JOURNAL OF FORENSIC SCIENCES

J Forensic Sci, September 2016, Vol. 61, No. 5
doi: 10.1111/1556-4029.13164
Available online at: onlinelibrary.wiley.com

TECHNICAL NOTE

DIGITAL & MULTIMEDIA SCIENCES

Abdullah Azfar,[1] *M.S.; Kim-Kwang Raymond Choo*,[2,1] *Ph.D.; and Lin Liu*,[3] *Ph.D.*

An Android Communication App Forensic Taxonomy

ABSTRACT: Due to the popularity of Android devices and applications (apps), Android forensics is one of the most studied topics within mobile forensics. Communication apps, such as instant messaging and Voice over IP (VoIP), are one popular app category used by mobile device users, including criminals. Therefore, a taxonomy outlining artifacts of forensic interest involving the use of Android communication apps will facilitate the timely collection and analysis of evidentiary materials from such apps. In this paper, 30 popular Android communication apps were examined, where a logical extraction of the Android phone images was collected using XRY, a widely used mobile forensic tool. Various information of forensic interest, such as contact lists and chronology of messages, was recovered. Based on the findings, a two-dimensional taxonomy of the forensic artifacts of the communication apps is proposed, with the app categories in one dimension and the classes of artifacts in the other dimension. Finally, the artifacts identified in the study of the 30 communication apps are summarized using the taxonomy. It is expected that the proposed taxonomy and the forensic findings in this paper will assist forensic investigations involving Android communication apps.

KEYWORDS: forensic science, digital forensics, Android forensics, communication app taxonomy, line app, mobile app, Viber app, WeChat app

The increasing popularity of smart mobile devices has resulted in a surge in the number of mobile application (app) users for both personal and business purposes (1). Mobile apps have become a key technological revolution in our daily life, from real-time communications with friends and relatives in different countries to finding the location of a restaurant, to social activities such as games and online dating. Communication apps such as Instant Messages (IMs) and Voice over IP (VoIP) are one popular app category used by the majority of mobile device users (2,3).

During the investigations of crimes or incidents involving mobile devices such as those involving the use of popular communication apps (4,5), there is usually some accumulation or retention of data on a mobile device that will need to be identified, preserved, analyzed, and presented in a court of law. Potential evidential data that could be recovered from a suspect's mobile device include login credentials for the communication service, messages, photographs, and videos exchanged between the suspect and his/her connections and the associated timestamps. Timely recovery and analysis of these artifacts is critical in developing an efficient discovery plan and investigative strategy (6).

Mobile device forensic is represented as a subcategory of small scale device forensics domain in the ontological representation for digital forensic disciplines by Karie and Venter (7). Recent reviews of mobile forensics literature have suggested that both mobile device forensics and mobile app forensics are emerging areas, but they are relatively less studied than traditional hard-disk forensics or mobile security (8–10).

A taxonomy provides an informative categorization of data remnants in the investigation of communication apps. In our recent work, a taxonomy incorporating artifacts of forensic interest from Android devices involving 40 mobile health (mHealth) apps (11) was proposed. Immanuel, Martini, and Choo (12) also noted the lack of research on forensic taxonomy and presented an Android Cache Forensic Process, designed to forensically classify, extract, and analyze Android caches. To the best of our knowledge, there is no published forensic taxonomy for Android communication apps. This is the gap this paper seeks to address.

In this paper, the communication apps are broadly categorized into three categories based on the services provided, and the forensic artifacts are classified based on our study of these communication apps. Then, a two-dimensional taxonomy of the communication apps is provided.

The contributions of this paper are twofold:

- Identification and analysis of data remnants of forensic interest to an investigator from the Android communication apps.
- Providing a two-dimensional taxonomy for Android communication apps with the communication app categories in one dimension and the forensic artifacts in the other dimension.

The rest of the paper is organized as follows. Review of existing work is outlined in the next section. The forensic analysis and experiment results are presented in the "Case study: 30 communication apps" section, prior to the presentation of the two-dimensional taxonomy in the "Proposed Forensic Taxonomy for

[1]Information Assurance Research Group, University of South Australia, Adelaide, SA 5001, Australia.
[2]Department of Information Systems and Cyber Security, University of Texas at San Antonio, One UTSA Circle — San Antonio, TX 78249-0631, USA.
[3]School of Information Technology and Mathematical Sciences, University of South Australia, Mawson Lakes, SA 5095, Australia.

Received 11 May 2015; and in revised form 28 Sept. 2015; accepted 17 Oct. 2015.

Communication Apps" section. The last section concludes the paper.

Related Work

Plachkinova, Andrés (13) proposed a security and privacy taxonomy for mobile health apps, without considering the forensic artifacts and any other category of apps in their taxonomy. As mentioned in the previous section, we proposed a two-dimensional forensic taxonomy of mHealth apps in our earlier work (11). However, the work did not examine any communication app. Alliano, Herriger (14) reviewed 21 Augmentative and Alternative Communication (AAC) apps for iPad and identified how individuals with complex communication needs can use them for a variety of communication purposes and to target a variety of treatment goals. However, the authors did not consider any forensic artifact remnants in their work.

In a forensic examination, user-generated data—for example, due to app usage, such as in cloud and dating apps (15–21) could be recovered in plain text format from the device's user data partition. Similarly, in the forensic examinations of the Android WhatsApp messenger by Anglano (22) and Thakur (23), various artifacts and data were recovered.

Mahajan, Dahiya (24) used Cellebrite UFED Classic Ultimate (V 1.8) to analyze WhatsApp and Viber apps and managed to extract files and folders from five Android phones running Froyo (2.2), GingerBread (2.3.x), and Ice-Cream Sandwich (4.0.x). The extracted data included chat messaging logs and history, sent and received images, and video files. They were also able to determine the location of data on the device. However, their study was limited to text messages, voice calls, and multimedia files (e.g., videos and audio), as group chat and hidden/blocked/deleted contacts were not considered. The authors also did not explain the structure of the databases or how to extract the information from the databases.

Lee and Chung (25) analyzed Line and Viber apps for Windows 8 mobile. The authors were able to determine the locations of the directories that stored the artifacts for these two apps. The authors were also able to identify the event type, direction, time, and the phone number that a user contacted through the Viber app. For Line app, the authors identified the user's access ID, last message, the type of chat, and timestamp from the database.

In the forensic analysis of three IM apps, namely AIM, Yahoo Messenger, and Google Talk, on iPhones, Husain and Sridhar (26) accessed the logical copies of the files on the iPhone through a Windows machine. Their investigation concluded that AIM screen name, plain text password, conversation detail with timestamp, and unique phrase could be found in the stored files.

In the study of Skype and MSN apps, Chu, Lo (27) were able to locate the target strings in the Android device's memory, even after the devices were rebooted. In the investigation of the Android Viber app, Chu, Yang (28) were able to create an image of the memory and recover the received and sent IMs using the sender's and receiver's mobile numbers.

Gao and Zhang (29) investigated the WeChat app to understand how the app stored user data on an iPhone 5 device. The authors recovered different types of data located in different subfolders of the device, such as user information, voice messages, and thumbnails of pictures and videos taken using the device's built-in camera.

A summary of communication app forensic research is outlined in Table 1.

Case Study: 30 Communication Apps

At the time of this research (April 2015), 540 free communication apps were available at the Google Play store (30). However, many of these apps are used as add-ons to other apps. Therefore, the total number of communication apps is less than the number shown in Google Play store. The 30 apps examined in this paper were among the 100 most popular communication apps. In this section, the findings of three of the most popular apps [Line (version 4.6.1), Viber (version 5.0.2), and WeChat (version 5.4)] are discussed and the interested reader is deferred to Tables 11 and 12 for the forensic taxonomy and summary of findings, respectively. Line was reportedly the most downloaded communication app in 52 countries on Google Play—with 400 million monthly active users in 2013 sending 50 billion messages per day (31). WeChat and Weixin (Chinese version of WeChat) had 355 million monthly active users, and Viber was reported having 200 million users in 193 countries with 100 million monthly active users in 2013 (31).

