관련된 유용한 정보를 얻을 수 있다. 특히 범죄 현장에서 스마트폰을 획득한 경우 많은 정보를 얻을 수 있는 이점이 존재하기 때문에, 모바일 포렌식의 비중은 증가하고 있다. 스마트폰 사용자들은 스마트폰을 통해 전화통화 및 문자 메시지를 이용할 뿐만 아니라, 다양한 콘텐츠를 이용해 활동하며 그에 따른 기록들은 스마트폰에 저장된다.

하지만 스마트폰의 특성상 높은 이동성과 데이터의 손실이나 위·변조 가능성이 크기 때문에 그에 맞는 포렌식 방법이 필요하다. 특히 위·변조된 타임스탬프는 디지털 증거의 신뢰성을 저하하며, 타임라인 분석에 어려움을 가져온다. 그러므로 이러한 스마트폰의 특성에 맞는 포렌식 방법이 필요하다.

본 논문에서는 iOS 운영체제를 기반으로 하는 디바이스 내에서 추출된 증거들을 기준으로 타임스탬프 위·변조 흔적 조사에 관한 연구를 진행하였다. Table 1과 같이 주요 사용자 정보에 대한 타임스탬프 위·변조의 가능성을 언급하고, 타임스탬프가 위·변조된 데이터를 통해 타임스탬프 위·변조 분석 방법에 대하여 언급하면서 탈옥(Jailbreak)된 기

† 준 회 원 : 고려대학교 정보보호대학원 정보보호학과 석사과정
†† 비 회 원 : 고려대학교 정보보호대학원 정보보호학과 박사과정
††† 종신회원 : 고려대학교 정보보호대학원 교수
Manuscript Received : April 11, 2016
First Revision : June 27, 2016
Accepted : June 30, 2016
* Corresponding Author : Sangjin Lee(sangjin@korea.ac.kr)

Table 1. iOS backup user data and Stored data

| Name | Domain - Path | Data |
|---|---|---|
| Address Book | HomeDomain-Library/AddressBook/AddressBook.sqlitedb<br>SHA1 : cd6702cea29fe89cf280a76794405adb17f9a0ee | Name, Phone Number etc. |
| Call History | HomeDomain-Library/CallHistory/call_history.db<br>SHA1 : 2b2b0084a1bc3a5ac8c27afdf14afb42c61a19ca | Caller, Callee, Time etc. |
| SMS | HomeDomain-Library/SMS/sms.db<br>SHA1 : 3d0d7e5fb2ce288813306e4d4636395e047a3d28 | Contents, Time etc. |
| Photos | CameraRollDomain-Media/PhotoData/Photos.sqlite<br>SHA1 : 12b144c0bd44f2b3dffd9186d3f9c05b917cee25 | Photo, Video information |
| Safari History | AppDomain-Library/Safari/History.plist<br>SHA1 : ed50eadf14505ef0b433e0c4a380526ad6656d3a | Visited URL, Searching keyword etc. |
| Note | HomeDomain-Library/Notes/notes.sqlite<br>SHA1 : ca3bc056d4da0bbf88b5fb3be254f3b7147e639c | Note Content |

기와 순정 기기를 가지고 2가지 환경에서 타임스탬프 위·변조 흔적에 대한 조사를 진행하였다.

탈옥은 iOS의 샌드박스 제한을 풀어 타 회사에서 사용하는 서명되지 않은 코드를 실행할 수 있게 하는 과정을 말한다. 일반적으로 탈옥하는 이유는 애플과 앱 스토어가 막아놓은 콘텐츠에 접근하기 위해서이다. 또한, 사용자들은 탈옥함으로써 앱스토어를 이용하지 않고 애플리케이션을 이용할 수 있으며, 사용자가 직접 제작한 앱이나 인터페이스를 사용할 수 있다[3]. 하지만 포렌식 관점에서는 기본적으로 백업 기능을 통해 얻을 수 있는 정보 이외에 추가적인 정보를 얻을 수 있으므로 탈옥을 수행한다[4]. 실제 수사기관에서도 기본인 정보 이외에 탈옥을 통해서 추가적인 정보를 얻는 경우가 많으며, 해당하는 사안의 중대성에 따라 수사기관은 탈옥의 여부를 결정하여 정보 수집 및 분석을 진행한다.

## 2. 연구 배경

### 2.1 관련 연구

모바일 포렌식 분야에서 타임스탬프 분석에 관한 논문은 2014년 M. Kaart와 S. Laraghy가 Android 운영체제 기반 디바이스를 대상으로 추출된 증거들에 대한 분석 방법을 제시하였다[5]. 2010년 S. Morrissey는 iOS 포렌식 분석에 관한 책을 발간하였지만, 타임스탬프 분석에 관한 부분은 언급하지 않았으며, G. Horsman의 경우 교통사고 분석 시 iOS의 Current Powerlog를 통해 운전자의 행위분석방법에 대하여 제시하였다[6, 7].

국내에서도 모바일 포렌식에 관한 연구는 지속하여 왔지만[8, 9], 앞서 언급한 타임스탬프 위·변조 분석에 관한 연구는 미미한 실정이다.

### 2.2 타임스탬프 위·변조 흔적 조사의 필요성

디지털 포렌식 관점에서 타임스탬프의 위·변조 흔적 조사는 다음의 2가지의 이유로 필요하다.

