Chapter 1

A FRAMEWORK TO DESCRIBE MULTIMEDIA CIRCULATION IN THE SMARTPHONE ECOSYSTEM

Panagiotis Andriotis, Theo Tryfonas, George Oikonomou and Irwin King

Abstract Contemporary mobile devices allow almost unrestricted sharing of multimedia and other types of files. But as smartphones and tablets can easily access the Internet or exchange files wirelessly, they've also transformed to useful tools for criminals, aiming at performing illegal activities such as sharing contraband or distributing child abuse images. Thus, the need to investigate the source and destination of a multimedia file that resides in the internal memory of a smartphone becomes apparent. In this paper we present a framework that illustrates and visualizes the flow of digital images as evidence obtained from the artefacts retrieved from Android smartphones during a forensic investigation. Our approach uses 'big data' concepts to facilitate the processing of diverse (semi-structured) evidence derived from mobile devices and extends the idea of Digital Evidence Bags (DEB). We obtained our data after running an experiment that included image exchanging through numerous channels such as Bluetooth, Internet and cloud services. Our study presents information about the locations where evidence resides and uses graph databases to store metadata and therefore, visualize the relationships that connect images with apps and events.

Keywords: Android, Forensics, Graph, Database, Content, Analysis, NoSQL

1. Introduction

The proliferation of smartphones in modern societies and the fast mobile telephony networks offer countless opportunities to their owners to exchange text messages, photos, videos and multimedia content in general. Unfortunately these smart applications, which are equipped with convenient interfaces allowing smooth and rapid flow of information, can

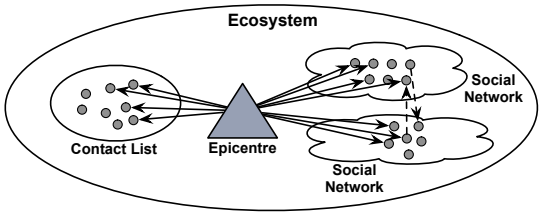


Figure 1: A Smartphone Ecosystem Representation.

also become tools in the hands of criminals who commit crimes such as child pornography sharing.

Smartphones can be held as evidence and used in courts to prove the innocence of defendants in many countries. In addition, smartphones are considered to hold valuable types of evidence because they contain a large amount of personal information, as they are fully integrated with peoples' lifestyle. These devices are equipped with a variety of sensors providing the capability to a developer to build smart applications (apps). They also became very efficient considering their processing power and the accuracy of their sensor measurements. Users are able to connect to the Internet, upload and download material through cloud services, capture images, sound and videos, monitor their health, create and edit documents and spreadsheets and get personalized information from various sources based on their location and interests. All these actions leave artefacts that could be mined and used in court.

Using the metaphor of an ecosystem, a smartphone can be viewed as defining a unique ecosystem in itself, where the owner of the device can be seen as a central entity, the epicenter. The ecosystem consists of smaller groups that include entities linked with one another with various relationship types. For example, the contact list of the phone defines an area where the analyst can find people that are connected with the epicenter. Another entity group might include those accounts that constitute the friends of a social network or the contacts of a professional network like *LinkedIn*. One of the problems a forensic analyst has to solve is which entity is a real person or represents a fake, a secondary or a parody account. In addition, a forensic analysis on a smartphone can reveal which entities (from different groups) are linked together, if we assume that the owner of the phone has used the contact syncing utility that most of the social network apps provide.

A representation of such an ecosystem can be seen in figure 1. Here the epicenter is linked with the 'contact list' group (neighborhood) and with two other social networks. Some of the entities in the ecosystem are

also linked together (dash line) through the automatic contact linking process. During a forensic analysis these different 'neighborhoods' can be linked via the artefacts we can find in various databases in the internal memory of the phone. In Android smartphones for example, the applications store data in specific folders in the data partition [15]. Most of the application folders consist of at least three directories (cache, databases, lib, shared_preferences, files). The 'databases' folder is usually the place where information about the specific user is kept and stored in SQLite databases. All this data is useful in order to reconstruct the profile and activities of the epicenter. Furthermore, a forensic analysis in general can be enhanced using reporting tools that provide visual metaphors of the underlying data [13].