In the case study, 30 popular free Android communication apps available on Google Play store (30) were examined. The apps were installed and registered on two Google Nexus 4 phones (Android version 5.0.1). WiFi network to WiFi network communication channel was used for the experiments. Both phones had WiFi enabled and were connected to the same WiFi network. MicroSystemation XRY (version 6.10.1), a popular commercial forensic tool, was used to extract a logical forensic

TABLE 1—*Summary of communication app forensic research.*

Communication Apps	Examined By	Platform(s)	Artifacts Recovered
WhatsApp	Anglano (22) Thakur (23) Mahajan, Dahiya (24)	Android	Database, chat history, contact information, group chat, images, timestamp
Viber	Mahajan, Dahiya (24) Lee and Chung (25) Chu, Yang (28)	Android Windows 8 mobile	Chat logs, chat history, received images
Line	Lee and Chung (25)	Windows 8 mobile	User's access ID, last message, type of chat and timestamp
WeChat	Gao and Zhang (29)	iPhone	User information, voice messages, thumbnails of pictures and videos
Skype MSN	Chu, Lo (27)	Android	Located target strings
AIM Yahoo messenger Google Talk	Husain and Sridhar (26)	iPhone	User credentials, timestamp, screen names, friend list

image (see the manual (32) for a step-by-step guide in conducting a logical acquisition using XRY). The use of XRY was based on the authors' access, and no personal recommendations or endorsement should be presumed from the tool selected.

A Windows 7 desktop machine was used to analyze the artifacts. Individual sets of experiments were conducted, one set for each app. After one set of experiments was concluded, the phone was wiped prior to installing the next app. The steps we undertook in the forensic wiping are as follows:

- Turn the Nexus 4 phone off;
- Press and hold the "VOLUME DECREASE" and "POWER" buttons simultaneously;
- In the recovery menu, press the "VOLUME DECREASE" volume twice;
- Press the "POWER" button, which will result in a red exclamation mark being displayed;
- Press the "POWER" and "VOLUME INCREASE" buttons simultaneously;
- Scroll down to the Wipe data/factory reset and press the "POWER" button; and
- Scroll to Yes and press the "POWER" button, which will result in the deletion of user data.

As noted by Leom et al. (33), other deleted data from the phone using Android factory reset may be recoverable using a physical extraction. However, this will not affect the findings in this paper, as app-related artifacts/databases are stored in their respective directories.

Databases

Files and data generated by the Line app, WeChat app, and Viber app were stored on the internal device memory, which is normally inaccessible by users. For example, the generated data stored in the databases of Line, WeChat, and Viber could be located in the following:

- /data/data/jp.naverline.android/databases
- /data/data/com.tencent.mm/MicroMsg/fa6dbec88b51fcb3f3cb82b2a4412c99, and
- /data/data/com.viber.voip/databases

However, the user profile pictures, sent and received pictures for Line, WeChat, and Viber apps were found to be stored respectively on the phone's memory in the following locations (which are accessible by the user without the need to root the phone):

- /sdcard/Android/data/jp.naver.line.android/storage,
- /sdcard/tencent/MicroMsg/fa6dbec88b51fcb3f3cb82b2a4412c99,
- /sdcard/viber, and/sdcard/Android/data/com.viber.voip.

The recovered artifacts from Line, WeChat, and Viber apps are listed in Table 2.

Line's main database, `naver_line.db`, contains 24 tables; and the nine tables containing artifacts of forensic interest are described in Table 3.

The main database of WeChat, `MicroMsg.db`, contains 73 tables. In Table 4, six tables containing artifacts of forensic interest are listed.

Viber's main database is `viber_data.db` and has eight tables. Five tables containing artifacts of forensic interest are listed in Table 5.

Contact Information

The `contacts` table in Line's `naver_line` database stores records of contacts. Table 6 describes the fields in the `contacts` table that were considered to have evidentiary value for an investigator.

Each user's contacts are assigned a unique ID (`m_id`), which is stored as a hash value. The contact is identified by the name saved by the user (`name` field). The `server_name` field indicates the contact profile name of the contact. The `picture_status` and `picture_path` fields store information about the contact's profile picture. A NULL value in `picture_status` indicates that the contact does not have a profile picture stored. In the event that a picture is stored, it will be stored in the `/sdcard/Android/data/jp.naver.line.android/storage/p` directory and can be located by searching for `m_id` (by default, the picture in `.thumb` uses `m_id` as the filename).

WeChat stores contact information in multiple tables. The `addr_upload2` table stores the phonebook contacts of the user. WeChat assigns a unique ID for each contact and stores a MD5 hash value of the contact information. The table stores the names, phone numbers, and email IDs (if stored in the phonebook) of the contacts (see Fig. 1). The contacts stored in the `addr_upload2` table are from the phonebook of the device, but the table does not indicate whether the contacts are WeChat app users.

The `rcontact` table stores the WeChat contacts of the user, which are each assigned a unique user name in the format `wxid_X...X`. However, there is no timestamp information indicating when the contact was added, removed, or blocked.

The third table associated with contact information in WeChat database is the `linkedInFriend` table, which stores the name, positions, URLs of profiles, and profile pictures of the user's LinkedIn contacts (see Fig. 2), while WeChat stores the profile pictures of the contacts in its server.

Viber database stores the contact information in three separate tables, namely `vibernumbers`, `phonebookcontact`, and `phonebookdata`. The `vibernumbers` table stores the phone numbers (in plain text), and both `photo` and `actual_photo` fields contain the names of the images used as profile pictures of the contacts stored in `/sdcard/viber/media/User photos` directory (see Fig. 3). Contact data were imported and stored in the `phonebookcontact` and `phonebookdata` tables (see Table 7).

Determining When a Contact was Added

Investigators may need to identify when a contact was added to the list. In the Line app, such information can be identified from the `contacts` table (see Table 6). The `created_time` and `added_time_to_friend` fields show the timestamp information of the sent and accepted add requests. The `updated_time` field shows any update (hidden, blocked) made with the contact. All times are shown in Unix millisecond epoch time.

For the WeChat app, information from the `type`, `createTime`, and `content` fields of the `message` table can help determine when a contact request was sent and when the contact was added (see Fig. 4). For a friend request sent to a user, the value of the `type` field is 10,000 and the `content` field contains a notification message. When the request

TABLE 2—*Line, WeChat and Viber artifacts.*

	Line		WeChat		Viber	
Content	Directory	File	Directory	File	Directory	File
Main Database	/data/data/jp.naverline.android/databases	naver_line.db (24 tables)	/data/data/com.tencent.mm/MicroMsg/fa6dbec88b51fcb3f3cb82b2a4412c99	MicroMsg.db (73 tables)	/data/data/com.viber.voip/databases	viber_data.db (8 tables), viber_messages.db (15 tables)
All images	/sdcard/Android/data/jp.naver.line.android/storage	gallery default	N/A	N/A	N/A	N/A
Sent and Received images	/sdcard/Android/data/jp.naver.line.android/storage/mo	mo	/sdcard/tencent/MicroMsg/fa6dbec88b51fcb3f3cb82b2a4412c99/image2	Files with different names	/sdcard/viber/media/Viber Images and/sdcard/viber/media/.thumbnails, and /sdcard/viber/media/Viber and/sdcard/viber/media/.thumbnails	Files with different names
Profile pictures of contacts	/sdcard/Android/data/jp.naver.line.android/storage/p	stored as .thumb, which can be changed to .png or .jpg to view the pictures	N/A	Not locally stored in the device	/sdcard/viber/media/User photos,/sdcard/Android/data/com.viber.voip/cache/image_fetcher_cache	Files with different names
Own profile picture	/data/data/jp.naverline.android/files	Profile_photo	/sdcard/tencent/MicroMsg/diskcache	cache	/sdcard/viber/media/User photos	Files with different names
Maps	N/A	N/A	/sdcard/tencent/MicroMsg/fa6dbec88b51fcb3f3cb82b2a4412c99/image2	Directory	N/A	N/A
LinkedIn profile pictures	N/A	N/A	/sdcard/tencent/MicroMsg/fa6dbec88b51fcb3f3cb82b2a4412c99/image	Directory	N/A	N/A
Deleted images	N/A	N/A	N/A	N/A	/sdcard/viber/media/.thumbnails	Files with different names
Voice messages	N/A	N/A	N/A	N/A	/sdcard/viber/media/.ptt	Audio Files

TABLE 3—*Tables in the* naver_line *database of Line.*

Table Name	Content
buddy_detail	List of added friends
Chat	List of ongoing chat
chat_history	Chat history
chat_member	List of participants in all chats
contacts	List of Line contacts
email_recommend	Email IDs of contacts
groups	List of group members
permanent_tasks	List of tasks
product	List of Line products purchased

TABLE 4—*Tables in the* MicroMsg *database of WeChat.*

Table Name	Content
imgInfo2	Contains the path of the thumbnail of sent or received images. The path is named as THUMBNAIL_DIRPATH://th_XXYY...YY. Here, XX (two numerical digits) represents the directory name in /sdcard/tencent/MicroMsg/fa6dbec88b51fcb3f3cb82b2a4412c99/image2 where the image is stored
linkedIn Friend	Stores details of LinkedIn friends of the user
addr_upload2	The contact list from the phonebook
message	Text messages between users
Rcontact	WeChat friend list
Userinfo	User phone number, location, email address, LinkedIn URL and etc.

