Forensic Analysis of the Recovery of Wickr's Ephemeral Data on Android Platforms

Thomas Edward Allen Barton and M A Hannan Bin Azhar
Computing, Digital Forensics and Cybersecurity
Canterbury Christ Church University
Canterbury, United Kingdom
Email: tb1150@canterbury.ac.uk; hannan.azhar@canterbury.ac.uk

***Abstract*—This paper documents anti-forensics techniques of a secure messaging application named "Wickr" on the Android platforms. Advertised as an application that focusses on security, Wickr provides many anti-forensics features, such as ephemeral messaging and end-to-end encryption. This paper analyses Wickr in detail using experimental research methods. The results revealed how Wickr's file deletion consisted of distinct stages beginning with a simple logical deletion and progressing to overwriting deleted files as the application operated.**

Keywords- wickr; android; ephemeral messaging; forensic analysis; data sanitization; mobile forensics

I. INTRODUCTION

The advent of smartphones has allowed for a huge range of third party applications to be developed [1]. Third party applications used to commit crimes present a challenge to digital forensic investigators as vital digital evidence is obscured by their unknown nature and structure. This challenge creates the need for research into how these applications operate. The opportunity for malicious actions presented by third party applications, especially those that emphasize security, is obvious. In the last two years, two large terror attacks were carried out in Europe [2] [3]. These acts are highly coordinated in a fashion that could not occur without technology. Reports on usage of the social media by the Islamic State group (ISIS), for example, have demonstrated how a small group of individuals can fully utilise technology to their advantage, for the purposes of recruitment [4], coordination [5] and communication [6].

One category of the communication applications is an Ephemeral Messaging Application (EMA) that uses transient data [7]. Transient data are only permitted to exist for a limited amount of time. An example of an EMA that rose to huge popularity among the eighteen to thirty-four year old age range (which happens to be statistically the largest demographic for smartphone ownership in the United States [8]) was SnapChat [9], an application that allowed users to send photos and images that would be deleted after opening. In SnapChat's case, the use of ephemeral messaging provided a novel social networking platform that quickly attracted many users. After SnapChat's popularity, other applications began to adopt the feature, including Wickr [10], which aims to provide a service that goes beyond simple social networking. Wickr's use of transient data is part of its professional dedication to security; other features concurrent with this theme include end-to-end encryption [10]. Third party applications with a focus on security thus pose a challenge to digital forensics investigators in retrieving artefacts especially when an application like Wickr incorporates anti-forensics features, such as ephemeral messaging and encryption, in its core functionality. To investigate the organization of the data and to understand how the deletion process works in secure messaging applications like Wickr, a series of experiments were conducted on the Android based platforms. A number of forensic tools were used to inspect different areas of the platforms including data storage, Random Access Memory (RAM) and the Wickr application itself. The experiments reported in this paper are listed in chronological order, each one building on from the last.

The remainder of this paper is organized as follows: Section 2 explains the objectives of this research in relation to previous work completed, and Section 3 goes on to detail the experimental setup and tools used in this project. Sections 4 to 8 detail the experiments, one per section. These experiments follow a pattern of acquisitions and analyses on various areas of interest on the target platform, including the data storage, RAM and the Wickr application itself. Finally, conclusions and future work are discussed in Section 9.

II. OBJECTIVES

Research into the forensic analyses of social media applications often focusses on the most popular applications at the time [11, 12]. Ephemeral data techniques first emerged after the previously mentioned SnapChat gained popularity. The methods used in [7] were successful in recovering artefacts, using physical image analysis of the test platforms. The work reported in [7] demonstrated the ability of forensic investigators to overcome the deletion and obfuscation of artefacts by an ephemeral messaging application. Since SnapChat, other applications have adopted ephemeral messaging as part of their features. Wickr, a secure messaging application, employs anti-forensics tactics, including ephemeral messaging [10]. In the face of such tactics, the use of standard forensic methodologies, such as string searches, may yield challenges in recovering artefacts. Previous investigations reported in [13] have focused on

 ISBN: 978-1-61208-512-8

using such plaintext searches to look for artefacts, but Wickr's extensive use of encryption hampered the results.