The metaphor serves as a scheme that distributes diverse data (existing in the internal memory of the phone) in distinct categories such as entities, groups, multimedia. It also focuses on the relationships the epicenter has with them. But despite the practicality of the aforementioned representations of social connections, these schemes have disadvantages too. They do not provide any further information about events that are linked to entities. For example, if a forensic analyst wanted to visualize which entity in the ecosystem is responsible for capturing and distributing an illegal photo, figure 1 would not provide any useful hints, because the entities are not linked with actions they performed through the different apps. Consequently, we need to enhance such schemes with capabilities that will provide the means to link actions with entities.

In this paper we focus our interest on cases, which involve the circulation of photographs and digital images in general in the smartphone ecosystem. We aim to reconstruct a network that will be able to depict the multimedia distribution in the environment ('ecosystem') defined by an Android mobile device. Our main contributions are as follows:

- We propose a framework that solves the problem of linking entities with events and digital artefacts during a forensic analysis on smartphones, highlighting the relationships between apps and multimedia. This approach uses 'big data' concepts and extends the idea of digital evidence bags, utilizing modern storing methods and focusing on the ecosystem epicenter and its actions.

- We augment the capability of storing information about a person under investigation with visualizations of the interactions that happen in their ecosystem using graph databases. Our conceptual design can be easily extended to cover all aspects of evidence that can be found in a smartphone. In addition, the proposed scheme

is able to represent more than one ecosystems-smartphones linked with a case.

2. Related Work

Despite the use of Internet was not as extensive as it is today, the importance of distinguishing traces that reveal potential access to illegal content has been highlighted in the past by various researchers. For example, Howard in [1] presents a technical analysis of the cache and the various methods forensic analysts use to disclose data stored in the Temporary Internet Files of a browser, in order to approach the problem of prosecuting possession of Child Pornography (CP).

More recently, the expansion and popularity of peer-to-peer (P2P) networks allowed for seamless information flow transforming these digital neighborhoods to blooming areas of illegal image and video trafficking. Hurley et al. [4] measure and analyze multimedia trafficking in two popular P2P networks ('eMule' and 'Gnutella') and study various subgroups such as: a) peers that use 'Tor' or b) those who bridge multiple P2P networks or c) those who contribute to file availability. They conclude that these groups are more active with respect to CP trafficking. Wolak et al. [9] also examine data from 'Gnutella' using the investigative 'RoundUp' tool [3] and propose that data should be systematically gathered and analyzed to prioritize investigations in P2P networks. However, these tools cannot be applied in smartphone examinations and therefore they are not able to reconstruct and present the exchange of information via different apps.

Traditional digital image forensics [8] can be employed to achieve tasks like device identification and linking, or detection of digital forgeries. Despite the plethora of known anti-forensics frameworks [5] that aim to misguide forensic algorithms, there exist sources of information able to provide indications (such as sensor data) that can be used during a forensic analysis on smartphones [11]. However, this data is usually volatile, thus not accessible during a post-mortem analysis.

Non-volatile data in Android devices can be found internally and some novel proactive approaches to automatically collect and analyze them have been proposed recently, with a special focus on sensitive enterprise environments [6]. The authors in [2] propose in their study a Machine Learning-based Triage scheme to automate digital media categorization, merging Digital Forensics with Machine Learning. In addition, Liu et al. [10] use Support Vector Machines to identify the smartphone camera source of digital images and reveal possible operations that might have been applied on them. Turner [17] presented his approach to unify digital

evidence from disparate sources using digital evidence bags (DEB). DEB is a universal container for capturing and processing digital evidence from different sources. The existence of diverse file formats for digital evidence preservation was highlighted in [19] and various solutions have been proposed to overcome this obstacle. For example, Garfinkel [18] presented the DFXML scheme to empower the exchange of structured forensic data from different sources.

Forensics researchers have studied the plurality of diverse data that occur from different sources existing in the smartphone ecosystem. Chung et al. [7] investigate and analyze artefacts from various sources (desktop machines and mobile devices) connected with cloud storage services, revealing traces of activity in file paths, xml files and databases. Huber et al.[12] collect and categorize information from social media networks including user data, posts, private messages, photos and associated metadata and Kontaxis et al. [16] use such information to detect if there exist clone profiles in various social networks. Anglano [14] presents an analysis of the traces that were left from the use of the popular chatting app *WhatsApp Messenger* on Android devices. However, there is no particular work in the literature describing the circulation of images (or multimedia) in the smartphone ecosystem. Our work here integrates all these valuable data a forensic analysis can provide to construct an extensible scheme to analyze and visualize them.