TABLE 5—*Tables in the viber_data database of Viber.*

Table Name	Content
blockednumbers	List of blocked numbers
Calls	List of Viber voice calls
phonebookcontact	All contacts imported from phonebook
Phonebookdata	Email addresses of contacts
Vibernumbers	Phone numbers of Viber users in phonebookcontact table

TABLE 6—*Structure of* contacts *table of Line.*

Field Name	Meaning
m_id	A unique ID stored as a hash value identifies each contact
contact_id	A unique numerical contact ID
contact_key	Key value for the contact
Name	Contact name as stored in the phonebook
server_name	Contact name as stored by the contact in the server during registration
addressbook_name	Contact name as stored in the phonebook
status_msg	Displays status message set by contact
picture_status	A non-null value indicates there is a profile picture of the contact in
picture_path	/sdcard/Android/data/jp.naver.line.android/storage/p directory
created_time	Unix millisecond epoch time indicating when the contact information was added and/or updated
updated_time	
Status	1 if unblocked contact, 2 if blocked contact, 6 if contact deleted
Hidden	0 if not hidden, 1 if hidden contact
Relation	0 if not yet added friend request, 1 if added as friend, 2 if deleted from contact list

has been accepted, the type field becomes 1 and the notification is observed in the content field. The createdTime field shows the times of the events in Unix millisecond epoch time.

Viber does not allow the sending of an add request to a contact. Viber adds the contacts from the imported phonebook of the user device. The time when a contact was added into the phonebook or when a contact joined Viber can be determined from the joined_date field of the phonebookcontact table in the viber_data database (see Table 7).

Dealing with Hidden, Blocked, and Deleted Contacts

The contact details of hidden, blocked, or deleted contacts in the Line app (see Fig. 5) were recovered by examining the status, hidden, and relation fields in the contacts table (see Table 6).

When a user deletes a contact from his/her list, the contact type field in the rcontact table of WeChat becomes 0. For blocked contacts, the type is 11, and for any normal contact, the type is 3 (see Fig. 6).

Viber does not allow user to hide a contact, but a user can block contacts. Contact information of blocked contacts can be recovered from the blockednumbers table in the viber_data database (see Fig. 7).

Determining a Contact Phone Number

The contact number can be determined from the phonebook of the device in the Line app. The contact names and numbers are stored in the realname and mobile fields of the addr_upload2 table in the WeChat app, respectively.

Two tables in the viber_data database, namely phonebookcontact and phonebookdata, store contact information (see Table 6). The former stores the name of the contact and the unique ID in the display_name and _id fields, respectively. The contact's phone number and ID are stored in the latter's data1 and contact_id fields, respectively. Therefore, by linking the IDs in both _id fields of phonebookcontact and phonebookdata tables, one could trivially obtain the contact name and number.

Exchanged Text Messages

The Line app stores all sent and received messages in the chat_history table (see Table 8), which facilitates the reconstruction of chronology of the exchanged messages.

WeChat and Viber store all sent and received messages in the message table (see Table 9) and messages table (see Table 10), respectively.

Multimedia Files

The images sent and received by a Line app user are stored in the /sdcard/Android/data/jp.naver.line.android/storage/mo directory. A separate directory with the m_id of the contact stores the associated images. A cached copy of the sent images can also be found in the /storage/emulated/0/Android/data/jp.naver.line.android/temp/ directory. The image is saved with a name identified with integer numeric value. Similarly, the voice messages are stored in the /sdcard/Android/data/jp.naver.

realname	realnamepyinitial	realnamequanpin	username	nickname	nicknamepyinitial	nicknamequanpir	type	moblie
Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
M[redacted]ed	M[redacted]HED	M[redacted]ed					0	[redacted]-22937
K[redacted]al	K[redacted]AL	K[redacted]					0	[redacted]7510
J[redacted]aj	J[redacted]AJ	J[redacted]aj					0	[redacted]1385
Ka[redacted]ma M	KA[redacted]AMAM	Ka[redacted]ma M					0	[redacted]74535

FIG. 1—*Phonebook contacts in WeChat addr_upload2 table.*

name	position	picUrl	echatUsernam	echatSmallHea	chatBigHe	linkedInProfileUrl
Filter	Filter	Filter	Filter	Filter	Filter	Filter
AK[redacted]kder	Graduate Research Assistant	https://media.licdn.com/mpr/mprx/0_zdf...				https://www.linkedin.com/pub/akm-khaled-...
Md[redacted]ullah	Manager Engineering	https://media.licdn.com/mpr/mprx/0_zdf...				https://www.linkedin.com/pub/md-tarique-...
Md[redacted] Khan	Manager, Insights & Intellige...	https://media.licdn.com/mpr/mprx/0_yrL...				https://www.linkedin.com/in/khanazim
Sh[redacted]Hasan	Postdoctoral Researcher in Se...	https://media.licdn.com/mpr/mprx/0_yq5...				https://www.linkedin.com/in/sheikhfaridulh...
H. M[redacted]ssan	Assistant Commissioner of C...	https://media.licdn.com/mpr/mprx/0_xrB...				https://www.linkedin.com/in/sharifhassan

FIG. 2—*LinkedIn contacts in WeChat linkedInFriend table.*

_id	canonized_number	photo	actual_photo	viber_name	clear
Filter	Filter	Filter	Filter	Filter	Filter
786	[redacted]3463				0
787	[redacted]10282				0
788	[redacted]33166	16d22bc92bac66f7e1f5ee5939c20ef207ce880969a1ef6c447639be37c8e00b	16d22bc92bac66f7e1f5ee5939c20ef207ce880969a1ef6c447639be37c8e00b		0
789	[redacted]5829	a79424c58550c2e97af5cdac9d5830bae683d6b3e0d9371b639402b8c3160441	a79424c58550c2e97af5cdac9d5830bae683d6b3e0d9371b639402b8c3160441		0
790	[redacted]8707	9fc23da1694e0896bc79711daef766f29f6ba780dddd103d9bc9284bf8fd3246	9fc23da1694e0896bc79711daef766f29f6ba780dddd103d9bc9284bf8fd3246		0
791	[redacted]3294	fce19a8c5a9e56183f2a141674cb2926018651d4eecf03fb22750a2f629a5ae1	fce19a8c5a9e56183f2a141674cb2926018651d4eecf03fb22750a2f629a5ae1		0

FIG. 3—*Vibernumbers table in viber_data database.*

TABLE 7—*Structure of* `phonebookcontact` *and* `phonebookdata` *tables of Viber.*

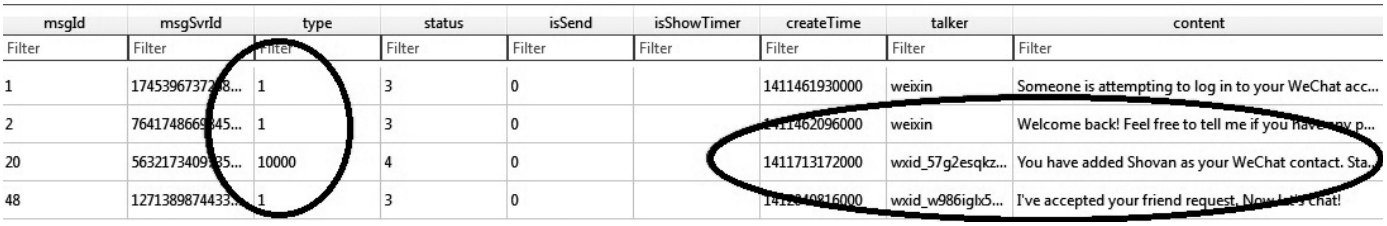
	Tables	
Field Name	`phonebookcontact`	`phonebookdata`
`_id`	A unique numerical ID for the contacts	
`contact_id`		Contact name as stored in phonebook
`display_name`	Contact name as stored in phonebook	
`low_display_name`	Contact name in lowercase characters (same name as display_name)	
`viber`	0 if contact does not use Viber, 1 if contact uses Viber	
`contact_lookup_key`	A key value to identify a contact	
`contact_hash`	Hash value of contact id	
`joined_date`	Unix millisecond epoch time indicating when the contact joined Viber	
raw_id		Contact name in lowercase characters (same name as display_name)
data1		Email addresses imported from synced Gmail account of the user

`line.android/storage/mo` directory. The naming convention of the voice messages is voice_xx.aac (where xx is integer numeric value). The storage location of the sent video files is the `/sdcard/Pictures/NAVER_LINE_MOVIE` directory, whereas the received video files are stored in the `/media/external/video/med7ia/` directory.