첫째, 사건 은폐나 위조의 가능성이 있으므로 분석이 필요하다. 디지털 증거의 경우 신뢰성이 중요하다. 디지털 증거의 신뢰성은 증거 데이터의 분석 등 처리 과정에서 디지털 증거가 위·변조되거나 의도되지 않은 오류를 포함하지 않았다는 것을 의미한다. 만일 디지털 증거가 위·변조되었다면, 디지털 증거의 신뢰성 문제와 수사 과정에서 혼선을 일으킬 수 있으므로 위·변조 흔적 조사가 필요하다.

둘째, 타임라인 분석에 어려움을 가져오기 때문에 분석이 필요하다. 타임스탬프가 위·변조된 데이터로 인하여 특정 시점의 행위에 대한 추적이 어려워지기 때문에 수사에 혼선을 줄 수 있다.

디지털 데이터에서 시간은 범죄 사실을 규명하기 위해 매우 중요한 정보이다. 파일 시스템 상에 저장되는 파일의 시간 정보, 파일 내부의 메타데이터에 저장되는 시간 정보 등 다양한 곳에 저장된 시간 정보를 이용해 타임라인을 구성함으로써 시스템 사용자의 행위를 추적할 수 있으며, 저장된 시간 정보가 연, 월, 일 등으로 분류되어 가시적으로 출력된다. 이러한 기능을 활용해서 시스템이 사용된 시간대를 알아내거나 특정 시점에서부터 시스템 사용자의 행위를 추적할 수 있다.

한편 파일 시스템에 저장된 시간과는 별도로 문서 파일, 사진 파일 등은 파일 내부에 고유한 형식으로 시간 정보를 저장하고 있다. 파일 자체에 저장된 시간 정보는 파일이 복사, 이동, 수정되는 과정에도 변경되지 않기 때문에, 전문 지식이 없는 경우에는 조작이 쉽지 않다.

하지만 모바일 환경에서 데이터의 타임스탬프는 간단한 시간 설정으로 쉽게 위·변조할 수 있다. 따라서 조작 여부를 판단할 수 있는 흔적에 관한 조사 연구가 필요하다.

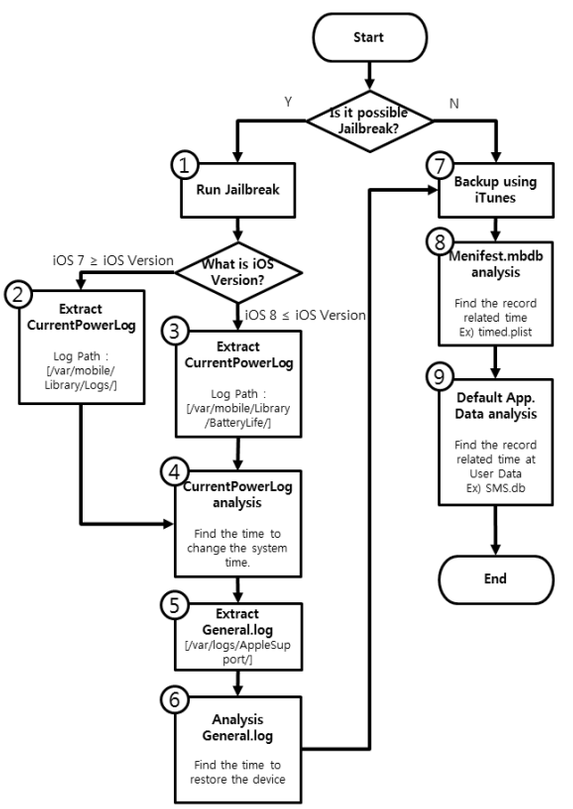
## 3. iOS 디바이스의 시간 조작 실험 및 검증

iOS 운영체제를 기반으로 하는 디바이스 내에서 타임스탬프 위·변조에 대한 흔적을 조사하는 방법에는 크게 2가지가 존재한다.

첫 번째는 아이튠즈의 백업기능을 이용하여 분석하는 방법이다. 아이튠즈를 통해 백업하면 iOS 디바이스 내의 각종

파일이 사용자 PC로 저장된다. 백업기능을 통해 사용자 PC에 저장된 파일 중 com.apple.timed.plist와 Menifest.mbdb를 통해 시간 조작 여부를 확인한다. 여기서 com.apple.timed.plist는 iOS디바이스의 시간설정 값이 저장된 파일이다. 해당 파일의 설정 값에 따라 디바이스의 시간 설정 방식을 달리하며, 기본적으로 시간을 설정하는 방식은 서버로부터 동기화하는 방식으로 이뤄진다. 또한, 기본적인 방식 외에 사람에 의해서 디바이스의 시간을 조정할 수 있는 기능을 기본적으로 제공한다. 인위적으로 시간을 조작하게 되면 최초 설정 파일의 생성 시간보다 최종 수정 시간이 달라질 수밖에 없다. 하지만 백업된 com.apple.timed.plist파일은 저장되어 있는 설정값만 볼 수 있으므로, 백업된 파일들에 대한 파일시스템 상의 메타데이터와 파일검증에 필요한 해시 값 등이 저장된 Menifest.mbdb를 이용해서 생성시간과 최종 수정시간을 확인 및 비교함으로써 시간 조작 여부를 확인한다.

