DOI 10.1007/s11036-016-0791-8

The Role of Mobile Forensics in Terrorism Investigations Involving the Use of Cloud Storage Service and Communication Apps

Niken Dwi Wahyu Cahyani[1,2] · Nurul Hidayah Ab Rahman[1,3] · William Bradley Glisson[4] · Kim-Kwang Raymond Choo[5,1]

Published online: 23 December 2016

Abstract Mobile technologies can be, and have been, exploited in terrorist activities. In this paper, we highlight the importance of mobile forensics in the investigation of such activities. Specifically, using a series of controlled experiments on Android and Windows devices, we demonstrate how mobile forensics techniques can be used to recover evidentiary artefacts from client devices. There are three simulation scenarios, namely: (1) information propagation, (2) information concealment and (3) communications. The experiments used three popular cloud apps (Google Drive, Dropbox, and OneDrive), five communication apps (Messenger, WhatsApp, Telegram, Skype and Viber), and two email apps (GMail and Microsoft Outlook). The evidential data was collected and analysed using mobile forensics and network packet analyser tools. The correlation of evidence artefacts would support to infer illegal use of mobile devices. This study also highlights the extent of acquired evidence between Android and Windows devices, in which Android presents more evidentiary value.

Keywords Android device forensics · Cloud app forensics · Mobile forensics · Terrorist investigations · Windows phone forensics

1 Introduction

Terrorism can be defined as "the use of violence by groups or individuals pursuing political objectives. Terrorists are frequently indiscriminate in their attacks and can deliberately target civilians and non-combatants, often seeking to inflict mass casualties" [1]. While smart mobile devices are increasingly popular with both individuals and businesses, their usage can be criminally exploited to facilitate terrorist activities, including financing of terrorism [2, 3].

A recent example of a mobile forensic challenge in terrorism investigations is the difficulty faced by the Federal Bureau of Investigation (FBI) in acquiring assistance from Apple Inc. to unlock an encrypted iPhone 5C [4]. It was alleged that this phone belonged to one of the key suspects, and the suspect had disabled iCloud backups several weeks prior to the incident. The challenges in this particular incident also demonstrate the potential role of mobile forensics in providing evidential data from mobile devices due to the use of the devices and their apps during terroristic activities, in particular, or other criminal activities in general.

Examining artefacts from mobile cloud services and mobile communication channels, including communication apps and emails, can provide useful information to reconstruct terrorist activities. This information is important for law enforcement in their investigation. Information such as chat logs, multimedia files, contact lists, and geo-tagged data can be used to determine the chain of events and identify terrorists and their associates.

✉ Kim-Kwang Raymond Choo
raymond.choo@fulbrightmail.org

[1] School of Information Technology & Mathematical Sciences, University of South Australia, Adelaide, Australia

[2] Telkom University, Bandung, Indonesia

[3] Universiti Tun Hussein Onn Malaysia, Kuala Lumpur, Johor, Malaysia

[4] School of Computing, University of South Alabama, Mobile, AL, USA

[5] Department of Information Systems and Cyber Security, The University of Texas at San Antonio, San Antonio 78249, TX, USA

Cloud storage apps are regularly used for file synchronisation and sharing activities. They are also commonly used to automatically backup a user's device. For example, Android Backup Service uses Google Accounts (e.g. Google Drive, Google Photos) to back up a user's data. As a result, this approach potentially leaves evidential data on both the cloud user's device and the cloud provider's storage area. Thus, new and/or refinements in mobile device digital evidence collection procedures are required. In this study, we demonstrate how our previously published integrated incident handling and digital forensics model can be used to guide a mobile forensic investigation [5]. The model consists of the following phases:

(i) Preparation and forensic readiness: Getting prepared with strategies and tools.
(ii) Identification: It commences after a suspicious event is detected and reported.
(iii) Assessment, forensic collection and analysis: Initial assessment is conducted to decide the scale of forensic analysis and appropriate response actions.
(iv) Act and monitoring: Involves containment and eradication activities of cybersecurity incidents.
(v) Recovery: Involves restore system disruption to normal and in a secure state.
(vi) Evaluation and forensic presentation: Delivering findings and recommendations.

2 An overview of terrorism activities and mobile forensics

As mobile devices continue to integrate into all aspects of society, it is conceivable that the importance attached to mobile device investigations will continue to escalate. The plausibility of this escalation coupled with increasing legal implications prompts the examination of information and communications technologies (ICT) and computing devices from the perspective of terrorist related activities. It also prompts an inspection of relevant research activities in mobile device forensics.

2.1 Terrorism

It is important to understand the common terrorist-related activities and how the emerging ICT, such as mobile computing infrastructure, affects them. Terrorist-related activities can be broadly classified into (1) information propagation, (2) information concealment, (3) fund raising, and (4) recruitment and training [6–8].

Information propagation concerns with the creation and dissemination of politically – or ideologically – motivated propaganda with the aims of influencing a particular segment of the community, radicalising potential supporters, and inciting "naïve" individuals to conduct terrorist and other criminal activities [7–9]. The dissemination use multimedia objects (e.g. videos, audios), usually via social media services such as social network sites, online forum, online games, video-sharing sites, and file-sharing sites.

Information concealment involves the misuse of (secure) communication platforms to disseminate information to circumvent law enforcement scrutiny and existing surveillance tools [10]. A method that can be used to conceal messages is steganography. A straightforward stenographic method modifies the least significant bit to hide messages within other forms of digital communication by embedding the true message within digital objects, such as text, image or audio [11]. The advancement of mobile device capabilities, coupled with the availability of freeware steganography, makes obscuring the true message from these devices very easy. Message obfuscation makes it difficult to identify and trace illegal communications regarding general activities and financial dealings.

Fundraising refers to the collection of funding to support terrorism and related operations. Source of funding includes donations from supporters, diverting funds raised by legitimate means (charity donations), and proceeds of crime [7, 12, 13]. The collection would come from a number of channels such as donations from supporters, money laundering approach from charitable institutions but diverted for terrorist intentions, and underground activities [8, 14]. Meanwhile, recruitment of terrorist members includes reaching out, communicating, influencing and radicalising like-minded individuals by utilizing ICT (e.g. social networking sites) [6].

The ease in which information is disseminated and hosted on devices coupled with the availability of tools to hide secret messages, plausibly, increases the effectiveness of terrorist fundraising and recruitment activities. Therefore in this paper, we focus on identifying artefacts that can be found in mobile devices after they were used in activities that are related to information propagation and concealment.

2.2 Mobile forensics

As defined by the National Institute of Standards and Technology (NIST), "[m] obile device forensic is the science of recovering digital evidence from a mobile device under forensically sound conditions using accepted methods" [15]. Data acquisition in mobile forensic activities involves physical, logical, and manual methods. Physical acquisition refers to recovering binary representations of the internal memory of mobile devices and dumping them into files, while logical acquisition interacts with a mobile device's operating system to recover the logical objects stored in the file system [16]. Manual acquisition involves viewing the data content stored on a mobile device that requires manual manipulation of the buttons, keyboard or touchscreen and may be recorded using an external digital camera [15].

Existing mobile forensic research can be broadly classified into: (1) examining the capabilities of acquisition methods, (2) undertaking detailed forensic procedures, and (3) conducting in-depth forensic analysis of mobile apps or mobile operating systems.

In examining acquisition methods, Tassone et al. [17] demonstrated that mobile forensic tools have different capabilities in recovering artefacts from different mobile Operating Systems (OS). The authors indicate that the amount of artefacts recovered varies for different OSs and that specific tool support for physical acquisitions of certain phone models is not always present. This is consistent with Glisson et al.'s [18] study, which concluded that there is a considerable variation in recovery results between recovery methods and among toolkits. They acknowledge that this variance can be caused by vendors having different designs, overall software engineering requirements, and practical implementation decisions. The authors go on to highlight the fact that this variance makes it, potentially, difficult to validate artefacts recovered by different toolkits. One study on Windows Phone devices highlights a number of challenges in data acquisition on the three phones with this operating system, includes unrecovered deleted contacts and messages in the physical acquisition process, and impact of reset operation on the acquisition result [19].

The implementation of specific procedures and techniques in digital forensic investigations ensures that evidence can be acquired in a forensically sound manner. Using several cloud storage services such as Amazon S3, Dropbox, Evernote, and Google Docs as case studies, Chung et al. [20] utilised iPhone backup files and rooted Android devices to collect evidence of interest. Based on McKemmish's framework [21], Martini et al. [22] proposed an evidence collection and analysis methodology for Android devices with detailed processes in the collection phase. Ariffin et al. [23] presented an operational technique to recover deleted image files by examining an iOS journaling file system. Leom et al. [24] demonstrated that the forensic collection and analysis of thumbnails in an Android Operating System (OS) would be significant for investigating steganography imagery.

