NUST Journal of Engineering Sciences, Vol. 10, No. 1, 2017, pp. 30-36

Forensic Investigation of Smartphone Cloud Storage Applications

Aiza Aqeel Abbasi[1*], Shahzad Saleem[1], Roha Zulqarnain[1]
[1]School of Electrical Engineering and Computer Science, National University of Sciences and Technology, Islamabad 44000, Pakistan
[*]14msisaabbasi@seecs.edu.pk
Received: Accepted:

Abstract

Advancement in technology allows people to access the data through smartphones regardless of the time and place. Because of widespread applications of users' interest, the dependency on the mobile devices has increased. Cloud storage applications are attracting user's attention rapidly and will continue enjoying this ever increasing popularity in the near future as well. This makes them an important potential container of evidence during the investigation. So, it is important for forensic practitioners to match their pace with technological advancements. This paper has addressed the above-mentioned problem as per NIST methodology; bit-by-bit image(s) of android phone is analyzed for exploring the containers for retrieving important artifacts of user activities. The study aims to possibly help the investigative process by scrutinizing cloud storage applications namely: Cubby and IDrive. As a result, interesting locations were identified from where security vulnerabilities and other short comings were exposed. Overall the study concludes that security of Cubby is far better than IDrive.
Keywords: cyber security, mobile forensics, storage applications, forensic investigators, cloud computing

Introduction

From 1973 to 2009, the telephonic technology evolved from 0G Mobile Radio Telephone to 4G cell phones. During this era, where the capacity and quality of technology has increased, an immense decrease is observed in the cost of a mobile phone. The reason for this can be named as no monopoly in the market due to tough competition between vendors. This triggered a boom in the use of mobile phones and now people highly depend on this for managing their daily affairs.

Another notable point is that despite having improvements in the power and storage capacity of mobile devices, the consumers are still using cloud storage applications. It provides them the ease of switching the mobile devices without worrying about their data but with this ease there come some disadvantages as well. One of the major drawback is that the applications are being used by every type of person including criminals and like-minded people. According to [1], data hidden in the form of incriminating files contain illegal drug, child abuse, and inappropriate material. [2]

It directs that exploring the cloud can give much needed material to the investigator. The most popular approach in the past, was to explore the server but due to jurisdictional limitations, such an approach is sometimes not viable. That is why usually it is not possible to physically access the server. The solution thus is to examine the client side of cloud. In the light of which this research work will examine cloud storage applications on client side i.e. the smartphones of criminals having cloud services installed on them. This process is a need of the hour. Initially, this was practiced by Hobson [3] but now other researchers are also turning towards this approach. Usually NIST [5] methodology is followed in this domain [1], [9], we have also followed the same.

As the storage is digital, it implies that the evidence will be in digital form. So, another concern with digital evidence is its forensic soundness because without this the evidence is not admissible in courts, no matter how strong the stance is. Forensic soundness is measured by checking whether the integrity of evidence is hurt or not? So, for digital evidence, it is a must to prove its integrity only then the court would accept it to accuse the guilty. Therefore, hashes like MD5 and SHA1 are being calculated by [1] [6] [7] [8] [9] and this research work will also follow it.

The target audience consists of end users, vendors, forensic practitioners and investigators. Goals of study are to help investigators and practitioners in learning; vendors in making their product's quality better and end users for wise selection of the applications according to their needs.

Objectives include to identify the relevant containers for better understanding of investigators, to provide a systematic approach for the practitioners, to expose the vulnerabilities and enlisting shortcomings of applications, plus, the comparative study which'll conclude the better choice for users at some given condition.

This research work will answer the following questions:

- Will investigator be able to extract the login credentials? Some or all? If yes, then what would be the location and format i.e. encrypted or plaintext etc.?
- Is it possible to retrieve the information about recently uploaded files i.e. Name, Size, Timestamp etc. or not?
- What happens if User has uploaded the information from any other device and browsed in phone? Will it make availability of item in phone?
- What if the file is deleted by the user? Examiner can trace its footprint to link it back or not?
- If a file was made available Offline will examiner be able to get it or not?
- If the user has shared files or folders with an individual, will examiners be able to find the other participants or not?
- In case of public availability of any file/folder, how much information can be retrieved?