또한, 백업된 각 애플리케이션들의 사용자 데이터의 분석을 통해서도 분석할 수 있다. 각 애플리케이션은 사용자 데이터를 SQLite 형식의 DB 파일로 저장하는데, 이때 각 데이터는 생성한 순서대로 저장된다. 이러한 점을 근거로 하여 생성된 순서와 생성시간의 역전현상이 존재하는 경우 시간 조작 여부를 확인한다.




1. Run Jailbreak to iOS device. Jailbreak is performed using a tool provided for each version.
2, 3. Extract the CurrentPowerLog in Jailbroken device.
4. Analyze whether the system time has been changed at extracted logs.
5. Extract General.log to investigate whether the device was recovered.
6. Analyzes whether the device was recovered.
7. Backup iOS device using iTunes tool.
8. Analyze Menifest.mbdb whether the records relating to the time.
9. Analyze whether there is a reversal of time from the default Application Data.

Fig. 1. Flow chart of timestamp analysis

두 번째는 탈옥(Jail-break) 상태의 디바이스를 가지고 하는 분석 방법이다. 이 방법은 순정상태에서 루트(root) 권한을 얻어내는 탈옥이 선행되어야 한다. 이렇게 탈옥된 기기를 가지고 임의적인 디바이스의 시간 조작이 이뤄졌을 때 남아있는 로그 분석을 통해 시간 조작 여부를 확인한다.

앞의 방법을 위해서 iPhone 4s, 5, 5S, iPad mini 총 4대의 기기를 사용하였고, iOS 7, iOS 8의 버전에서 수행하였다. iOS 7과 iOS 8 버전에 대한 차이를 현 논문에서 다루는 타임스탬프 위·변조 흔적과 관련해서 얘기한다면, CurrentPowerLog의 운영방식이 바뀌었다는 점을 볼 수 있다. iOS 7 이하 방식에서는 CurrentPowerLog를 일반 파일형식으로 저장하였으나, iOS 8부터는 DB를 사용하여 저장하는 방식으로 바뀌었다. 이에 따라 탈옥 시 조사 방법이 달라지는데, 그 방법은 3.2절에서 소개한다.

이 장에서는 개념 설명과 2가지 환경에 따른 타임스탬프 위·변조 분석 방법을 실험을 통해 알아볼 것이며, 해당 처리 순서도는 Fig. 1과 같다.

### 3.1 아이튠즈의 백업기능을 이용한 분석

이 절에서는 아이튠즈의 백업 기능을 이용한 타임스탬프 위·변조 흔적에 대한 조사 방법을 제시한다. 시간 설정 파일인 com.apple.timed.plist의 생성 및 수정시간을 통해 조사하는 방법과 백업된 애플리케이션의 사용자 데이터를 조사하는 방법을 제시한다.

#### 1) 시간 설정 파일을 이용한 시간 조작 흔적 조사

iOS 디바이스는 기본적으로 서버로부터 시간을 동기화하는 방식으로 설정되어 있다. 그러나 사람에 의해 시간을 임의로 변경할 수 있는데, 이러한 시간 설정에 관한 내용을 담고 있는 파일이 com.apple.timed.plist이다. 일반적으로는 사람들이 시간을 임의로 변경할 일이 없으므로, com.apple.timed.plist 파일은 수정되지 않는다. 따라서 해당 파일에 대한 생성 시간과 최종 수정시간이 같은 것이 기본적이다. 하지만 시간을 임의로 변경한다면, 시간 설정 파일의 최종 수정 시간은 변경된다. 따라서 com.apple.timed.plist의 최종수정시간을 통해 시간 변경 여부를 확인해야 하지만, 백업된 com.apple.timed.plist 파일만 가지고 해당 파일의 생성 시간과 최종 수정 시간을 확인할 수 없다. 이것을 확인하기 위해 백업 파일들에 대한 메타데이터를 가지고 있는 Menifest.mbdb 파일을 이용한다.

아이튠즈를 이용하여 백업을 수행하면 각 디바이스에 해당하는 백업 경로에 백업 파일들이 생성되게 된다. 각각 사용자 PC에 백업된 파일들은 일정 규칙으로 해시(hash)된 파일 이름으로 명명된다. 이 파일들은 각각 디바이스의 애플리케이션 데이터 및 사용자 데이터이며, 해당 파일들에 대한 메타데이터를 레코드 단위로 가지고 있는 파일이 Menifest.mbdb이다. 해당 파일은 백업 폴더 안에 저장되어 있는 파일들의 파일명, 파일의 도메인, 파일 경로, 파일의 해시값, 파일시스템상의 시간 등이 레코드 단위로 기록되어 있다.