Recent research by Berman et al. [25] and McMillan et al. [26] indicate that the introduction of GPS and mobile device artefact evidence is escalating and impacting court cases. Hence, the legal relevance, from an evidentiary value perspective, is, generally, based on the ability to locate and extract residual data in a forensically sound manner.

The relevance and admissibility of residual data is dependent on an in-depth forensic analysis of extracted artefacts. An analysis of mobile cloud apps by Martini et al. [22] on Android; and Grispos et al.'s [27] analysis on both iOS and Android identified various types of evidence artefacts along with their locations on the devices' file system. Al Mutawa et al.'s [28] research showed different extraction results from social networking apps such as Facebook, Twitter, and MySpace found on Blackberry, Android, and iPhone. The authors observed that no traces of social networking activities could be recovered from Blackberry devices whereas, iPhone and Android phones stored significant amounts of evidentiary data. Farhood et al. [29] examined social network app artefacts left in Android's internal memory and iOS's internal storage that produced evidence of interest which include login, username, password, name, contact information, profile picture, work and education, location, friend list, posts, messages, comments, and IP addresses.

Focusing mainly on the in-depth forensic analysis of the artefacts left by WhatsApp messenger, Anglano [30] demonstrated how to interpret the data stored in the contacts and chat databases in order to reconstruct the list of contacts and the chronology of the messages that have been exchanged by the user. Another study emphasised an in-depth analysis to produce a taxonomy of artefacts. Azfar et al. [31] examined 40 popular Android mHealth apps and proposed forensic taxonomy that comprises databases, user credentials, personal details of users, user activities, user location, activity timestamps, and images.

Sgaras et al. [32] analysed WhatsApp, Viber, Skype, and Tango in Android and iOS that produced target artefacts such as installation data, traffic data, content data, user profile data, user authentication data, contact database, attachment or files exchanged, and location data. Most of the studies were highlighted to simulate common user activities to the particular applications and examine the evidentiary values of these artefacts. Glisson et al. [18] actually acquired devices from secondary markets to mimic situations faced by a forensic investigator when recovering data from an unknown device. Conducting common activities through social networking applications such as logging into apps, modifying personal information, uploading posts, uploading photos, posting comments, sending emails, and chatting promotes a real-world understanding of the artefacts that are generated from these activities [28, 29].

The literature clearly presents the extent to which acquired artefacts depend on acquisition techniques, types of mobile operating systems, and support features of forensic tools. This research indicates that file system architectures require particular techniques that pose challenges in mobile forensic investigations. It also indicates that the validation of extracted artefacts is not a trivial undertaking. Therefore, an in-depth understanding of acquisition techniques, a file system's architecture, forensic tools features, an artefact's taxonomy, and the users' activities that trigger cybersecurity incidents are key points that need to be acknowledged and addressed in effective investigation practices. Understanding these points will aid in the development and re-construction of event scenarios.

3 Experimental environment

The benefits to these event scenarios were investigated through a series of exploratory controlled experiments as defined by Oates [33]. In these experiments, mobile devices were used to act as a sender and a receiver for both Android and Windows platforms. Details of the hardware and software utilized in these experiments are available in Table 1.

It should be noted that the Samsung devices were tested with the Android OS and the Nokia devices were tested with the Windows OS. Addition information pertinent to the experiment is that, for all of the experiments, the Samsung P3100 was always the sender and the Samsung 9300 was always the receiver.

Our experiments simulate three scenarios of common terrorism activities: (1) information propagation activities that use public cloud storage services, (2) information concealment activities that are associated with steganography apps, and (3) communication using available communication apps and emails. Data acquisitions were conducted at the end of each scenario to ensure that all residual data are acquired for a particular scenario.

In the first scenario, Sender (S) prepares files and saves them in a phone for further actions. The files are uploaded to particular cloud storage services. At Receiver (R) side, R runs two activities in accessing the files: (i) read files without download, and (ii) download files to a phone and read. After we conducted data acquisition on S devices, S started to clear his traces by uninstalling cloud storage apps and clearing browsing data; then we conducted data acquisition again on these devices.

For the second scenario, a steganography technique is used to hide secret messages. S prepares image files and they are processed using a steganography tool (i.e. Stegais). S send the image files with and without saving it to phone's internal memory. Then the files are sent using cloud storage service (i.e. OneDrive), email (i.e. Gmail for Android device and Outlook for Windows Phone device) and messaging (i.e. WhatsApp) application. Similar to the information propagation scenario, R access the stego files with and without downloading the files.

These two scenarios, mainly, involve cloud storage apps and a pre-defined dataset containing 246 files which were created to be used in these scenarios. Information pertaining to the file types, file formats and the number of files for each file format is available in Table 2. The dataset used in both of these scenarios is Govdocs1 and is downloadable from the DigitalCorpora server (in http://digitalcorpora.org/corp/files/govdocs1/).

During these scenarios, five files were downloaded for each format and the remaining files were read on the cloud storage. This translates into a total of 50 files being downloaded and 196 files being read on the receiver's side.

The third scenario involves communication apps where common communication activities, such as adding a friend, conducting chat conversations and sharing media content were simulated. Network packets were captured on the receiver side after connecting the receiver's phones to a hotspot.

The high-level mobile device acquisition process implemented in this experiment is illustrated in Fig. 1. The details of an initial inspection and manual acquisition procedures are presented in Figs. 7 and 8, Appendix A. Initial inspection refers to early examination of a device's condition by collecting information such as manufacturer of device, model name and International Mobile Equipment Identification (IMEI) number.

The device's power status was used to determine which acquisition technique to implement. If the device was powered on and functioning, a logical acquisition was conducted using

Table 1 Hardware and software specifications

	Sender (S)	Receiver (R)
Mobile device	Samsung GT-P3100 Galaxy Tab 2 7.0; Nokia Lumia 625	Samsung GT-i9300 Galaxy SIII; Nokia Lumia 735
Device's operating system	Android 4.1.2, kernel version: 3.0.31–1,189,459; Windows Phone 8.1, v8.10.14234.375	Android 4.1.2; kernel version: 3.0.31–1,042,642; Windows Phone 8.1, v8.10.14157.200
XRY mobile forensics	v6.15	v6.15
Stegais	v1.2.2; v1.2.0.0	v1.2.2; v1.2.0.0
Dropbox	v3.0.6.0.2; v1.2.0.0	v3.0.6.0.2; v2015.1125.747.0
Google Drive	v2.3.474.23.24	v2.3.474.23.24
OneDrive	v3.6; v3.6.3.0	v3.6; v4.15.0.0
Messenger	v68.0.0.22.67; v11.01	v63.0.0.10.56; v11.01
WhatsApp	v2.12.510; v2.12.222.0	v2.12.510; v2.12.226
Telegram	v3.8.0; v1.24.8.0	v3.7.0; v1.23.9.0
Skype	v6.31.0.709; v2.32.0.48	v6.25.0.1107; v2.32.0.48
Viber	v6.0.1.13; v4.5.4	v5.8.0.1736; v4.5.4

Table 2 Dataset

Files type	Files format
Document	.pdf (25 files), .xls (25 files), .ppt (25 files), .doc (25 files), .txt (25 files), .ps (25 files)
Audio	.mp3 (25 files)
Image	.gif (25 files),.jpeg (25 files)
Video	.mp4 (21 files)

XRY. The logical acquisition focused on the identification of missed calls and unread messages along with associated date/time stamp information. Once the logical acquisition had been conducted, a physical acquisition was attempted on the device using XRY. If XRY did not support a physical acquisition of the device then a manual acquisition of data on the device was conducted. Manual examinations were conducted through direct interaction with the device's screen. To mitigate data alteration risks in powered on mobile devices, the flight mode was enabled and the GPS receiver on these devices was disabled. This acquisition utilized video recording and/or photography to capture important data which could become digital evidence.

For example, in our experiments, we identified that the Android sender's device was not supported for physical acquisition. It should be noted that the device was rooted before the manual acquisition started. A rooting approach is applied to show that the sender's account information is available and that there is a need to develop a proper method to acquire this information in a forensically sound manner. We are, however, aware that data alteration issues might occur when rooting a device.

If the device was powered off, a physical acquisition was attempted at the beginning. If XRY did not support a physical acquisition of the device, a manual acquisition of the data was conducted. In the real-world, manual acquisitions are optional if results from logical acquisition are limited and/or physical acquisition is not supported. However, for the purposes of this research, combinations of all three acquisition procedures (physical, logical and manual) were applied in this study; where manual acquisition was utilised to complete the result of logical acquisition, to confirm data validity of physical acquisition result.

4 Findings

The experimental finding are presented from the perspectives of information propagation, information concealment and information communication.