When a multimedia file is sent or received, WeChat stores them on the device. The `imgInfo2` database contains the path of the thumbnail of sent or received images in the `thumbImgpath` table, and the path is THUMBNAIL_DIRPATH://th_XXYY...YY. XX (two numeric digits) represents the directory name where the image is stored. The sent and received images are stored in `/sdcard/tencent/MicroMsg/fa6dbec88b51fcb3f3cb82b2a4412c99/image2` directory.

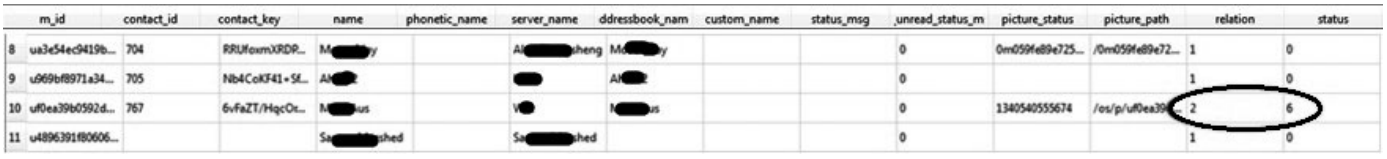
A video message of WeChat is indicated by the `type` field (with the value 43) of the `message` table (see Table 9). Whether the video message has been sent or received that can be determined from the `isSend` field (similar as text messages). The video messages are stored in the `/sdcard/tencent/MicroMsg/fa6dbec88b51fcb3f3cb82b2a4412c99/video` directory, and the file name is indicated by the `imgPath` field in the `message` table (see Fig. 8)

Further details about the video messages can be found in `videoinfo2` table of `microMsg` database. The `downloadtime` field indicates when the video message was downloaded by the receiver. A value of 0 indicates the files has been received, but has yet to be downloaded by the receiver (see Fig. 9).



msgId	msgSvrId	type	status	isSend	isShowTimer	createTime	talker	content
Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1745396737[redacted]8...	1	3	0		1411461930000	weixin	Someone is attempting to log in to your WeChat acc...
2	7641748669[redacted]45...	1	3	0		1411462096000	weixin	Welcome back! Feel free to tell me if you have any p...
20	5632173409[redacted]35...	10000	4	0		1411713172000	wxid_57g2esqkz...	You have added Shovan as your WeChat contact. Sta...
48	1271389874433[redacted]	1	3	0		1412640816000	wxid_w986iglx5...	I've accepted your friend request. Now let's chat!

FIG. 4—*Adding a new contact and request acceptance in message table of WeChat.*

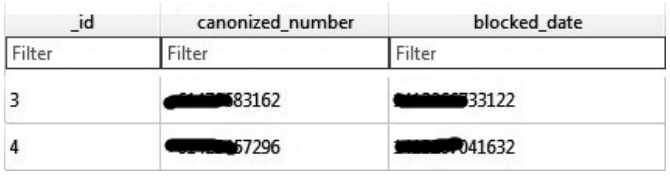


m_id	contact_id	contact_key	name	phonetic_name	server_name	ddressbook_nam	custom_name	status_msg	unread_status_m	picture_status	picture_path	relation	status
8 ua3e54ec9419b...	704	RRUfoxmXRDP...	M[redacted]y		Al[redacted]heng	M[redacted]y			0	0m059fe89e725...	/0m059fe89e72...	1	0
9 u969bf8971a34...	705	Nb4CoKF41+Sf...	Al[redacted]		[redacted]	Al[redacted]			0			1	0
10 uf0ea39b0592d...	767	6vFaZT/HqcOt...	M[redacted]us		W[redacted]	M[redacted]us			0	1340540555674	/os/p/uf0ea39[redacted]	2	6
11 u4896391f80606...			Se[redacted]shed		Se[redacted]shed				0			1	0

FIG. 5—*Deleted contacts in Line app.*

	username	alias	conRemark	domainList	nickname	pyInitial	quanPin	showHead	type
24	weixin				[redacted]	W[redacted]	w[redacted]ui	87	3
25	v1_f644c3b21f8dd29ab2f61d...				Ti[redacted]N			84	0
26	wxid_57g2esqkziiu22	o[redacted]an			S[redacted]n	SH[redacted]N	S[redacted]n	83	11
27	wxid_w986iglx5y9022	ia[redacted]a2			Azfar	A[redacted]R	A[redacted]	65	0

FIG. 6—*Blocked or deleted contact in rcontact table of WeChat.*



_id	canonized_number	blocked_date
Filter	Filter	Filter
3	[redacted]83162	[redacted]33122
4	[redacted]57296	[redacted]041632

FIG. 7—*Blocked numbers in Viber.*

TABLE 8—*Structure of* `chat_history` *table of Line app.*

Field Name	Meaning
`id`	Sequence number of the text messages. This is an incremental value and if any message is deleted, then that value will be missing
`type`	1 if text message, 4 if voice chat, 9 if group chat
`chat_id`	Identifies the contact with whom the conversation was done
`from_mid`	Contains the `m_id` of the sender of the message. This value is null when messages are sent from the device under examination
`content`	The plain text messages exchanged between the sender and receiver
`created_time` `delivered_time`	Time in Unix millisecond epoch when the message was created/delivered
`attachment_ image`	0 if no image attached, 1 if an image attached

In Viber, the type of a multimedia file is indicated by the `extra_mime` field of the `messages` table (see Table 10). The images and voice messages are stored in different

TABLE 9—*Structure of* `message` *table of WeChat.*

Field Name	Meaning
`msgId`	Sequence number of the text messages. This is an incremental value and if any message is deleted then that value will be missing
`type`	1 if text message, 3 if multimedia image, 43 if recorded video message, 48 if location data, 50 if voice chat
`status`	2 if message sent, 3 if message received, 4 if a new contact add request sent, 6 for VoIP message,
`isSend`	1 if message sent, 0 if received
`createTime`	Time in Unix millisecond epoch when the message was created
`talker`	Unique user name of the person with whom the message is exchanged in the format wxid_XXXXXX XXXXXXXX
`content`	Plain text messages, location coordinates, image detail or type of VoIP content (video/voice)
`imgpath`	Contains the path of the thumbnail of sent or received images. The path is named as THUMBNAIL_ DIRPATH://th_XXYY...YY. XX (two decimal digits) represents the directory name in/sdcard/tencent/ MicroMsg/fa6dbec88b51fcb3f3cb82b2a4412c99/ image2, where the image is stored

locations. For images, the `extra_uri` field indicates the name and location of the image file, whereas the `extra_uri` field indicates the file name of the voice message. The images sent and received are stored in the `/sdcard/viber/media/.thumbnails` and `/sdcard/viber/media/Viber Images` directories, respectively, and stored voice messages are located in `/sdcard/viber/media/.ptt` directory.

TABLE 10—*Structure of* `messages` *table of Viber.*

Field Name	Meaning
`_id`	Sequence number of the text messages. This is an incremental value and if any message is deleted, then that value will be missing
`address`	Viber number (a phone number) of the contact
`date`	Time in Unix millisecond epoch when the message was created
`read`	0 if message read by receiver, 1 if message delivered but not yet read by the receiver
`type`	0 if message is received by the user, 1 if message sent by the user
`body`	Contains plain text message or voice call information (outgoing_call, incoming_call, missed_call)
`location_lat`	Latitude of the location of the sender/receiver
`location_lng`	Longitude of the location of the sender/receiver
`extra_uri`	Name and location of the image files sent/received. Name of the file received as a voice message in the sdcard/viber/media/.ptt directory
`extra_mime`	Indicates the content type of the message (text, call, image, sticker, sound)

Group Chats

The contents of group chats in the Line app can be determined from the `chat_history` table as discussed earlier. There is a difference in the `chat_id` field of the `chat_history` table between a group chat and a one-to-one chat. In a group chat, the `chat_id` field is a unique ID for the group chat and does not correspond to the `m_id`s of the contacts. The latter can be recovered from the `chat_member` table of the `naver_line` database, in addition to the timestamp information of the group's creation.