Table 2. Example of Menifest.mbdb record structure

| No. | Offset | Data | Explanation |
|---|---|---|---|
| 1 | 2byte | 0x000A | Domain string length |
| 2 | String | HomeDomain | Domain name |
| 3 | 2byte | 0x0012 | File Path string length |
| 4 | String | Library/SMS/sms.db | File Path |
| 5 | 2byte | 0xFFFF | End of Domain and File Path |
| 6 | 2byte | 0x0014 | File Hash string length |
| 7 | String | 0xC5B0....7DBD | File Hash string |
| 8 | 2byte | 0xFFFF | End of File Hash |
| 9 | 2byte | 0x81A4 | Mode(File, Directory and Authority) |
| 10 | 4byte | 0x000000 | NULL |
| 11 | 4byte | 0x00000099 | Node Value |
| 12 | 4byte | 0x000001F5 | User ID |
| 13 | 4byte | 0x000001F5 | Group ID |
| 14 | 4byte | 0x54FD8061 | Last modified time (UNIX Time) |
| 15 | 4byte | 0x5502C62F | Last access time (UNIX Time) |
| 16 | 4byte | 0x53F73BA5 | Created time (UNIX Time) |
| 17 | 8byte | 0x0000000000807000 | File size (byte) |

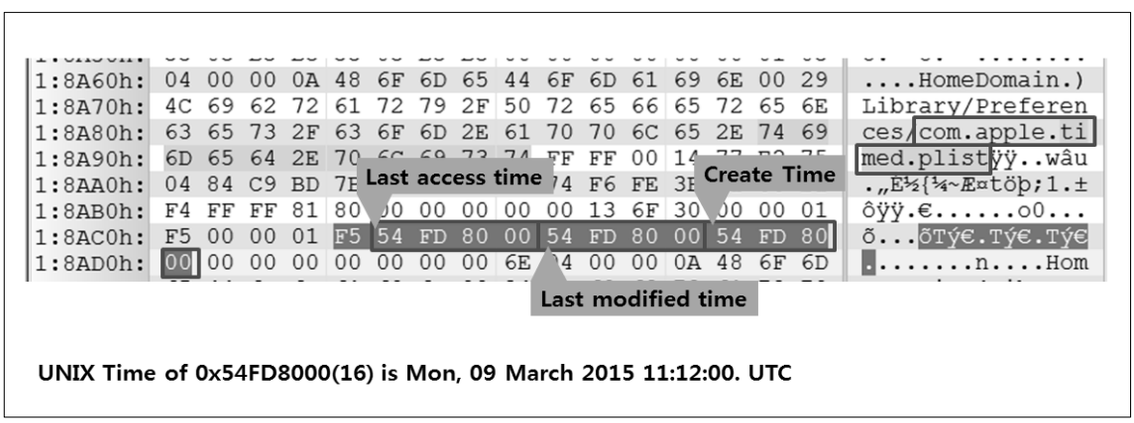



Fig. 2. Analysis com.apple.timed.plist through Menifest.mbdb file

Menifest.mbdb 내에 존재하는 파일별 레코드 구조를 분석해보면 Table 2와 같다. 이 정보를 통해 백업된 파일의 생성 시간과 수정 시간 그리고 접근시간을 알 수 있다. 따라서 디바이스의 시간 설정 파일인 com.apple.timed.plist의 생성 시간, 접근 시간 및 수정 시간을 Menifest.mbdb 내에 저장된 데이터를 통해 알 수 있다. Menifest.mbdb에서 com.apple.timed.plist에 해당하는 레코드를 찾고 그중에서 그 파일에 대한 생성, 수정, 접근 시간을 조사한다. Menifest.mbdb에서 com.apple.timed.plist에 해당하는 레코드는 Fig. 2와 같다. 만일 생성 시간과 수정 시간이 같다면, 디바이스의 로컬 시간을 임의로 변경하지 않았다고 판단하며, 그렇지 않으면 임의로 변경한 것으로 판단한다.

2) 애플리케이션 사용자 데이터를 이용한 시간 조작 흔적 조사

디바이스 내에 설치된 iOS 운영체제가 탈옥을 지원하지 못할 경우 Apple 사에서 제공하는 아이튠즈를 이용하여 백업된 각 애플리케이션의 사용자 데이터 파일을 가지고 분석을 한다. 이 파일들은 SQLite 형식의 DB 파일로써 데이터가 저장될 때, 각 데이터는 디바이스의 시간과 관계없이 생성된 순서대로 저장되기 때문에 데이터의 생성 순서와 저장된 타임스탬프 사이의 역전을 찾음으로써 시간 조작 여부를 확인한다.

여기서 분석할 애플리케이션(이하 앱)은 기본 앱인 SMS(문자메시지 및 아이메시지), PhoneCall, Camera 앱의 사용자 데이터에 대하여 분석을 진행한다. 해당 앱은 기본 앱 중에서 시간 조작과 연관이 있는 앱을 선정하였으며, 3rd Party 메신저 앱의 사용자 데이터를 분석해본 결과 사용자 데이터가 저장된 DB 파일에 메시지별 송수신 시간을 저장할 때, 디바이스의 시스템 시간을 저장하는 것이 아닌 서버 시간을 가져오기 때문에 시간 조작 분석과 무관하여 제외하였다.

a) SMS 앱 분석

SMS 앱으로 송·수신한 메시지들은 sms.db라는 파일로 파일시스템 내부에 존재하는데, 이것을 아이튠즈의 백업 기능을 통해 백업했을 경우, sms.db에 해당하는 해시값으로 백업 폴더 내에 저장한다.