4.1 Information propagation

A description of the successful actions of upload, read and download files are presented in Fig. 2. The actions were executed using both mobile client apps and mobile web browsers. Observation for Google Drive on mobile web browsers is discarded as the interface does not display properly in both Android and Windows' Phones. Additionally, a mobile client app for Google Drive is not available in the Microsoft Apps store.

In Android devices, all file types and formats can be uploaded using both mobile client apps and a mobile web browser, while read and download actions present dissimilar results. The mobile client apps for Dropbox, Google Drive, and OneDrive allow R to read files without downloading for all file types and formats. It should be noted that opening files in .docx and .pdf format from OneDrive required the use of a Microsoft Office Mobile app.

Using a mobile web browser for Dropbox, read actions cannot be undertaken without download, but OneDrive interface allows the actions for all file types and formats (without required Microsoft Office Mobile).

In Windows devices, on the other hand, it was identified that only image files were successfully uploaded. This was due to a limitation of the attachment menu to handle other types of files. Read and download actions were, therefore,

Fig. 1 Acquisition procedure

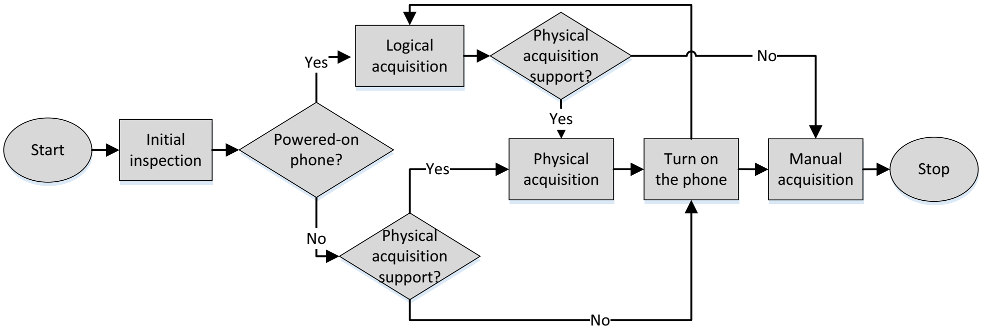
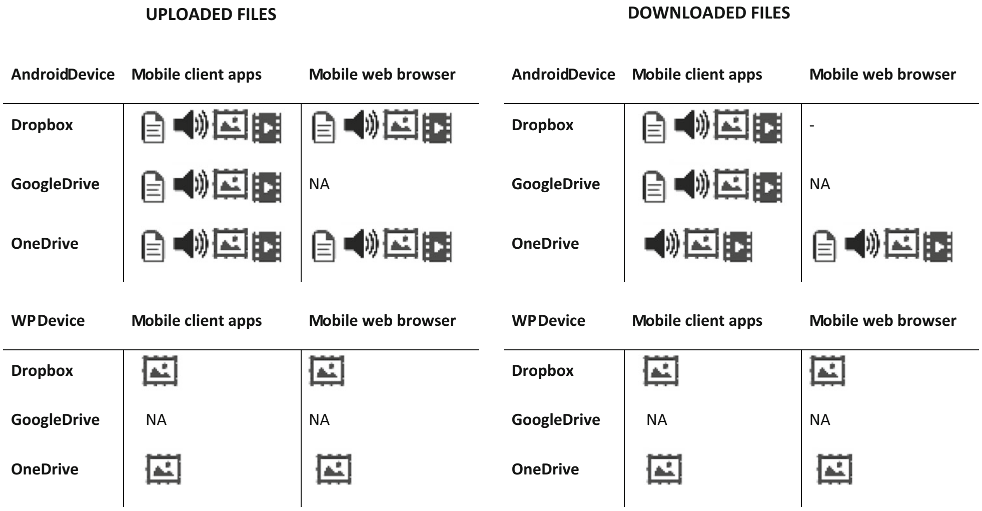


Fig. 2 Upload and download activities



	UPLOADED FILES		DOWNLOADED FILES	
AndroidDevice	Mobile client apps	Mobile web browser	Mobile client apps	Mobile web browser
Dropbox	📄 🔊 🖼 🎞	📄 🔊 🖼 🎞	📄 🔊 🖼 🎞	-
GoogleDrive	📄 🔊 🖼 🎞	NA	📄 🔊 🖼 🎞	NA
OneDrive	📄 🔊 🖼 🎞	📄 🔊 🖼 🎞	🔊 🖼 🎞	📄 🔊 🖼 🎞

	UPLOADED FILES		DOWNLOADED FILES	
WPDevice	Mobile client apps	Mobile web browser	Mobile client apps	Mobile web browser
Dropbox	🖼	🖼	🖼	🖼
GoogleDrive	NA	NA	NA	NA
OneDrive	🖼	🖼	🖼	🖼

undertaken for image files. It was observed that 7 out of 13 image files could be downloaded using a mobile web browser for Dropbox.

4.1.1 Artefacts on android devices

Artefacts on Android devices were collected using a combination of physical, logical and manual acquisition methods. Logical and manual acquisitions were undertaken on the sender's device whereas a physical acquisition was undertaken on the receiver's device. It should be noted that XRY did not support physical acquisition of the sender's device. Therefore, the dataset artefacts on the sender's device were analysed from a logical acquisition perspective and its account information was mainly analysed using manual acquisition techniques after we rooted the device.

Sender and receiver accounts The cloud storage service account ID that connected devices is key information for further investigations. Accounts.db is a local SQLite database that contains account ID metadata for its associated component apps and encrypted passwords (see Table 3). We noted that the only app that does not keep users' passwords in the table is Dropbox.

Table 3 Account artefacts in Android devices

Name – Sender	Type	Password
ia…a02@gmail.com	Dropbox	-
ia…a02@gmail.com	Google Drive	oauth2rt_1/…
ia…a02@gmail.com	OneDrive	MCTlvExq…
Name – Receiver	Type	Password
vic…g@gmail.com	Dropbox	-
vic…g@gmail.com	Google Drive	oauth2rt_1/…
vic…g@gmail.com	OneDrive	MCX5!538…

Each of the cloud storage services has their own SQLite databases to preserve their account information. The information for Dropbox is stored in prefs.db; GoogleDrive information can be found from doclist.db, in table account168 (on the sender's device) and account164 (on the receiver's device); and OneDrive keeps the information in a metadata.db that can be located from table item.

Cloud storage activities One database that gives general information about programs' executions is launcher.db. It also gives information about the execution of the Dropbox, GoogleDrive and OneDrive. This information can be used to confirm that the apps have been executed on the device. Another important database in Android is /data/com.android.providers.media/databases/external.db. It consists of three relevant files which are (1) files – it presents all folders on the internal and external memory; (2) image thumbnails – it presents thumbnails of images; and (3) video thumbnails – it presents thumbnails of videos.

In detail, each of the apps store their activities' log in their own records. Using these records, we can compare the metadata that is generated by uploading, sharing, reading and downloading files between the sender's device and the receiver's device. This comparison provides insight into overall relevance.

For the Dropbox app, an upload_log table in db.db provides key information in reference to the sender's upload activity. This record contains a log of operations, relevant timestamps, the local file path, the file size and the upload status. To map the relevance between activities on the sender and receiver sides, Table 4 shows the examples of uploaded and downloaded files for the .mp3 file dataset. There is no information indicating the location of the uploaded file on the sender side. Conversely, on the receiver side, the file will be stored using the specified path in the field _data once the receiver execute download action and the timestamp is

Table 4 Dropbox's log examples

	Data	Modified	Display_name	Local_modified	Size (MB)
S		Sat, 26 Mar 2016 11:05:05 + 0000	Track 5.mp3		2.6
R	/storage/sdcard0/Android/data/com.dropbox.android/…/Track 5.mp3	Sat, 26 Mar 2016 15:33:12 + 0000	Track 5.mp3	145906942…0	2.6

denoted in the modified field. Additional storage information of interest includes: the file name _display_name, the path is the folder path on the Dropbox, local_modified contains the time of the last access to the file and the field local_modified holds timestamp for actions conducted on the files once the files are in the cache folder.

Although both GoogleDrive and One Drive do not have a specific upload log table, evidentiary metadata can be located from key database tables. In GoogleDrive, for instance, the Entry149 table contains file owner metadata that can be identified from the field labelled 'owner' while timestamps of upload, share, and recent access activities are represented in the creationTime, sharedWithMeTime and lastOpenedTime fields, respectively. This information is displayed in Table 5.

Table 6 presents an example of the metadata generated by OneDrive when files are uploaded and downloaded from the table item. Metadata that link to the sender is ownerCid that comprises 16-digit of hexadecimal and ownerName denotes the registered name for the account. Creationdate and dateShared denotes the upload timestamp and shared timestamp respectively.

Other tables of interest on sender's device include the permission_scopes (see Table 7) and permission_entity (see Table 8) tables. These tables contain receiver metadata that is relevant to specific file transactions. In the permission scopes table, the PermissionScopeResourceName refers to the file name while permissiononEntityName denotes the file's receiver user ID. This data indicates that a particular file was shared with particular user.

