International Journal of Computer Science and Software Engineering (IJCSSE), Volume 5, Issue 12, December 2016
ISSN (Online): 2409-4285 www.IJCSSE.org Page: 272-282

DATA: Survey of Data Acquisition Techniques on Android Devices

Muhammad Shahbaz[1], Awais Bilal[2] and Usman Ahmad[3]

[1, 2] SEECS, National University of Sciences and Technology, Pakistan

[3] College of Computing, Riphah International University, Pakistan

[1]13msccsmshahbaz@seecs.edu.pk, [2]13msccsabilal@seecs.edu.pk, [3]usman715@gmail.com

ABSTRACT

The usage of Android smartphones as cellular mobile phones is very popular and dramatically increasing around the globe because of the rich functionality and an open source android operating system developed by Google Inc. In this report different digital forensics techniques are explored on android smartphones considering different scenarios of the android devices based on exiting methods. Also an android application was developed which retrieves contacts from smartphone and dumps on the external SD card in a unique case scenario when the android device has enabled adb while the device has obstructed screen lock present. A small survey was conducted based on how users use their android devices from a forensic perspective in this course class room for the possibilities of different scenarios of the android device. An android smartphone was tested for digital forensics with a unique scenario and a custom recovery was flashed and got the root access and using adb in recovery mode obstructed screen with pattern lock was disabled on the stock rom. These forensics techniques can be used for digital forensics on android smartphones when an android device is retrieved which is used either in a criminal act or the owner of the device is under investigation.

Keywords: *Android, Smartphone, Digital Forensics, adb.*

1. INTRODUCTION

From past several years smartphones market is exponentially increasing. In 2014, there are many major players in the market but android operating system dominates all other smartphones by holding the highest shipment volume according to a recent study conducted by IDC [1] [2]. The reason for popularity of android operating system is because of open source platform for mobile device manufactures and an application friendly environment developed by Google Inc. [3]. Android smartphones provide many features under one umbrella including mass storage for big files, internal contact storage, camera to image capture or use as a video recorder, GPS navigation, internet with support of Wi-Fi or cellular data Edge or 3G even 4G LTE, emails, a wide range of applications, games and many more. Android devices are also cost effective which provides different ranges from high value devices with state of the art hardware including multi core processors and gigabytes of ram to low hardware cheap devices which attracts consumers from all price ranges in the smartphone market from different manufactures and independent of carrier providers.

If an android device is used either in a criminal act, any unlawful activity or the owner of the android device is under investigation by a legal entity and the device is captured as evidence, the data inside the android device can be helpful in resolving a critical situation. In this report a procedural flow chart is presented based on different conditions of the captured android device that is android device is locked with obstructed screen or unlocked. Several digital forensics techniques are discussed to retrieve data from android devices for lawful purposes.

An android java application was developed and tested to collect contacts from android smartphone and saved on the external SD card of the device for a unique scenario when an android device had obstructed screen locked but adb was enabled. The saved file then can be retrieved from external memory card.

A small survey was conducted in the computer security course class about the use of android smartphones for different scenarios from digital forensics perspective and how different forensics techniques can be applied on different devices. A test android smartphone manufactured by HTC model One V was used for digital forensic purpose with obstructed screen lock present and bootloader was unlocked. A custom recovery was flashed and nand backup was performed and dumped on the external SD card. Then root access was gained and using adb in recovery mod obstructed screen lock was disabled on a Windows 8.1 operating system.

The next section 2 describes working of android operating system architecture. In section 3 related work is discussed on digital forensics. Section 4 will talk about procedural design for several scenarios to collect data with different digital forensic techniques.

2. ANDROID ARCHITECTURE

Android platform is represented in 0The next subsections describe android operating system (subsection 0), Working of Android Java Application (subsection 0), How Android Application Security Works (subsection 2.2), ADB (Android Debug Bridge) in subsection 2.3.

2.1 Android Operating system

Android Operating system is an open source operating system which is based on linux kernal also an open source operating system with built-in C/C++ libraries. All android application are java based applications runs inside a sandbox with reserved area for applications a java virtual machine that is specially designed for Android DVM abbreviated Dalvik Virtual Machine optimized for android smartphones [4].