The aim of the research presented in this paper was to analyse the data storage, removal and sanitization techniques used by Wickr, in order to provide digital forensics investigators some insight on how Wickr operates and also to provide some useful techniques for analysing a secure mobile application like Wickr. The methodology adopted in this investigation took into account that previous attempts to retrieve artefacts, for example searching a backup of the Android data directory for matching strings were not successful as string searches for known artefacts in Wickr revealed no matches [13]. Our investigation overlooked the encryption problem in Wickr and focused instead on how Wickr stores and treats its transient data. To achieve the aim of this research, a number of experiments and analyses were performed on Wickr and its relevant data. These actions can be categorized into two distinct classes. Firstly, forensically sound analysis, which refers to techniques used by forensic investigators in a real-world case with the aim of presenting evidence in court [14], in conjunction with the Association of Chief Police Officers (ACPO) guidelines [15]. Secondly "experimental" analysis, which takes advantage of the freedom of academic research. Results presented in this paper identified the files that Wickr used to store its transient data. Experiments were also conducted to observe how these files could change as the app removed them.

III. EXPERIMENTAL SETUP

In order to establish an experimental setup, Wickr had to be installed on two different Android Platforms. The Samsung Galaxy S4 Mini [16] was the primary platform used. The AllWinner A13 Android tablet [17] was used as a backup platform to ensure the repeatability of all experiments performed. The choice of platform in this case, a phone from Samsung's flagship galaxy range, reflects the current state of the worldwide smartphone market, which is dominated by Android [18], the market for which is in turn dominated by Samsung [18]. Another reason Android was chosen was its huge online developer community, which stems from its open source status. Table 1 lists the detail of platforms, including the versions used in the experiments for Android, kernel and Wickr.

TABLE I. ANDROID PLATFORMS

Name	Specifications			
	Model Number	***Android Version***	***Kernel Version***	***Wickr Version***
Samsung Galaxy S4 Mini [16]	GT-I9195I	4.4.4 (KitKat)	3.10.28-5334500	2.6.4.1
AllWinner A13 [17]	Q8	4.4.2 (KitKat)	3.4.39	2.6.4.1

In order to access all areas of the platform's data storage systems, they needed to be rooted. In the Android developer community, rooting refers to the process of gaining administrator privileges on the device through exploitation of weaknesses in the operating system [13]. This was done via an application called Kingo Root [19], which offers a simple rooting solution. In a real-world scenario, the option of rooting may not be available as it involves changing data on a captured device, which goes against the ACPO guidelines for handling digital evidences [15], and should only be used as a last resort. Investigators may need to use another method to gain access, such as "chip-off" forensics [20]. This involves removing the memory chip of a device and reading the stored data using a bespoke hardware interface. This bypasses any restrictions as it can be performed directly and independently of the platform's operating system.

A. Forensic Worksation and Software tools

Two forensic workstations were used, listed in Table 2, the first with Windows, which makes accessing specific tools easier. The second had a distribution of Linux named Kali which came with forensic tools pre-installed on the system.

TABLE II. FORENSIC WORKSTATIONS

Name	Specifications	
	Operating System	***Installed Software***
RM Desktop Core i3	Windows 7	Android Debug Bridge [21] Dex2Jar [22] Java Decompiler [23] File Manager (ZIP Extractor) Autopsy 3.0.8 [24]
Toshiba Sattelite L450D	Kali	Android Debug Bridge [21] Cat (Linux) Strings (Linux) BASH (Linux) Sleuth Kit [24] Mount (Linux)

Most of the software tools used to perform analyses were specific to the experiments. Android Debug Bridge (ADB) is a tool that allows for access to the mobile platform via USB cable [21]. This is a useful tool as it allows command line execution on the platform from the forensic workstation. The Wickr application itself was examined once the "classes. dex" artefact was extracted and converted to a Java Archive (see Section 4). Java Archives have a custom format and are not easily readable in text editors, so a specific tool for Windows, Java Decompiler [23], was used to examine this artefact. Java Decompiler presents the archive in a tabulated format that makes analyses easier.

To examine the data storage, Sleuthkit was used. Sleuthkit is an open-source digital forensics toolkit that revolves around the recovery of deleted files [24]. It is a set of command line tools for Linux. Autopsy was developed by creating a Graphical User Interface (GUI) for Sleuthkit, combining all of the included tools into one seamless package. Autopsy performs very much the same functionality as Sleuthkit, but in GUI form, offering advantages such as file previews. The analysis of acquired data with standard forensics tools previously reported such as Hex Workshop [13], Encase and DCode [11] was not suitable in our experiments, as initial tests confirmed the lack of any artefacts available for recovery. Instead, the

 ISBN: 978-1-61208-512-8

experiments searched for alternate artefacts, such as the location and status of files. To do this, simple terminal commands, such as "ls", "strings" and "cat", which are all included in the forensic workstations' distribution of Linux, were used to examine acquired files and directory structures.