3. Using Graph Databases

Forensic investigations on smartphones consist of 4 basic steps; 1) Phone seizure, 2) (physical or logical) data acquisition, 3) data analysis, 4) data presentation and preservation. Our study focuses on the data presentation step and proposes a methodology to automate the forensic investigation considering the photo trafficking in the ecosystem. We are particularly interested in images that entered or exited the ecosystem through various paths. These paths include: a) Wireless technologies, such as Bluetooth, Wi-Fi Direct and Near Field Communication (NFC), b) Emails, c) Downloads from the Internet, d) Cloud storage services, e) Applications (Messaging and Other Apps, e.g. *Facebook Messenger*, *Twitter*).

The mapping method we propose highlights the relationships that link photos and their sources or destinations. For example, the most obvious relation between a JPEG image and the *Camera* application would show if the photo was captured using the app or if it was downloaded or distributed through another app. This information is critical to the

analyst to determine if a person under investigation produces or just distributes illegal content.

In modern mobile devices the storage capacity is constantly improving and the possibility to find a large amount of information while we are performing a forensic analysis is high. As a result, we need a medium (such as a database) where we can safely migrate and store all the acquired data and its relationships for further examination. Relational databases are based on traditional storing models and in most of the cases require very complicated design strategies to depict relations using foreign keys, joins and tables that act as reference points, linking different entities. However, in the social media and 'big data' era, new types of databases have been introduced. NoSQL and graph databases are typical examples of non-relational models and some (among a plethora) widely used storage systems are: 'Cassandra', 'MongoDB' and 'Neo4j'. The strongest advantage of a graph database, compared to relational databases, is that it has the ability to easily handle and depict relationships that link entities.

Our study investigates relations that occur between linked entities. If, for example, we want to search about all these images that were downloaded from the Internet, we are targeting connections-relationships with the attribute 'DOWNLOADED'. Thus, a graph database is the most appropriate storage medium for this study because it has the ability to search for patterns inside paths created by nodes, which are connected together. In a graph database we can store nodes and connect them together using different attributes that characterize the type of relationship these nodes have. Hence, a graph database is a graph that is expandable and can store different kinds of information. If a forensic investigator aims to store and analyze photos (but also videos, music or sound clips and documents) the graph database can hold all this new information without the need for database refactoring.

4. Use Cases - Experiments

In order to reveal any traces the use of various applications leave in the internal memory of the mobile device, we designed a scenario that involves numerous user activities. The phone we used was a Samsung Galaxy Fame GT-6810P equipped with an external Secure Digital (SD) card and Super User privileges (su). The phone was running the Android Operating System (version 4.1).

The actions we performed to simulate photo exchanging and trafficking in the smartphone ecosystem are as follows: a) Images were emailed to the user's email account. The user read and downloaded them in

the device. We used the standard Android *Email* and the *Gmail* apps. b) Pictures were taken (and stored) using the phone's camera. c) Pictures were exchanged using specific wireless interfaces (Bluetooth and NFC). d) Applications like *Snapchat* (chatting through image exchanging) were used. We also experimented with the popular *Facebook Messenger*, *WhatsApp* and *Google Hangouts* apps. e) Images were uploaded and downloaded from the phone via cloud services (*Dropbox* and *Google Drive* apps). f) JPEG images were downloaded to the phone via applications like *Twitter* and *Instagram*. g) Digital images were downloaded from the Internet using the standard Android *Internet* browser and *Google Chrome*, which is also shipped with most of the recent versions of the Operating System (OS).

After the execution of our scenario, we gathered the data doing a physical acquisition of the phone's data partition. For this process, the USB Debugging option on the phone was enabled and the Android Debug Bridge (adb) tool was utilized. This is a common way to extract the physical image of the phone's partitions as explained in [20]. The method of physical data acquisition provides the possibility to find deleted images using open source tools like 'scalpel'. Deleted images can be signed in our graph database as nodes connected to the epicenter with relationships flagged as 'DELETED'.

5. Results

This section discusses the locations (in the form of lists of folders) where we can find information about the circulation of the images in the ecosystem. It also describes the traces that were found in the internal memory of the phone after our scenario execution. We are mostly interested in the data folder (data/data/) of the phone, where most applications store locally significant amount of information. We should also note that these databases are not encrypted and an individual with super user privileges can access and view them in an open source SQLite browser like the 'sqliteman' tool. (Note that '[SQLite]' in figure 1 denotes an SQLite3 database.)