Fig. 3과 Fig. 4는 SMS 앱을 이용하여 문자 메시지 및 아이메시지를 보낸 화면을 보여주고 있다. 여기서 네모 상자를 보면 시간의 순서가 오름차순이 아닌 무작위로 섞여 있는 것을 발견할 수 있는데, 이것은 디바이스의 시간을 조작하여 문자메시지를 조작된 시간으로 발송한 것이다. 해당 문자 메시지는 sms.db 파일 내에서 역전 현상 없이 그대로 화면에 보여주고 있음을 확인할 수 있다.

해당 메시지를 sms.db를 통하여 확인해 본 결과는 Fig. 5와 같다. 발송된 메시지들은 해당 DB의 형식에 맞춰 저장되고, 차례로 ROW ID의 생성 순서에 따라 메시지가 저장되기에 디바이스 상에서는 SMS의 타임스탬프 위·변조가 불가능하다. 하지만 Menifest.mbdb에 저장된 해시 값을 조작된 sms.db에 대한 해시 값으로 대체한 뒤 아이튠즈의 복원 기능을 통해 복원하면 조작할 수 있다.

b) PhoneCall 앱 분석

PhoneCall 앱을 통해 사용한 전화 사용 기록은 파일시스템 내부에 CallHistory.storedata라는 이름으로 존재한다. 이 파일은 백업 폴더 내에 해시값으로 지정된 이름으로 저장되어 있다.

Fig. 6은 PhoneCall 앱을 이용하여 전화를 송·수신한 화면과 CallHistory.storedata 파일을 보여주고 있다. SMS 앱에서 봤던 것과 달리 여기서는 시간의 흐름대로 나열되어 있음을


Fig. 3. Text Message on SMS Application

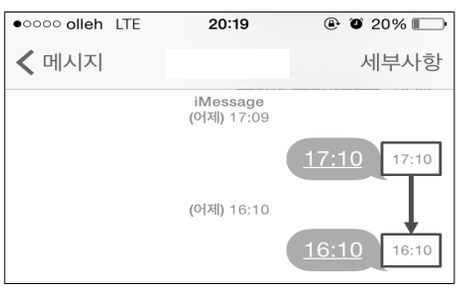



Fig. 4. iMessage on SMS Application

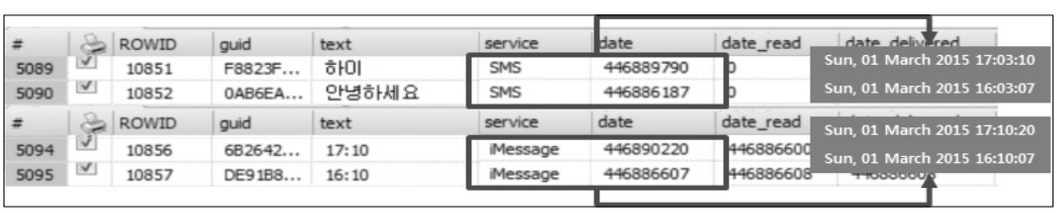



Fig. 5. Database of SMS Application

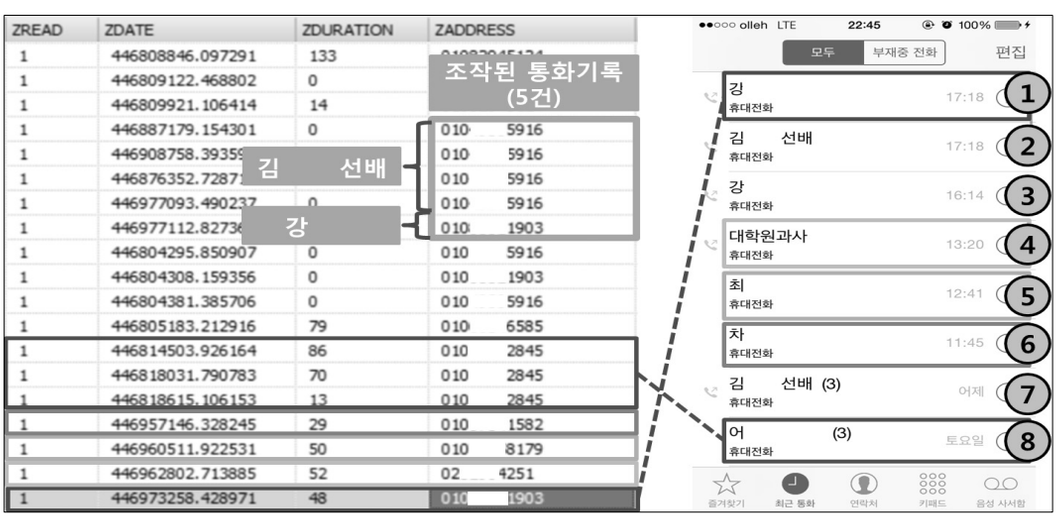



Fig. 6. Comparing Call history to CallHistory.stroedata file

볼 수 있다. 하지만 Fig. 6의 우측 번호 중 2,3,7번은 디바이스의 시간을 조작한 후에 전화를 시도한 부분이다. 조작한 부분을 찾기 위해 전화 사용 기록에 대한 정보를 담고 있는 DB 파일인 CallHistory.storedata를 분석한다.