In WeChat, the contents of group chats can be determined from the `message` table (see Table 9). The chat ID in the `talker` field is a unique identifier for a group chat, and the contacts' `m_id`s can be revealed from the `content` field. This field contains the user names of the group chat members when the group is created. The `createTime` field shows the time in Unix millisecond epoch. The messages of individual members are identified by the IDs of the users preceding the text message in the `content` field (see Fig. 10).

For Viber, the contents of group chats can be determined from the `messages` table (see Table 10). The `address` field is always empty when a message is sent to a group. In the `conversations` table, the `group_id` field stores the unique ID for the group (same `group_id` in the `messages` table). There can be a maximum of eight participants in a group, and each participant is assigned an integer ID in the `participant_1`, `participant_2`, …, `participant_8` fields. These numeric values correspond to the `_id` field of the `participants_info` table. From the `participants_info` table, the phone number and Viber contact name can be obtained from a corresponding `_id` of a group member (see Fig. 11).

Voice Chats

The history of voice chat conversations in Line, WeChat, and Viber is stored in the `chat_history` (see Table 8), `message` (see Table 9), and `messages` (see Table 10) tables, respectively. The `m_id` of the recipient of the voice chats in Line and WeChat can be obtained from the `chat_id` and `talker` fields, respectively.

In both Line and WeChat, the `content` field shows the duration of a call in milliseconds, and the `created_time` field shows the time when the call was initiated or received. In Viber, the `body` field indicates `incoming_call`, `outgoing_call`, or `missed_call`, and the `extra_duration` field shows the duration of a call in seconds.

msgSvrId	type	status	isSend	isShowTimer	createTime	talker	content	imgPath
6140999076610...	43	3			1411975702000	wxid_w986igbk5...	:5:0	1658252...
5981136827091...	1	2	1		1411975742844	wxid_w986igbk5...	Received	
8599179312064...	43	2	1		1411975751121	wxid_w986igbk5...	wxid_t09o60hxzub222:5:0	165910290914991b13d7662
2090672389508...	43	3			1411975806000	wxid_w986igbk5...	:0:0	17001129091411193
8384195965044...	1	2	1		1411975859672	wxid_w986igbk5...	Not yet downloaded	
2597985009340...	43	2	1		1411975869672	wxid_w986igbk5...	wxid_t09o60hxzub222:13:0	1700342...1414eec3a42864

FIG. 8—*Video messages in WeChat.*

	filename	clientid	msgsvrid	netoffset	filenowsize	totallen	thumbnetoffset	thumblen	status	createtime	lastmodifytime	downloadtime
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	16582529091453080		6140999076610642431	0	14672833	14672833	0	0	199	1411975702	1411975730	1411975708
2	165910290914991b13d7662		8599179312064847243	4271516	0	7516136	0	9605	199	1411975751	1411975757	1411975751
3	17001129091411193		209067238950870810	0	0	7373919	0	0	111	1411975806	1411975811	0
4	17003429091414eec3a42864		2597985009340708094	2324705	0	2324705	0	17675	199	1411975869	1411975873	1411975869

FIG. 9—*Timestamp information of downloaded video message in WeChat.*

msgId	msgSvrId	type	status	isSend	isShowTimer	createTime	talker	content
23		10000	4	2		1412057558449	451027899@chatroom	You invited Shovan, Azfar to the group chat
24		10000	4	2		1412057558524	451027899@chatroom	<a href="weixin://findfriend/verifycontact/451027899@chatroom/wxid_57g2esqkzii...
25	1212533720676...	1	2	1		1412057585819	451027899@chatroom	This is a test message. Pls do not reply
26	7773207540416...	1	3	0		1412057602000	451027899@chatroom	wxid_w986igbk5y9022:Ok, i wont

FIG. 10—*Group chat in WeChat.*

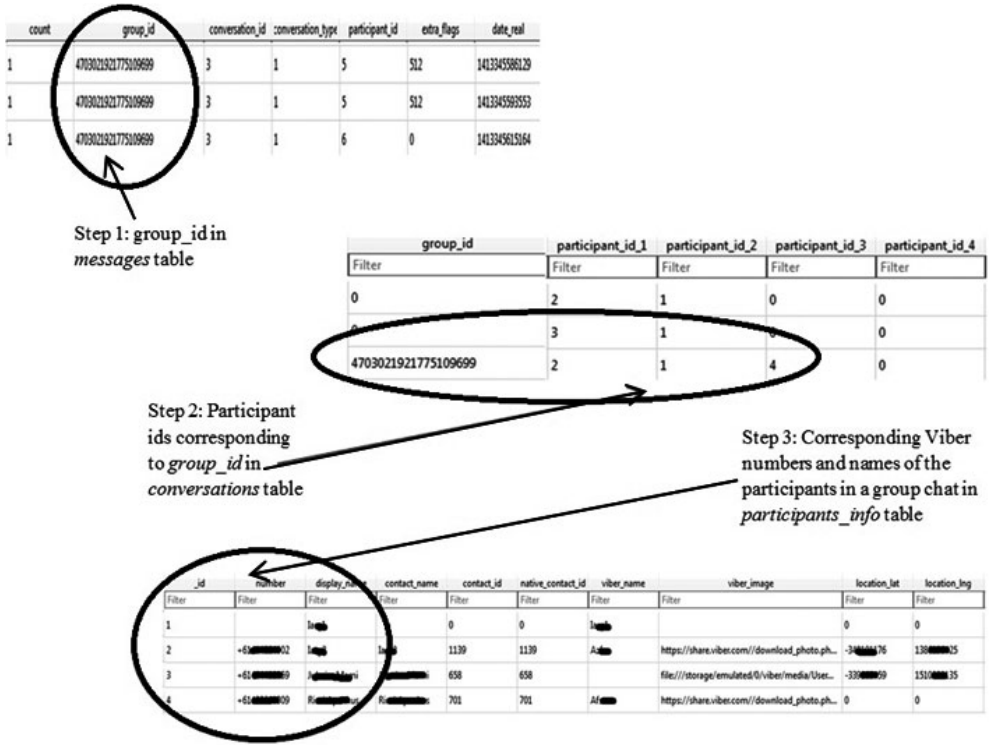


FIG. 11—*Obtaining participant's phone number from a Viber group chat.*

Proposed Forensic Taxonomy for Communication Apps

Due to the increasing number of communication apps, it is important to have a forensic taxonomy for all existing communication apps. Therefore, based on the case studies of the 30 apps, a forensic taxonomy is proposed.

The communications apps are broadly categorized into Instant messaging (IM) apps, Voice over IP (VoIP) apps, and Augmentative and Alternative Communication (AAC) apps:

IM apps offer real-time text transmission between users. Some of the IM apps allow the users to create a friends list (e.g., Skype, Yahoo messenger), while the others create a friend list from the phonebook directory of the users (e.g., Line, Viber). The latter takes into consideration phonebook contacts and places those users in the friend list who are using the same app.

VoIP apps provide an easier and more affordable way to chat with individuals who are located around the world. Most of the VoIP apps also provide video chat facility to the users.

AAC apps encompass the communication apps used to express thoughts, needs, wants, and ideas for those with impairments in the production or comprehension of spoken or written language. As noted by Higginbotham and Jacobs (34), Android AAC apps have come a long way to support the needs of individuals who have speech and language disabilities.

From the artifacts determined from the forensic analysis of the 30 most popular Android communication apps, the artifacts are broadly categorized into four groups namely User and contact information, Exchanged messages, Timestamps, and others.

User and contact information: This group of artifacts contains the data remnants related to the user identity and list of contacts.

- *Find phone number from a contact ID*: It is not always possible to determine the phone number of the user from his/her contact ID. However, some apps store the phone number with the user ID.
- *Recover user credentials*: Apps may require users to login using their user credentials (e.g., username and password, PIN, and authentication tokens) in order to use the apps. Therefore, user credentials should be an artifact that forensic investigators seek to locate during the app forensic process (e.g., determine whether the credentials are stored in and can be recovered from the app's databases).
- *View address book*: Communication apps generally import the phone address book into their local database. The address book contains the user names and phone numbers of all contacts saved by the user.
- *View contact ID:* Communication apps generally require a user to create a personal contact ID. This artifact reveals the contact ID of the user to the forensic investigator.
- *View contact Status messages*: The status message includes the publicly shared status of the users.
- *View blocked/hidden contacts*: A user may block or hide someone from the contact list and deny having any contact with the other individual. Therefore, recovered hidden or blocked contacts could prove useful to an investigator.
- *View deleted contacts*: Similar to blocked or hidden contacts, it would be useful for the investigators if the deleted contacts could be recovered.