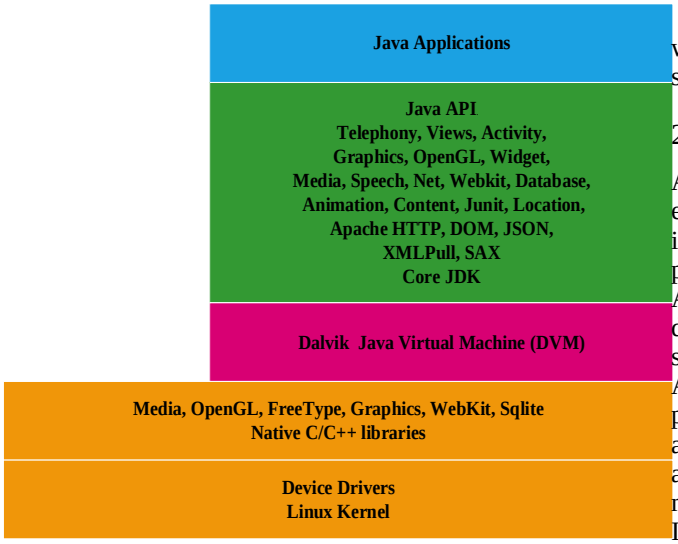

Fig. 1. Android Platform

2.2 Working of Android Java Application

In android operating system applications are written with Java language and android provides a rich application environment to build creative applications and fun games [5].
Android applications uses android's built-in Java APIs to access different components of the system running inside the DVM. For android application development Google provides SDK (Software Development Kit) tool with ADT plugin for Eclipse. To test android applications Eclipse ADT plugin provides an android emulator. Android application during development phase can be tested and debugged inside the android emulator before releasing or publishing the application.

Android application has four essential application components which considers as building blocks [6]. These are:

1) Activities: To display information on the screen with a user interface activities are responsible for these task. For example a dial pad on the screen is an activity another activity could be to reading your personal SmS or locating your location on the maps [6].
2) Services: These run on the background and does not require a user interface or screen display and can perform its task quietly without user interrupt [6].
3) Content providers: When two application require data to be shared between them, then the content provider an app component is used to access data from another application with providing interface [6].
4) Broadcast receivers: Another app component that are used when the message or a response require to broadcast system wide on an android operating system.

In this report an android application was developed which can capture contacts from android device and saves a file with .VCF format on an external SD card.

2.2 How Android Application Security Works

Android operating system provides an isolation environment to run applications with a unique system identity. Every application is assigned with a different process id and are completely isolated from each other.
Android application security model consists of a declaration permissions if an application require specific system components to access inside a file AndroidManifest.xml packed with android application package [7]. If there is no permission is declared then the application will not have any access to the system. Also android security model prevents application to access or modify android operating system files.
In this report the application uses READ_CONTACTS permission declared in the AndroidManifest.xml.

2.3 ADB (Android Debug Bridge)

ADB is command line tool provided by Android SDK to interface an emulator or communicate an android device with client server basis. ADB runs on Window operating system or any other specified operating system. In this report ADB was used with Windows 8.1 operating system. ADB is used for development debug purposes. It can also install android applications without any notification and even can read data logs of the applications. But ADB doesn't have root access to alter or modify system files. ADB also can be used in the recovery mode with limited functionality to copy data from device etc. ADB can also be used to make a backup of the device with ADB backup command [8].

In this research an application was installed using ADB with obstructed screen locked and USB debugging was enabled without any notifications.

3. RELATED WORK

In the following subsections several research ideas are discussed which influence this report. DroidWatch is discussed in subsections 23.1, General Collection in subsection B and acquisition of digital evidence in subsection C.

3.1 Droid Watch

DroidWatch [9] is a first application of its kind which provides enterprise monitoring system for Android smartphones if an android smartphone is allowed to use inside the corporate area or corporate network [9]. DroidWatch runs as a service in the background of the android smartphone with user consent and it periodically collect different data sets for monitoring purposes to help incident responders, proactive security monitors, security auditors and forensic investigators. It stores monitoring data sets locally in SQLite database inside the android smartphone in the plain text form and uploads periodically with https on an enterprise remote server. On a successful transfer of monitoring datasets, DroidWatch clears locally stored values from SQLite database. DroidWatch works without any root privileges [9].

Tabl e 1: Pros and cons of DroidWatch

	PROS AND CONS OF DROIDWATCH	
	Pros	**Cons**
1	Large data sets can be sent for monitoring.	DroidWatch needs user Consent before using this application.
2	no root privileges are required.	It works on a network and is totally dependent on the internet connection.
3	no exploiting of the Android architecture.	Locally stored data sets are in plain text and can easily be altered.
4	first open-source android based enterprise monitoring solution.	A user under monitoring of its android smartphone using DroidWatch can uninstall the application anytime he wants.
5	Privacy	DroidWatch is vulnerable to anti-forensics techniques which mislead the results obtained from data sets.

Pros and cons of DroidWatch are presented in 0Monitoring user behavior on smartphones inside an enterprise or corporation is a challenge while allowing BYOD. DroidWatch provides enterprise a monitoring system for Android smartphones with capability to capture large data sets, does not require root privileges to work and runs the application with android architecture compatibility with no exploitation of the android architecture to properly work.

3.2 General Collection [10]

In this research author discussed and represented a general collection method for data acquisition from android smartphones regardless any manufacture specific device [10]. Author explored the android partitions and exploited on android recovery portioning by flashing the recovery partition with custom build recovery partition. The custom build partition was built specifically for a specific device. Using custom built recovery, android device was booted in recovery mode and data was collected with dumping on the memory [10].
Pros and cons of General Collection are presented in 0 A general process for data collection of Android devices independent of manufacturer system architecture. This method does not require root access to collect data. Using this procedure collected data is unchanged. It can work even with obstructed screen lock. Dumping data is the exact copy. On the other hand, to flash a custom recovery on an android device, it requires bootloader unlocked. For every specific android device a custom rom built. Unique to different devices. Android 2.0. No data analysis on collected data.