CallHistory.storedata는 SQLite 형식의 DB 파일로써 해당 전화 송·수신 기록이 index의 순서대로 저장된다. 따라서 앱에서 보이는 전화 송·수신 목록의 순서와 DB 상의 순서를 비교하여 조작된 통화를 찾아낼 수 있다.

c) Camera 앱 분석

Camera 앱을 통해 찍은 사진 정보는 파일시스템 내부에 Photos.sqlite라는 DB 파일에 존재한다. 사진들의 기본 정보를 담고 있는 테이블은 ZADDITIONALASSETATTRIBUTES로써 파일 저장 경로 및 이름 등을 기록하고 있다.

Fig. 7은 Camera 앱을 통해 찍은 사진을 보여주는 Photos 앱이다. 여기서 제일 상단에 보이는 사진이 시간 조작을 통해 찍은 사진으로써 Photos 앱에서는 시간의 흐름에 따라 정렬해서 보여주는 것을 볼 수 있다. 따라서 해당 증거의 시간 조작 여부를 판단하기 위해서 사진들의 정보를 DB형태로 저장된 Photos.sqlite를 통해 흔적을 조사한다. Fig. 8은 해당 DB 형식의 파일을 열어본 것이다. iOS 기반 디바이스의 경우 사진을 찍었을 경우 해당 테이블의 ZEXIFTIMESTAMPSTRING 칼럼에 사진을 찍은 시간이 저장되어 있는 것을 확인할 수 있으며, 이 사진이 현재 조사 중인 디바이스에서 찍었음을 판단함과 동시에, index 순서대로 DB 상에 기록되기 때문에 사진의 시간 위·변조 여부를 판단할 수 있다.


Fig. 7. Photos Application

Table: ZADDITIONALASSETATTRIBUTES

| | ZEXIFTIMESTAMPSTRING | ZORIGINALFILENAME | ZORIGINALPATH |
|---|---|---|---|
| | Filter | Filter | Filter |
| 2529 | 2015:08:25 21:03:55 | IMG_4362.JPG | DCIM/104APPLE |
| 2530 | 2015:08:26 11:56:06 | IMG_4363.JPG | DCIM/104APPLE |
| 2531 | 2015:08:25 11:56:34 | IMG_4364.JPG | DCIM/104APPLE |

Fig. 8. DB Table of Photos.sqlite

3.2 탈옥된 디바이스의 CurrentPowerLog를 이용한 분석

CurrentPowerLog는 시스템 이벤트를 나타내는 레코드들에 대한 로그를 저장하는 시스템용 파일이다. 해당 로그는 iOS 버전에 따라 달라지는데, iOS 7 이하 버전의 경우는 CurrentPowerLog.powerlog라는 파일의 형태로 저장되어 있으며, iOS 8 이상 버전의 경우 CurrentPowerLog.PLSQL로 DB 파일로 저장되어 있다. 이 파일을 추출하기 위해서는 탈옥(Jailbreak)해야 한다.

1) CurrentPowerLog.powerlog 분석(iOS 7 이하)

이 파일은 앞에서 언급한 바와 같이 iOS7 이하의 버전에서 사용되고 있는 시스템 이벤트에 해당하는 로그 파일이다. 이 파일에서 각 레코드 항목은 속성 태그에 고정되어 있으며, 이 속성 태그는 수행된 활동의 유형을 나타낸다. 해당 파일이 저장되어 있는 경로는 [/var/mobile/Library/Logs]이며 Fig. 9는 CurrentPowerlog.powerlog의 일부분이다. 그림을 보면 네모 상자 위의 시간인 [03/04/15 19:01:15]에서 네모 상자 내에 존재하는 시간인 [03/01/15 19:01:57]로 바뀐 것을 확인할 수 있다. 이것은 time zone 설정 부분에서 시간 값을 변경한 것에 대한 로그가 기록된 것이다. 이것을 통해 현재 인위적으로 아이폰 내의 시간이 조작되었음을 확인할 수 있다. 하지만 이 로그 파일은 하루에 한 개의 로그만을 기록하고 있으며, 최대 10일간의 로그파일을 [/var/mobile/Library/Logs/PLArchive]의 하위 디렉터리에 저장한다. Fig. 10은 PLArchive 디렉터리 하위에 저장된 PowerLog의 모습이다.

2) CurrentPowerLog.PLSQL 분석(iOS 8 이상)

이 파일은 iOS 8 버전으로 바뀌면서 해당 로그의 형태가 바뀌게 되었다. DB 형식으로 바뀌게 되었으며, 수행된 활동에 대한 유형을 207개의 테이블 별로 나눠 로그를 기록하고 있다. 각 테이블은 유형에 따라 다른 칼럼들을 가지고 있다. 해당 파일이 저장된 경로는 [/var/mobile/Library/BatteryLife]이다. Fig. 11은 CurrentPower log.PLSQL의 일부분이며, PLStorageOperator_EventFoward_Timeoffset이라는 테이블을 보여주고 있다. 이 테이블은 시간과 관련된 이벤트에 대한 기록이며, 이것을 통해 시간 조작 행위가 일어났음을 알아낼 수 있다. Fig. 11을 보면 System이란 칼럼은 시스템 내부 시간을 나타내는데 표현 형식은 Unix Time이다. 여기서 17, 18번 index를 비교하였을 때, [1436511806.23503에서 1436475757.27837]과 같이 시간이 줄어들었음을 확인할 수 있다. 이 로그도 [/var/mobile/Library/BatteryLife/Archives]라는 하위 디렉터리에 최대 5일 분량의 로그파일을 저장하고 있다.