IV. EXPERIMENT 1: WICKR.APK DECONSTRUCTION

The objective of the first experiment was to explore the installation package for Wickr in order to understand how it functioned. Android uses installation packages to install new software on the platform. The installation packages used are file archives with the extension ".apk". These contain all the compiled code that is needed to run the application, including both core and third party libraries. Included in these libraries are the functions that Wickr uses to store data. This experiment accessed the data storing functions contained in these libraries so they can be comprehensively understood. A useful artefact contained in the archive is the "classes.dex" file, which contains all the definitions for classes used by Wickr. The "classes.dex" file is composed of compiled code that cannot be analysed using a plaintext analysis method. However, the tool "dex2jar" from the dex2jar project [22] was used to convert this file into a Java archive. The Java Archive was opened using a specialized tool, "Java Decompiler" [23]. There were four steps to this procedure: acquiring the ".apk" using an ADB pull command as seen in Fig. 1, extracting the "classes.dex" file from the archive, converting this file to a Java archive using the tool "dex2jar", and lastly examining the archive using the windows tool "Java Decompiler".

```
Windows PowerShell
Copyright (C) 2015 Microsoft Corporation. All rights reserved.

PS C:\Users\Thomas> adb pull /data/app/com.mywickr.wickr2-1.apk <Local destination>
```

Figure 1: ADB Pull command for Wickr's APK

Android stores applications under "/data/app" [25] and the filename for Wickr is "com.mywickr.wickr2-1.apk". The resulting parameters for the "adb pull" command are displayed in Fig. 1. After opening the "com.mywickr.wickr2-1.apk" file using a ".zip" archive extractor (for example 7-zip or Windows Explorer) the "classes.dex" file was extracted. Using the tool "dex2jar" [22] the "classes.dex" file was converted into a Java archive.

WickrDBAdapter.class

```
package com.mywickr.wickr;

import android.app.Application;
import android.content.Context;
import android.content.SharedPreferences;
import android.content.SharedPreferences.Editor;
import com.mywickr.helpers.SharedPreferencesHelper;
import java.io.File;
import net.sqlcipher.Cursor;
import net.sqlcipher.database.SQLiteDatabase;
import net.sqlcipher.database.SQLiteDatabaseHook;
import net.sqlcipher.database.SQLiteOpenHelper;
import timber.log.Timber;
```

Figure 2. WickrDBAdapter.class header

Data was stored in a database file, which was managed by an SQL Helper class. Shown in Fig.2, the SQL helper class found in the system was "net.sqlcipher.database.SQLite OpenHelper". SQLCipher is an extension to the SQLite database engine that incorporates encryption into its functionality [26]. An extract from the "WickrDBAdapter. class" directory, as shown in Fig. 3, includes variable names such as DATABASE_NAME and their respective values.

```
private static final String DATABASE_NAME = "wickr_db";
private static final int DATABASE_VERSION = 12;
static final String Partner_User_KEY_Wickr_User = "wickrUser";
static final String Partner_User_KEY_avatarUrl = "avatarUrl";
static final String Partner_User_KEY_id = "_id";
static final String Partner_User_KEY_isHidden = "isHidden";
static final String Partner_User_KEY_status = "status";
static final String Partner_User_KEY_userName = "userName";
static final String TABLE_Partner_User = "Partner_User";
public static final String TABLE_Wickr_Account = "Wickr_Account";
```