The Android OS is installed in a large number and variety of devices with different characteristics. A smartphone for example might be equipped with external disk storage (an SD or microSD), internal emulated storage or both. Our reference phone had two storage folders: emulated and external SD. When an SD card is inserted to the phone, the system stores usually any photos captured by the *Camera* app in the SD card. A logical copy of the folders in the external or emulated media will reveal only the photos that can be seen by the operating

Table 1: Resources Providing Information about Image Sharing.

Path	**Type**	**Details**
com.android.bluetooth/ /databases/btopp.db [SQLite]	Sent, received, deleted.	At table 'btopp', see 'uri' & 'direction'.
com.android.email/ /cache/[folder_e.g._1.db_att)]	Received, attached (via *Email* app).	Might not be visible via *Gallery* app.
com.android.email/databases/ /EmailProvider.db [SQLite]	Sent, received, downloaded.	At 'attachment' and 'message' tables.
com.android.providers.downloads/ /databases/downloads.db [SQLite]	Downloaded.	Downloaded (Internet) & chat apps (e.g. *Hangouts*).
com.dropbox.android/ databases/db.db [SQLite]	Deleted, uploaded.	Tables 'dropbox' 'photos'.
com.facebook.orca/cache/fb-temp/	Uploaded.	Uploaded content.
com.facebook.orca/cache/image/ /v2.ols100.1/[folders]/	Sent, received, seen via *Gallery* app.	Thumbnails from *Gallery* (if accessed by *Messenger*).
com.google.android.apps.docs/cache/ /diskCache/fetching/account_cache_1/	Uploaded, downloaded.	Images residing in *Google Docs* app.
com.google.android.apps.docs/ /databases/DocList.db [SQLite]	Deleted, data 'owners'.	Tables like 'entry111'.
com.google.android.apps.docs/ /files/fileinternal/[folders]/	Downloaded, 'pinned'.	'Pinned' images to be viewed offline.
com.google.android.gm/ /cache/ [user's_gmail_address]	Unknown, sent or received.	Images and other attachments.
com.google.android.talk/ /cache/scratch/	Sent.	Sent images via *Hangouts* app.
com.google.android.talk/ /databases/babel1.db [SQLite]	Sent, received.	At 'messages', attribute 'local_uri'.
com.instagram.android/cache/	Pending, captured, sent, received.	Names flagged with timestamps.
com.sec.android.gallery3d /databases/picasa.db [SQLite]	Various.	Table 'photos' (if auto-uploading is on).
com.sec.android.providers.downloads/ /databases/sisodownloads.db [SQLite]	Downloaded.	Downloaded via *Internet* browser.
com.snapchat.android/cache/ /received_image_snaps/	Received (.nomedia files)[1].	Might be encrypted (version depended).
com.snapchat.android/cache/ /stories/received/thumbnail/	Received (.nomedia files)[2].	Might be encrypted (version depended).
com.snapchat.android/ /databases/tcspahn.db	Various.	Entries about sent, received images.

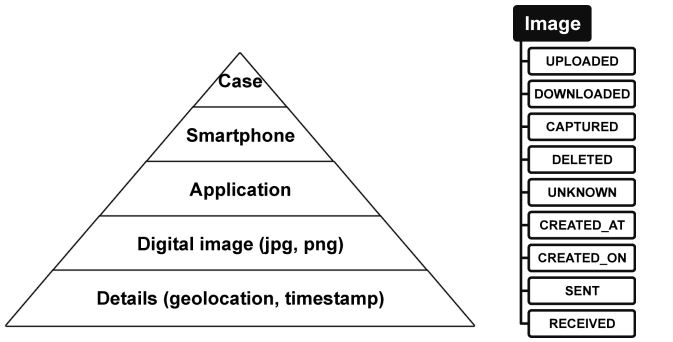
Table 2: Digital Images Stored in the External or Emulated Storage.