From a wealth of similar application, the reason for selecting the listed applications are as follows:

a. Popularity among users – determined by number of downloads on Google Play Store.
b. Positive Feedback from users – determined by rating on Google Play Store.
c. No or insufficient research on these applications.
d. Availability in Pakistan

First step listed more than 10 applications. Moving on second helped finalizing 7 from them. Third step however reduced it to

only 5 applications. And last one shortlisted Cubby[16] and IDrive[17].

Other sections of the paper include section 2, where the literature review is discussed. The third section will define the methodology by NIST in context to this research. Section four will bear the investigative framework. The fifth section will present experiment setup. Sixth section will scrutinize all the applications one by one. In the seventh section, the comparative analysis will be done. Eighth section will contain conclusion. Ninth section will throw some light on the future directions for research in this domain.

Literature Review

The focus of this research is to perform forensics on the client side. Client side includes both the computer and mobile device forensics. The following section will first discuss a little about desktop cloud applications before moving on towards mobile device forensics of cloud storage applications.

Exploring Desktop Applications

Where researchers were exploring only the client side of cloud environment, authors of [9] originated a more challenging path; examining the server along with client. Artifacts retrieved from server consist of SQL databases, data directory, information about blowfish encryption, web server logging data and file versioning metadata. Client side as expected revealed more information about file metadata, synced files, SQlite databases, URL parameters, page titles and recently accessed files. They concluded at end that by following this methodology [1] time can be saved. Also, the file handling metadata can be helpful in this regard as it will help the investigators to limit the research scope and produce better results.

In [6], authors tried to figure out data remnants on Windows 7 by analyzing the hard drive, network traffic and live memory. Dropbox was analyzed for building artifacts. Authors succeeded in listing artifacts like traces of application itself, file names along with content and location, credential details, full files and metadata. Plus, study figured out that application maintains a log of PCs IP address with whom Dropbox was accessed or synced.

According to [10] the un-installation or deletion in MAC OS is very different than Windows. So, they have tried two ways i.e. standard and recommended, for finding the artifacts. A total of seven experiments were carried in seven different VMs. Analysis helps people by telling about important containers and identifying other significant files, logs and databases. When live memory was looked up the username was found but no password.

In [11], authors proposed a methodology for investigating and analyzing the artifacts stored in MAC OS. Storage applications like Evernote, Dropbox, Google Docs and Amazon S3 has been analyzed as case study here. Results were somehow different for each app based on the techniques used by authors. When a user edits the file in HTML editor, screenshots are being saved in drive.

Authors of [15] used Amazon Drive as a case study in windows environment. Research work contributed in two ways. One was to determine file transfer between application and computer and other was lookup of artifacts from unallocated space. The scope was same as other studies [6], [14] i.e. interaction from browser and client application. Study was less like an analysis or evaluation and more like a guide for forensic examiners. It had no to-the-point results but artifacts were correlated and discussed for other options. Authors also introduced two scripts written in Perl for automating the whole process. This made it easier to follow and understand.

Exploring Mobile Applications

Authors of [6] investigated the containers for various activities of Dropbox in smartphone. Access to DB was made by browser as well as client application. For performing experiments several VMs were created with Internet Explorer, Firefox, Google Chrome and Apple Safari as browsers. Other tools for examination were Guidance Software EnCase and Access Data FTK Imager. Authors concluded at end that live memory and network traffic is more important when talking about mobile forensics.

In [10], Authors concluded that no matter what approach you follow for uninstalling SugarSync, it's more likely that a forensic examiner will be able to find traces of application along with content of files.

In [11], the study gave following remnants. Amazon S3 gave no valuable evidence. Dropbox had some more data for examiners like e-mail address used for login process, info about recently accessed apps and timestamps of creation and modification. Evernote artifacts consists information about location, account ID, authentication and note status i.e. deleted or not. Plus, a screenshot of note and snapshot of note content was also retrieved. Artifacts of Google Docs have a huge collection of pictures as all pages of files were retrieved in the form of images.

According to [12], usage of social networking applications is increasing and that's why investigators must investigate these for remnants. For study Facebook, Twitter and MySpace were installed and then phone was imaged for analysis. Rooting the phone helped authors to access the device fully. Facebook left remnants like detailed information about user, his friends and their contact numbers, messages, and pictures viewed on timeline. Twitter gave information about user, his tweets, people he follows, and photos he has uploaded and device metadata. MySpace however produced a small number of valuable artifacts like user credentials, cache files and cookies.