We also noted that the cloud storage apps store the artefacts of cached files in a specific cache folder path. Cache files refer to files that have been accessed without being downloaded. In Dropbox, there are two important cache folder paths:

- /storage/sdcard0/Android/data/com.dropbox.android/files/scratch/, and
- /data/com.dropbox.android/files/log.txt.

The last path provides the location of the log file that keeps track of synchronised files.

In Google Drive, a log file located at the path /USERDATA/data/com.google.android.apps.docs/cache/ documents files that have been accessed and read. For OneDrive, cache folders are located at /USERDATA/data/com.microsoft.skydrive/no_backup/stream_cache/ia…a02@gmail.com/…/streams/. However, all OneDrive files and related formats can only be viewed by using a mobile web browser. In addition to the web browser restriction, associated OneDrive residual data was located using the path: /USERDATA/data/com.android.vending /databases.

Clearing traces Clearing trace activities by the sender was simulated by uninstalling cloud storage apps and clearing browsing data. Data acquisition procedures were repeated and the same sources of evidence were observed. The system's collection of cloud app databases are still in the phone's memory but are generally lacking data in particular tables.

However, accounts.db retained the username IDs for both Dropbox and Google Drive accounts along with the encrypted password needed to access the Google Drive (see Table 9).

Event reconstruction An example of event reconstruction, for the information propagation activity, using Android devices is illustrated in Fig. 3. We present the upload details of an executable file from the sender's side and report all artefacts on the receiver's side.

4.1.2 Artefacts on windows phone devices

Sender and receiver accounts It should be noted that XRY version 6.15 did not support a physical acquisition of a Windows Phone when this study was conducted. Logical acquisitions obtained general information from sender and receiver devices such as device name, device manufacturer and model name.

Table 5 An example of GoogleDrive logs

	Owner	CreationTime	LastModifiedTime	LastOpenedTime	SharedWithMeTime
S	vic…g@gmail.com	145896895…1	145899312…0	145897775…8	
R	vic…g@gmail.com	145896895…1	145899312…0	145917068…6	145899312…2

Table 6 OneDrive’s log examples

	OwnerCid	CreationDate	DateShared	OwnerName
S	41…66	145897…3	145899…5	Vic…g
R	41…66	145897…3	145899…7	Vic…g

Cloud storage activities Logical acquisitions are conducted on both sender and receiver devices to collect the artefacts of upload/share – read/download activities. Media files such as documents, pictures, audios, videos, and archive files that were intact in the phone’s internal memory and/or located on the memory card were the primary artefacts that were acquired.

Artefacts from cloud storage accounts, cache files, and xml documents (related to installation and app usage) were not found. Manual acquisitions were conducted in an attempt to identify any artefacts of interest in reference to cloud storage app activities. Both of the sender and receiver phones were not protected with screen password, thus a complete list of installed apps, including cloud storage apps, was obtained. Examinations of archive folders (e.g. downloads) is recommended as the Windows Phone operating system allows users to side load an application. The archive folder is used to store .xap files of side load apps that might be useful in identifying current or attempted installations of apps. A file’s metadata such as name, type, size, created time and hash value were examined to reconstruct information propagation activities. Comparison of file metadata from senders and receivers suggested that cloud storage app integrity, for both uploaded and downloaded files, is maintained (i.e. same file names and hash values).

In these specific experiments, when a receiver only viewed uploaded picture files, no residual artefact of the viewed pictures was identified. This indicates that users who only view an uploaded picture, without downloading it first, may be more difficult to track in terms of viewing activities.

Clearing traces We noted that there is no difference in logical acquisition results between, before and after uninstallation of cloud storage apps. Clearing browsing data activities did not appear to impact the extraction results.

Event reconstruction Event reconstruction for information propagation activities on Windows Phone devices is shown in Fig. 4. From our findings, only image files were successfully uploaded and no artefacts were found, even if the user viewed the files.

4.2 Information concealment

4.2.1 Artefacts on android devices

OneDrive and Gmail accounts were used to illustrate sending and receiving activities along with facilitating communication.

Hide/send – receive/unhide activities Installation artefacts from the Stegais apps were collected from the path that was created by the Android operating system: /USERDATA/data/com.romancinkais. Stegais/files/. Generated steganography images can be located in the following path: /storage/sdcard0/stegais/.

To illustrate sending and receiving activities, we used OneDrive to represent cloud storage services, Gmail for email services and WhatsApp apps to facilitate communication. Artefacts of shared steganography images on OneDrive are identified in the metadata.db in the table labelled item. Evidence of interest from WhatsApp was located in the chat_list table in message.db. No artefacts are found for activities using Gmail. Moreover, no artefact was found if the sender did not store the steganography image to the device’s internal memory before sending it.

We found downloading traces from OneDrive and Gmail in the /storage/sdcard0/Download path on the receiver’s device. WhatsApp generated its own folder in the path /storage/sdcard0/WhatsApp/Media /WhatsAppImages/ in order to store all downloaded image files.

Files integrity File integrity is maintained during transmission activities via OneDrive and Gmail services as evidenced by the same file name and hash values. WhatsApp modified the file name and may apply compression to its transmitted data. We acknowledged that WhatsApp, most likely, does not compress small size file (e.g. 461,136 bytes) but probably does compress larger files (e.g. 2,926,316 bytes). As a result, it is probably that hidden messages, in stego images that use WhatApp, cannot be revealed due to the compression process.

Event reconstruction An example of an event reconstruction for information concealment activities on Android devices is shown in Fig. 5. We confirmed that OneDrive and Gmail did not change the integrity of the sent files.

Table 7 Permission_scopes table

PermissionScopeResourceId	PermissionScopeResourceName	PermissionScopeEntityCount
41FC970250EAAA66!211	Android extension dataset	1

Table 8 Permission_entity table

PermissionEntityName	PermissionEntityCanUsrChg	PermissionEntityEmail	PermissionEntityId
ia…a02@gmail.com	1	ia…a02@gmail.com	3E201203B2CF34B6

4.2.2 Artefacts on windows phone devices

Logical acquisition results did not provide sufficient data to identify sender and receiver accounts for cloud storage services (i.e OneDrive), email (i.e. Outlook) and WhatsApp.

Hide/send – receive/unhide activities Original images can be created by taking pictures using the phone's camera and saving them in the Camera Roll folder in the phone's memory. As expected, logical acquisitions successfully extracted images from the Camera Roll folder.

The use of the Stegais app was identified from artefacts in documents and unrecognised files that were extracted from the sender's phones. README_FIRST.txt is an example of a Stegais app installation artefact that was extracted from Lumia c625/SDard/WPSystem/Apps/{D414A421-403A-4FC C-9069-7583604390BD}/Install, where {D414A4 21-403A-4FCC-9069-7583604390BD}is a code for Stegais app in the Windows Store. Another indication that Stegais was installed on the device was found in a collection of unrecognized files; Steganography.ni.exe was extracted from: Lumia 625/SDcard/WPSystem/AppRepository/29636 DharmendraMauryaRajp.Steganography_1.0.0.0_neutral__d0xnxt1pzcw50/NI. Meanwhile the use of WhatsApp was identified from the existence of messages.db that was saved in Lumia 625/SD card/WhatsApp/WinPhoneBackup/2015-12-09-0000.

Although the example files showed that a sender concealed information and delivered it, there is no artefact recovered from the hidden information that indicates the receivers' identity. Furthermore, the logical acquisition could only extract steganography images if the sender saves the images before sending them to the receiver. Manual acquisition was undertaken to identify the Stegais and WhatsApp installations on the receiver's device; however, from logical acquisition results, neither documents nor unrecognized files were extracted to identify the installation.

File integrity We observed that cloud storage and email services do not modify the content of a sent steganography file. Uploaded and downloaded files have an identical hash value, but they have different file names. WhatsApp did compress the original files to enable light communication between their users. The compression made hidden messages unreadable.

Table 9 Clearing traces artefacts

Name	Type	Password
ia…a02@gmail.com	com.dropbox.android.account	-
ia…a02@gmail.com	com.google	oauth2rt_1…

Event reconstruction Figure 6 presents an event reconstruction of information concealment activity on Windows Phone devices. The findings are similar to the findings generated in the Android scenario in the context of files integrity.

4.3 Communication

Five mobile communication apps (i.e. Viber, Skype, WhatsApp, Telegram and Messenger) were chosen based on their current popularity as communication channels. Evidentiary values of digital objects on the communication apps were examined based on the simulation activities undertaken on Android devices. Those activities included sending and receiving text messages, documents, location, images, and making or accepting voice and video calls. In addition, two email apps (i.e. GMail and Microsoft Outlook) were studied. Sending and receiving emails and attachments activities were simulated.

