Forensic artifacts of the ChatON Instant Messaging application

Conference Paper · November 2013

DOI: 10.1109/SADFE.2013.6911538

CITATIONS
11

READS
2,540

3 authors:

Asif Iqbal
KTH Royal Institute of Technology
17 PUBLICATIONS **68** CITATIONS
SEE PROFILE

Andrew Marrington
Zayed University
46 PUBLICATIONS **483** CITATIONS
SEE PROFILE

Ibrahim Baggili
University of New Haven
75 PUBLICATIONS **828** CITATIONS
SEE PROFILE

Forensic artifacts of the ChatON Instant Messaging application

Asif Iqbal [1], Andrew Marrington [2] , Ibrahim Baggili [3]

Athena Labs [1] , Zayed University [1, 2], University of New Haven [3]
Dubai, UAE
asif@babariqbal.com, andrew.marrington@zu.ac.ae, ibaggili@newhaven.edu

***Abstract*— Instant Messaging (IM) is one of the most used types of applications across all digital devices, and is an especially popular feature on smartphones. This research is about the artifacts left by Samsung's ChatON IM application, which is a multi-platform IM application. In this work, we acquired forensic images of a Samsung Galaxy Note device running Android 4.1 and an iPhone running iOS 6. The acquired images were analyzed and the data relevant to the ChatON application were identified. This research resulted is a map of the digital evidence left by ChatON on these mobile devices which assists digital forensics practitioners and researchers in the process of locating and recovering digital evidence from ChatON.**

Keywords— Digital Forensics; ChatON; artifacts; Instant Messaging; IM; Mobile Forensics; Small Scale Digital Devices

I. INTRODUCTION

In 1996 ICQ, one of the well-known instant Messaging (IM) applications was introduced to the desktop environment [1]. Other instant messaging applications surfaced as well such as AOL Instant Messenger (AIM), Skype, MSN Messenger, Google Talk, and Yahoo! Messenger [1]. These applications gained popularity as a result of features such as the synchronous nature of communication, awareness of the presence of the person with whom the user wants to correspond, and the user's ability to select the contacts to add to their own contact list [3]. With the increased popularity of IM applications in personal and professional usage, it was just a matter of time until they were utilized to perform cyber crimes such as phishing, social engineering, threatening, cyber bullying, hate speech and crimes, child exploitation, sexual harassment, and illegal sales and distribution of software [2].

Before the introduction of common IM application Short Messaging Service (SMS) was a popular instant messaging platform on mobile devices.With the evolution of computing devices and the move towards mobility, IM applications evolved to provide the concept of real time messaging. It was one of the first and most adopted applications that migrated to the smart mobile platform. The move consisted of downsizing the existing IM applications such as Yahoo! Messenger, MSN Messenger and Skype to work with mobile devices as well as creating IM applications that were designed specifically for devices such as BlackBerry Messenger, WhatsApp, Viber, Tango, Nimbuzz and ChatOn. The portability of mobile devices and the availability of a mobile Internet connection provided a medium for these applications to spread across a wider user demographic.

This paper studies the artifacts left by ChatON which is one of the IM services for mobile devices. It was released in 2011 by Samsung Electronics covering a range of platforms. These platforms include Android smartphones and tablets, iOS devices, BlackBerry, Windows Phone as well as Windows 8 and bada smartphones. Additionally, a web client is offered for access to the service via a web browser [8]. ChatON is preloaded in several Samsung products such as Galaxy S3 and Galaxy Note2 Smartphones as well as the Galaxy Camera and it officially serves more than 120 countries in 60 languages which makes it available to a wide number of users [8]. It has around 50,000,000 + installs from Google Play, and the fact that it is available on other platforms gives it an edge over IM applications like BBM or iMessage, which are limited to one smartphone manufacturer only.

The research intends to explore the digital evidence that may be left on both Android and iOS from the ChatOn application. In this work, we examine two mobile devices a) A Samsung Galaxy Note running Android 4.1 and b) An iPhone running iOS 6. The purpose of this analysis was to locate the artifacts left by ChatON on the devices and identify their significance to the forensic investigation process.

II. RELATED WORK

A. Mobile Device Forensics

As described by Al Mutawa et. al. [5], the initial research work in this field has focused on acquisition techniques and general forensic analysis of smart devices. This was shown in Burnette's work in 2002 where he discussed the forensic examination of older versions of the BlackBerry and the hardware and software used for acquisition [7]. In 2003 Willassen [9] discussed the items of interest in GSM mobile phone based analysis such as location, SMS and contacts.

It has also been observed that the hetereogenity in device hardware, software, ports, connectors, and so on poses a significant challenge to forensic investigators because Small Scale Digital Devices (SSDs) are more proprietary than the traditional personal computer [6]. Illustrating this point, in a

later work, Willassen explored the collection methods for commodity mobile phones, in particular the use of physical access to the circuit board interface (e.g., JTAG port) or physically removing memory chips for later data collection via a chip programmer [10]. Mokhonoana and Olivier [11] as well as Distenfano [12] worked on the collection methods for Symbian OSv7 and Symbian OSv8 respectively, illustrating disparities between versions even of the same operating system software.