In [13] WhatsApp messenger was used for the case study. Authors did an incredible two-fold work of finding valuable artifacts for investigators and by telling them how to interpret and correlate the stored data. Research shows that a complete list of received and sent messages can be reconstructed with the help of artifacts like msgstore.db. Internal tables of this and other important databases are explained well. Log files when retrieved gave the information like when a contact was being added and which ones are being blocked or unblocked by the user. But it wasn't possible to know whether the user was being blocked by some contact or not. These along with more artifacts were correlated for better understanding WhatsApp messenger.

In [14] SkyDrive was being used for case study. Experiments were carried out for iPhone. Again, live memory, network traffic and hard disk drive was being examined. Authors stated that main purpose of study was to be aware of types of artifacts

and identification of data sources from where valuable evidence can be retrieved by practitioners. Live memory when searched gave password in unencrypted form. Study concluded that with the help of live memory and network traffic it is easier to find artifacts like login credentials, file metadata, original content along with time stamps. Like [6], it was noted that SkyDrive also saves the record of connected PCs for accessing or syncing files. So, decision about a computer can be made that either it was connected or not.

Forensics Methodology

Martini et al. [1] presented a framework for cloud forensic process, which is later followed by [9]. This research work is also based on same methodology along with NIST publication about mobile device forensics [5] details of which are listed below:

Identification & Collection

Before starting the process of collection, it was decided to collect the evidence by getting the bit-by-bit image of internal storage of device i.e. QMobile LT700 with Android Lollipop version 5.0. For the preservation of evidence, it was finalized to make copies of the original image and use them for further research i.e. (Analysis and Examination) and for later use too. Identification was performed as following:

First of all, possible sources of data were identified. Then, for data acquisition, there was a three step approach consisting of following; proper planning was done telling what to do (shown in Fig.1.) for acquiring data. After acquiring the data, it was saved to a secure place and two generated copies were used for the further procedure. Next, hashes of both (original and copy) were calculated and then compared for verifying the integrity which concluded that data was not tempered during the process. Other steps consisted of log maintenance with complete information about all the tools used in the whole process. Documentation – which allows others to follow the procedure and get the same results. Also, chain of custody was followed properly to avoid tempering; the notable points were:

- Who had the device?
- Actions performed along with time.
- Copying of evidence for use.
- Secure location where evidence was stored.

Preservation

As discussed above, forensic soundness is very important when it comes to digital evidence. For achieving this goal, MD5 of images were calculated for verifying later that there were no unwanted changes caused by examination. To prevent any kind of damage to the device it was isolated from external influences. The perimeter of the examination was made short and secure.

Throughout the collection process, use of the device was limited to authorize personals only. For preventing the remote access to the device, network connection of all types including Wi-Fi was kept off.