To examine artefacts of user account, we first checked the accounts.db as it may contain account information (e.g. username and encrypted password) of installed apps. We found users' email, they were identified from Skype and GMail (i.e. vic…g@gmail.com) and Microsoft Outlook (i.e. Vic…g07@yahoo.com.au:Yahoo) apps; and users' encrypted password for Viber and GMail, 25b1…47 and oauth2rt_1/L-… I, respectively. We also found launcher.db that comprises installed apps, component name (e.g. org.telegram.messenger) provide clue that the apps have been launched.

Similar to cloud storage apps, initial survey on the accounts.db lead to additional metadata. Table 10 presents a snapshot of potential evidence sources for communication apps.

To identify contacts, Skype uses the registered email or Skype name, Telegram and Messenger assign a unique user ID in integer format, while WhatsApp and Viber use the registered phone number. In terrorism investigations, for example, this metadata could lead to other terrorist actors, new members to be recruited, and/or probable targets.

Chat log artefacts present key information that may provide insight into terrorism perspectives, including target, motivation, attack tools, domain, method of action and perceived impacts. Accessing chat log tables allows investigators to identify metadata from communication activities, including actors (sender and recipient), message body, call logs from VoIP voice and video, logs of attachments (type, file location, size) and timestamp. Telegram, however, keeps chat log

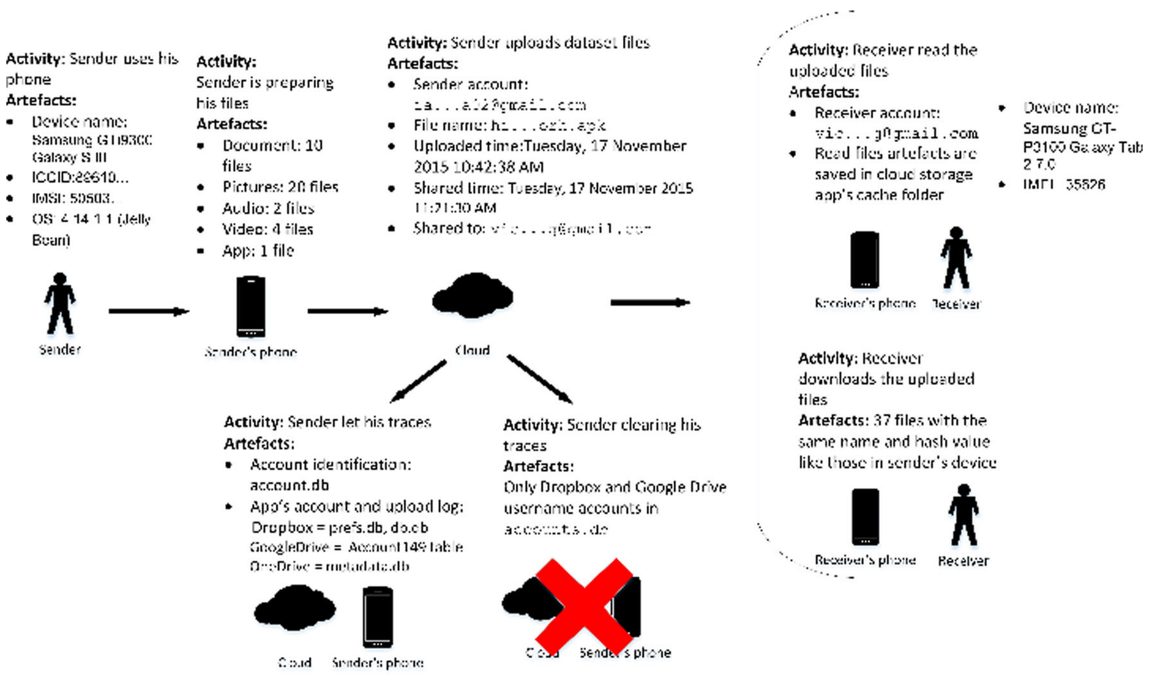

Activity: Sender uses his phone
Artefacts:
Device name: Samsung GT-I9300 Galaxy S III
ICCID:89610...
IMSI: 50503...
OS: 4.14.1.1 (Jelly Bean)
Sender
Activity: Sender is preparing his files
Artefacts:
Document: 10 files
Pictures: 20 files
Audio: 2 files
Video: 4 files
App: 1 file
Sender's phone
Activity: Sender uploads dataset files
Artefacts:
Sender account: ta...a12@gmail.com
File name: h2...ozk.apk
Uploaded time:Tuesday, 17 November 2015 10:42:38 AM
Shared time: Tuesday, 17 November 2015 11:21:30 AM
Shared to: vic...q@gmail.com
Cloud
Activity: Receiver read the uploaded files
Artefacts:
Receiver account: vic...g@gmail.com
Read files artefacts are saved in cloud storage app's cache folder
Device name: Samsung GT-P3100 Galaxy Tab 2 7.0
IMEI: 35826
Receiver's phone
Receiver
Activity: Receiver downloads the uploaded files
Artefacts: 37 files with the same name and hash value like those in sender's device
Receiver's phone
Receiver
Activity: Sender let his traces
Artefacts:
Account identification: account.db
App's account and upload log:
Dropbox = prefs.db, db.db
GoogleDrive = Account149 table
OneDrive = metadata.db
Cloud
Sender's phone
Activity: Sender clearing his traces
Artefacts:
Only Dropbox and Google Drive username accounts in accounts.db
Cloud
Sender's phone

Fig. 3 Event reconstruction for information propagation activities on Android devices

metadata in an encrypted format and BLOB objects format for attached media files. Examples of available artefacts are user ID, timestamp, and chat states (sent or received).

Group chat metadata was observed in WhatsApp and Telegram. Group chat is normally created for particular themes (e.g. organising events) or members (e.g. family, friends), thus the logs would give a greater overview of suspects' motivation and for tracing connections between participants. WhatsApp provides detail information on group members including their phone number. Table of group_participants presents a list of group members [phone number – group id@g.us]where group_participants_history denotes the latest activity (in integer value) of each participant in a particular group. A chat log of a group chat was found in the messages table by observing key_remote_jid, for example as 614…48–1,459,248,221@g.us. The acquired metadata for Telegram includes group id and group name. There did not appear to be a group administrator for Telegram.

The four communication apps, apart from Telegram, record logs of the media attachments during chat conversations in the chat log tables along with the media format. WhatsApp, Telegram and Viber create file paths in the internal memory

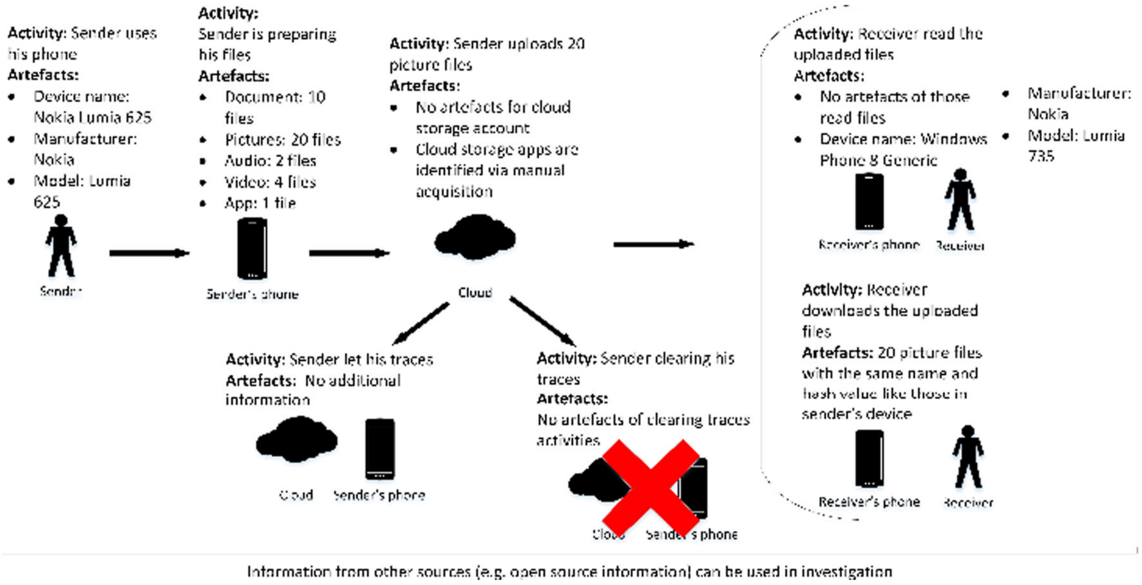

Activity: Sender uses his phone
Artefacts:
Device name: Nokia Lumia 625
Manufacturer: Nokia
Model: Lumia 625
Sender
Activity: Sender is preparing his files
Artefacts:
Document: 10 files
Pictures: 20 files
Audio: 2 files
Video: 4 files
App: 1 file
Sender's phone
Activity: Sender uploads 20 picture files
Artefacts:
No artefacts for cloud storage account
Cloud storage apps are identified via manual acquisition
Cloud
Activity: Receiver read the uploaded files
Artefacts:
No artefacts of those read files
Device name: Windows Phone 8 Generic
Manufacturer: Nokia
Model: Lumia 735
Receiver's phone
Receiver
Activity: Receiver downloads the uploaded files
Artefacts: 20 picture files with the same name and hash value like those in sender's device
Receiver's phone
Receiver
Activity: Sender let his traces
Artefacts: No additional information
Cloud
Sender's phone
Activity: Sender clearing his traces
Artefacts:
No artefacts of clearing traces activities
Cloud
Sender's phone
Information from other sources (e.g. open source information) can be used in investigation