As the smartphone market matured, many third-party applications (Apps) became available for multiple smartphone platforms. Apps for social media platforms and IM protocols are available for every different major smartphone platform. Consequently, researchers began to examine activity traces left behind on different devices running third-party applications intended for the same purposes, such as Al Mutawa et al.'s work on social networking applications on iPhone, Android and BlackBerry devices [5]. This work is similar in its outlook as it examines the same application (ChatON) on two competing smartphone platforms (Samsung's own Android-based Galaxy and the Apple iPhone).

With respect to Android device forensics, in 2010 Lessard and Kessler identified a method to acquire a physical image of an Android device by obtaining a dd image of its memory. Their research also included studying and locating information that can be considered as evidence using several tools such as FTK and CelleBrite UFED to perform the analysis [13]. In order to completely image the Android device, Lessard and Kessler had to root the device.

Rootkits are often installed by users to circumvent manufacturer restrictions on the device, but may undermine the trustworthy status of the device both because the process of "rooting" the device necessarily modifies its memory, and because it may undermine the trusted status of the Android device's kernel (if the rootkit has not been thoroughly tested). Motivated in part by the desire to avoid rooting the Android device, Vidas et al. explored Android's boot mode and partitioning schema in order to develop an acquisition methodology based on overwriting the "Recovery" partition on the Android device's SD card with specialized forensic acquisition software [14]. The aim of this research was to design a general process for data collection of Android devices. Other work was also done with regard to Android device forensics [15] [16].

As the iPhone has been on the market longer, there is slightly more diversity in the approaches to acquiring digital evidence from iOS devices described in the literature when compared to Android devices. The research methods are divided into several areas such as:

1. Logical acquisition via iTunes backup of an iPhone device [21],
2. Physical acquisition via jailbreaking the device and executing dd via a remote shell on the device,
3. Physical acquisition via disassembly of the device,
4. Physical acquisition without jailbreaking the iPhone [18].

Zdziarski [17] work in 2010 was evaluated by the National Institute of Standard and Technology (NIST) and the results showed that it is one of the best forensic methods for acquiring iPhone evidence at the time. Although early versions of the Zdziarski method required jailbreaking the device (analogous to rooting on an Android device), for most iOS hardware/software version combinations jailbreaking is not required for successful physical acquisition of the device. Gomez-Miralles et al. [19] developed a method similar to the Zdziarski method but used the camera connection kit to acquire the image, reducing the imaging time. In 2012 Iqbal et. al. [18] investigated the acquisition and analysis of iDevices (iPhone, iPad, iPod). Their developed method was designed to acquire data from iDevices quickly and securely. This method required less than 30 minutes for the acquisition process while ensuring that little or no footprint are left on the device as a result of the acquisition process.

B. Instant Messaging and Social Networking Forensics Artifacts

Artifacts of instant messaging have long been of interest in many different digital investigations. This interest has been reflected in a steady stream of research publications on the topic. Early work focussed on artifacts left behind by particular instant messaging applications, such as AOL Instant Messenger [25][26]. With "Web 2.0" came web-based clients for the popular instant messaging protocols, and Kiley et al. examined the traces left behind on a host PC from using such web-based IM clients [27]. Popular social networking platforms like Facebook also incorporated their own instant messaging services, and these have also been investigated on the traditional personal computers [28]. The focus of our research, however, is in the small SSDD domain (although ChatON does have a web-based client available for PC).

As noted earlier, the research done in the realm of SSDD evolved with the increased usage of third party applications. An analysis of the artifacts left by IM applications on iPhone such as AIM and Yahoo! Messenger as well as Google Talk that was used through the built in browser was performed in 2010 by Hassan and Sridhar [20]. The result of this analysis showed that there are many artifacts that can be gathered from AIM and Yahoo! Messenger. These artifacts included username, password, buddy list, last log-in time, and conversation timestamp as well as conversation details. But they could not retrieve many artifacts regarding Google Talk as it was used through the built in browser. The only extracted information was located in the history of the browser indicating the time the user signed in.

In 2010 Bader and Baggili [21] investigated and examined the logical backup acquisition of the iPhone 3GS mobile device using the Apple iTunes backup utility. Using tools such as a plist Editor, SQLite Database Browser, and iPhone Backup Extractor they analyzed the iPhone 3GS mobile and identified data related to the Facebook application in the phone's memory. The located database contained a log of the friends on the Facebook profile of the device user. The friend table within the database stored the full name, first name, last name, unique id and phone numbers for each friend in the list. This

table also contains the URL address pointing to the friend's profile picture on Facebook.

Levinson et. al. [22] discussed in their work the use of third party applications that included social networking apps as well as an instant messaging apps in order to create a time line of the suspect activities. The studied third party applications were iBooks,Yelp, FourSquare, Twitter, BrightKite, Skype, Where.com as well as Facebook.

Other work was done with regard to Windows Phone 7 in 2012 by Schaefer et. al. [23] The authors discussed the acquisition and analysis process of this device and indicated the retrieval of Facebook artifacts. The retrieved artifacts included the user's profile name and a link to the user's profile and profile picture. Every picture, which the user viewed via the Facebook app was located. Other interesting artifacts were information about the user's home page and last location if the feature was enabled as well as a list of the user's friends, including their birthday and their ID.

Tso et. al. [24] discussed diversification of the backup files with five popular social networking applications (Twitter, WhatsApp, Windows Live Messenger, Viber and Skype) via the comparison between after-installed-and-used and afterdeleted by iTunes backup process. Their work indicated that the five kinds of applications will output a fixed number of backup files and the fixed file names. Al Mutawa et. al. [5] studied social networking application such as Facebook, Twitter, and MySpace, which were used on BlackBerry, iPhone, and Android. The study explored the forensic acquisition, analysis and examination of the logical backup copies of the three Smartphones. The results showed that no traces of social networking activities could be recovered from BlackBerry devices. However, iPhones and Android phones stored a significant amount of valuable data that could be recovered and used by the forensic investigator.