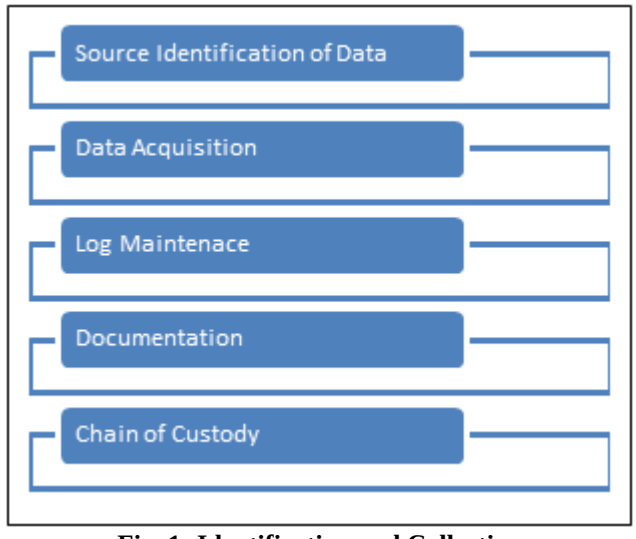

Fig. 1: Identification and Collection

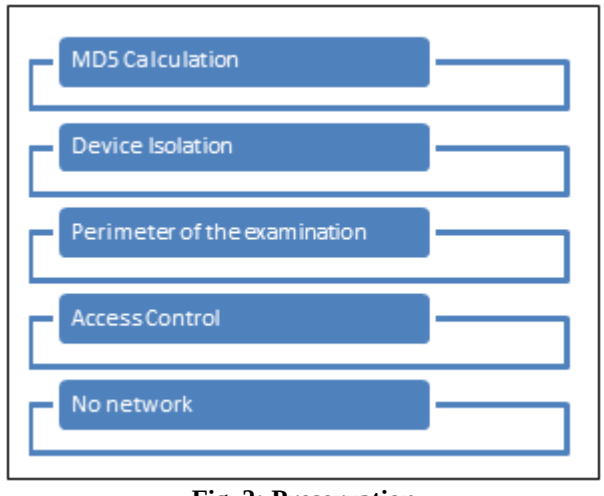

Fig. 2: Preservation

Examination and Analysis

Bit-by-bit image of the device was then examined and analyzed in search of remnants related to activities performed on each app. The process details include:

- Uncovering of digital evidence where various most important containers of information were identified as /data, /lib, and other unallocated spaces etc.
- Tools selection for every little step which concluded in use of FTK Imager, Android Debug Bridge, Rooting Application, Root Checker, Super SU, BusyBox, SQLite database reader form Firefox, tool for calculating and then verifying MD5 etc.
- Data reduction which limited the scope of research by implementing various controls upon containers.

Presentation

This section is all about summary on the basis of which the results will be concluded. The presentation format of document base on .docx, .ppt and .pdf.

Forensics Methodology

This research study follows the following framework for perusing the research in an iterative manner. The most basic

Table 1: Naming Convention

Application Name	Activity Performed	Activity Code
IDrive	Installation	AII
	Login	AIL
	Uploading file(s)	AIU
	Downloading file(s)	AID
	Deleting file(s)	AIR
	Shared file(s)	AIS
Cubby	Installation	ACI
	Login	ACL
	Uploading file(s)	ACU
	Downloading file(s)	ACD
	Deleting file(s)	ACR
	Shared file(s)	ACS

need here is to root the device to access full storage details. So, if the mobile phone is not rooted already, the foremost task of investigator will be to root it for further research as depicted in Figure 3.

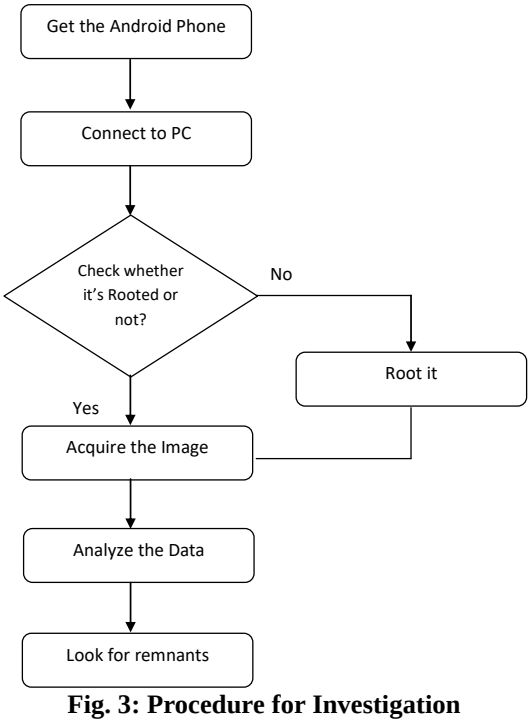

Fig. 3: Procedure for Investigation

Permeate quality analysis was made for each incremental change in the feed flowrate. The TSS rose to 13.5 ºBx upon increasing the feed flow rate to 450 $L.h^{-1}$. Increasing the feed flow rate beyond this point did not affect the TSS anymore. Viscosity remained between 1 to 1.2cP for the feed flowrate variations. Turbidity was reduced to 1.7 NTU. No change was noticed in permeate pH.

Experiment Setup

To extend the research started by [1], [9] a total no of 6 experiments have been done for each app. Details of the whole setup are as follows:

Android phone (QMobile LT700) with android version 5.0 Lollipop has been used as an experimental device. For collection phase, as described above, images have been acquired right after using the applications and populating the device in a controlled manner.

Naming Convention

For each activity, there is a code assigned to it in Table 1. The process of naming the activities is based on the following technique.