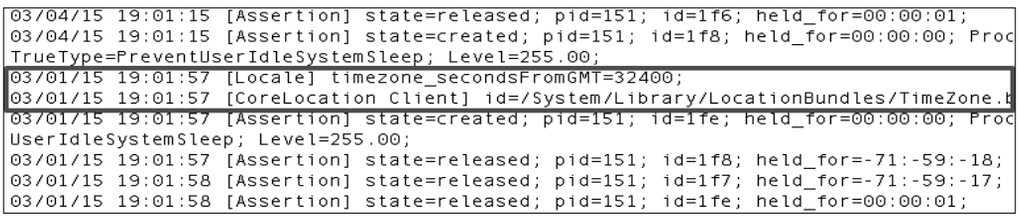



Fig. 9. CurrentPowerLog.powerlog file

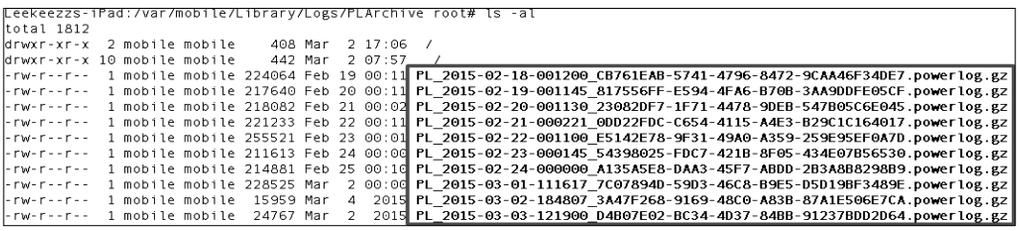



Fig. 10. Archive files of CurrentPowerlog.powerlog

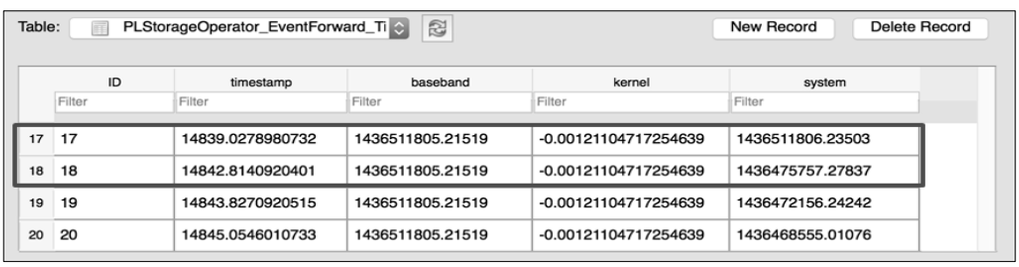



Fig. 11. PLStorageOperator EventForward Timeoffset Table on CurrentPowerLog.PLSQL

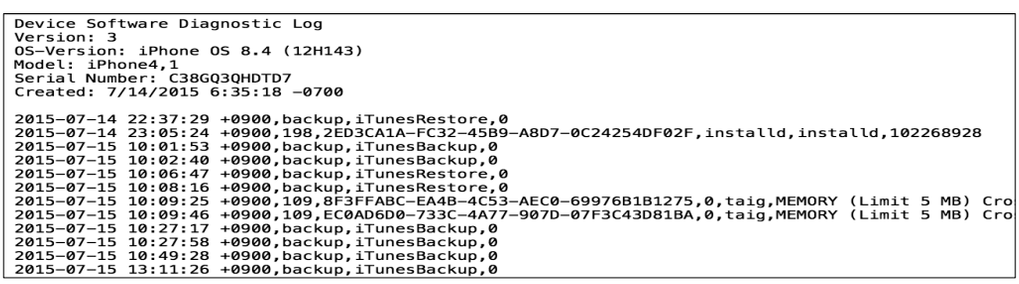



Fig. 12. general.log

### 3.3 백업된 파일의 Restore 기록을 통한 타임스탬프 위·변조 흔적에 대한 타당성 검증

디바이스의 타임스탬프 위·변조 여부에 관한 확인은 앞에서 설명한 것과 같이 아이튠즈를 통해 디바이스의 데이터를 백업한 뒤, 이렇게 백업된 파일을 조작하여 Restore(복구)한 경우에는 위·변조의 여부를 파악할 수 없지만 백업된 파일이 조사 대상 디바이스에 언제 Restore가 되었는지 확인할 방법이 존재한다[10]. 이것은 디바이스 내에 존재하는 general.log를 통해서 확인할 수 있는데, 해당파일이 저장된 경로는 [/private/var/logs/AppleSupport/]와 같다. 이렇게 추출된 general.log파일을 확인하면, 백업 및 복구 날짜를 확인할 수 있다. Fig 12.는 general.log를 확인한 모습이다. 해당 로그를 통해 iOS 디바이스의 백업 및 복구 날짜를 확인함으로써 실제 조사한 디바이스의 타임스탬프 위·변조 흔적에 대한 무결성을 검증할 수 있다.