Path	**Type**
mnt/extSdCard/DCIM/Camera/	Photos captured from the *Camera* app.
mnt/sdcard0/Download/	Downloaded using *Chrome* browser.
mnt/extSdCard/Download/	Downloaded using *Internet* browser.
mnt/sdcard0/Pictures/Twitter/	Uploaded and downloaded via *Twitter*.
mnt/sdcard0/Pictures/Facebook/	Captured and Uploaded via *Facebook*.
mnt/sdcard0/Pictures/Messenger/	Downloaded from *Messenger*.
mnt/sdcard0/Beam/ or /sdcard0/Bluetooth/	Exchanged through wireless media.
mnt/sdcard0/Snapchat/	Downloaded photos from *Snapchat*.
mnt/sdcard0/WhatsApp/Media/ /WhatsApp Images/	Received and sent images via *WhatsApp*.
mnt/sdcard0/Android/data/	Folders containing photos from distinct apps.

system. However, these photos might contain important information in their *Exif* metadata headers, such as the user's location (if the GPS facility was enabled, when the photo was captured). We can obtain a logical copy of the storage media using the 'pull' command of the 'adb' tool when the phone is connected with a computer via the USB cable. This data enhances our scheme with more information about the type of connections that link the nodes of our graph. Finally, the analyst will be able to find digital images in the folders highlighted in figure 2.

6. System Design

A graph database, as already mentioned, is a graph and a database at the same time. Entities, photos and applications can be represented as nodes in the graph. This feature makes the whole system expandable and able to hold diverse types of information in the future (e.g. videos, sound recordings or other documents). In our scheme, each case is represented as a node in the graph, because in the future we might want to link different cases with unique ecosystems. For example, a person under investigation might be related with more than one case. In the graph database, nodes are connected together with relationships like 'DOWNLOADED', 'UPLOADED', 'DELETED' and contain information from the original SQLite databases. Both nodes and relationships contain



(a) Hierarchical Node Structure.

(b) Main Relationship Tags for Images.

Figure 2: Graph Database Conceptual Design.

attributes-properties that can be used by the analyst to find patterns and paths inside the graph, performing SQL-like queries.

In this work we use the 'Neo4j' graph database[3] and its 'Cypher' query language. Cypher is a graph database language that uses ASCII art expressions and only a limited number of commands to perform queries to the graph. An example of the ASCII art writing style used by Cypher is the following:

(a)->[:UPLOADED]-(b)-[:DELETED]->(c)

In this case (a), (b) and (c) represent nodes and [:UPLOADED], [:DOWNLOADED] represent relationships. Thus, (a) is linked with (b) with the [:UPLOADED] relationship which points from (a) to (b) and so on. In a large graph the output of a query can be shown either graphically, through a browser, or with the traditional form of a table. Such schemes (graph databases) would be beneficial to a forensic analysis because they provide the opportunity to visualize data and at the same time the analyst can search for patterns and paths in the ecosystem using a common infrastructure (which is the graph database itself).

Figure 2a presents the conceptual hierarchical design of a system that incorporates the findings of our case study presented at Section 5. The graph consists of nodes that represent: i) a case, ii) a seized smartphone (linked with a case), iii) applications that exist in the smartphone's ecosystem, iv) photos and images found in the cache and other storage media and, finally, v) other important details like geolocation and timestamps. The decision to depict geolocation and timestamps as distinct nodes and not as node attributes was made because of the importance of

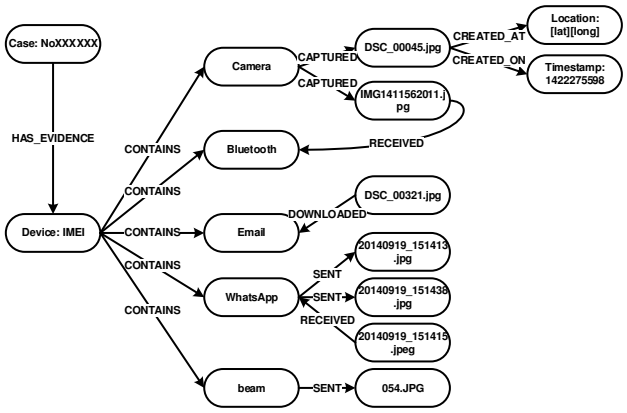


Figure 3: Basic Nodes and their Relationships in our Graph Database.

such information and also, because we might want to link more actions in the future with specific timestamps and locations. Figure 2b illustrates the main relationships that link images with other entities of the graph. 'CAPTURED' is related to the *Exif* metadata and discloses the camera type (or the smartphone) that was used to capture the photo. 'CREATED_AT' is a relationship that links photos and locations and 'CREATED_ON' connects photos with timestamps. Figure 3 shows an example that contains the most critical entities and their relationships in the graph database.