- First Letter indicates that it is an android device.
- The second letter is based on initials of the respective app.
- The last or Third letter shows the activity like L for Login, U for Uploading the file, D for downloading, R for deletion and S for Sharing.

Analysis of Applications

As visible in Table 1, six experiments were performed for each application making a total of 12 experiments for two applications. The process was broken down into two parts, one for each application; after completing activities on one application the phone was reset to factory settings to make things clearer as depicted in Figure 2. After then, next application was installed.

Cubby

Under the six activities various experiments were performed using Cubby. All detailed points are listed in Table 2.

Table 2: Detailed activities performed on Cubby

Activity Code	Experiments Performed
ACI	Application Installed
ACL	Registered/Logged in by using aizaabbasi@gmail.com
ACU	Created Cubbies
ACD	Uploaded data in them (.png, .jpg, .docx, .pdf)
ACS	Uploaded images from PC and downloaded in Mobile
ACR	Made Offline Access Available

1. Remnants of ACI

When image of phone was analyzed for ACI activity, we found that Cubby app was installed as 'com.logmein.cubby' under '/data/data'. Other changes made on phone were at '/data/app/com.logmein.cubby-1/lib' which is path of native library. This information was retrieved by examining '/userdata/root/system' directory after viewing packages.xml, packages.list and a file named as 81_task.

2. Remnants of ACL

Analysis /cache gave the user id as 'registered id=aizaabbasi@gmail.com' shown in Fig. 3. The file 81_task.xml helped here too for retrieving the date when the user interacted with the application for the very first time and also most recent interaction between both. The timestamp was in epoch format which when converted gave the accurate information as noted.

3. Remnants of ACU

Almost every required detail about uploaded files / folders was retrieved by exploring the database named 'cubby.db' in SQLite Manager. It was retrieved from '/data/data/com.logmein.cubby/databases'. The IDs were assigned to folders or cubbies. In each of them the files name

had two parts like Id1_Id2.ext where Id1 was folder's ID and id2 refers to ID of that specific file. Every file had an associated hash with a length of 64 ASCII characters. Analysis of cache showed that SHA-256 is used in Cubby for hashing Figure 5. In database, path of actual files was also displayed which was then followed for retrieving the images successfully.

4. Remnants of ACD

For performing this activity, account was logged in using some other PC. There a new Cubby named PC Data was created and then two pictures were added to it namely *beyuk_adaa.jpg* and *mehndi.jpg*. When browsed from phone, the newly created cubby along with data was there. So, pictures were then downloaded in Mobile. Their presence was observed after analysis of database as shown Figure 6. A unique ID was assigned to all three of them. Other information was also available and when followed the path found pictures too. Files whose access was made available offline had a value '1' under 'savedForOffline' attribute of above mentioned database.

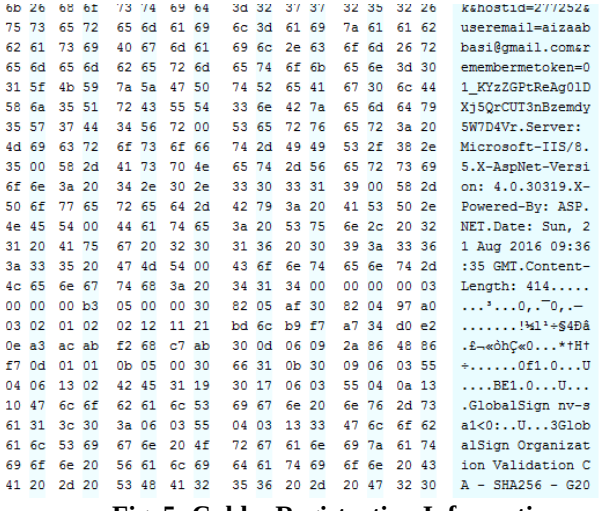

Fig. 5: Cubby Registration Information

5. Remnants of ACS

Cubby allows two types of sharing; public and private. Both experiments were carried out for this activity. The ID invited for sharing a Cubby named 'MyCubby' was 14msisaabbasi@seecs.edu.pk.In above mentioned database, 'numberOfMembers' attribute told how many members other than owner (which was aizaabbasi@gmail.com) has access to this cubby. For 'MyCubby', the count was 2 which means three persons were sharing this folder. When the cache was investigated a network invite ticket was sent to email id 14msisaabbasi@seecs.edu.pk along with link for invited URL. So concluded result is that when an invite has been sent to a person Cubby generates a ticket and send it to associated email id. So, number of participants and email addresses was recovered by investigators. Plus, when that link was followed it returned the message that either that invitation was accepted by the recipient or its expired now. For public sharing, the link of public address was visible under 'publiclink' attribute of the DB.