Table 2: Pros and cons of General Collection

	PROS AND CONS OF GENERAL COLLECTION	
	Pros	**Cons**
1	A general process for data collection of Android devices independent of manufacturer system architecture.	Android 2.0
2	This method does not require root access to collect data.	For every specific android device a custom rom built.
3	Using this procedure collected data is unchanged.	Unique to different devices
4	It can work even with obstructed screen lock.	To flash a custom recovery on an android device, it requires bootloader unlocked.
5	Dumping data is the exact copy.	No data analysis on collected data.

3.3 Acquisition of digital evidence [11]

In this research paper author proposed a data collection method with flow chart procedure if an android device is retrieved with a unique condition [11]. In this paper possible conditions of the captured android device are discussed and how digital forensics can be applied to collect data with maximum data acquisition possible. This paper discussed more specific conditions and what

procedure to follow next from a specific condition of the device.

Table 3: Pros and cons of Acquisition of digital evidence

	PROS AND CONS OF ACQUISITION OF DIGITAL EVIDENCE	
	Pros	**Cons**
1	A general workflow for digital forensic procedures for acquisition of data.	There are some methods which are android device specific.
2	This workflow can be applied on any android device manufacture.	Access control are more secure now, the mentioned forensic techniques might deprecate for newer systems.
3	Using it maximum data can be collected from possible ways.	
4	The existed digital forensic techniques for android can be applied.	

A general workflow for digital forensic procedures for acquisition of data. This workflow can be applied on any android device manufacture. Using it maximum data can be collected from possible ways. The existed digital forensic techniques for android can be applied. On the other hand, there are some methods which are android device specific. Access control are more secure now, the mentioned forensic techniques might deprecated for newer systems.

3.4 Digital forensic tools for android devices

There are many digital forensic tools available for android smartphones which provide aid and boost in data acquisition process. Also free forensic tools available provided by viaForensics lab for community and law enforcement entities

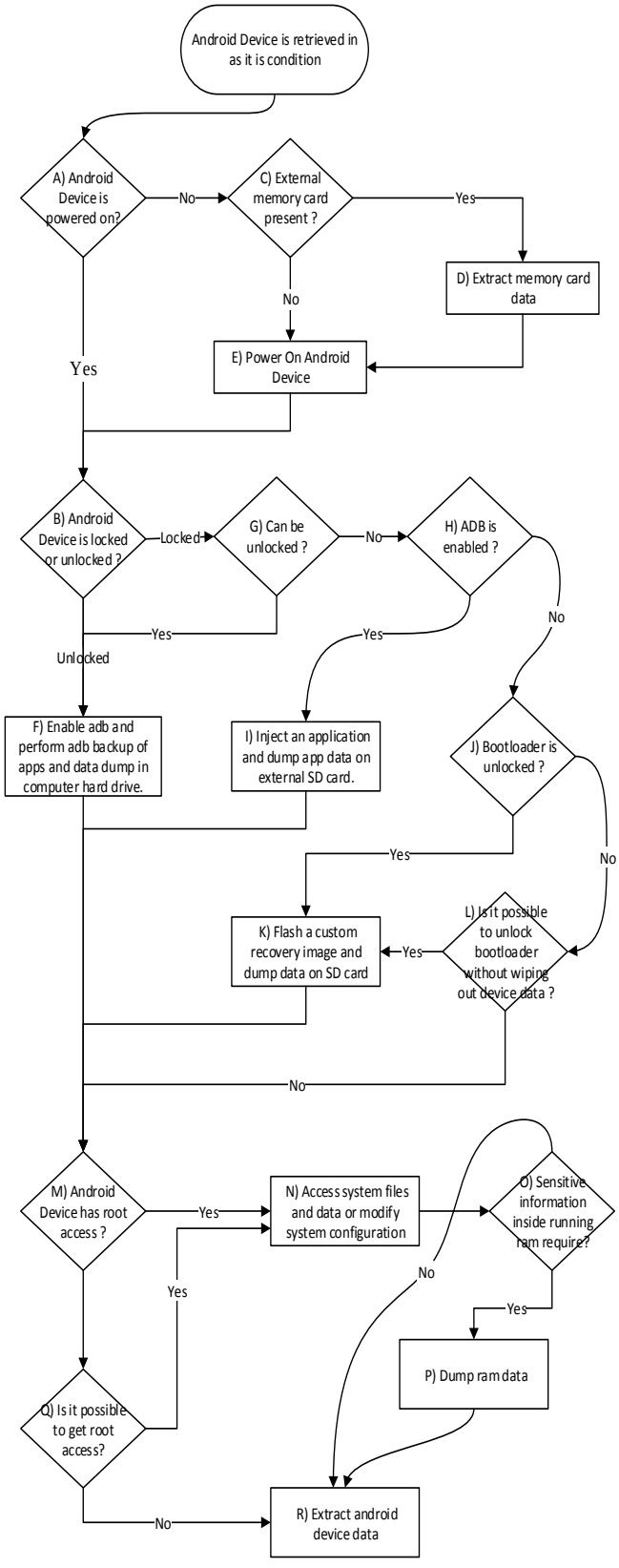

Fig. 2. Design flowchart for android device digital forensics

[12]. With community contribution there are some android specific tools available which can also be used for android forensic analysis, xda-developers [13] has the biggest community for android specific devices tools that can be helpful to boost up data acquisition and forensics.