Mahajan et. al. focused on conducting forensic data analysis of two widely used IMs applications on Android phones: WhatsApp and Viber [4]. Five Android phones were analyzed covering three different versions of Android OS: Froyo (2.2), GingerBread (2.3.x) and Ice-Cream Sandwich (4.0.x). The research identified many useful artifacts and their location in the Android file system. With regard to Viber information such as Viber numbers, the total number of calls made by user, date and duration of call, messages to Viber users in plain text, phone numbers to whom messages were sent and other valuable information. The analysis of WhatsApp also provided valuable information such as all the chat messages in plain text, the phone numbers of chat participants, as well as the time and date of chat sessions.

III. METHODOLOGY

The aim of this research was to identify artifacts left by the ChatON IM application on the investigated devices. We followed this basic methodology as we:

- Implemented a scenario of pre-defined activities on the investigated devices (See Fig 1)
- Acquired an image of these devices
- Performed manual analysis of the acquired image
- Identified our findings

ChatON is a multiplatform application, and for our experiment we selected two widely used platforms from the – the Apple iPhone (running iOS 6) and the Samsung Galaxy Note (running Android 4.1).

A forensic workstation was used to perform the acquisition and analysis of the investigated devices. The tools used to perform the manual analysis and acquisition were as follows:

- SQLite Database Browser
- plutil
- Apple iTunes Application

IV. IMPLEMENTATION AND ANALYSIS

Following the methodology in section 3 this section will provide a detailed description of the used scenario, acquisition process of each device and finally the analysis of the acquired image for each device.

A. Scenario

Two different accounts were created for the purpose of testing with screen names "Babar Iqbal" and "Asif Iqbal". Messages were exchanged between these accounts, "Asif Iqbal" using an iPhone (iOS version 6) device and "Babar Iqbal" using an Android (version 4.1) device. A variety of messages were sent and viewed including a video message and several photos taken using each device's camera. The apps also featured an animated drawing message that was also sent from the Android Device to the iPhone device.

B. iPhone Forensic Examination

This section describes the process of acquiring an image of the iPhone running iOS6 device along with the results acquired from the manual analysis.

1) Acquisition Methodology

The acquisition was done using Apple iTunes application which is a technique that was published by Bader and Baggili [21]. iTunes is a synchronization and management application that is freely available on Apple's official website. It is designed to synchronize data, applications, and media files between Apple devices (e.g., iPhone, iPad, and iPod) and the host computer. It is also used to create backup copies of data from the Apple device and save them on the host computer [5].

Using the backup feature of iTunes we were able to create a logical copy of the directories and various types of files found within the iPhone file system. Bader [21] identified that iTunes is not designed for forensic acquisition but it can be utilized for such a task. During the process of creating a backup of the iPhone the iTunes syncs the device by default with the computer. Then it will copy data from the iPhone to the PC and vice versa to ensure that content is same on both. This process might compromise the integrity of the acquired backup copy as data can be transferred to the phone's memory. Hence, in a forensic examination it is important to invoke the backup

Fig. 1 Screenshot of conversation between users "Babar Iqbal" and "Asif Iqbal" on Android and iPhone respectively

process independently without initiating the synchronization to avoid the risk of data cross contamination during the forensic logical acquisition [21].

2) Analysis Process

After acquiring the logical image of the iPhone device using Apple's iTunes application backup feature, the logical image was analyzed using the SQLite Database Browser and plutil. The most relevant evidence for this application resided in the main database for ChatON, which was found in "Library/Application Support/ChatOnClientApp.sqlite". Table 1 shows the key tables in this database. Tables that we did not identify as being of forensic value have been omitted in this list. The full database schema is published online at (http://students-cafe.com/asif/publications/sadfe2013).

TABLE I. Structure of database ChatOnClientApp.sqlite

Table	***Content***
ZBUDDY	Information about all of the user's contacts or "buddies"
Z_1GROUP	This table defines relationship of buddy groups
ZPRIMARYDATA	This table contains miscellaneous information related to user.
ZSERVERADDRESS	This table is used to store information about Samsung's ChatOn servers.
ZCONTACT	Information about user's phone contacts. This table is populated with information when Contact sync feature is enabled. All contacts that have a ChatOn account are automatically added to buddy list.
ZGROUP	This Table holds information about groups created by the user.
ZINBOX	Each chat session is assigned an inbox. Multiple buddies can be part of a single inbox.
ZMYPROFILE	Contains user's profile.
ZONMESSAGE	Contains a list of all messages and their contents

Telephone contacts are associated with ChatON buddies through a foreign key in the ZCONTACT table referring to a corresponding user record in the ZBUDDY table. The ZCONTACT table also records if a telephone contact which was associated with a ChatON buddy has been deleted. Since ZCONTACT also records if a buddy has been deleted, the reverse situation (telephone contact still present, ChatON contact deleted) can also be detected. These relationships helps link ChatON identities with phone contact identities.