Figure 3. Extract from WickrDBAdapter.class

V. EXPERIMENT 2: ACQUIRING AND ANALYSING WICKR DATA DIRECTORY

A key part of how an application functions is how and where it stores data in permanent secondary storage. Analysing the data stored by Wickr revealed exactly how the data was stored. To analyse this data, it first had to be acquired. On Android platforms, all data for Wickr is stored in the "com.mywickr.wickr2" directory within the "/data/data" directory [25]. This area of storage is inaccessible unless the investigator has administrative privileges, i.e. the platform is rooted (see Section 3). To acquire this directory and its contents, an external SD card was mounted in the phone. Fig. 4 shows how the UNIX "cp" (copy) command was used recursively to acquire the directory.

```
1|root@serranovelte:/data/data/com.mywickr.wickr2 # cp -av /data/data/com.mywi>
/data/data/com.mywickr.wickr2 -> /mnt/sdcard/Documents/wickr.directory/com.mywick
cp: symlink: /data/app-lib/com.mywickr.wickr2-1: Operation not permitted
/data/data/com.mywickr.wickr2/cache -> /mnt/sdcard/Documents/wickr.directory/com.
/data/data/com.mywickr.wickr2/cache/bugsnag-errors -> /mnt/sdcard/Documents/wickr
/data/data/com.mywickr.wickr2/cache/bugsnag-errors -> /mnt/sdcard/Documents/wickr
/data/data/com.mywickr.wickr2/cache/com.android.opengl.shaders_cache -> /mnt/sdca
s_cache
```

Figure 4. Dumping Wickr Data Directory to SD card

Upon inspection of the contents of these files using the UNIX "*cat*" command, it becomes clear that files such as "wickr_db" and the ".wic" files lack normal file headers. The resulting conclusion was that Wickr uses encryption, confirming the hypothesis of experiment 1. There were two areas of interest in the directory, including "databases" and "files", both were the subdirectories of the "com.mywickr. wickr2" directory, as seen in Fig. 5.

app_sfs	06/06/2016 15:15	File folder
app_webview	06/06/2016 15:15	File folder
cache	06/06/2016 15:15	File folder
databases	06/06/2016 15:15	File folder
files	06/06/2016 15:15	File folder
no_backup	06/06/2016 15:15	File folder
shared_prefs	06/06/2016 15:15	File folder

Figure 5. Wickr Data Directory Structure

 ISBN: 978-1-61208-512-8

The "databases" directory, shown in Fig. 5, contained a file called "wickr_db", which was an encrypted database. Upon examination of the "WickrDBAdapter.class" file shown in Fig. 3, it became clear that this database was used to store account information, such as usernames of the user and their contacts, IDs, public and private keys. The "files" directory is where Wickr stores its received messages, including both text and attachments. This will be explored in the next experiment.

VI. EXPERIMENT 3: WICKR'S DATA REMOVAL AND SANITIZATION METHODS

The nature of transient data requires an ongoing process consisting of at two logical stages. Firstly, the data must be created and stored. Wickr's data is generated by receiving messages. The second stage is the removal of data. This experiment will explore how Wickr achieves the second logical stage, data removal, by examining the data both when they are present and after they have been removed. One of Wickr's features is a "secure shredder" that offers removal of deleted data. Copies of the data directory were be taken when messages were present (opened) in Wickr, after messages had expired, and after the "secure shredder" function had been executed. This experiment used the acquisition method described in experiment 2. Table 3 shows the disparities in the structure and contents of the data directories that revealed how the data was being treated during the handling of transient data.

TABLE III. DATA DIRECTORY ANALYSIS RESULTS

Stage of File Removal (Arbitrary)	*Secure Shredder Status*	*Copy Taken*	*Files Directory and Further Observations*
1	Before	Before images were received	Only two ".wic" files, "pcc.wic" and "pcd.wic" were present in the files directory.
2	Before	After images were received	Two ".wic" files, each with 64 character string file names, were present in the files directory. Their sizes were 47488 and 54136 bytes respectively.
3	Before	After images were removed	Two ".wic" files were not present.
4	After	After images were removed	Two ".wic" files were not present.

These disparities can be monitored using general-purpose UNIX tools such as "ls" for directory listing and "cat" for the examination of the contents of files. The concurrence of the amount of received messages and the amount of "*.wic" files in the files subdirectory reveals that Wickr stores its received attachments as ".wic" files. From the observations in Table 3, a record of Wickr's file decay were established, as seen in Table 4. After the received messages had been deleted, the corresponding ".wic" files were no longer present. The type of acquisition performed in this experiment by using the command "cp" relied on the file headers to locate blocks of data on the storage media. The files did not show up after expiry, which indicates that Wickr had removed the files, at least from the logical filesystem. To check if the files were completely removed without any trace of the data anywhere in the device, a low-level data acquisition had to be performed; the outcome of this will be discussed in the next experiment.