One of the advantages of this system is that it can be extended easily to include new nodes with various files like videos, sound recordings, and documents. In addition, it allows for integration with other ecosystems that might exist in a case. We can also add nodes that represent entities like social media accounts or people from the contact list. Thus, we can link data and evidence found in different devices, which are involved in the same case. Another advantage is that the proposed framework is a graph; hence, we can apply metrics used for graphs to extract information that describe our data. As an example, we can use metrics like *degree*, *indegree* and *outdegree* for a node that represents an application (e.g. *Facebook Messenger*), describing the usage frequency of the app and the incoming and outgoing digital evidence, respectively. In a larger scale these metrics will provide information about which was the most preferred application for image exchanging. Using the timestamp nodes we can also focus on different periods of time and highlight possible alterations in the user's behavior. Finally, data from Neo4j graph databases can be easily stored as GraphML files. GraphML files can feed dedicated graph visualization tools, which are equipped with functions

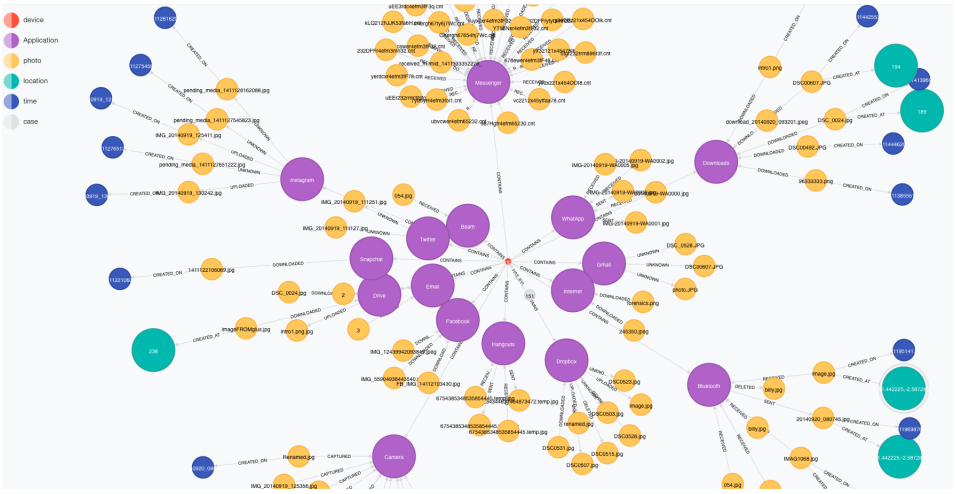


Figure 4: A Screenshot from our Experimental Graph Database.

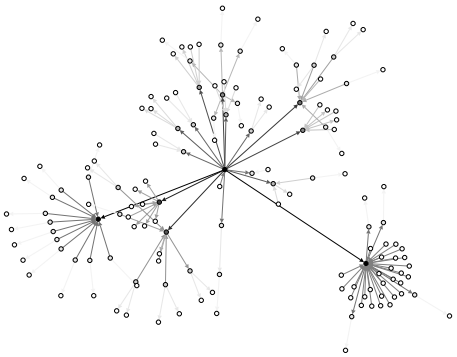


Figure 5: The Graph Analyzed with 'Gephi' Highlighting the Most 'Busy' Nodes.

that analyze the graph in more detail. Thus, our scheme provides the possibility for further analysis of the data.

Figure 4 depicts the emerged graph database after the execution of our scenario. The analyst has to establish a connection with *http://localhost:7474/browser* when the database server is running and type the command "MATCH (n) RETURN n" in order to be able to see the graph. Figure 5 shows an example of further analysis of the graph using an open source visualization tool[4]. We first extracted the graph as a GraphML file and then we plotted the nodes according to their degree to easily show which app was mostly used for digital image sharing.