4. DESIGN AND ARCHITECTURE

In this report a procedural workflow is suggested for digital forensics on android devices based on existing forensic techniques and exploring android operating system vulnerabilities that can be used to retrieve data from an android smartphone. This flowchart considers several unique scenarios of a retrieved android device.
This design is based on [11] with advanced techniques are included like data acquisition using custom recovery image flashed and used in recovery mode using adb in an android device if the bootloader is unlocked. The suggested design is presented in 0
In the following subsections the design flowchart shown in 0, are described. Workflow starts when an android is retrieved in as it is condition to collect data and digital forensics.

1. Android Device is powered on?

When an android device is retrieved first thing check that this device is powered on or off. If the device power is on then follow to the next step to B or C if the device is powered off.

2. Android Device is locked or unlocked?

The next thing to check the android device is obstructed screen lock or unlocked. If the device is unlocked then this could be the easiest scenario to perform data collection as much as possible and follow to step F or to step G if not unlocked.

3. External memory card present?

In this step check external SD card of the android device. If present then follow to step D or to step E if not.

4. Extract memory card data

In this process extract SD card from android device and using memory card reader extract all data. After extracting all data follow to step E.

5. Power on Android Device

In this step power on the android device in a control environment and follow to step B.

6. Enable adb and perform adb backup of apps and data dump in computer hard drive.

7. Can be unlocked?

In this step check that android device can be unlocked. If the android device is pattern locked on screen then the smudge attack on the screen can be applied to unlock the device [14]. If the device unlocks then follow to step F otherwise to step H.

8. ADB is enabled?

To check adb is enable with the obstructed screen lock, connect android device with computer via usb and run ADB command line tool comes with android SDK and the following command:

```
E:\myproject\sdk\platform-tools>adb devices
List of devices attached
SH24YTV00640    device


E:\myproject\sdk\platform-tools>
```

Fig. 3. Check adb enabled

If the android device has adb enabled then device will be listed as shown in 0Ensure that correct device drivers are installed. Then follow to the next step I otherwise to step J.

9. Inject an application and dump app data on external SD card.

This method is a proposed method specifically for this case scenario when an android device has obstructed screen lock and adb is enabled.

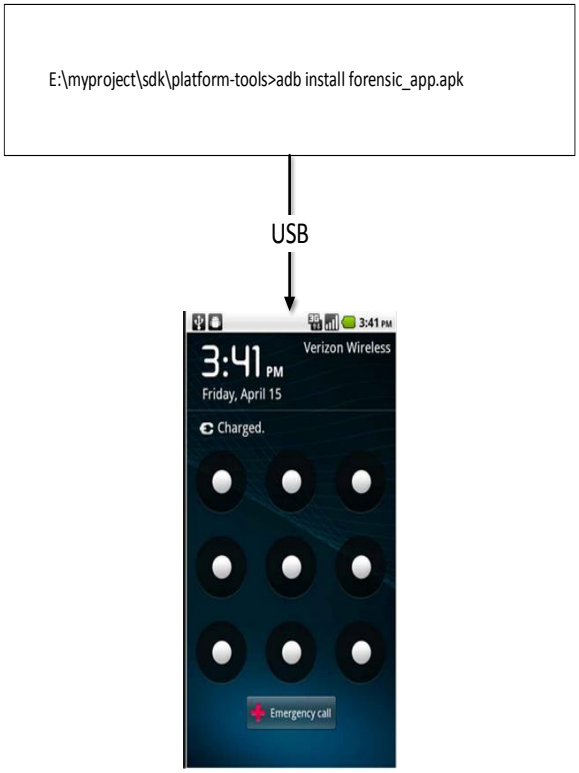

Fig. 4. Injecting app using adb with device locked

Using adb command line tool inject an android application [8] which has system access permission declared inside its AndroidManifest.XML file as shown in 0, captured all possible data and dump on external SD card. Then dumped data extract from the SD card using memory card reader.

10. Bootloader is unlocked?

To load an OS in the device or a run time environment bootloader is used. The initial state of the device is configured with the bootloader and to start executing the device's kernel [15]. To flash a custom rom or a custom operating system in an android device, bootloader must be unlocked. To check bootloader unlock reboot android device and trigger bootloader with pressing both power button and volume down key then enter fastboot mode with pressing power button. If the bootloader is unlocked then it would be displayed on top of the screen and then follow to step K otherwise to step L.

11. Flash a custom recovery image and dump data on SD card.

Flash a custom recovery image as described in [10] and perform data dump by booting device in the recovery image. Recovery image is unique to specific android device and require an expert to perform these task as it can cause a soft brick.