Fig. 4 Event reconstruction for information propagation activities on Windows Phone devices

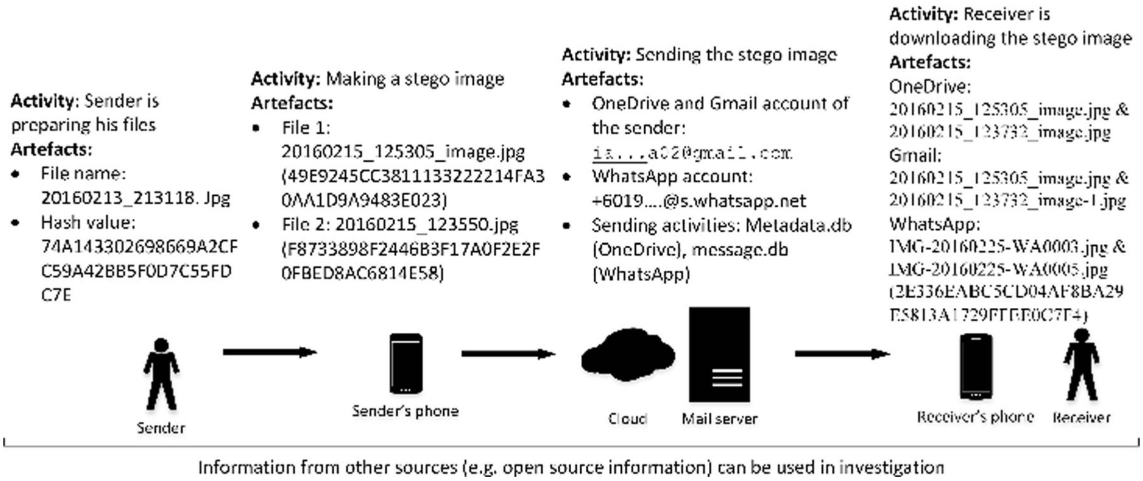

Activity: Sender is preparing his files
Artefacts:
• File name: 20160213_213118. Jpg
• Hash value: 74A143302698669A2CFC59A42BB5F0D7C55FDC7E
Activity: Making a stego image
Artefacts:
• File 1: 20160215_125305_image.jpg (49E9245CC3811133222214FA30AA1D9A9483E023)
• File 2: 20160215_123550.jpg (F8733898F2446B3F17A0F2E2F0FBED8AC6814E58)
Activity: Sending the stego image
Artefacts:
• OneDrive and Gmail account of the sender: ia....a02@gmail.com
• WhatsApp account: +6019....@s.whatsapp.net
• Sending activities: Metadata.db (OneDrive), message.db (WhatsApp)
Activity: Receiver is downloading the stego image
Artefacts:
OneDrive: 20160215_125305_image.jpg & 20160215_123732_image.jpg
Gmail: 20160215_125305_image.jpg & 20160215_123732_image-1.jpg
WhatsApp: IMG-20160225-WA0003.jpg & IMG-20160225-WA0005.jpg (2E336EABC5CD04AF8BA29E5813A1729FFEE0C7F4)
Sender
Sender's phone
Cloud
Mail server
Receiver's phone
Receiver
Information from other sources (e.g. open source information) can be used in investigation

Fig. 5 Event reconstruction for information concealment activities on Android devices

directory to store the received and sent media. Messenger and Skype keep the shared media in their cache, but are limited to only a few media types such as images and audio messages for Messenger, and images and video for Skype. Traces of audio messages were found in Messenger and WhatsApp, while video messages were found on WhatsApp, Skype and Viber.

Besides shared media, the files of profile pictures were observed for WhatsApp, Viber, and Telegram. The naming convention for WhatsApp profile pictures is [phone number]@s.whatsapp.net.j. Telegram and Viber keep a history of timestamped profile pictures for users and their contacts; not just the most recent ones.

GPS data may play an important role in providing information such as location of target, suspects' whereabouts during communication, and any shared location data during conversations. In communication apps, location data is recorded, if a user turns on the GPS menu and opts to share location or send their location data as an attachment during chat conversation. Whatsapp, Skype and Viber have specific columns for latitude and longitude data in their main chat logs tables, while data on Messenger can be observed from the embedded URL as attachments, for instance [{"name":"Iaxx's Location", "caption": null, "... markers = −34.81076900%2C138.62047000&language = en.... }].

Network analysis Mobile forensics alone may not provide adequate digital evidence, although other forensic techniques, such as network forensics, could address this inadequacy by correlating evidence from different sources, for example network logs [34]. Let us consider on 29/03/2016 at 10:26:31 PM, a user shared one image file, file:///storage/sdcard0/viber/media/Viber%20Images/image-5…4aa-V.jpg. Table 11 presents an example of DNS responses between a client and a server that provide clues to file sharing activities using Viber on Android.

Since minimal significant artefacts were found on Windows devices, network-based evidences can, potentially,

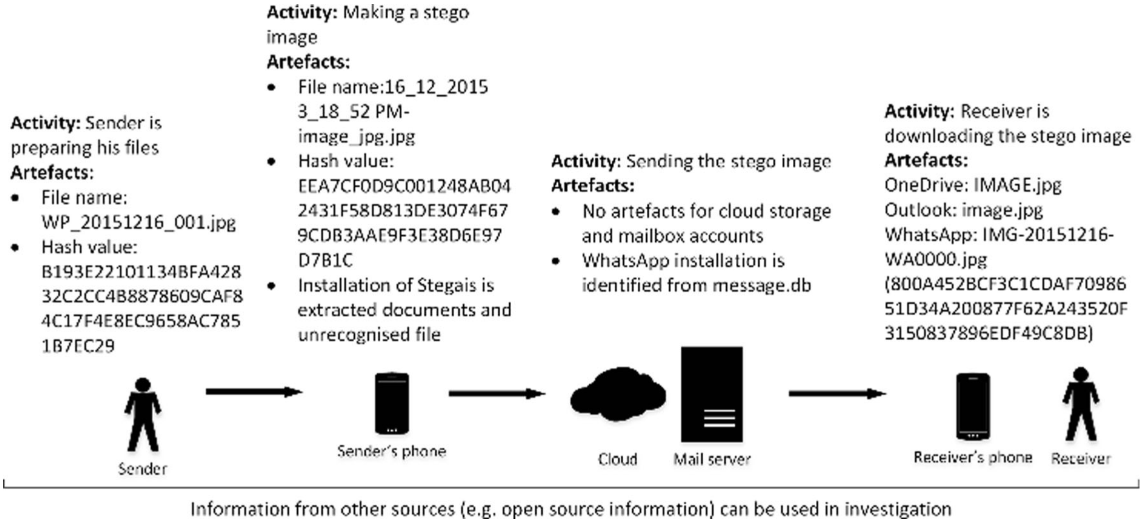

Activity: Sender is preparing his files
Artefacts:
• File name: WP_20151216_001.jpg
• Hash value: B193E22101134BFA42832C2CC4B8878609CAF84C17F4E8EC9658AC7851B7EC29
Activity: Making a stego image
Artefacts:
• File name:16_12_2015 3_18_52 PM-image_jpg.jpg
• Hash value: EEA7CF0D9C001248AB042431F58D813DE3074F679CDB3AAE9F3E38D6E97D7B1C
• Installation of Stegais is extracted documents and unrecognised file
Activity: Sending the stego image
Artefacts:
• No artefacts for cloud storage and mailbox accounts
• WhatsApp installation is identified from message.db
Activity: Receiver is downloading the stego image
Artefacts:
OneDrive: IMAGE.jpg
Outlook: image.jpg
WhatsApp: IMG-20151216-WA0000.jpg (800A452BCF3C1CDAF7098651D34A200877F62A243520F3150837896EDF49C8DB)
Sender
Sender's phone
Cloud
Mail server
Receiver's phone
Receiver
Information from other sources (e.g. open source information) can be used in investigation

Fig. 6 Event reconstruction for information concealment activities on Windows Phone devices