The ZINBOX table has records for every open chat session, and messages within those sessions are recorded in ZONMESSAGE. Using the ZINBOX table, the investigator can find the download location on the iPhone's file system for files (e.g. photographs) associated with the chat. The investigator will be directed to the URL of the other participant's profile picture from this table. The investigator can also see which chats (in ZINBOX) and messages (in ZONMESSAGE) are marked as read and unread, potentially informing determinations about whether particular chat sessions were welcome or unwelcome to the user. Messages can be retrieved in plaintext. For graphical messages, thumbnails are recorded in ZONMESSAGE, and corresponding full-sized images will be present on the file system in a storage location referred to in ZONMESSAGE if the thumbnail has been clicked on (which may be relevant to the user's intent in receiving the image). Especially in cyber harassment, stalking or grooming cases, it is easy to see the value of the ZINBOX and ZONMESSAGE tables to the investigator.

The ZMYPROFILE table includes information about the user. It records whether the ChatON profile is linked to Facebook, and if so, it records the Facebook ID – allowing investigators to cross-reference ChatON and Facebook records. It also contains the email address/es, date of birth, status message and nickname for the user.

Apart from the main database, a plist file was also found on the iPhone device. The tables shown in section VI show a detailed description of the ChatON database structure and indicate the content of this database. This analysis showed that useful evidence can be gathered from the ChatON application on iPhones such as the plaintext content of the sent and received messages as well as the path to images or videos received in these messages etc.

C. Android Forensic Examination

This section describes the process of acquiring an image of the Samsung Galaxy Note running Android 4.1 device along with the result acquired from the image manual analysis.

1) Acquisition Methodology

Using an acquisition method similar to that described in [13], an image of the Samsung Galaxy Note (Android 4.1) device was acquired.

The acquisition process required the preparation of the forensic workstation and the Samsung Galaxy Note device. The device was connected to a forensic workstation that contains a payload to be injected into it using an Android Debug Bridge (ADB) to a temporary location. The payload included "known-good" binaries – crucially the dd utility.

This payload was then executed on the device in order to reboot the device with temporary root access using an exploit in Android Gingerbread. A bitstream image of the data partition was acquired using dd. Through this process we

created a physical image of the data partition that can be used to analyze the data on the Samsung Galaxy Note device.

2) Analysis Process

Using the methodology mentioned above the data was acquired from the Android device. The application uses "com.sec.chaton" as its class id, so the data of interest was found under directories "/data/data/com.sec.chaton" and "/sdcard/Android/data/com.sec.chaton". These directories hereon will be referred to as the application data directory and the userdata directory respectively.

The file most relevant to this investigation was the sqlite database "ChatOn.db" found under application data directory. Table 2, below, lists the most relevant tables mentioned in "ChatOn.db". The full schemas of all relevant tables found on the Android device is online at (http://students-cafe.com/asif/publications/sadfe2013).

TABLE II. List of relevant tables in ChatOn.db database file

Table	*Content*
buddy	Information about user's "buddies"
participant	Describes relationship between participants (buddies) of a chat and their inboxex (inbox_no of inbox table)
buddy_group	Information about groups of "buddies"
contacts	List of user's phone contacts, these can be used with contacts sync feature and can also be used to search for contacts that have ChatOn account.
inbox	Every chat session is assigned an inbox. Relationship between inbox and buddies is defined in participants table.
message	Contains all information related to message including content and timestamps.
inbox_buddy_relation	Table being used to define a relation between inbox and buddy.

The "message" and "inbox" tables on the Android device contain similar content to the ZONMESSAGE and ZINBOX tables on the iPhone. Again, one can see whether particular messages or chat sessions have been marked as read or not, the location of downloaded files from non-plaintext messages (e.g. images or audio messages), plaintext of text messages, timestamps, and other relevant information. Inboxes can be related back to their relevant chat partner by reference to the "inbox_buddy_relation" table and then to the "buddy" and "buddy_group" tables.

Records for all ChatON buddies are recorded in the "buddy" table. These can be related, where a connection has been made by the user, to mobile phone contacts in the Android Contact Provider service, via the "contacts" table. The "buddy_raw_contact_id" field in the "buddy" table serves as a foreign key to the "contacts" table if the user has established such a link by adding their telephone contact to ChatON.

Our analysis of the ChatON database files showed many valuable evidentiary data that can be extracted from a Samsung Galaxy Note running Android 4.1. Similar to the analyzed iPhone data we were able to identify all information related to messages including content in plain text and timestamps as well along with information about the files transferred during the conversation and the location where they are stored on the phone.

V. Conclusion and Future Work

In this research we performed a detailed manual analysis of ChatON artifacts that are left on an iPhone running iOS 6 and a Samsung Galaxy note running Android 4.1. Using tools such as SQLite Database Browser and plutil to perform the analysis of the investigated devices we were able to identify valuable evidentiary data that can aid in the forensic investigation process. An example of these data was the sent and received messages in plain text along with their timestamps. We were also able to identify the location of the files sent during the conversation and the timestamp of when they were sent. Full schema of the tables identified in our analysis of the application on each platform is online at (http://students-cafe.com/asif/publications/sadfe2013).

ChatON is a multiplatform application available on Android smartphones and tablets, iOS devices, BlackBerry, Windows and bada devices. In addition to these applications there is a web client. Possible work can be done to identify its artifacts that are left on other devices through their clients and/or through the web client. Although at a protocol level ChatON must function more or less the same in all of its variants, the application itself could operate differently on devices and different platform development teams might select different storage locations or database schemas entirely.