2	1829299183	1002963980	1	CFP_CSDiFo2016.doc
3	1829297584	1002963981	1	Screenshot_2016-08-2
4	1829297619	1002963981	1	Screenshot_2016-08-2
5	1829297634	1002963981	1	Screenshot_2016-08-2
6	1829297636	1002963981	1	Screenshot_2016-08-2
7	1829297686	1002963981	1	Screenshot_2016-08-2
8	1829297036	1002963978	1	facebook_ringtone_p
9	1829297370	1002963978	1	five.jpg
10	1829296109	1002963979	1	one.jpg
11	1829296176	1002963979	1	two.jpg
12	1829296290	1002963979	1	four.jpg
13	1829724588	1002963979	1	IMG-A.jpg
15	1829949012	1002964389	1	mehndi.jpg
16	1830026991	1002964389	1	beyuk_adaa.jpg
17	1834535959	1002963978	1	14102644_1320303278

Fig. 6: Downloaded images

6. Remnants of ACR

After the activity ACU, some of the pictures were deleted namely *three.jpg* and *screenshot.png*. Deleted pictures were not recovered but information about them was sufficient enough to conclude that user have deleted some of his pictures recently. Database when viewed in SQLite manager, it didn't show any of deleted picture or its path. But when the path was followed for retrieving other pictures, a slight change was observed in both views. The hex view showed four files but in files window there was name of three images. One of them was three.jpg, the deleted file. Same strategy was followed for the other cubby from where another image was deleted. Results were same too as expected. The information we've found for a deleted file consists of name along with extension and calculated hash. The third point in this regard was that when thumbnails folder was located and searched, we've found thumbnails of deleted pictures too. So, the pictures were also visible.

IDrive

Different investigations were performed using IDrive under those 6 exercises specified above in Table 1. A point by point rundown of those is depicted underneath:

Table 3: Detailed activities performed on IDrive

Activity Code	Experiments Performed
AII	Application Installed
AIL	Registered/Logged in by using aizaabbasi@gmail.com
AIU	Created Folders Uploaded data in them (.png, .jpg, .docx, .pdf)
AID	Uploaded images from PC and downloaded in Mobile Made Offline Access Available
AIS	Invited other users Shared files publically
AIR	Deleted files

1. Remnants of AII

At the point when image of device was dissected for ACI, we found that IDrive was introduced under '/data/data' with a directory named "com.prosoftnet.android.idriveonline". The file under /shared_prefs/IDrivePreffile.xml gave "accountCreation_date" as 2016-08-21 03:03:35 which is accurate. The path of local library was noted as

'/data/data/com.prosoftnet.android.idriveonline.-1/lib' with the help of packages.xml, packages.list and 99_task.xml. All of these were accessible at '/userdata/root/system' registry.

2. Remnants of AIL

Analysis of /shared_prefs/ IDrivePrefFile.xml provided the user id as '<string name="emailid">14msisaabbasi@seecs.edu.pk</string>' shown in Figure 7. Not only this, file also revealed the password for IDrive in plaintext. The file '357_task.xml' aided here too to retrieve the date when first and last interaction between client and application was made. Again, when timestamp of epoch format was converted it presented the accurate information.

```
<string name="gcmRenewalRegistrationId">APA91bGUTTJUjAC
<string name="shareevsserver">evsweb983.idrive.com</str
<string name="encstatus">DEFAULT</string>
<string name="external">/storage/sdcard1</string>
<string name="idrivesync_username"></string>
<string name="usedquota">6308222</string>
<string name="encFlag">N</string>
<string name="configtype">DEFAULT</string>
<boolean name="isbackupallfrag" value="false" />
<string name="password">12345$$</string>
<int name="activeTab" value="1" />
<int name="upload_filecount" value="0" />
<string name="new_backup_approach_android">yes</string>
<string name="idrive_sync_server_address"></string>
<string name="international_add_flag">0</string>
<boolean name="startautoupload" value="true" />
<string name="acctype">Evs</string>
<boolean name="canautoupload" value="true" />
<string name="filecount">41</string>
<string name="username">14msisaabbasi@seecs.edu.pk</str
<string name="optionid"></string>
<string name="servername">evsweb983.idrive.com</string>
<string name="enddate"></string>
<string name="new_backup_approach_date">2014-10-01 00:1
ap>
```