Table 10 A summary of potential evidence sources in communication apps

Apps	Artefacts
Messenger	Database path: com.facebook.orca/databases Account: prefs_db.db (database) and preferences (table) Contact: contacts_db2 (database) and contacts (table) Chat log: •chat history: threads_db2.db (database) and messages (table) •video and voice calls: call_log.sqlite (database) and person_summary (table) Shared media path: /data/data/com.facebook.orca/cache/fb_temp (for images and audio) Location: threads_db2.db (database), messages (table)
WhatsApp	Database path: com. Whatsapp/databases Account: axolotl.db (database) and identities (table), com.whatsapp/shared_prefs/RegisterPhone.xml and com.whatsapp/shared_prefs/com.whatsapp_preferences (XML files) Contact: wa.db (database) and wa_contacts (table) Chat log: msgstore.db (database), messages and chat_list (tables) Shared media paths: •/data/com.whatsapp/files/Avatars (profile picture) •/media/WhatsApp/Media/ (for audio, voice notes, video and documents) Location: msgstore.db (database), messages (table)
Telegram	Database path: org.telegram.messenger/files Account and Contact: cache4.db (database) and users (table) Chat log: cache4.db (database), messages (table), enc_chats and chats (tables, encrypted) Shared media paths: /USERDATA/media/Android/data/org.telegram.messenger/cache (for profiles pictures and images) •/storage/sdcard0/Telegram/Media/ (for images, video, audio and documents)
Skype	•Database path: com.skype.raider/files/live#3 < username> Account: main.db (database) and accounts (table), qik_main.db (database) and settings (table) Contact: eascache.db (database) and fullobjects (table), main.db (database) and contacts (table) Chat log: main.db (database) Contact list: conversations (table) Chats history: messages (table) •Video calls: videos (table) •Voice calls: calls (table) Shared media path: com.skype.raider/files/live#3 < username>/media_messaging/media_cache (for images and video) Location: main.db (database), messages (table)
Viber	Database path: com.viber.voip/databases Account: (No data) Contact: viber_data.db (database) and phonebookcontact (table) Chat log: viber_messages.db (database) •Chats history: messages (table) •Latest events: adx (table) Shared media path: /storage/sdcard0/viber/media/ (for profile pictures, images and video) Location: viber_messages (database), messages (table)
GMail	Database path: com.google.android.gm/databases Mailbox: mailstore.iargunisa02@gmail.com.db Attachment: com.google.android.gm/cache/username@gmail.com
Microsoft Outlook	Database path: com.microsoft.office.outlook/databases Account: acompliAcct.db:mailAccounts Mailbox: accompli.db Attachment: com.microsoft.office.outlook/app_1…. and com.microsoft.office.outlook /app_attachment-staging

Table 11 Related network packets for Viber on 29/03/2016

Timestamp (PM)	Client; DNS query	Server; DNS answer
10:26:32	10.xxx.xx.198; share.viber.com	130.xxx.x.2; share.media…viber.com
10:26:32	10.xxx.xx.198; share.media…viber.com	130.xxx.x.2; lb.-share-lbshare-1bm50fb…amazonaws
10:26:34	10.xxx.xx.198; share-b…viber.com	130.xxx.x.2; du3y6….cloudfront.net

facilitate investigations. For instance, both the source and destination of IP addresses observed on network packets would have evidentiary value to trace the suspect (see Fig. 7).

Our findings have shown that network metadata supports the occurrence of mobile communication activities. The limitation, however, is that network monitoring is, normally, a practice that takes place at organisational levels. Hence, it, generally, requires a warrant to undertake this practice on personal devices.

5 Discussion

The increasing adoption of cloud and mobile technology infrastructures by organisations present plausible abuse opportunities for terrorist activities. Cooperative counter-terrorism at organisational and national levels is needed to increase overall understanding and mitigate infrastructure abuse.

In this paper, we demonstrated the intricacies associated with investigating terrorist activities on mobile devices. Our experiments interact with cloud storage services (i.e. Dropbox, Google Drive and OneDrive) and communication apps (Messenger, WhatsApp, Telegram, Skype and Viber, including email client services). The findings show that we can reconstruct more complete activities on Android devices as compared to Windows Phone devices.

On Android devices, we identified an account's username, reconstructed a user's activities and downloaded data. While on a Windows Phone only downloaded data; no database and XML files were extracted using logical acquisition. Similarly, while we can acquire viewed-only data in a cache folder for Android devices, only downloaded data is acquired on Windows Phone. The limited results derived from Windows Phone devices makes it necessary for a manual acquisition highly probable moreover when physical acquisition is not applicable. One study reported that more data was extracted including installed apps, databases and locations on Windows Phone device (i.e. Nokia Lumia 625) using physical acquisition method [19].

Focusing on communication apps, the extent of extracted data depends on the communication apps themselves. In other words, how individual apps store their data impacts extraction success, i.e., either in plaintext or encrypted. Telegram, which implements encryption, provides the fewest clues on users and metadata of events. In the future, similar encrypted data issues might be encountered for newer versions of WhatsApp.

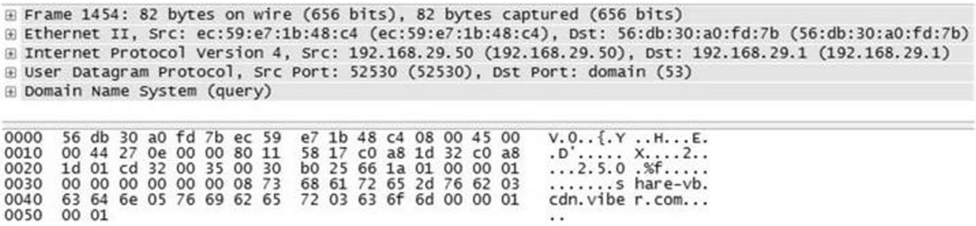
6 Conclusion

This research highlighted the importance of mobile device forensics in investigations involving the use of cloud storage services and communication apps along with the necessity and potential utility of the integrated incident handling and digital forensics models to investigate and reconstruct terrorist incidents [5]. Key findings could inform future similar investigations.

Future research include extending the research to a wider range of mobile apps and app categories, as well as newer versions of mobile devices and operating systems.

Acknowledgements The authors thank the anonymous reviewers for providing constructive and generous feedback. Despite their invaluable assistance, any errors remaining in this paper are solely attributed to the authors. This paper is an extended conference version [35], with more than 50% new content.

Appendix A



```
⊞ Frame 1454: 82 bytes on wire (656 bits), 82 bytes captured (656 bits)
⊞ Ethernet II, Src: ec:59:e7:1b:48:c4 (ec:59:e7:1b:48:c4), Dst: 56:db:30:a0:fd:7b (56:db:30:a0:fd:7b)
⊞ Internet Protocol Version 4, Src: 192.168.29.50 (192.168.29.50), Dst: 192.168.29.1 (192.168.29.1)
⊞ User Datagram Protocol, Src Port: 52530 (52530), Dst Port: domain (53)
⊞ Domain Name System (query)

0000  56 db 30 a0 fd 7b ec 59  e7 1b 48 c4 08 00 45 00   V.0..{.Y ..H...E.
0010  00 44 27 0e 00 00 80 11  58 17 c0 a8 1d 32 c0 a8   .D'..... X....2..
0020  1d 01 cd 32 00 35 00 30  b0 25 66 1a 01 00 00 01   ...2.5.0 .%f.....
0030  00 00 00 00 00 00 08 73  68 61 72 65 2d 76 62 03   .......s hare-vb.
0040  63 64 6e 05 76 69 62 65  72 03 63 6f 6d 00 00 01   cdn.vibe r.com...
0050  00 01                                              ..
```