References

[1] Christensen, K., Levinson, D.: Encyclopedia of Community: From the Village to the Virtual World, SAGE, California (2003)

[2] Orebaugh, A., Allnutt, J.: Data Mining Instant Messaging Communications to Perform Author Identification for Cybercrime Investigations. Digital Forensics and Cyber Crime, vol. 31, pp. 99—110. Springer, Heidelberg (2010)

[3] To, P. L., Liao, C., Chiang, J. C., Shih, M. L., Chang, C. Y.: An empirical investigation of the factors affecting the adoption of Instant Messaging in organizations. In: Computer Standards & Interfaces, vol. 30, issue 3, pp. 148-156, March (2008)

[4] Mahajan, A., Dahiya, M. S., Sanghvi, H. P.: Forensic Analysis of Instant Messenger Applications on Android Devices. In: International Journal of Computer Applications, vol. 68, No.8, April (2013)

[5] Al Mutawa, N., Baggili, I., Marrington, A.: Forensic analysis of social networking applications on mobile devices. In: Digital Investigation, vol.9, pp. S24–S33 (2012)

[6] Al Zarouni M. Mobile handset forensic evidence: a challenge for law enforcement. In: Proceedings of the 4th Australian Digital Forensics Conference; 2006. Perth, Australia.

[7] Burnette, M. W. (2002). *Forensic Examination of a RIM (BlackBerry) Wireless Device*. Rogers & Hardin LLP. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.64.2385

[8] Samsung ChatON, https://web.samsungchaton.com/

[9] Willassen, S.: Forensics and the GSM mobile telephone system, In: International Journal of Digital Evidence, vol. 2, Spring (2003)

[10] Willassen, S. Y. (2005). Forensic Analysis of Mobile Phone Internal Memory. In M. Pollitt & S. Shenoi (Eds.), *Advances in Digital Forensics* (Vol. 194, pp. 191–204). Springer US. doi:10.1007/0-387-31163-7_16

[11] Mokhonoana, P., Olivier M.: Acquisition of a Symbian smart phone's content with an onphone forensic tool, In: Proceedings of the Southern African Telecommunication Networks and Applications Conference

2007 (SATNAC 2007), Sugar Beach Resort, Mauritius, September (2007)

[12] Distefano, A., Me, G.: An overall assessment of Mobile Internal Acquisition Tool. In: Digital Investigation, vol 5, Supplement, pp. S121-S127 September (2008)

[13] Lessard J, Kessler GC.: Android forensics: simplifying cell phone examinations. In: Small Scale Digital Device Forensics Journal, vol. 4, No. 1, September (2010)

[14] Vidas, T. Zhang, C., Christin, N.: Toward a general collection methodology for Android devices, In: Digital Investigation, vol 8, Supplement, pp. S14-S24, August (2011)

[15] Hoog, A.: Android Forensics Investigation, Analysis,and Mobile Security for Google Android [Book], Elsevier, ISBN: 978-1-59749-651-3, (2011)

[16] Sylve, J., Case, A., Marziale, L., Richard, G.: Acquisition and analysis of volatile memory from Android devices. In: Digital Investigation, vol 8, Issues 3–4, pp. 175-184 February (2012)

[17] Zdziarski J.: iPhone forensics: recovering evidence, personal data, and corporate assets. Sebastopol, CA: O'Reilly; (2010)

[18] Iqbal, Babar, Iqbal, Asif, Al-Obaidli, Hanan: A Novel Method of iDevice (iPhone,iPad,iPod) Forensics without Jailbreaking. In: International Conference on Innovations in Information Technology (IIT), pp. 238-243, Al Ain, Abu Dhabi, IEEE, (2012)

[19] Gómez-Miralles, L., Arnedo-Moreno, J.: Universal, fast method for iPad forensics imaging via USB adapter. In: Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, (2011)

[20] Husain, M. I.,Sridhar, R.: iForensics: Forensic Analysis of Instant Messaging on Smart Phones," In: Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering: Digital Forensics and Cyber Crime, vol.31, pp. 9-18, (2010)

[21] Bader M., Baggili I.: iPhone 3GS forensics: logical analysis using Apple iTunes backup utility. In: Small Scale Digital Device Forensics Journal, vol.4 No. 1 September (2010)

[22] Levinson, A., Stackpole, B., Johnson, D.: Third Party Application Forensics on Apple Mobile Devices, In: Proceedings of the 44th Hawaii International Conference on System Sciences, (2011)

[23] Schaefer, T., Höfken, H., Schuba, M.: Windows Phone 7 from a Digital Forensics' Perspective. In: Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering: Digital Forensics and Cyber Crime, vol. 88, pp. 62-76, (2012)

[24] Tso, Y.C., Wang, S.J., Huang, C.T., Wang, W. J.: iPhone social networking for evidence investigations using iTunes forensics. In: Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication. pp. 62:1--62:7, New York, ACM, (2012)

[25] Reust, J. (2006). Case study: AOL instant messenger trace evidence. *Digital Investigation*, *3*(4), 238–243. doi:10.1016/j.diin.2006.10.009

[26] Dickson, M. (2006). An examination into AOL Instant Messenger 5.5 contact identification. *Digital Investigation*, *3*(4), 227–237. doi:10.1016/j.diin.2006.10.004

[27] Kiley, M., Dankner, S., & Rogers, M. (2008). Forensic Analysis of Volatile Instant Messaging. In *Advances in Digital Forensics IV* (Vol. 285, pp. 129–138). Boston: Springer.