## 4. 결 론

본 논문에서는 iOS에서의 타임스탬프 위·변조 흔적 조사에 관한 방법을 탈옥된 기기 혹은 탈옥이 가능한 기기를 분석하는 방법과 탈옥이 되지 않은 기기를 분석하는 방법 2가지를 제시하였다. 특히 iOS 7 이하 버전에서 iOS 8로 버전이 상향되면서 변경된 CurrentPowerlog에 대한 분석 방법도 제시함으로써 타임스탬프 위·변조에 대한 탐지를 높이고자 하였으며, 실제 애플리케이션에 남는 타임스탬프 데이터의 분석을 제시함으로써 로그 분석의 방법과 비교하여 분석할

Table 3. Summarized in the table for the analysis

| Filename | Type | Table name | Referenced column or line | Details |
|---|---|---|---|---|
| sms.db | Database | message | service | Whether used for SMS or iMessage |
| | | | date | Unix Time value |
| CallHistroy.storedata | | ZCALLRECORD | ZDATE | Unix Time value |
| | | | ZADRESS | Target phone number |
| Photos.sqlite | | ZADDITIONALASSETATTRIBUTES | ZEXIFTMESTAMPSTRING | Taken time |
| | | | ZORIGINALFILENAME | File name |
| CurrentPowerLog.PLSQL | | PLStorageOperator_EventForward_Timeoffset | ID | Event procedure |
| | | | system | Unix Time value |
| CurrentPowerLog.powerlog | Text | - | Locale | Notification that there is a change in the Timezone |
| general.log | | - | iTunesRestore | Notification that a recovery occurred at that time. |

수 있도록 하였다. Table 3은 분석해야 할 파일과 참조해야 할 데이터에 대한 정리 표이다.

타임라인 분석은 분석 데이터를 시간순으로 나열하여 분석하는 방법을 말한다. 이런 타임라인 분석은 포렌식 조사에서 특정 이벤트 발생 시점 전, 후로 시스템상에서 어떤 일이 발생했는지 쉽게 파악할 수 있으며, 정밀 분석 대상을 빠르게 선별할 수 있게 해준다. 하지만 이런 타임스탬프가 위·변조될 경우, 디지털 증거의 신뢰성 저하뿐만 아니라 타임라인 분석 시 범죄자의 의도대로 수사의 흐름이 바뀔 수 있다. 예를 들어, 범죄자가 자신의 거짓 알리바이를 위해 디바이스의 시스템 시간을 바꾼 뒤 조작된 시간 아래에서 사진을 찍어 증거로 제시할 가능성이 있다. 따라서 본 논문을 통해 기기 내 타임스탬프 위·변조 여부를 분석함으로써 사건해결에 중요한 타임라인을 구성할 때 큰 역할을 할 수 있으리라 판단된다.

## References


[1] Business Post [Internet], “The market share of smartphone,” 2015, http://www.businesspost.co.kr/news/articleView.html?idxno=9866.

[2] ZD Net Korea [Internet], “Domestic smartphone subscribers,” 2015, http://www.zdnet.co.kr/news/news_view.asp?artice_id=20150120151312.

[3] Wikipedia [Internet], “iOS Jailbreak,” https://ko.wikipedia.org/wiki/IOS_ %ED%83%88%EC%98%A5.

[4] SANS, “Forensic analysis on iOS devices,” 2012.

[5] M. Kaart and S. Laraghy, “Android forensics: Interpretation of timestamps,” *Digital Investigation*, Vol.11, Issue 3, pp. 234-248, 2014.

[6] Sean Morrissey, “iOS Forensic Analysis for iPhone, iPad and iPod touch,” Apress, 2010.

[7] Graeme Horsman and Lynne R. Conniss, “Investigating evidence of mobile phone usage by drivers in road traffic accidents,” *Digital Investigation*, Vol.12, pp.S39-S37, 2015.

[8] SungKyoung Un and WooYoun Choi, “A Trend of Smartphone Forensic Technology,” Tech Report of ETRI, 2013.

[9] JaeHyun So, “Search evidence of the iPhone Backup file,” AhnLab Tech Report [Internet], 2012, http://www.ahnlab.com/kr/site/securityinfo/secunews/secuNewsView.do?seq=20118.

[10] Sangah Kim, “The Analysis of Synchronized or Restored Pictures in Smart Devices,” Graduate School of Information Security, Korea University, 2014.



**이 상 현**

e-mail : leekeezz@korea.ac.kr

2011년 가톨릭대학교 컴퓨터공학과(학사)

2015년～현 재 고려대학교 정보보호대학원 정보보호학과 석사과정

관심분야 : Digital Forensic, Mobile Forensic, Reverse Engineering



**이 윤 호**

e-mail : yuno21@korea.ac.kr

2014년 고려대학교 정보보호대학원 정보보호학과(석사)

2014년～현 재 고려대학교 정보보호대학원 정보보호학과 박사과정

관심분야 : Digital Forensic, Mobile Forensic, Information Security



**이 상 진**

e-mail : sangjin@korea.ac.kr

1987년: 고려대학교 수학과(학사)

1989년: 고려대학교 수학과(석사)

1994년: 고려대학교 수학과(박사)

1989년～1999년 ETRI 선임연구원

1999년～현 재 고려대학교 정보보호대학원 교수

2008년～현 재 고려대학교 디지털포렌식연구센터 센터장

관심분야 : Digital Forensic, Steganography, Hash Function