12. Is it possible to unlock bootloader without wiping out device data?

Bootloader is locked by default to prevent flashing a custom rom or an operating system on android device and my void warranty of the android device. Unlocking bootloader may cause a complete factory reset, it depends upon the vendor of the android device. For example the android device manufactured and developed by HTC Corporation has bootloader locked by default and it has developer support for unlocking bootloader an HTC android developer center but unlocking bootloader of a HTC android device using their developer center wipes out all the device data with factory reset [16]. If there is a method to unlock bootloader without device data wipe then use this but be sure before implementing it. If bootloader successfully unlocked then follow the step K otherwise to step M.

13. Android Device has root access?

With root access user can have access to system files and all device partitions which is usually not allowed for non-root android device. Check the android root access if it has the root access then follow to step N otherwise to step Q.

14. Access system files and data or modify system configuration

In this step using root access copy all data that can be accessible and mirror android device partitions for forensic analysis. System read write permissions are also available to change system configuration to enable system hardware functionality or disable obstructed screen lock. Then follow to step O.

15. Sensitive information inside running ram require?

If android device uses encryption techniques on run time applications to protect user data then the sensitive information such as encryption keys for cryptographic and password may require to be extracted from running ram. If requires the follow step P otherwise to step R incase no sensitive information is present in android device ram.

16. Dump ram data

The running application process' data can be extracted from ram of the android device by dumping running process data from ram as describe in [11]. This method requires root access to work and access android device's running processes. Then follow to step R.

17. Is it possible to get root access?

Rooting a device with super user to get access system files may alter or delete some data files during the

rooting process. For rooting a device bootlock must be unlocked. If rooting a device is not harmful to the device's data then root the device and follow to step N otherwise to step R.

18. Extract android device data

Finally, extract android device's internal data if it's possible to collect call logs, sms text messages, emails etc. Also do a thorough visual check using android file explorer for a key information collection.

5. IMPLEMENTING DIGITAL FORENSICS ON ANDROID

For implementing of the suggested design presented above in 0,for digital forensics techniques on an android device, three case scenarios are considered as shown in 0Android smartphone manufactured by HTC model ONE V is used for implementation of digital forensic techniques with a Dell laptop N4050 running with Windows 8.1 Operating system. Android device connected with laptop using a USB cable.

Table 4: Use case scenarios of Android Device

	Scenarios of Android devices		
	Obstructed screen lock	***ADB enabled***	***Bootloader***
1	Yes	Yes	Locked
2	Pattern lock is present but device is unlocked	No	No
3	Yes	No	Unlocked

1. ADB enabled but device is locked.

To perform digital forensics in a situation when an android device has obstructed screen lock present either with pattern lock or any other authentication mechanism and adb is enabled with bootloader locked.
With following presented design for forensics in subsection 9 an android Java application was developed and installed using adb command line tool. The android application gets contacts from android smartphone's internal storage and saves a file with .vcf format on an external SD card. This file includes contact detail retrieved from the locked android device.

```
ContactsContract.Contacts.LOOKUP_KEY
```

Fig. 5. Android Java API for contacts lookup

Android Java API for contacts is used to get access to contact list as shown in 0 Also uses permission to read contacts and write permission to save a file on an external SD-card are declared in AndroidManifest.xml as shown in 0

```
<uses-permission
android:name="android.permission.READ_CONTACTS" />
                        <uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

Fig. 6. Uses-permission to read and save contacts

2. Access control is present but device is unlocked.

If there is a case scenario when an android device is retrieved and has some access control is enabled like pattern lock but at the time of occupying the device, it is in unlocked condition and switching off the device can enable obstructed screen lock as a case scenario discussed in [11]. In that case first enable USB debugging in the developer options and connect with adb command line tool keeping the device unlock. To remove android device lock there is a vulnerability in android operating system till 4.3 found by curesec GmbH [17]. This vulnerability is removed from android 4.4 kitkat. This vulnerability was tested in android 4.0.3 with following adb command:

```
E:\myproject\sdk\platform-tools>adb shell am start -n com.android.settings/
com.android.settings.Choo
seLockGeneric --ez confirm_credentials false --ei lockscreen.password_type 0 --activity-
clear-task
Starting: Intent { flg=0x8000 cmp=com.android.settings/.ChooseLockGeneric (has
extras) }