[28] Al Mutawa, N., Al Awadhi, I., Baggili, I., & Marrington, A. (2011). Forensic artifacts of Facebook's instant messaging service. In *Internet Technology and Secured Transactions (ICITST), 2011 International Conference for* (pp. 771–776).

VI. iPHONE TABLE SUMMARIES

TABLE III. Table ZCONTACT of ChatOnClientApp.sqlite database file

Column	*Content*
ZDELETED	Indicates if original Phone contact associated is deleted
ZORIGINALNAME	Original name of the contact as it was stored in Phone's contact list
ZORIGINALNUMBER	Original number stored in Phone's contact list.
ZPHONENUMBER	This number is used to associate user's Phone contact with buddy of matching mobile number.

TABLE IV. Table ZBUDDY of ChatOnClientApp.sqlite database file

Column	*Content*
Z_PK	Primary Key
Z_ENT	
Z_OPT	
ZCONTACTSID	
ZIMAGESTATUS	
ZINTERACTIONRANK	
ZINTERACTIONRANK_PRE	
ZINTERACTION_POINT	
ZINTERACTION_RECEIVED	
ZINTERACTION_SENT	
ZISADDEDBUDDY	Indicates if this user is added as buddy. Messages can also be sent to users that are not added as buddies. 1 = Added as Buddy null or 0 = Not added as Buddy
ZISAUTHUSER	
ZISBLOCKEDBUDDY	Indicates if buddy is blocked null = Not Blocked 1 = Blocked
ZISBUDDYSSAYBLIND	
ZISDELETEDBUDDY	Indicates if user has been deleted from Buddy List
ZISFAVORITE	Indicates if user is part of favorites group 1 = Is Favorite 0 or null = Is not favorite
ZISFIXEDDISPLAYNAME	Indicates if the user has changed this buddy's name to their choice
ZISINTERACTIONHIDE	
ZISNEWBUDDY	Indicates if the buddy is new. When no messages has been exchanged the buddy is said to be new 0 = Not New 1 = New Buddy
ZISNOBUDDY	Indicates if this user is not a buddy 1 = This user is not a buddy 0 = This user is a Buddy
ZISSHOWPHONENUMBER	Indicates if this user allows service to disclose mobile number.
ZISSPECIALBUDDY	
ZTIMESTAMP	

ZCONTACTLIST	
ZFOLLOWERCOUNT	
ZLIKECOUNT	
ZSERVICE_STATUS	
ZBIRTHDAY	
ZCOUNTRYCODE	
ZDISPLAY_NAME	Display name of the buddy. By default this is the same as buddy's nickname but can be changed to user's liking. Changing this will set ZISFIXEDDISPLAYNAME to true.
ZEMAIL	Email address of the buddy
ZMSISDN	Unique ID of the buddy
ZMSISDN_LIST	List of IDs associated with the buddy
ZNICKNAME	Nickname of the buddy
ZORG_NAME	Name of the buddy according to Samsung Account associated with the buddy
ZORG_NUMBER	Phone Number of the buddy according to Samsung Account associated with the buddy
ZORG_NUMBER_LIST	List of Phone Numbers of the buddy according to Samsung Account associated with the buddy
ZSAMSUNGEMAIL	Email of the buddy according to Samsung Account associated with the buddy
ZSECTION	
ZSTATUS	Status Message of the buddy
ZCONTENT_URL	
ZMESSAGE	
ZPROVIDER_URL	
ZSBUDDY_DESCRIPTION	
ZTEL	
ZWEBVIEW_URL	

TABLE V. Table ZGROUP of ChatOnClientApp.sqlite database file.

Column	***Content***
ZISNEW	Indicates whether the group is new. N = Group is not new
ZGROUPID	Unique id of the Buddy Group
ZGROUPIMAGETIMESTAMP	
ZNAME	Name string of the group
ZSECTION	

TABLE VI. Table ZINBOX of ChatOnClientApp.sqlite database file

Column	***Content***
Z_PK	Primary Key of the table
Z_ENT	
Z_OPT	
ZINBOX_CHAT_MEMBER_TIMESTAMP	
ZINBOX_CHAT_TYPE	
ZINBOX_ISCHAGEDTITLE	Indicates if title of Inbox was changed from Original

ZINBOX_LASTSESSIONMERGETIMESTAMP	
ZINBOX_LAST_MSG_ID	Unique id of last message sent or received. The unique id is ZMESSAGE_ID in ZONMESSAGE table.
ZINBOX_LAST_MSG_TIME	Integer Timestamp of last message sent or received.
ZINBOX_PUSH	
ZINBOX_STATUS	
ZINBOX_UNREAD_MSG_COUNT	Number of unread messages in inbox
ZINBOX_USE_TIME	Timestamp of last usage of the inbox i.e. time of last message sent
ZINDEX	
ZINBOX_BUDYGROUPID	If the inbox is associated with a group this field will hold the unique id of that group specified in ZGROUP table.
ZINBOX_DOWNLOAD_PATH	Unique path to save file downloaded in this inbox, files are saved under /var/mobile/Applications/{application-id}/Library/Caches/{ZINBOX_DOWNLOAD_PATH}/
ZINBOX_LANG_FROM	If translation feature is enabled this field contains the original language of the message.
ZINBOX_LANG_TO	If translation is enabled this field contains the language that message should be translated to.
ZIBOX_LAST_MESSAGE	Content of last message received in the inbox
ZINBOX_OLD_NO	
ZINBOX_PROFILEIMAGE_URL	If the buddy associated with the inbox has a profile picture set, this will be the path to that image.
ZINBOX_SERVER_IP	IP address of Samsung Server on which this inbox is stored
ZINBOX_SERVER_PORT	Port number of Samsung Server inbox is stored on.
ZINBOX_SESSION_ID	Unique session id associated with the inbox
ZINBOX_TILE	String title of the inbox. This title is generated by concatenating the names of buddies associated with the inbox.
ZINBOX_TITLE_ARRANGE	
ZINBOX_TITLE_FIXED	If user replaces the generated title with a new one, this field holds the new title.
ZINBOX_UNCOMPLETED_MSG	