TABLE IV. WICKR'S FILE DECAY

Stage of file removal	*1*	*2*	*3*	*4*
Status of received file	N/A	File present, encrypted, stored in ".wic" file	File present, encrypted, filesystem header removed	File overwritten with random or null data
Process required to recover file	N/A	Logical level acquisition, for example copy.	Low level acquisition, such as device data dump or chip-off analysis	File unrecoverable .

VII. EXPERIMENT 4: LOW LEVEL DATA ANALYSIS

The results from the experiment 3 showed that there were multiple stages to Wickr's removal of files. A crucial stage in the recovery of expired files is when their filesystem headers have been removed, so they cannot be accessed via the application, but their contents still reside in unallocated space, as they have not been overwritten. To examine unallocated space on the "userdata" partition, a low-level logical acquisition was performed. Low-level acquisition captures deleted data that resides in unallocated space, something that could not be achieved using the filesystem alone.

Figure 6. Cellular Phone Tool Levelling Pyramid [27]

Fig. 6 shows the cellular phone tool-levelling pyramid [27], which is a model used to describe the increasingly complex, expensive and forensically sound levels of mobile forensic acquisitions and analyses. The previous experiments performed have relied on the filesystem which is also a type of logical analysis situated the second level of the pyramid.

In Fig. 7, the listing for "/dev/block/bootdevice/by-name/", shows the mount points of various android partitions. For this experiment, the target partition was the

 ISBN: 978-1-61208-512-8

data directory, which contains all user created data [25]. In this case, the "userdata" partition was mounted on the block device "/dev/block/mmcblk0p27".

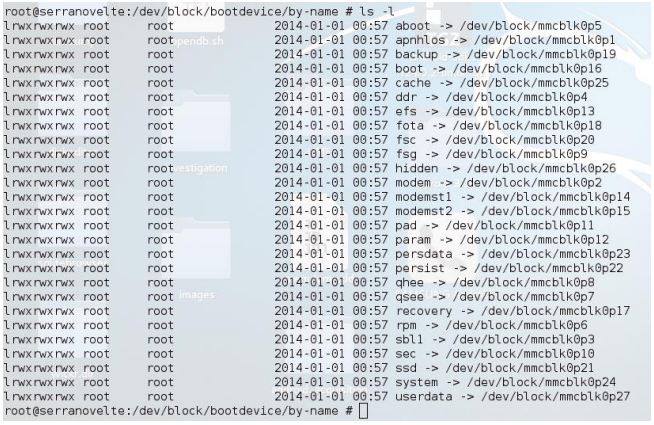

Figure 7. List of partitions mounted on the Android platform

To acquire this partition, the UNIX tool "dd" (data dump) was used to create a bit-for-bit copy on an external SD card, using the command "dd if=/dev/block/mmcblk0p27 of=/mnt/extSdCard/userdata.dd". While performing this type of acquisition it is important to have a correctly formatted SD card, because some filesystems impose upper size limits with the file creation. The appropriate filesystem to use in this case was exFAT, which has no limits on file size. The resulting raw data file came to around 5 GB, which was then transferred to a forensic workstation for analysis using an "adb pull" command. The "userdata.dd" image's filesystem was analysed using the Autopsy forensic suite, which uses Sleuthkit. Upon mounting the "userdata" image in Autopsy and navigating to the "/data/com.mywickr. wickr2" directory, several references to deleted ".wic" files were found in unallocated space, as seen Fig. 8. Using the deleted files' meta addresses (i-nodes), the forensic examiner could see where these files were previously stored, and use this information to recover their contents.