E:\myproject\sdk\platform-tools>
```

Fig. 7. adb shell command to remove lockscreen

By running the above command a pattern unlock setup initiates without authentication of the previously set pattern lock. When new draw pattern for unlock was established then device can be unlocked or even security lock can be removed. So a new pattern was established and then in android setting => security and lock was disabled to prevent device lock. Now data can be extracted from devices internal storage using adb backup.

3. Device locked and bootloader in unlocked.

In this scenario the HTC One V android device has obstructed screen lock with pattern and adb is not enabled but bootloader is unlocked.

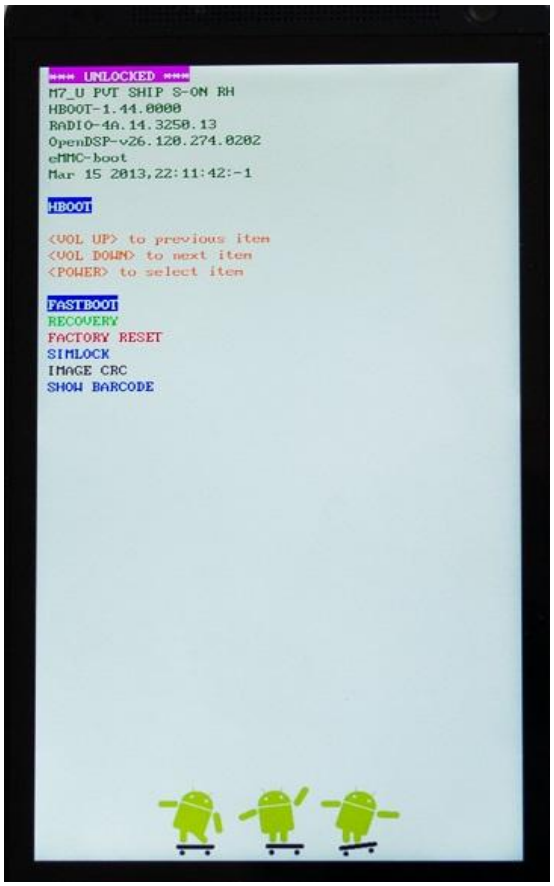

Fig. 8. Showing bootloader is unlocked

To confirm the device's bootloader is unlocked, android device was rebooted in bootloader mode by pressing power button and volume down button. On loading the android device into bootloader confirms that the device has bootloader unlocked as shown in 0

Then using power button fastboot mode was selected to flash a custom recovery image a method described in [10].

The custom recovery image specially designed for HTC One V was provided by clockworkmode [18]. Using fastboot command line tool provided with Android SDK the clockworkmode recovery image was flashed in the android device's recovery partition. The commands are shown in 0Although using Windows 8.1 caused many obstacles for example the android device was not detecting in adb tools even after installing the HTC sync manager somehow the device was detected in adb mode but was not detectable again in fastboot mod even installing correct device drivers. Device was successfully detected in fastboot mod when windows usb device signature checking drivers was replaced with old operating system drivers that is Windows 7.

```
E:\myproject\sdk\platform-tools>fastboot flash recovery recovery.img
sending 'recovery' (5168 KB)...
OKAY [  4.461s]
writing 'recovery'...
OKAY [  1.597s]
finished. total time: 6.060s

E:\myproject\sdk\platform-tools>fastboot reboot-bootloader
rebooting into bootloader...
OKAY [  0.157s]
finished. total time: 0.158s

E:\myproject\sdk\platform-tools>fastboot erase cache
erasing 'cache'...
OKAY [  0.383s]
finished. total time: 0.384s

E:\myproject\sdk\platform-tools>
```

Fig. 9. Flashing a custom recovery

Then the device was rebooted into bootloader and using power button revovery mode was selected. After booting into receovery mode using volume keys up and down backup and restore option selected. Using volume up and down keys backup option was selected and backup procedure was initiated as describe in [10] and dumped in SD card inside clockwork folder. The backup data was more than a GB including data, application, application data and system files.

After performing backup using recovery mode the backup image was confirmed by using restore option in recovery mode. When restore option was selected using volume up and down keys a restore image with timestamp was showing which showed that the backup image was created successfully.

Then the device was rebooted into bootloader and fastboot mode was selected to flash a custom superuser image to gain root access using method provided by a user on xda-developers [19]. After gaining root access successfully the device was rebooted into recovery mode and adb command line tool was attached.

```
E:\myproject\sdk\platform-tools>adb devices
List of devices attached
SH24YTV00640    recovery


E:\myproject\sdk\platform-tools>
```

Fig. 10. ADB in recovery mode

Android device was connected with adb command line tool in recovery mode as shown in 0Now android device is rooted, so adb can perform task with root privileges.

```
E:\myproject\sdk\platform-tools>adb shell mv /data/system/
gesture.key /data/system/gesture.key.bak

