Acquisition and analysis of digital evidence in Android Smart phones

By: André Morum de L. SimãoAndre, Fábio Caús Sícoli, Laerte Peotta de Melo, Flávio Elias de Deus, Rafael Timóteo de Sousa Júnior

Presented By: Abubakar Bala (g201201620)

Outline:

- Abstract
- Introduction
- Brief History of mobile phones
- Android Platform
- Data Acquisition Methods for Android Phones
- Examination and analysis
- Validation of the proposed method
- Conclusion/future work
- Questions

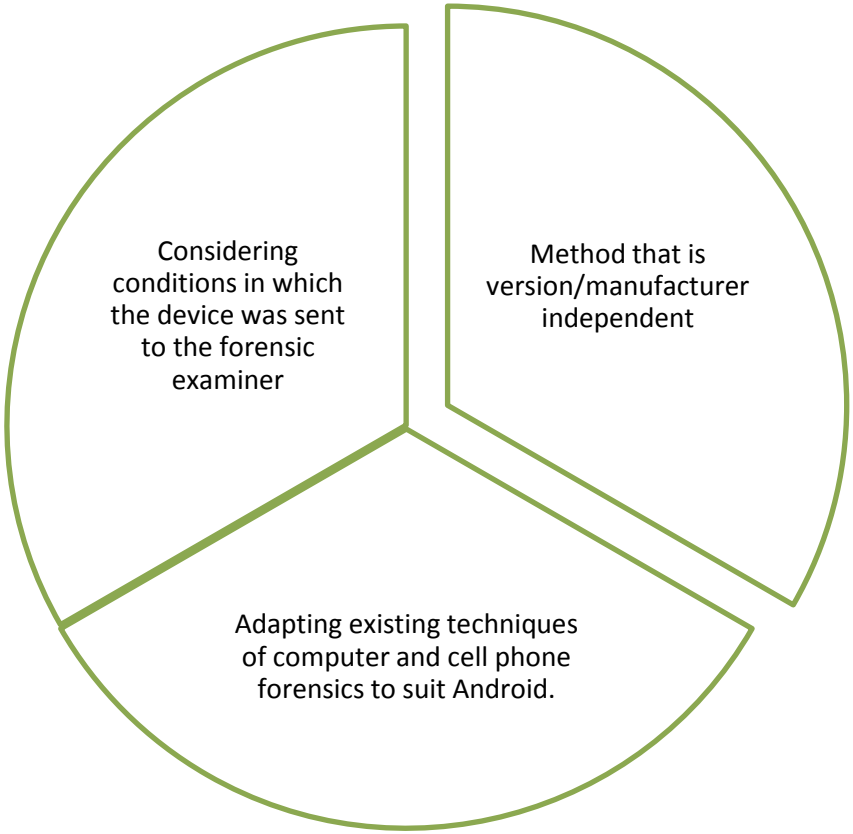
Abstract

- Android phone has a large data capacity.
- Data can be stored either locally or remotely
- Its platform supports extracting data and evidence
- Existing documented procedures are not detailed /specific to be conducted on Android phones