Fig. 7 Example of a Viber's packet from a Windows device

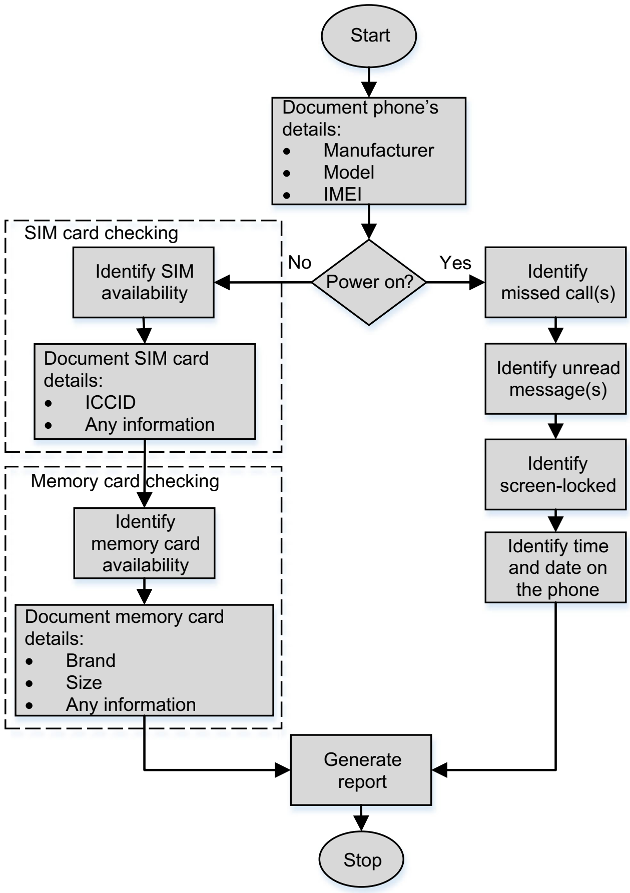


Fig. 8 Initial inspection

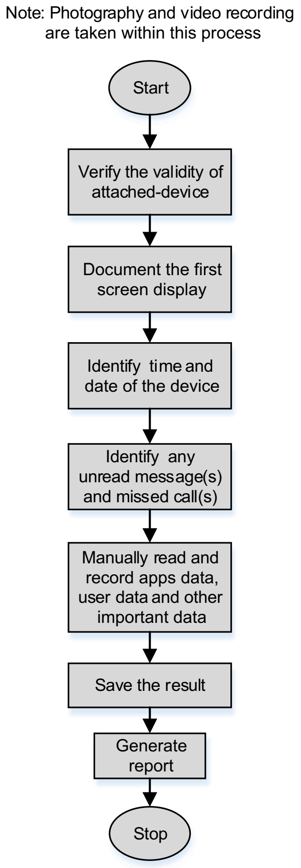

Note: Photography and video recording are taken within this process
Start
Verify the validity of attached-device
Document the first screen display
Identify time and date of the device
Identify any unread message(s) and missed call(s)
Manually read and record apps data, user data and other important data
Save the result
Generate report
Stop

Fig. 9 Manual acquisition

References

1. Australian Government (2010) Securing Australia: Protecting Our Community. https://www.asio.gov.au/img/files/counter-terrorism_white_paper.pdf. Accessed 28 February 2016
2. Choo K-KR (2013) New payment methods: a review of 2010–2012 FATF mutual evaluation reports. Comput Secur 36:12–26
3. Choo K-KR (2014) Designated non-financial businesses and professionals: a review and analysis of recent financial action task force on money laundering mutual evaluation reports. Secur J 27(1):1–26
4. Federal Bureau of Investigation (2016) Statement to Address Misleading Reports that the County Of San Bernardino Reset Terror Suspect's Iphone without Consent of the FBI https://assets.documentcloud.org/documents/2716811/Statement-from-the-FBI-Feb-20-2016.pdf. Accessed 28 Februari 2016
5. Ab Rahman N, Choo K (2015) Integrating digital forensic practices in cloud incident handling: A conceptual cloud incident handling model. In: KO R, CHOO K-KR (eds) Cloud Security Ecosystem. Syngress, an Imprint of Elsevier, Waltham, pp. 383–400
6. Amble JC (2012) Combating terrorism in the new media environment. Stud Conf Terror 35(5):339–353
7. UNODC (2012) The Use of the Internet for Terrorist Purposes. https://www.unodc.org/documents/frontpage/Use_of_Internet_for_Terrorist_Purposes.pdf. Accessed 28 February 2016
8. Ogun MN (2012) Terrorist use of internet: possible suggestions to prevent the usage for terrorist purposes. J Appl Secur Res 7(2):203–217
9. Choo K-KR (2008) Organised crime groups in cyberspace: a typology. Trends in Organized Crime 11(3):270–295
10. Choo K-KR, Smith RG, McCusker R (2007) Future directions in technology-enabled crime: 2007–09. Research and public policy no 78. Australian Institute of Criminology, Canberra
11. Zielińska E, Mazurczyk W, Szczypiorski K Trends in steganography. Commun ACM 57(3):86–95
12. Choo K-KR SR, Walters J, Bricknell S (2013) Perceptions of money laundering and financing of terrorism in the Australian legal profession. Research and public policy no 122(1). Australian Institute of Criminology, Canberra
13. Walters J, Budd C, Smith R, Choo K, McCusker R, Rees D (2012) Anti-money laundering and counter-terrorism financing across the globe: a comparative study of regulatory action. Research and public policy no 113. Australian Institute of Criminology, Canberra
14. Mishra S (2003) Exploitation of information and communication technology by terrorist organisations. Strateg Anal 27(3):439–462
15. Ayers R, Brothers S, Jansen W (2014) Guidelines on mobile device forensics. NIST Special Publication 800 (101 Revision 1)
16. Grispos G, Storer T, Glisson WB (2011) A comparison of forensic evidence recovery techniques for a windows mobile smart phone. Digit Investig 8(1):23–36
17. Tassone C, Martini B, Choo K-KR, Slay J (2013) Mobile device forensics: a snapshot. Trends issues crime Crim. Justice no. 460: 1–7. Australian Institute of Criminology, Canberra
18. Glisson WB, Storer T, Buchanan-Wollaston J (2013) An empirical comparison of data recovered from mobile forensic toolkits. Digit Investig 10(1):44–55
19. Cahyani NDW, Martini B, Choo KKR, Al-Azhar A (2016) Forensic data acquisition from cloud-of-things devices: Windows smartphones as a case study. Concurrency and Computation: Practice and Experience
20. Chung H, Park J, Lee S, Kang C (2012) Digital forensic investigation of cloud storage services. Digit Investig 9(2):81–95
21. McKemmish R (1999) What is forensic computing? Trends issues crime Crim. Justice no. 118:1–6. Australian Institute of Criminology, Canberra
22. Martini B, Do Q, Choo K-KR (2015) Mobile cloud forensics: An analysis of seven popular Android apps. In: KO R, CHOO K-KR (eds) Cloud Security Ecosystem. Syngress, an Imprint of Elsevier, Waltham, pp. 309–345
23. Ariffin A, D'Oorazio C, Choo K-KR, Slay J (2013) iOS Forensics: How can we recover deleted image files with timestamp in a forensically sound manner? In: Proceedings of the 8th International Conference on Availability, Reliability and Security, Regensburg, Germany, Sept 2–6, 2013 (IEEE), 375–382
24. Leom MD, DOrazio CJ, Deegan G, Choo K-KR (2015) Forensic Collection and Analysis of Thumbnails in Android. In: Proceedings of the 14th IEEE International Conference on Trust, Security and Privacy in Computing and Communication, Helsinki, Finland, Aug 20–22, (IEEE) 1059–1066
25. Berman KJ, Glisson WB, Glisson LM (2015) Investigating the Impact of Global Positioning System Evidence. In: Hawaii International Conference on System Sciences, Hawaii, Jan 5–8, 2015 (IEEE), 5234–5243
26. McMillan JER, Glisson WB, Bromby M (2013) Investigating the increase in mobile phone evidence in criminal activities. In: Hawaii International Conference on System Sciences, Wailea, Hawaii, Jan 7–10, 2013 (IEEE), 4900–4909
27. Grispos G, Glisson WB, Storer T (2015) Recovering residual forensic data from smartphone interactions with cloud storage providers. In: KO R, CHOO K-KR (eds) Cloud Security Ecosystem. Syngress, an Imprint of Elsevier, Waltham
28. Al Mutawa N, Baggili I, Marrington A (2012) Forensic analysis of social networking applications on mobile devices. Digit Investig 9: S24–S33
29. Farhood ND, Dehghantanha A, Eterovic-Soric B, Choo K-KR (2015) Investigating social networking applications on smartphones detecting Facebook, twitter, LinkedIn and Google + artefacts on android and iOS platforms. Aust J Forensic Sci:1–20
30. Anglano C (2014) Forensic analysis of WhatsApp messenger on android smartphones. Digit Investig 11(3):201–213
31. Azfar A, Choo K-KR, Liu L (2015) Forensic Taxonomy of Popular Android mHealth Apps. In: Proceedings of the 21st Americas Conference on Information Systems
32. Sgaras C, Kechadi M-T, Le-Khac N-A (2015) Forensics Acquisition and Analysis of Instant Messaging and VoIP Applications. In: Garain U, Shafait F (eds) Computational Forensics. Springer, Switzerland, pp. 188–199
33. Oates BJ (2005) Researching information systems and computing. Sage Publications, London, p. 341
34. Ab Rahman NH, Cahyani NDW, Choo KKR (2016) Cloud incident handling and forensic-by-design: cloud storage as a case study. Concurrency and Computation: Practice and Experience
35. Cahyani NDW, Ab Rahman NH, Xu Z, Glisson WB, Choo KKR (2016) The role of mobile forensics in terrorism investigations involving the use of cloud apps. In: Proceedings of the 9th International Conference on Mobile Multimedia Communications