The graph database is able to provide information about data sharing in a more concise way, projecting results (degree, indegree, outdegree) on tables in order to observe the data circulation in the ecosystem. For this reason we provide some formal definitions. We define $\mathcal{R}$ as the set of relationships that link apps with photos: $\mathcal{R}$ = {RECEIVED, SENT, DOWNLOADED, CAPTURED, UNKNOWN, DELETED, UPLOADED}. Also we define $\mathcal{I}$ as the incoming relationships and $\mathcal{O}$ the outgoing, thus $\mathcal{I}$ = {RECEIVED, DOWNLOADED, CAPTURED} and $\mathcal{O}$ = {SENT, UNKNOWN, DELETED, UPLOADED}. The set of apps-sources in our ecosystem is defined as: $\mathcal{A}$ = {Bluetooth, Beam, Email, Downloads, Messenger, Drive, Gmail, Hangouts, Instagram, Dropbox, Snapchat, Camera, Facebook, Twitter, WhatsApp}. Generally, we define $\mathcal{A}$ as the set $\mathcal{A} = A_i$, i = 1, 2, ..., n, where n is the number of different apps in the ecosystem. Subsequently, if m is the number of different relationships that link app A_i, then we represent these various relationships as ${r^{(i)}}_j$, j = 1, 2, ..., m and ${r^{(i)}}_j \in \mathcal{A}$. For simplicity we set ${r^{(i)}}_j$ as:

$$ {r^{(i)}}_j = \begin{cases} \kappa & \text{if } {r^{(i)}}_j \in \mathcal{I} \\ \lambda & \text{if } {r^{(i)}}_j \in \mathcal{O} \end{cases} \tag{1} $$

Thus, $indegree = \sum_{i=1} \kappa, outdegree = \sum_{j=1} \lambda$ and

$$ degree = indegree + outdegree \tag{2} $$

After the execution of our scenario the graph database provides the following results (table 3) using equation 2.

Table 3 informs that *Messenger* stored more information compared to other apps. Also, the user of the phone exchanged images using Bluetooth and NFC (*Beam*) ($indegree \neq 0$). In addition, we can derive other information related to the use of chatting apps for digital image exchanging (*WhatsApp*, *Hangouts*). For example, the person under investigation cannot claim that no photo was left the device via *WhatsApp* because the evidence shows that outdegree for *WhatsApp* equals 2. This means that our graph indicates that there were at least 2 images that were sent via *WhatsApp* to another ecosystem. Additionally, if the 'image' nodes are connected to 'time' nodes, we will be able to see when these transactions happened.

7. Conclusion

This paper investigated a novel approach to capture and analyse the flow of photos in the ecosystem defined by a smartphone. We are using information that can be collected from the existing SQLite databases in the cache and the data partition of Android phones (and tablets) during

Table 3: Number of Relationships Among Photos and Apps.

App	Indegree	Outdegree	Degree
Bluetooth	5	2	7
Beam	1	0	1
Email	2	0	2
Downloads	6	0	6
Messenger	28	4	32
Drive	2	1	3
Gmail	0	3	3
Hangouts	2	2	4
Instagram	0	5	5
Dropbox	2	6	8
Snapchat	1	0	1
Camera	11	0	11
Facebook	3	0	3
Twitter	0	2	2
WhatsApp	3	2	5

a forensic analysis. The migration of this data can be achieved using well-tested environments like the 'Neo4j' graph database. Its use allows for fast and accurate searches and has the advantage that creates results utilizing its pattern matching functions. The proposed methodology is extensible and can be adopted to provide big data functionality adding diverse, semi-structured data from various sources. Finally, we discovered somewhat unexpectedly that applications like *Facebook* have access to the smartphone's *Gallery* and they also keep copies of images that were present in the particular folder. This means that even if the user deleted the original photos from the *Gallery*, the cached files of the app can reveal the deleted content.

Our future work will take into consideration the advantages of graph databases we demonstrated in this paper and propose methods to accommodate various entities in the same graph. Entities can be fake or secondary social media accounts or they can be friends, colleagues or family members existing in the contact list of a seized phone. These entities are linked with actions and events that might produce additional evidence. For example, if the person under investigation exchanges a photo having embedded geolocation data via Bluetooth taken at a specific time, then the recipient's location will be also derived by the action. Thus, our proposed scheme can be extended to hold diverse data that produce additional information for other entities, which share actions with the person under investigation.

8. Acknowledgement

This work has been supported by the EU DG Home Affairs - ISEC (Prevention of and Fight against Crime) / INT (Illegal Use of Internet) programme and the Systems Centre of the University of Bristol.