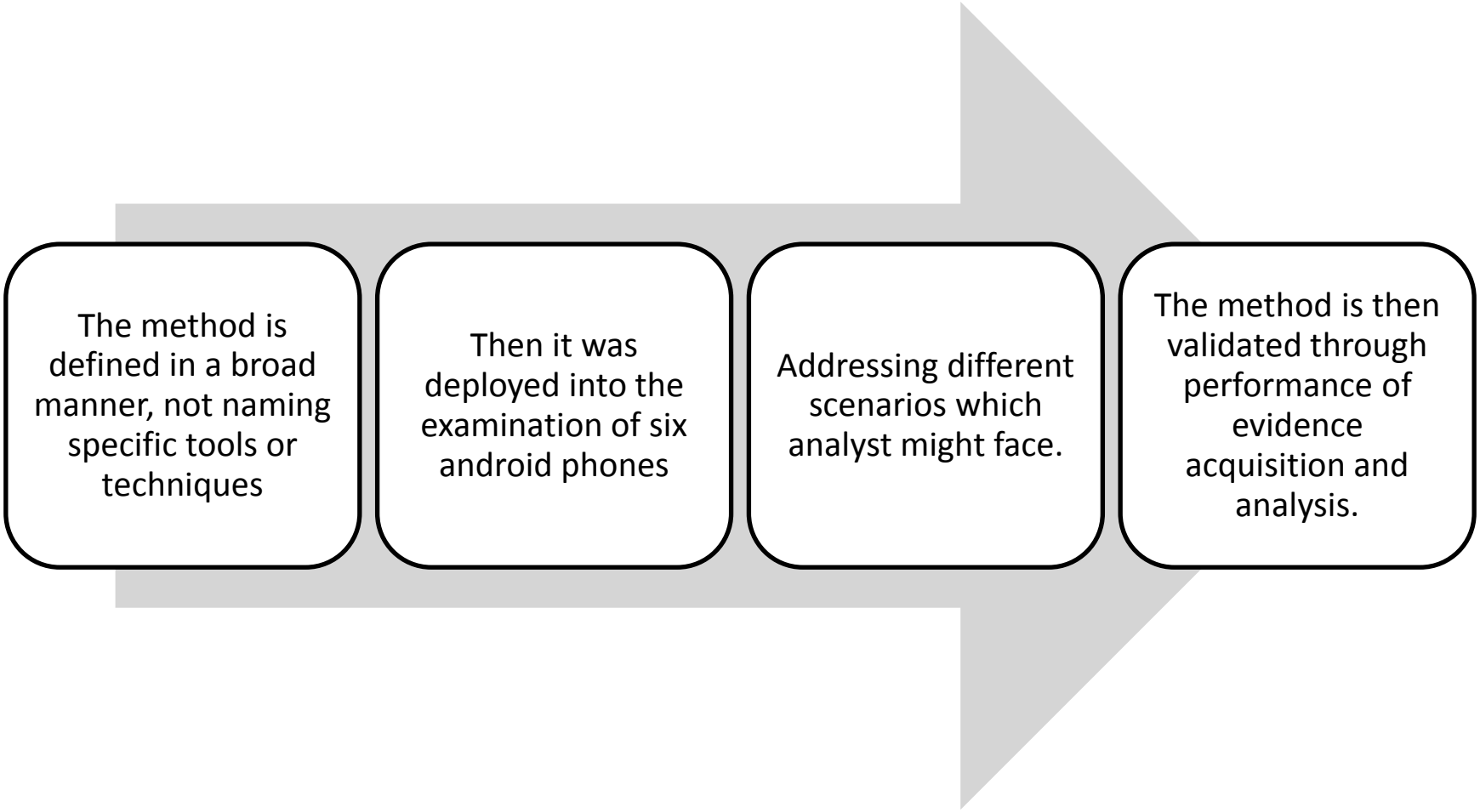
..Abstract

- Existing forensic tools do not support YAFFS2
- Each Smartphone has a unique feature
- Copying or mirroring of its internal memory can be invasive or rather complex due to difficulty in having direct hardware access.

.. Abstract


Considering conditions in which the device was sent to the forensic examiner
Method that is version/manufacturer independent
Adapting existing techniques of computer and cell phone forensics to suit Android.

.. Abstract


The method is defined in a broad manner, not naming specific tools or techniques
Then it was deployed into the examination of six android phones
Addressing different scenarios which analyst might face.
The method is then validated through performance of evidence acquisition and analysis.

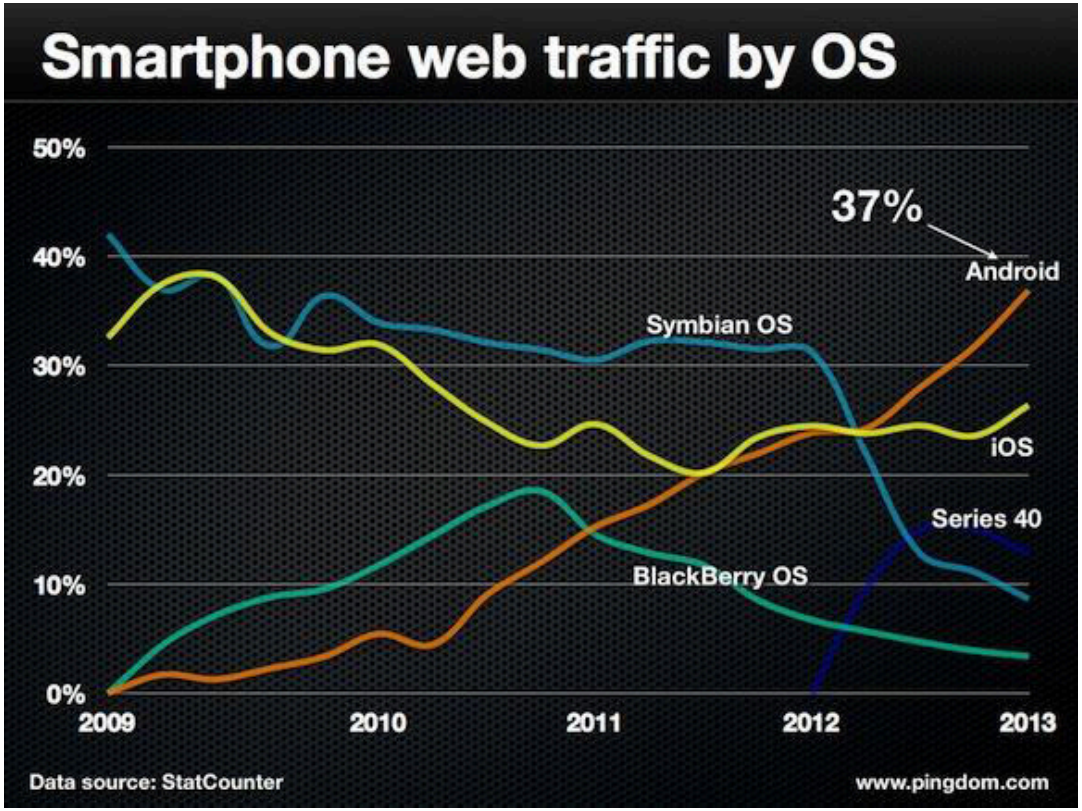
Introduction

The Android operating system have gotten a very wide acceptance.