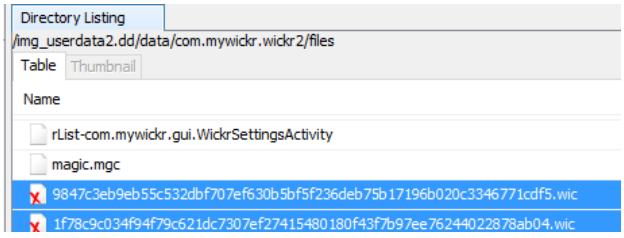

Figure 8. Wickr's "files" directory in Autopsy

VIII. EXPERIMENT 5: WICKR RAM DUMP

As no plaintext artifacts were recovered from the internal storage so far, our next approach was to analyse device's RAM to look for evidences which could be in the un-encrypted form [28]. The analysis of RAM involved the acquisition of data while the application was running and this falls into the category of "live forensics" [29], which refers to any actions taken when the device is in full operation. This has a huge amount of risk involved as an investigator could accidentally remove key bits of data or change data so that it is no longer viable in court. The important aspect with this is to follow ACPO guidelines for handling digital evidence [15]. The second and third principles of the ACPO guidelines state that any competent investigator must be able to explain their actions and keep an audit trail of any actions taken, for the sake of accountability. In the case of this experiment, the Android tool Memory Dump [30] was used while the test platform was turned on. Eventually the Wickr application was running while the acquisition occurred. During this process to preserve authenticity and integrity we followed ACPO guidelines using supporting documentation with minimal change of the original evidence where possible.

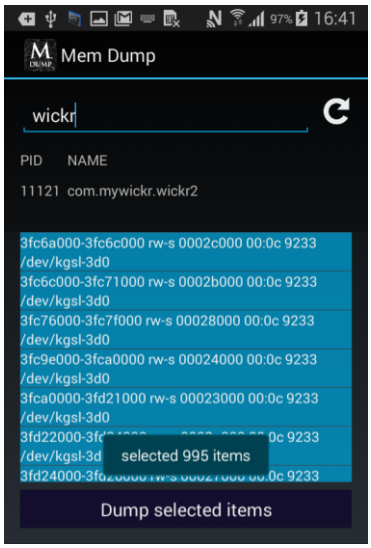

Figure 9. Memory Dump process

Memory Dump, as shown in Fig. 9, is an Android tool that allows the investigator to dump the information used by any specific running process. The resulting "MEMDUMP" directory was transferred to the Linux forensic workstation and analysed with a string search. To do this, the Linux tool "strings" was run on all files in the directory. A simple Linux bash script, as seen in Fig. 10, was created to search the output from a list of keywords. This list includes the username and password used to sign up to Wickr, as well as other accounts that had been used to communicate using the test scenario, names of files transmitted, and excerpts of transmitted messages.

```
#!/bin/bash

for ITEM in $(cat item_list.txt)
do

cat /media/xubuntu/7AEF-B34E/MEMDUMP/memdump_strings.txt | \
grep $ITEM > ~/searchresults/memdump_strings_$ITEM.txt

done
```

Figure 10. Bash Script to search strings

The files in the resulting output directory showed matches with the list of keywords used by the search script. These files were viewed in a text viewer. Although most of the search terms returned no matches, there was a match for the account name used to sign up to Wickr in the "dumped__7428b000-7428e000_rw-p" dump file. This is a pertinent artefact as if acquired could be used, in co-operation with Wickr and telecomms services, to locate the user that signed up to Wickr using a captured device.

IX. CONCLUSIONS

The results of the experiments documented in this paper give insight into the function of Wickr, a highly secure application, as well as the exploration of mobile digital forensics techniques that revolve around third party applications. The results found that Wickr stores data using extensive encryption with the CryptSQL extension for SQLite, and storing received messages in encrypted ".wic" files. Wickr removes its data by removing file headers. The experiments in this paper provided understanding of the manner in which Wickr stores its data by analyzing artefacts recovered from the Wickr application itself, as well as understanding the ephemeral messaging function by analyzing directory structures on the test Android platform's internal storage and the RAM. Interesting lines of research for the future include the recovery of encrypted artifacts, as well as the application of methods used in this paper to analyse similar applications on other platforms, such as iOS and Windows Phone.

REFERENCES

[1] Statista, "Number of apps available in leading app stores as of June 2016," http://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/, [retrieved: August, 2016].

[2] S. Almasy, P. Meilhan, J. Bittermann, "Paris Massacre: At least 128 killed in gunfire and blasts, French officials say," http://edition.cnn.com/2015/11/13/world/paris-shooting/, November 2015 [retrieved: August, 2016].

[3] M. Madi, S. Ryder, J. Macfarlane, A. Beach, and V. Park "As it happened: Charlie Hebdo attack" January 2016 [retrieved: August, 2016].

[4] A. Roussinous, "The social media Accounts of British Jihadis in Syria just got a lot more distressing," http://www.vice.com/en_uk/read/british-jihadis-beheading-prisoners-syria-isis-terrorism, April 2014 [retrieved: August, 2016].

[5] R. Torok, "How social media was key to Islamic State's attacks on Paris," http://theconversation.com/how-social-media-was-key-to-islamic-states-attacks-on-paris-50743, November 2015 [retrieved: August, 2016].