Exchanged messages: This group of artifacts identifies the text, multimedia, and group message communications.

- *Determine the type of exchanged message (text, voice, group)*: This allows an investigator to determine the message type.
- *View unencrypted exchanged text messages*: The unencrypted exchanged messages stored in the app database.
- *View unencrypted exchanged multimedia images*: Most apps store the sent or received multimedia messages (e.g., images) in their database, and in some cases, unencrypted.

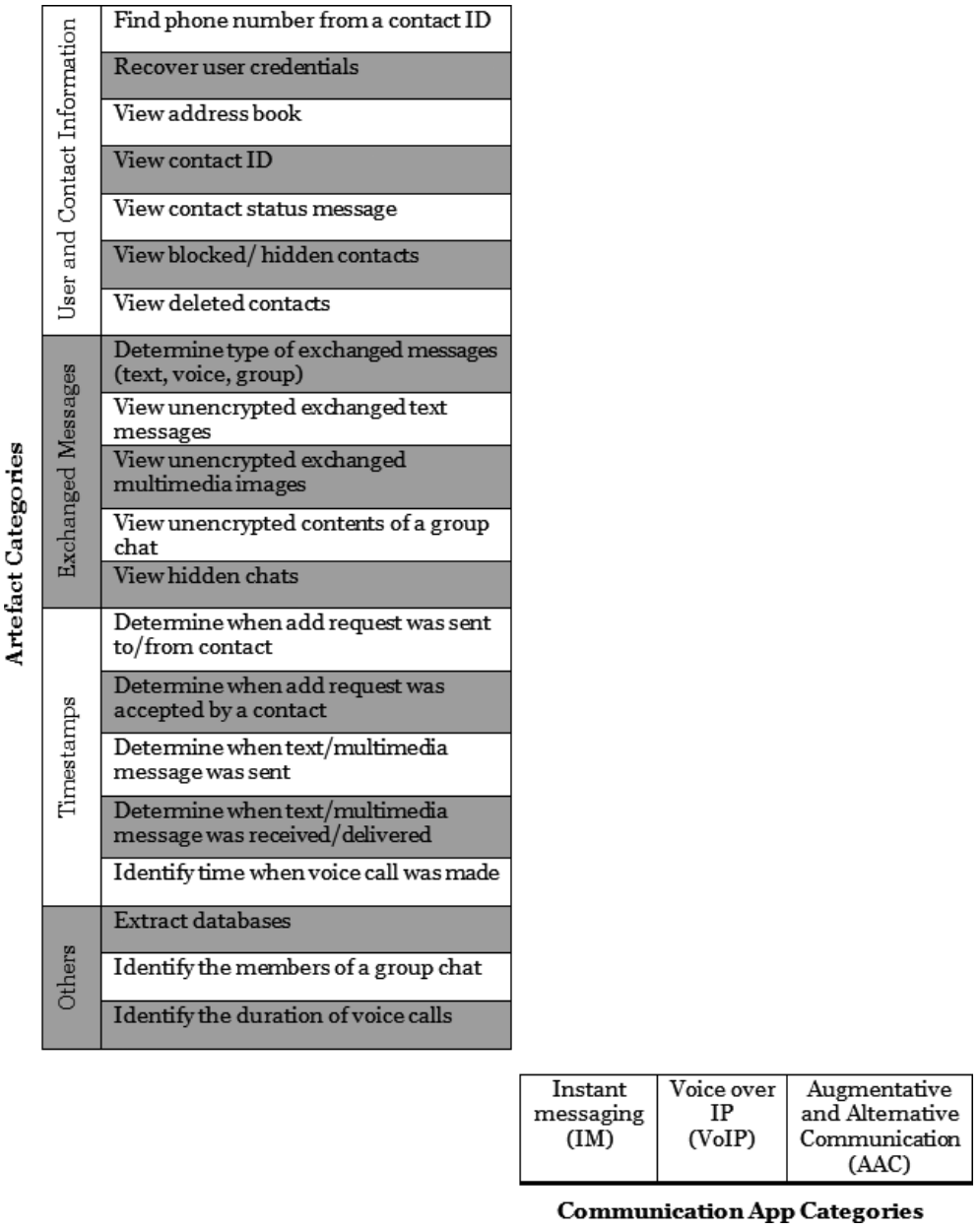


FIG. 12—*A two-dimensional communication app forensic taxonomy model.*

TABLE 11—*Forensic taxonomy of communication apps.*

App Category/Artifact Category		Instant Messaging (IM)	Voice Over IP (VoIP)	Augmentative and Alternative Communication (AAC)
User and contact information				
Find the phone number from a contact ID	Full	App2,App3,App5,App10,App12 App13,App14,App15,App16,App21 App22, App23 App24,App25,App30	App2,App3,App5,App10,App12 App13,App15,App16,App21, App22 App23,App24,App25,App30	
	Partial	App17	App17	
Recover user credentials	Full			
	Partial			
View address book	Full	App1,App2,App3,App4,App5 App6,App7,App10,App16,App18 App19,App21,App22,App23,App24 App30	App1,App2,App3,App5,App6 App10,App16,App18,App19,App21 App22,App23,App24,App30	
	Partial	App8,App12,App13,App15	App8,App12,App13,App15	

TABLE 11—*Continued.*

App Category/Artifact Category		Instant Messaging (IM)	Voice Over IP (VoIP)	Augmentative and Alternative Communication (AAC)
View contact ID	Full	App1,App4,App5App6,App10 App11,App16, App18,App19,App20 App21,App30	App1,App5,App6,App10,App11 App16 App18,App19,App20,App21 App30	
	Partial			
View contact status message	Full	App1,App23	App1,App23	
	Partial			
View blocked/hidden contacts	Full	App1,App2,App3,App4,App6 App7,App8,App10,App11,App13 App15, App18,App20,App22,App23 App24,App25, App30	App1,App2,App3,App6,App8 App10, App11,App13,App15,App18 App20, App22,App23,App24 App25,App30	
	Partial			
View deleted contacts	Full	App1,App2	App1,App2	
	Partial			
Exchanged messages				
Determine type of exchanged messages	Full	App1,App2,App3,App6,App8 App10,App12, App13,App14 App15,App16,App17,App19,App20 App21, App22,App23,App24 App25,App30	App1,App2,App3,App6,App8 App10,App12,App13,App15,App16 App17,App19,App20,App21,App22 App23,App24,App25,App30	App28
	Partial	App4,App18	App18	
View unencrypted exchanged text messages	Full	App1,App2,App3,App4,App6 App10,App11, App12,App13,App15 App16,App17,App18, App19,App20 App21,App22,App23 App24, App25 App30	App1,App2,App3,App6,App10 App11,App12,App13,App15,App16 App17,App18,App19,App20,App21 App22,App23,App24,App25,App30	App28
	Partial			
View unencrypted exchanged multimedia images	Full	App1,App2,App3,App4,App6 App7,App9,App10,App13,App15 App18,App19,App20,App21,App22 App23,App24,App25,App30	App1,App2,App3,App6,App9 App10, App13,App15,App18,App19 App20, App21,App22,App23 App24,App25,App30	
	Partial	App5,App8	App5,App8	
View unencrypted contents of a group chat	Full	App1,App2,App3,App4,App6 App10,App13, App15,App16,App19App20,App21,App22, App23,App24 App25,App30	App1,App2,App3,App6,App10 App13, App15,App16,App19,App20 App21, App22App23,App24,App25 App30	
	Partial	App8	App8	
View hidden chats	Full	App1,App15	App1,App15	
	Partial	App8	App8	
Timestamps				
Determine when add request was sent to/from contact	Full	App1,App2,App3,App18	App1,App2,App3,App18	
	Partial	App9	App9	
Determine when add request was accepted by a contact	Full	App1,App2,App18	App1,App2,App18	
	Partial	App9	App9	
Determine when text/ multimedia message was sent	Full	App1,App2,App3,App4,App6 App8,App10, App11,App12,App13 App14,App15,App16, App17,App18 App19,App20,App21,App22 App23 App24,App25	App1,App2,App3,App6,App8 App10, App11,App12,App13,App15 App16, App17,App18,App19,App20 App21, App22App23,App24,App25	
	Partial			
Determine when text/ multimedia message was received/delivered	Full	App1,App2,App4,App6,App8,App30	App1,App2,App6,App8,App30	
	Partial			
Identify when voice call was made	Full	App1,App3,App11,App12,App16 App21, App24,App30	App1,App3,App11,App12,App16 App21,App24,App30	
	Partial			
Others				
Extract databases	Full	App1,App2,App3,App4,App5 App6,App8,App9,App10,App11 App12, App13,App14,App15,App16 App17, App18,App19,App20,App21 App22 App23,App24,App25,App30	App1,App2,App3,App5,App6 App8 App9,App10,App11,App12 App13,App15,App16,App17,App18 App19,App20,App21,App22,App23 App24,App25,App30	App26 App28 App29
	Partial	App7		
Identify the members of a group chat	Full	App1,App2,App3,App4,App6 App10, App13,App14,App15,App16 App19, App20,App21,App22App23 App24,App25,App30	App1,App2,App3,App6,App10 App13,App15,App16,App19,App20 App21,App22App23,App24,App25 App30	
	Partial	App12	App12	
Identify the duration of voice calls	Full	App1,App2,App3,App6,App8 App10, App11,App12,App13,App16 App17, App19,App20,App21,App22 App23, App24,App30	App1,App2,App3,App6,App8 App10, App11,App12,App13,App16 App17, App19,App20,App21,App22 App23, App24,App30	
	Partial			