E:\myproject\sdk\platform-tools>
```

Fig. 11. Renaming gesture.key

Using adb shell command the gesture file that stores pattern lock was renamed as shown in 0Then the device was rebooted into system and the device was unlocked with any pattern draw successfully.

6. EVALUATION AND RESULTS

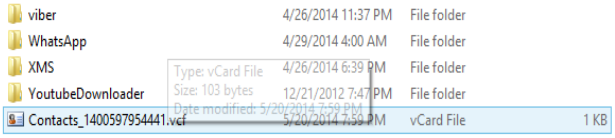

Fig. 12. Contacts saves in SD card.

Using adb an android application was installed described in subsection 1 and the application a concept of proof saved the contacts in the external memory card as shown in 0

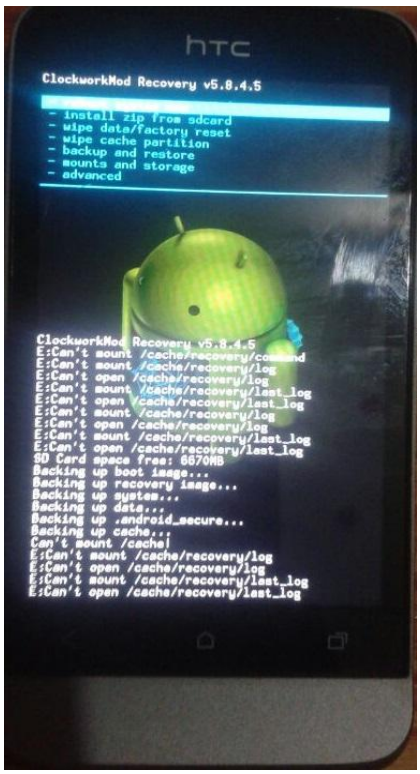

Fig. 13. Dumping device data in recovery mode.

In recovery mode backup dumping as described in subsection 3 was performed as in 0The data was dumped in the external memory card and can be accessible using memory card reader.

6.1 Survey about android device usage

A small survey was conducted in computer security course class IS-820 F'14 MS (CCS)-6 about android device usage by the users. Which can prove that the suggested design architecture for digital forensics on android devices can be applied in different and unique scenarios.

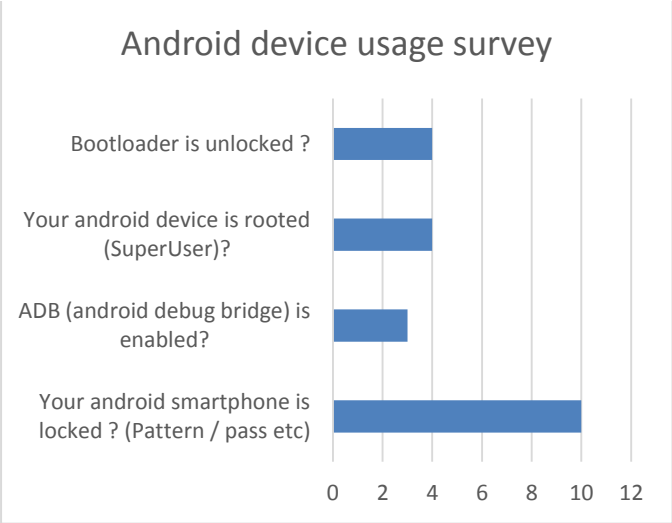

Fig. 14. Survey about android device usage.

There were sixteen responses from students in which thirteen students possess an android smartphone or device. Most of the android users uses access control mechanism to protect their device with pattern lock or password. Survey shows that ten android devices are obstructed screen lock out of thirteen devices. Only three android devices has enabled USB debugging and four android devices out of thirteen has rooted their android devices. Out of thirteen android devices only four android devices were bootloader unlocked.

7. CONCLUSION

This report shows that how to implement digital forensics with different scenarios of an android device for an effective data acquisition even on stock roms. In order to use adb an application can be installed even obstructed screen lock and data can be collected. Using recovery mode data backup was dumped on the external SD card and gained root access. Using adb in recovery mode with root privileges to change gesture.key file to disable access control on android device. And survey shows different scenarios of android device usage.

REFERENCES

[1] "Worldwide Quarterly Mobile Phone Tracker," IDC, [Online]. Available: http://www.idc.com/tracker/showproductinfo.jsp?prod_id=37. [Accessed 18 5 2014].
[2] D. Kerr, "Android dominates 81 percent of world smartphone market," CNET, [Online]. Available: http://www.cnet.com/news/android-dominates-81-percent-of-world-smartphone-market/. [Accessed 18 5 2014].
[3] "Android," Google Inc, [Online]. Available: http://www.android.com/. [Accessed 19 5 2014].
[4] "Porting Android to Devices," Google Inc., [Online]. Available: http://source.android.com/devices/index.html. [Accessed 19 5 2014].
[5] "Introduction to Android," Google Inc., [Online]. Available: http://developer.android.com/guide/index.html. [Accessed 19 5 2014].
[6] "Application Fundamentals," Google Inc., [Online]. Available: http://developer.android.com/guide/components/fundamentals.html. [Accessed 19 5 2014].
[7] "System Permissions," Google Inc., [Online]. Available: http://developer.android.com/guide/topics/security/permissions.html. [Accessed 20 5 2014].
[8] "Android Debug Bridge," Google Inc., [Online]. Available: http://developer.android.com/tools/help/adb.html. [Accessed 20 5 2014].