Fig. 7: Login Information

3. Remnants of AIU

Database of application named IDrive.db was investigated to have an insight about uploaded files and folders. It was recovered from '/data/data/com.prosoftnet.android.idriveonline /databases'. The IDs were assigned to content. Unlike Cubby, it displayed the timestamps in Human readable format under tags of 'lastmodifieddate' & 'savedate'. But, a column of 'capturedate' was in epoch format. Path of files was displayed under the tag of 'reference folder'. Hash algorithm used here is mentioned unlike Cubby. IDrive used MD5 for calculating hashes as shown in Figure 8. Pictures were recovered after following the mentioned path.

4. Remnants of AID

Like Cubby, for this experiment, account was logged in using some other PC. After downloading the pictures in Mobile phone the phone was imaged.

_id	md5filepath	md5filechecksum
3	/LT700_354872070237619/Photos/Screens...	97090
7	/LT700_354872070237619/Photos/Blueto...	224624
9	/LT700_354872070237619/Photos/Blueto...	149862
10	/LT700_354872070237619/Photos/Blueto...	74334
11	/LT700_354872070237619/Photos/Screens...	425741
12	/LT700_354872070237619/Music/notificat...	4440
13	/LT700_354872070237619/Music/Rington...	30056
14	/LT700_354872070237619/Music/Rington...	76425
15	/MyStuffHere/CFP_CSDiFo2016.docx	-880508451
16	/MyStuffHere/CSDiFo Poster.pdf	-1407307360
17	/MyStuffHere/MBA Dissertation.pdf	-1241528810
18	/Latest/Screenshot_2016-08-21-15-11-26....	114490
19	/Latest/Screenshot_2016-08-21-15-00-33....	45016
20	/Latest/Screenshot 2016-08-21-15-05-37....	171332

Fig. 8: MD5 Calculation

Analysis of IDrive.db showed their presence. Like other images, unique ID was assigned to these too. Information like 'hasthumbnail' and 'filetype' was also displayed which helped recognizing files. The reference folder path when followed lead to original images. Another attribute named 'offlineinfo' had a list of all files whose offline availability was enabled.

5. Remnants of ACS

As like Cubby, IDRive allow public and private sharing. Again both experiments were carried out for this activity. A link file was found in which a copy of invitation was kept. Form where we discovered the recipient which in this case was aizaabbasi@gmail.com. Unlike cubby, it doesn't display any public link or don't tell how many people are sharing some folder.

6. Remnants of AIR

Pictures were deleted during this activity, and we failed to recover the deleted pictures. But, as like Cubby, we were able to found the information about them from thumbnails and missing ID of IDrive.db. It was enough for understanding that some pictures were deleted by user. Pictures were visible due to thumbnails.

Comparative Analysis

The research questions were set as a base and after the analysis of each application final results were likened with those questions. For login credentials, Cubby is not revealing at all whereas IDrive is very revealing as it discloses very significant information like password in plaintext; no security at all. As far as file upload timestamp information is concerned, in Cubby these were not human readable unlike IDrive. However, File deletion timestamps were not available for both applications even when both gave information about deleted files.

Table 4: Comparative Analysis of Cubby and IDrive

Research Question	Cubby	IDrive
Username + Password?	Yes (Registered Id) + No	Yes (Registered Id) + Yes
Form of Password	Password was encrypted	Password was in Plaintext
Recently uploaded files with timestamps?	Yes + Epoch Format of Timestamps	Yes + Timestamps were human readable
Recently downloaded files + timestamps?	Yes + Yes	Yes + Yes
Participants of shared files?	Yes	Yes
Recently deleted files + timestamps?	Yes + No	Yes + No
Public Availability	Yes	No
Revealed the encryption Technique?	No; due to 64 characters SHA-256 is concluded	Yes; MD5

Information about participants of shared file and folders was available for both applications. Even the shared information was available for Cubby. On contrary, when it comes about

Public Availability of files, IDrive is more sophisticated as it doesn't give a clue about it. Last and least, as discussed above, Cubby provides more security for login credentials. This is considered true because Cubby displays much privacy for encryption algorithms it used; it does not expose it to end user unlike IDrive. However, by investigating in detail it gave clues which was confirmed by examining internal file.