[6] L. Vidino, S. Hughes, "ISIS in America: From retweets to Raqqa," http://www.stratcomcoe.org/download/file/fid/2828, December 2015 [retrieved: August, 2016].

[7] C. Wu, C. Vance, R. Boggs, and T. Fenger, "Forensic Analysis of Data Transience Applications on IOS and Android," http://www.marshall.edu/forensics/files/Wu-Poster.pdf, September 2013 [retrieved: August, 2016].

[8] M. Anderson, "The demographics of device ownership," http://www.pewinternet.org/2015/10/29/the-demographics-of-device-ownership/, October 2015 [retrieved: August, 2016].

[9] M. Wilbourn Partners, "Snapchat is now the third most popular social network among millennials," http://mwpartners.com/snapchat-is-now-the-third-most-popular-social-network-among-millennials/, 2014 [retrieved: August, 2016].

[10] Wickr Official Website, https://www.wickr.com [retrieved: August, 2016].

[11] D. Walnycky, I. Baggili, A. Marrington, J. Moore, and F. Breitinger, "Network and device forensic analysis of Android social-messaging applications," Digital Investigation, vol. 14, pp. 77-84, August 2015.

[12] K. M. Ovens and G. Morrison, "Forensic analysis of Kik Messenger on iOS devices," Digital Investigation, vol. 17, pp. 40-52, 2016.

[13] T. Mehrota and B. M. Mehtre, "Forensic Analysis of Wickr on Android Devices," IEEE International Conference on Computational Intelligence and Computing Research, December 2013.

[14] D. B. Garrie, "Digital Forensic Evidence in the Courtroom: Understanding Content and Quality," Northwestern Journal of Technology and Intellectual Property, vol. 12, pp. 121-128, 2014.

[15] Association of Chief Police Officers, "ACPO Good Practise Guide for Digital Evidence," https://www.cps.gov.uk/legal/assets/uploads/files/ACPO_guidelines_computer_evidence[1].pdf [retrieved: August, 2016].

[16] Samsung Galaxy Mini, http://www.samsung.com/uk/consumer/mobile-devices/smartphones/galaxy-s/GT-I9195ZKABTU, [retrieved: August, 2016].

[17] Allwinner A13 User Manual, http://linux-sunxi.org/A13, [retrieved: August, 2016].

[18] V. Woods and R. V. D. Meulen, "Gartner Says Worldwide Smartphone Sales Grew 3.9 Percent in First Quarter of 2016," http://www.gartner.com/newsroom/id/3323017, Feburary 2016 [retrieved: August, 2016].

[19] Kingo Root Tool, https://www.kingoapp.com, [retrieved: August, 2016].

[20] S. Bommisetty, R. Tamma, and H. Mahalik, "Practical Mobile Forensics," Packt Publishing, 2014.

[21] ADB tool, https://developer.android.com/studio/command-line/adb.html, [retrieved: August, 2016].

[22] Dex2Jar tool, https://github.com/pxb1988/dex2jar, [retrieved: August, 2016].

[23] Java Decompiler tool, http://jd.benow.ca, [retrieved: August, 2016].

[24] SleuthKit tool, http://www.sleuthkit.org, [retrieved: August, 2016].

[25] Wei-Meng Lee, "Beginning Android 4 Application Development," John Wiley & Sons, 2012.

[26] SQLCipher tool, https://www.zetetic.net/sqlcipher/, [retrieved: August, 2016].

[27] S. Brothers, "How Cell Phone "Forensic" Tools Actually Work - Proposed Leveling System," Mobile Forensics World Conference, Chicago, Illinois, 2009.

[28] E. Casey, G. Fellows, M. Geiger, and G. Stellatos, "The growing impact of full disk encryption on digital forensics," Digital Investigation, vol. 8, no. 2, pp. 129-134, 2011.

[29] A. Shortall and M. A. H. B. Azhar, "Forensic acquisitions of WhatsApp data on popular mobile platforms," Sixth International Conference on Emerging Security Technologies (EST). IEEE Press, Technische Universitaet Braunschweig, Germany, pp.13-17, 2015.

[30] Memory Dump tool, https://play.google.com/store/apps/details?id=com.cert.memdump&hl=en, [retrieved: August, 2016].

 ISBN: 978-1-61208-512-8