TABLE VII. Table ZMYPROFILE of ChatOnClientApp.sqlite database file

Column	***Content***
ZBIRTHDAY_MODE	
ZFACEBOOK_STATUS	Indicates whether Facebook is associated with user account 1 = associated 0 = not associated
ZIMAGESTATUS	
ZISSHOWPROFILEIMAGE	1 = Show profile image 0 = Don't show profile image
ZISSHOWSAMSUNGACCOUNT	1 = Show Samsung account 0 = Don't show Samsung account
ZPRIVACY_INFO	
ZDATEOFBIRTH	Date of birth of the user.
ZEMAIL	Email address associated with user account

ZFACEBOOK_ID	If there is a Facebook account associated with user's ChatOn account this field will contain the id of the associated account. If no Facebook account is associated this field is set to string “Signin”
ZNICKNAME	Nickname specified by user during creation of the account
ZSAMSUNG_EMAIL	If user account is associated with a Samsung account, this field will hold the email address used to create the Samsung account.
ZSTATUS_MSG	This field contains the status message set by the user

TABLE VIII. Table ZONMESSAGE of ChatOnClientApp.sqlite database file

Column	***Content***
ZMESSAGE_ID	Unique id of the message
ZMESSAGE_RECEIVER_COUNT	Indicates whether the user is receiver of the message or the sender 1 = Reaciver 0 = Sender
ZMESSAGE_SEND_STATUS	Indicates if message was successfully sent. 1 = Message was sent successfully 0 = Messages was failed to be sent
ZMESSAGE_TIME	Timestamp of message
ZMESSAGE_TYPE	Integer indicating the type of message. Following are the valid values: 0 = Plaintext Message 1 = Mixed, Image or Video Message
ZMESSAGE_WILL_DELETE	Indicates if message is marked to be deleted.
ZINBOX	Foreign key to Z_PK column of ZINBOX table. This field indicates which inbox the message belongs to.
Z5_INBOX	Foreign key to Z_OPT column ZINBOX table. This field indicates which inbox the message belongs to.
ZSENDER	Foreign key to Z_PK column of ZBUDDY table. This field indicates which buddy sent the message. This field is null when the message is outgoing.
Z1_SENDER	Foreign key to Z_OPT column of ZBUDDY table. This field indicates which buddy sent the message. This field is null when the message is outgoing.
ZMESSAGE_READ_STATUS	Indicates whether the message is read. 1 = Message is read 0 = Message is not read
ZMESSAGE_ISANIMATIONMESSAGE	Integer flag that indicates whether the message is an ChatOn Animated Message
ZMESSAGE_MEDIA_DURATION	Integer that contains the duration of media if the message is an audio or video message.
ZMESSAGE_CONTENT	Contains plaintext content of the message. If message is of type image or video this field contains string value associated with the type.
ZMESSAGE_SECTION	
ZMESSAGE_TID	
ZMESSAGE_ANSWERER	Unique id of buddy who replied to the message. This is same as ZBUDDY_ID in ZBUDDY table.
ZMESSAGE_TRANSLATED	Indicates whether the message has been translated.
ZMESSAGE_FILE_TYPE	This field holds string representation of the message type
ZMESSAGE_MIXED	If message is mixed message, this field hold the plaintext part of the message

ZMESSAGE_ORIGINAL_DATA_PATH	This field holds path to image or video received in the message. These files are stored in /var/mobile/Applications/{application-id}/Library/Caches/{ZINBOX_DOWNLOAD_PATH}/{unique_file_id}.{file_extension}
ZMESSAGE_THUMBNAIL	If message is an image, this field holds binary blob of image's thumbnail
ZMESSAGE_THUMBNAIL1	If message is a video, this field holds a thumbnail of first frame of the video

I. ANDROID TABLE SUMMARIES

Please note that these pages are provided as an appendix for the reviewers only – they are over the page limit and will be removed from the final print version. Instead we will make them available online for practitioners to access freely.