TABLE 12—*Summary of findings.*

			App Categories			User and Contact Information						
App ID	App Name	Version	Instant Messaging	VoIP	Augmentative and Alternative Communication	Find the Phone Number from a Contact ID	Recover User Credentials	View Address Book	View Contact ID	View Contact Status Message	View Blocked/ Hidden Contacts	View Deleted Contacts
App1	Line	4.6.1	F	F	N	N	N	F	F	F	F	F
App2	WeChat	5.4	F	F	N	F	N	F	N	N	F	F
App3	Viber	5.0.2	F	F	N	F	N	F	N	N	F	N
App4	Kik	8.0.0.1	F	N	N	N	N	F	F	N	F	N
App5	imo	–	F	F	N	F	N	F	F	N	N	N
App6	Hangout	–	F	F	N	N	N	F	F	N	F	N
App7	Telegram	2.5.2	F	N	N	N	N	F	N	N	F	N
App8	Kakao	4.7.6	F	F	N	N	N	P	N	N	F	N
App9	Tango	–	F	F	N	N	N	N	N	N	N	N
App10	ICQ	5.12	F	F	N	F	N	F	F	N	F	N
App11	Azar	2.6.6	F	F	N	N	N	N	F	N	F	N
App12	Fring	4.5.2.2	F	F	N	F	N	P	N	N	N	N
App13	Talkray	2.4.4	F	F	N	F	N	P	N	N	F	N
App14	Text Secure	2.6.4	F	N	N	F	N	N	N	N	N	N
App15	Hike	3.8.0	F	F	N	F	N	P	N	N	F	N
App16	Nimbuzz	3.5.1	F	F	N	F	N	F	F	N	N	N
App17	Voxofon	–	F	F	N	P	N	N	N	N	N	N
App18	Yahoo	1.8.8	F	F	N	N	N	F	F	N	F	N
App19	Skype	5.2.0	F	F	N	N	N	F	F	N	N	N
App20	Voxer	2.5.1	F	F	N	N	N	N	F	N	F	N
App21	Free PP	3.6.5	F	F	N	F	N	F	F	N	N	N
App22	Bigo	1.3.2	F	F	N	F	N	F	N	N	F	N
App23	U&Me	1.3.31	F	F	N	F	N	F	N	F	F	N
App24	Mypeople	4.8.4	F	F	N	F	N	F	N	N	F	N
App25	4talk	2.0.69	F	F	N	F	N	N	N	N	F	N
App26	Tap to Talk	3.0.3	N	N	F	N	N	N	N	N	N	N
App27	Alexicom AAC	1.1.1	N	N	F	N	N	N	N	N	N	N
App28	AAC Speech	1.5	N	N	F	N	N	N	N	N	N	N
App29	LetMe Talk	1.3.5	N	N	F	N	N	N	N	N	N	N
App30	WhatsApp	2.12.5	F	F	N	F	N	F	F	N	F	N

"F", detailed information was recovered; "P", only partial information was recovered (e.g., artifacts from some apps provided partial timestamp such as only the date of the activity rather than the time in hours, minutes, and seconds); "N", unsupported category.

- *View unencrypted contents of a group chat*: Group chats are similar to IM where more than two people are involved. A user can create a group and broadcast messages to the group.
- *View hidden chats*: Some apps allow the user to have hidden chats.

Timestamps: This group of artifacts identifies the timestamp of a communication.

- *Determine when add request was sent to/from a contact*: This is the timestamp when a friend request was sent to or received from someone from the device.
- *Determine when add requested was accepted by a contact*: This is the timestamp when a friend request was accepted by a contact using the device.
- *Determine when text/multimedia message was sent*: The timestamp of the sent text or multimedia (e.g., image) message.
- *Determine when text/multimedia message was received/delivered*: The timestamp when the text or multimedia (e.g., image) message was received or delivered.
- *Time when voice call was made*: This identifies the timestamp of a voice call.

Others: The remaining three artifacts are considered under this category.

- *Extract Databases*: Android apps typically generate their own databases in the internal device memory, where the latter is normally inaccessible by users. Information from these databases could be used to locate useful user information.
- *Identify the members of a group chat*: The group members are important to an investigator to find out who were sending/receiving messages to/from a particular group.
- *Identify the duration of voice calls*: This identifies the duration of a voice call.

In the proposed taxonomy, the communication app categories are represented in one dimension, and the forensic artifact categories are represented in the other dimension (Fig. 12). The findings of the 30 case study apps using the two-dimensional taxonomy is summarized in Table 11.

Concluding Remarks

The increasing use of Android apps and the capability of apps to access and store sensitive and personally identifiable

Artifact Categories												
Exchanged Messages					Timestamps					Others		
Determine Type of Exchanged Messages (Text, Voice, Group)	View Unencrypted Exchanged Text Messages	View Unencrypted Exchanged Multimedia Images	View Unencrypted Contents of a Group Chat	View Hidden Chats	Determine When Add Request was Sent to/from Contact	Determine When Add Request was Accepted by a Contact	Determine When Text/ Multimedia Message was Sent	Determine When Text/ Multimedia Message was Received/ Delivered	Identify Time When Voice Call was Made	Extract Databases	Identify the Members of a Group Chat	Identify the Duration of Voice Calls
F	F	F	F	F	F	F	F	F	F	F	F	F
F	F	F	F	N	F	F	F	F	F	F	F	N
F	F	F	F	N	F	N	F	N	F	F	F	F
P	F	F	F	N	N	N	F	F	N	F	F	N
N	N	P	N	N	N	N	N	N	N	F	N	N
F	F	F	F	N	N	N	F	F	F	F	F	N
N	N	F	N	N	N	N	N	N	N	P	N	N
F	P	P	P	P	N	N	F	F	F	F	N	N
N	N	F	N	N	P	P	N	N	N	F	N	N
F	F	F	F	N	N	N	F	N	F	F	F	N
N	F	N	N	N	N	N	F	N	F	F	N	F
F	F	N	N	N	N	N	F	N	F	F	P	F
F	F	F	F	N	N	N	F	N	F	F	F	N
F	N	N	N	N	N	N	F	N	N	F	F	N
F	F	F	F	F	N	N	F	N	N	F	F	N
F	F	N	F	N	N	N	F	N	F	F	F	F
F	F	N	N	N	N	N	F	N	F	F	N	N
P	F	F	N	N	F	F	F	N	N	F	N	N
F	F	F	F	N	N	N	F	N	F	F	F	N
F	F	F	F	N	N	N	F	N	F	F	F	N
F	F	F	F	N	N	N	F	N	F	F	F	F
F	F	F	F	N	N	N	F	N	F	F	F	N
F	F	F	F	N	N	N	F	N	F	F	F	N
F	F	F	F	N	N	N	F	N	N	F	F	N
N	N	N	N	N	N	N	N	N	N	F	N	N
N	N	N	N	N	N	N	N	N	N	N	N	N
F	F	N	N	N	N	N	N	N	N	F	N	N
N	N	N	N	N	N	N	N	N	N	F	N	N
F	F	F	F	N	N	N	N	F	F	F	F	F

information (PII) have resulted in an increasing need for the forensic research and practitioner communities to have an up-to-date and in-depth understanding of the types of terrestrial artifacts that are likely to remain on the devices. In this paper, 30 popular communication apps downloaded from Google Play store were analyzed. The findings of the analysis are summarized in Table 12.