[9] J. Grover, "Android forensics: Automated data collection and reporting from a mobile device.," Digital Investigation, vol. 10, pp. S12-S20, 2013.
[10] T. Vidas, C. Zhang and N. Christin, "Toward a general collection methodology for Android devices.," digital investigation, vol. 8, pp. S14-S24., 2011.
[11] A. Simao, F. Sicoli, L. Melo and R. Junior., "Acquisition of digital evidence in android smartphones.," in Australian Digital Forensics Conference, Perth, 2011.
[12] "FREE TOOLS," viaForensics, [Online]. Available: https://viaforensics.com/resources/tools/. [Accessed 20 5 2014].
[13] xda-developers, [Online]. Available: http://forum.xda-developers.com/.
[14] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze and J. M. Smith, "Smudge attacks on smartphone touch screens.," in In Proceedings of the 4th USENIX conference on Offensive technologies, pp. 1-7, 2010.
[15] "Bootloader," xda-developers, [Online]. Available: http://forum.xda-developers.com/wiki/Bootloader. [Accessed 2014 5 2014].
[16] "Unlock Bootloader," HTC Corporation, [Online]. Available: http://www.htcdev.com/bootloader/. [Accessed 20 May 2014].
[17] "CVE-2013-6271: Remove Device Locks from Android Phone," curesec GmbH, [Online]. Available: https://cureblog.de/2013/11/cve-2013-6271-remove-device-locks-from-android-phone/. [Accessed 20 May 2014].
[18] "ROMs and Recovery Images," Clockworkmod, [Online]. Available: http://clockworkmod.com/rommanager. [Accessed 20 May 2014].
[19] ckpv5, "[tutorial] How to Unlock Bootloader and Root your Htc One V," xda-developers, [Online]. Available: http://forum.xda-developers.com/showthread.php?t=1598964. [Accessed 15 May 2014].

Appendix A: Installation of Android SDK.

Software Development Kit for Android operating system runs using Eclipse IDE with Android Development Tool (ADT) Plugin; and Java Development Kit (JDK).

1. Java Development Kit.
2. Eclipse IDE.
3. ADT Plugin for Eclipse.
4. Android SDK.

1. Download the "ADT Bundle" Zip File

Goto "Android Developer" @ http://developer.android.com/index.html and click on "Get the SDK".

For Windows Operating System: Click "Download the SDK - ADT Bundle for Windows". Choose either "32-bit" or "64-bit" and "Download".

For Mac Operating System: Expand "Download for Other Platforms" ⇒ Under "ADT Bundle", Select "Mac OS X 64-bit".

2. Unzip

Unzip the download zip bundle file into a folder of your choice, e.g., "C:\project" (for Windows) or "/Applications" (for Mac).

It is not allowed to use a directory name containing space or special characters.
Android Emulators or Android Virtual Devices (AVDs) allow an application to test without the real device shown in 0The application describes in 1 was developed using android SDK and tested on Android Emulator using Eclipse.

Appendix B: Configuring "AndroidManifest.xml"

Android Application Descriptor File - "AndroidManifest.xml"
Each Android application has a manifest file named AndroidManifest.xml in the project's root directory. It describes the application.

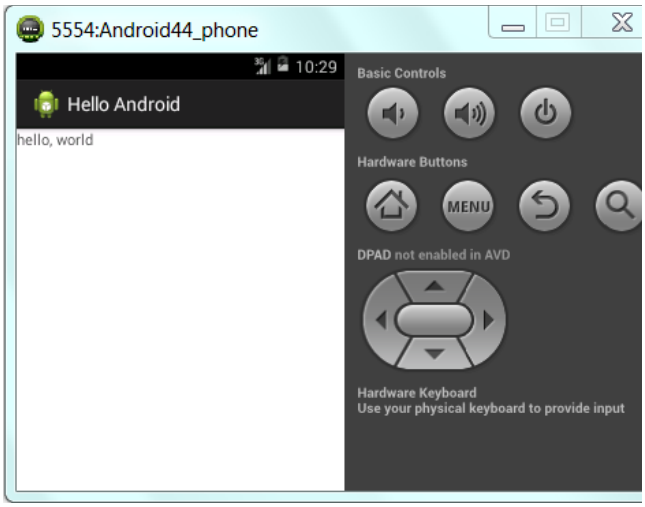

Fig. 15. Android Emulator (or Android Virtual Device)

The application describes in 1 requires permission to read contacts and write permission to save a file on an external SD-card. Using Eclipse IDE permissions are declared in "AndroidManifest.xml":

"android.permission.READ_CONTACTS"

"android.permission.WRITE_EXTERNAL_STORAGE"

Manifest file is selected from "Package Explorer" in Eclipse as shown in 0

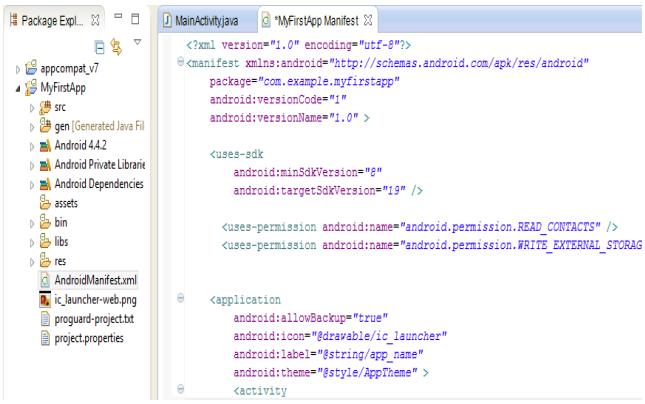

Fig. 16. "AndroidManifest.xml" opened in Eclipse IDE