Comparative Analysis

Security of Cubby is better than IDrive as they are following latest tools like SHA256. On the contrary, IDrive maintains the password in plaintext which is not a secure practice. Thus, for users Cubby is more secure while for examiners IDrive is more useful as it reveals very important information like password. For future, research can be extended for other platforms like MAC, iOS and Windows. Plus, the research area can be expanded as by examining internal memory as well.

REFERENCES

1. Martini, B., Do, Q., & Choo, K.-K. R. (2015). Conceptual evidence collection and analysis methodology for Android devices. arXiv:1506.05527 [Cs], 285–307. http://doi.org/10.1016/B978-0-12-801595-7.00014-8
2. Daryabar, F., Dehghantanha, A., Eterovic-Soric, B., & Choo, K.-K. R. (2016). Forensic investigation of OneDrive, Box, GoogleDrive and Dropbox applications on Android and iOS devices. Australian Journal of Forensic Sciences, 1–28. http://doi.org/10.1080/00450618.2015.1110620
3. Webb Hobson E. Digital investigations in the cloud. Farnborough, UK: QinetiQ Digital Investigations Service; 2010
4. Garfinkel, S. L. (2010). Digital forensics research: The next 10 years. Digital Investigation, 7, Supplement, S64–S73. http://doi.org/10.1016/j.diin.2010.05.009
5. Ayers, R., Jansen, W., & Brothers, S. (2014). Guidelines on mobile device forensics (NIST Special Publication 800-101 Revision 1), 1, 85. http://doi.org/10.6028/NIST.SP.800-101r1
6. Quick, D., & Choo, K.-K. R. (2013b). Dropbox analysis: Data remnants on user machines. Digital Investigation, 10(1), 3–18. http://doi.org/10.1016/j.diin.2013.02.003
7. Quick, D., & Choo, K.-K. R. (2014). Google Drive: Forensic analysis of data remnants. Journal of Network and Computer Applications, 40, 179–193. http://doi.org/10.1016/j.jnca.2013.09.016
8. Daryabar, F., Dehghantanha, A., & Choo, K.-K. R. (2016). Cloud storage forensics: MEGA as a case study. Australian Journal of Forensic Sciences, 1–14. http://doi.org/10.1080/00450618.2016.1153714
9. Martini, B., & Choo, K.-K. R. (2013). Cloud storage forensics: ownCloud as a case study. Digital Investigation, 10(4), 287–299. http://doi.org/10.1016/j.diin.2013.08.005
10. Shariati, M., Dehghantanha, A., & Choo, K.-K. R. (2016). SugarSync forensic analysis. Australian Journal of Forensic Sciences, 48(1), 95–117. http://doi.org/10.1080/00450618.2015.1021379
11. Chung, H., Park, J., Lee, S., & Kang, C. (2012). Digital forensic investigation of cloud storage services. Digital Investigation, 9(2), 81–95. http://doi.org/10.1016/j.diin.2012.05.015
12. Al Mutawa, N., Baggili, I., & Marrington, A. (2012). Forensic analysis of social networking applications on mobile devices. Digital Investigation, 9, Supplement, S24–S33. http://doi.org/10.1016/j.diin.2012.05.007
13. Anglano, C. (2014). Forensic analysis of WhatsApp Messenger on Android smartphones. Digital Investigation, 11(3), 1–13. http://doi.org/10.1016/j.diin.2014.04.003
14. Quick, D., & Choo, K.-K. R. (2013a). Digital droplets: Microsoft SkyDrive forensic data remnants. Future Generation Computer Systems, 29(6), 1378–1394. http://doi.org/10.1016/j.future.2013.02.001
15. Hale, J. S. (2013). Amazon Cloud Drive forensic analysis. Digital Investigation, 10(3), 259–265. http://doi.org/10.1016/j.diin.2013.04.006
16. https://play.google.com/store/apps/details?id=com.logmein.cubby&hl=en
17. https://play.google.com/store/apps/details?id=com.prosoftnet.android.idriveonline&hl=en