Based on the analysis, a forensic taxonomy was presented. The taxonomy outlines the various data of forensic interest that could be recovered from the examination of different categories of communication apps. Such information will allow forensic practitioners to act and secure such data in a timely fashion. For example, based on the findings, investigators will be able to compile a list of contacts, determine when a contact was added, find out the blocked/hidden/deleted contacts, and reconstruct the chronology of exchanged text and multimedia messages.

Findings are accurate at the time of this research, but new releases of communication apps may change the way data are stored on the devices, as well as the type of data that can be forensically recovered from the devices. Therefore, future work would include examining other and new releases of communication apps and potentially include additional artifact categories to the proposed taxonomy. Future work would also include examining user activities such as deletion/uninstallation of communication and other apps to determine whether any data remnants could be recovered.

Acknowledgments

The authors would also like to 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.

References

1. Chang YF, Chen CS, Zhou H. Smart phone for mobile commerce. Comput Stand Inter 2009;31(4):740–7.
2. Azfar A, Choo K-KR, Liu L. A study of ten popular Android mobile Voip applications: are the communications encrypted? Proceedings of the 47th Annual Hawaii International Conference on System Sciences (HICSS); 2014 Jan 6–9; Waikoloa, HI. Piscataway, NJ: Institute of Electrical and Electronics Engineers, 2014;4858–67.
3. Jung J, Kim Y, Chan-Olmsted S. Measuring usage concentration of smartphone applications: selective repertoire in a marketplace of choices. Mobile Media Commun 2014;2(3):352–68.

4. Bicchierai LF. Iran prosecutor really wants president to block Viber, WhatsApp; http://mashable.com/2014/09/23/iran-prosecutor-wants-rouhani-to-block-viber-whatsapp/ (accessed January 6, 2015).
5. Yin C. Court officials warn of crimes using WeChat app; http://english.people.com.cn/90882/8318954.html (accessed January 19, 2015).
6. Casey E, Ferraro M, Nguyen L. Investigation delayed is justice denied: proposals for expediting forensic examinations of digital evidence. J Forensic Sci 2009;54(6):1353–64.
7. Karie NM, Venter HS. Toward a general ontology for digital forensic disciplines. J Forensic Sci 2014;59(5):1231–41.
8. Keith MJ, Babb J, Lowry PB. A longitudinal study of information privacy on mobile devices. Proceedings of the 47th Hawaii International Conference on Systems Sciences (HICSS); 2014 Jan 6–9; Waikoloa, HI. Piscataway, NJ: Institute of Electrical and Electronics Engineers, 2014;3149–58.
9. Barmpatsalou K, Damopoulos D, Kambourakis G, Katos V. A critical review of 7 years of mobile device forensics. Digit Invest 2013;10(4):323–49.
10. La Polla M, Martinelli F, Sgandurra D. A survey on security for mobile devices. IEEE Commun Surv Tutorials 2013;15(1):446–71.
11. Azfar A, Choo K-KR, Liu L. Forensic taxonomy of popular Android mHealth apps. Proceedings of the 21st Americas Conference on Information Systems (AMCIS); 2015 Aug 13–15; Fajardo, Puerto Rico. Atlanta, GA: Association for Information System, 2015.
12. Immanuel F, Martini B, Choo K-KR. Android cache taxonomy and forensic process. Proceedings of the 14th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom 2015); 2015 Augt 20–22; Helsinki, Finland. Hoboken, NJ: IEEE Computer Society Press, 2015;1094–101.
13. Plachkinova M, Andrés S, Chatterjee S. A taxonomy of mHealth apps–security and privacy concerns. Proceedings of the 48th Hawaii International Conference on System Sciences (HICSS); 2015 Jan 5–8; Kauai, HI. Piscataway, NJ: Institute of Electrical and Electronics Engineers, 2015;3187–96.
14. Alliano A, Herriger K, Koutsoftas AD, Bartolotta TE. A review of 21 iPad applications for augmentative and alternative communication purposes. Perspect Augment Altern Commun 2012;21(2):60–71.
15. Shariati M, Dehghantanha A, Martini B, Choo K-KR. Ubuntu one investigation: detecting evidences on client machines. In: Ko R, Choo KKR, editors. The cloud security ecosystem: technical, legal, business and management issues. Waltham, MA: Syngress, an Imprint of Elsevier, 2015;429–446.
16. Shariati M, Dehghantanha A, Choo K-KR. SugarSync forensic analysis. Aust J Forensic Sci 2016;48:95–117. DOI:10.1080/00450618.2015.1021379
17. Farnden J, Martini B, Choo K-KR. Privacy risks in mobile dating apps. Proceedings of the 21st Americas Conference on Information Systems (AMCIS); 2015 Aug 13–15; Fajardo, Puerto Rico. Atlanta, GA: Association for Information Systems, 2015.
18. Martini B, Do Q, Choo K-KR. Mobile cloud forensics: an analysis of seven popular Android apps. In: Ko R, Choo K-KR, editors. The cloud security ecosystem: technical, legal, business and management issues. Waltham, MA: Syngress, an Imprint of Elsevier, 2015;309–345.
19. Quick D, Martini B, Choo K-KR. Cloud storage forensics. Waltham, MA: Syngress, an Imprint of Elsevier, 2013;8.
20. Quick D, Choo K-KR. Google drive: forensic analysis of data remnants. J Netw Comput Appl 2014;40:179–93.
21. Martini B, Choo K-KR. Cloud storage forensics: own cloud as a case study. Digit Invest 2013;10(4):287–99.
22. Anglano C. Forensic analysis of WhatsApp messenger on Android smartphones. Digit Invest 2014;11(3):201–13.
23. Thakur NS. Forensic analysis of WhatsApp on Android smartphones [Master's thesis]. New Orleans, LA: University of New Orleans, 2013.
24. Mahajan A, Dahiya M, Sanghvi H. Forensic analysis of instant messenger applications on Android devices. Int J Comput Appl 2013;68(8):38–44.
25. Lee C, Chung M. Digital forensic analysis on Window 8 Style UI instant messenger applications. In: Park JJ, Stojmenovic I, Jeong HY, Yi G, editors. Computer science and its applications. Berlin Heidelberg: Springer, 2015;1037–42.
26. Husain M, Sridhar R. iForensics: forensic analysis of instant messaging on smart phones. In: Goel S, editor. Digital forensics and cyber crime. Berlin Heidelberg: Springer, 2010;9–18.
27. Chu HC, Lo CH, Chao HC. The disclosure of an Android smartphone's digital footprint respecting the instant messaging utilizing Skype and MSN. Electron Commer Res 2013;13(3):399–410.
28. Chu HC, Yang SW, Wang SJ, Park JH. The partial digital evidence disclosure in respect to the instant messaging embedded in Viber application regarding an Android smart phone. In: Park JH, Kim J, Zou D, Lee YS, editors. Information technology convergence, secure and trust computing, and data management. Dordrecht, The Netherlands: Springer, 2012;171–8.
29. Gao F, Zhang Y. Analysis of WeChat on IPhone. Proceedings of the 2nd International Symposium on Computer, Communication, Control and Automation; 2013 Dec 1–2; Singapore, Singapore. Paris, France: Atlantis Press, 2013;278–81.
30. Google. Google Play; https://play.google.com/store/apps (accessed February 2, 2015).
31. Meeker M. Internet trends 2014 – code conference Kleiner Perkins Caufield Byers; http://www.kpcb.com/internet-trends (accessed January 10, 2015).
32. Retrieving data from Android OS devices Using XRY. Patrick Leahy Center for Digital Investigation (LCDI); www.champlain.edu/Documents/LCDI/Android_OS_Tutorial_Final_PDF.pdf (accessed September 4, 2015).
33. Leom MD, D'Orazio C, Deegan G, Choo K-KR. Forensic collection and analysis of thumbnails in Android. Proceedings of the 14th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom 2015); 2015 Aug 20–22; Helsinki, Finland. Hoboken, NJ: IEEE Computer Society Press, 2015;1059–66.
34. Higginbotham J, Jacobs S. The future of the Android Operating System for augmentative and alternative communication. Perspect Augment Altern Commun 2011;20(2):52–6.

Additional information and reprint requests:
Kim-Kwang Raymond Choo, Ph.D.
University of Texas at San Antonio
Department of Information Systems and Cyber Security
One UTSA Circle — San Antonio
TX 78249-0631
USA
E-mail: raymond.choo@fulbrightmail.org