TABLE IX. Table message of ChatOn.db database file

Column	***Content***
id	Unique id of the message
message_server_id	Unique id of message on Samsung's Server
message_inbox_id	Corresponds to inbox_id in inbox table of the ChatOn.db
message_session_id	Each message is stored with session_id, each chat buddy has a unique session id stored in inbox_session_id column of inbox table. This was not set during a "broadcast" type message.
message_read_status	Indicates if message is read or not.
message_content_type	Integer value indicating content type of the message. Following are the valid values: 0 = plain text message 1 = image 2 = video in mp4 container 12 = ChatOn animated message
message_time	An unformatted timestamp integer. add time format
message_content	This column holds content of the message. If message is not plaintext it holds the type string of message: "image" if message is an image "ams" for ChatOn animated message "mixed" for everything else
message_translated	If translation feature is enabled this field contains the translated content of the message
message_type	
message_sender	Stores unique id of message sender same as buddy_no in buddy table
message_download_uri	This field is used to store path to downloaded file in case that the message is not plaintext. These files are stored in {Userdata Directory}/files/{message_inbox_no}/{unique_name}.{file_extionsion}.
message_formated	
message_tid	
message_time_text	Message sent or received time stored in formated string
message_stored_ext	
message_need_update	
message_is_failed	Indicates if message has failed delivery. 0 = message successfully sent 1 = message delivery failed
message_is_file_upload	Indicates if sent message is a file to be uploaded, it is true(1) if user sends a image or video message but not when user receives one. 0 = false 1 = true

message_is_truncated	Indicates if message is truncated. N = not truncated send long message
message_old_session	
message_old_sender	
message_content_translated	Holds translated content of the message if translation feature is enabled
message_from_lang	Original language of message
message_to_lang	Language of translation
message_is_spoken	Indicates if message is an audio message

TABLE X. Table inbox of ChatOn.db database file

Column	***Content***
id	
inbox_no	Each chat session is assigned a unique inbox number. All messages exchanged with one buddy have a single inbox_no associated to them . Messages with a group also have a unique inbox_no.
inbox_chat_type	
inbox_unread_count	Integer value indicates number of unread messages in inbox
inbox_last_message	Holds message_id (message table) of last received or sent message.
inbox_title	String title of the displayed in the app. This string contains names of buddies in chat session concatenated using a comma.
inbox_last_time	Integer that holds timestamp of last sent or received message
inbox_lang_from	If translating is enabled this field holds the Language of original message
inbox_lang_to	If translating is enabled this holds the language of the user
inbox_translated	Indicates the translating status of inbox. N = Not being translated
inbox_server_ip	IP address of server being used to store the inbox
inbox_server_port	Port of server being used to store the inbox
inbox_participants	Number of participants in the inbox
inbox_session_id	Session id associated with the inbox , no session id is set when sending a broadcast type message.
inbox_last_msg_no	message_no (message table) of last message.
inbox_last_msg_sender	buddy_no (buddy table) of sender of last message
inbox_title_fixed	
inbox_last_chat_type	
inbox_last_temp_msg	
inbox_is_new	
inbox_trunk_unread_count	
inbox_valid	
inbox_enable_noti	
inbox_last_timestamp	
inbox_is_change_skin	
inbox_background_style	
inbox_send_bubble_style	
inbox_receive_bubble_style	

inbox_is_entered	
inbox_web_url	
profile_url	
lasst_session_merge_time	
inbox_old_session_id	
inbox_old_no	
inbox_translate_my_language	Language of the user.
inbox_translate_buddy_language	Language of chat participant or “buddy”.
inbox_enable_translate	Indicates if incoming messages are to be translated.
translate_outgoing_message	Indicates whether outgoing messages are to be translated.
inbox_last_tid	

TABLE XI. Table buddy of ChatOn.db database file

Column	***Content***
_id	
buddy_no	Unique number of user's contact or “buddy”
buddy_name	Name string of the buddy
buddy_status_message	Status message set by the buddy
buddy_email	Email address of the buddy
buddy_samsung_email	If the buddy has associated a Samsung account to their account, this field will hold the address to their Samsung account.
buddy_orignal_number	Mobile number of the buddy
buddy_birthday	Birthday of the buddy
buddy_msg_send	
buddy_msg_received	
buddy_relation_hide	
buddy_raw_contact_id	If the buddy was added from user's phone contacts, this field will hold unique integer value of that contact (contact_raw_id in contacts table). Else this value will be set to zero.
buddy_push_name	
buddy_is_new	
buddy_profile_status	
buddy_is_profile_updated	
buddy_is_status_updated	
buddy_show_phone_number	
buddy_extra_info	
buddy_is_name_updated	
buddy_updated_timestamp	
buddy_orignal_numbers	
buddy_msisdns	
buddy_multidevice	
buddy_sainfo	
buddy_account_info	

buddy_original_name	If buddy was added using Conact Sync feature, this field will hold name as specified in user's phone contacts.
buddy_using_contact_name	

TABLE XII. Table buddy_group of ChatOn.db database file

Column	***Content***
_id	Primary integer key
group_name	Name of the group.
group_type	
group_is_new	Indicates age of the group. N = Group is not new

TABLE XIII. Table contacts of ChatOn.db database file

Column	***Content***
contact_id	Unique id generated by the ChatOn application.
contact_raw_id	Unique id of contact as specified in Android Contact Provider service used by application to connect to the contact store of the device.
conatct_number	Mobile number of the contact.
contacts_name	Name of the contact.

TABLE XIV. Table participant of ChatOn.db database file. This table Relates Entries in table “buddy” with entries in table “inbox"

Column	***Content***
participants_buddy_no	buddy_no as specified in buddy table
participants_inbox_no	inbox_no of the participants. This connects inbox_no in inbox table to buddy_no in buddy table. In case of a group chat or broadcast type chat single inbox_no can be associated with multiple buddies.
participants_buddy_name	Name of the buddy (buddy table)
participants_country_code	
participants_is_auth	
participants_status	