Notes

1. [14] provides information about how to decode such images.
2. As above.
3. http://www.neo4j.org
4. Gephi: http://gephi.github.io

References

[1] T.E. Howard, Don't Cache out Your Case: Prosecuting Child Pornography Possession Laws Based on Images Located in Temporary Internet Files. *Berkeley Tech. LJ*, vol. 19, pp. 1227–1273. 2004.

[2] F. Marturana and S. Tacconi, A Machine Learning-based Triage methodology for automated categorization of digital media. *Digital Investigation*, vol. 10 (2), pp. 193–204, 2013.

[3] M. Liberatore, R. Erdely, T. Kerle, B.N. Levine and C. Shields, Forensic investigation of peer-to-peer file sharing networks. *Digital investigation*, vol. 7, S95–S103, 2010.

[4] R. Hurley, S. Prusty, H. Soroush, R.J. Walls, J. Albrecht, E. Cecchet, B.N. Levine, M. Liberatore, B. Lynn, and J. Wolak, Measurement and analysis of child pornography trafficking on P2P networks, *Proceedings of the 22nd international conference on World Wide Web*, pp. 631–642. International World Wide Web Conferences Steering Committee, 2013.

[5] M.C. Stamm and K.R. Liu, Anti-forensics of digital image compression, *IEEE Transactions on Information Forensics and Security*, vol. 6 (3), pp. 1050-1065, 2011.

[6] J. Grover, Android forensics: Automated data collection and reporting from a mobile device. *Digital Investigation*, vol. 10, pp. S12–S20, 2013.

[7] H. Chung, J. Park, S. Lee and C. Kang, Digital forensic investigation of cloud storage services. *Digital investigation*, vol. 9 (2), pp. 81–95, 2012.

[8] J. Fridrich, Digital image forensics, *IEEE Signal Processing Magazine*, vol, 26 (2), pp. 26–37, 2009.

[9] J. Wolak, M. Liberatore and B.N. Levine, Measuring a year of child pornography trafficking by US computers on a peer-to-peer network. *Child abuse & neglect*, vol. 38 (2), pp. 347–356, 2014.

[10] Q. Liu, X. Li, L. Chen, H. Cho, P.A. Cooper, Z. Chen, M. Qiao, and A.H. Sung, Identification of smartphone-image source and manipulation, in *Advanced Research in Applied Artificial Intelligence*, J. He, D. Wei, A. Moonis and W. Xindong (Eds), pp. 262–271. Springer Berlin Heidelberg, 2012.

[11] A. Mylonas, V. Meletiadis, L. Mitrou and D. Gritzalis, (2013). Smartphone sensor data as digital evidence, *Computers & Security*, vol. 38, pp. 51–75, 2013.

[12] M. Huber, M. Mulazzani, M. Leithner, S. Schrittwieser, G. Wondracek and E. Weippl, Social snapshots: Digital forensics for online social networks, *Proceedings of the Twenty-Seventh ACM Annual Computer Security Applications Conference*, pp. 113–122, 2011.

[13] S. Teelink and R.F. Erbacher, Improving the computer forensic analysis process through visualization. *Communications of the ACM*, vol. 49 (2), pp. 71–75, 2006.

[14] C. Anglano, Forensic analysis of WhatsApp Messenger on Android smartphones, *Digital Investigation*, vol. 11(3), pp. 201–213, 2014.

[15] A. Hoog, Android forensics: investigation, analysis and mobile security for Google Android, *Elsevier*, Waltham, MA, USA, 2011.

[16] G. Kontaxis, I. Polakis, S. Ioannidis and E.P. Markatos, Detecting social network profile cloning, *Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pp. 295–300, 2011.

[17] P. Turner, Unification of digital evidence from disparate sources (digital evidence bags). *Digital Investigation*, vol. 2 (3), pp. 223–228, 2005.

[18] S. Garfinkel, Digital forensics XML and the DFXML toolset. *Digital Investigation*, vol. 8 (3), pp. 161–174, 2012.

[19] A.O. Flaglien, A. Mallasvik, M. Mustorp and A. Årnes, Storage and exchange formats for digital evidence, *Digital Investigation* vol. 8 (2), pp. 122–128, 2011.

[20] P. Andriotis, G. Oikonomou and T. Tryfonas, Forensic analysis of wireless networking evidence of Android smartphones, *Proceedings of the WIFS'12 IEEE International Workshop on Information Forensics and Security*, pp. 109–114, 2012.