Partly because it is open source

Supports latest features and applications available

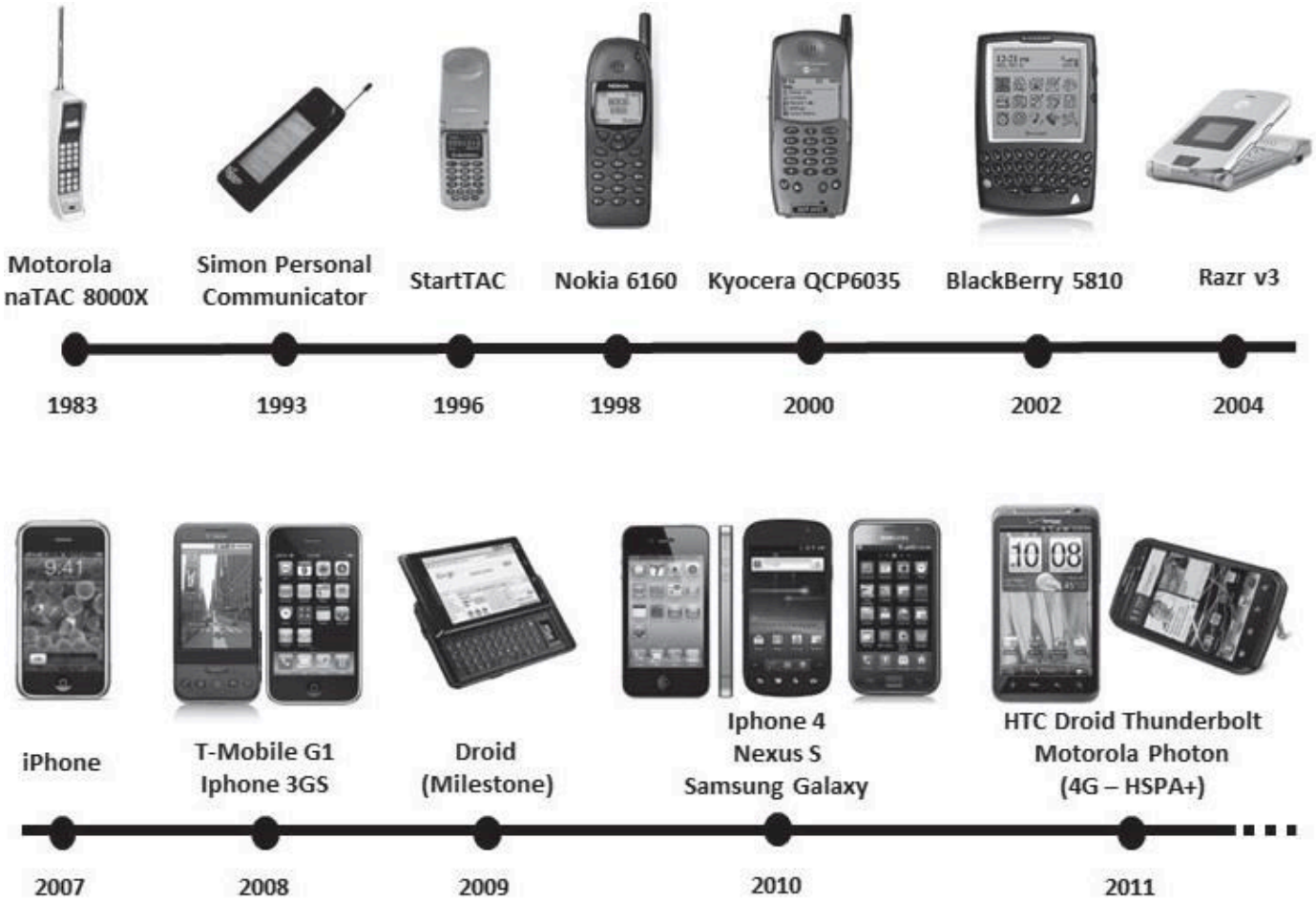
Its large storage capacity provides the forensic examiner with ample data to rely open.



.. Introduction

- Challenges of Smartphone forensics
  - Data cannot be maintained in the same format they were found.
  - The use of embedded memories whose direct hardware access is delicate and complex.
  - Sometimes involves installing applications/tools on the device-INVASION

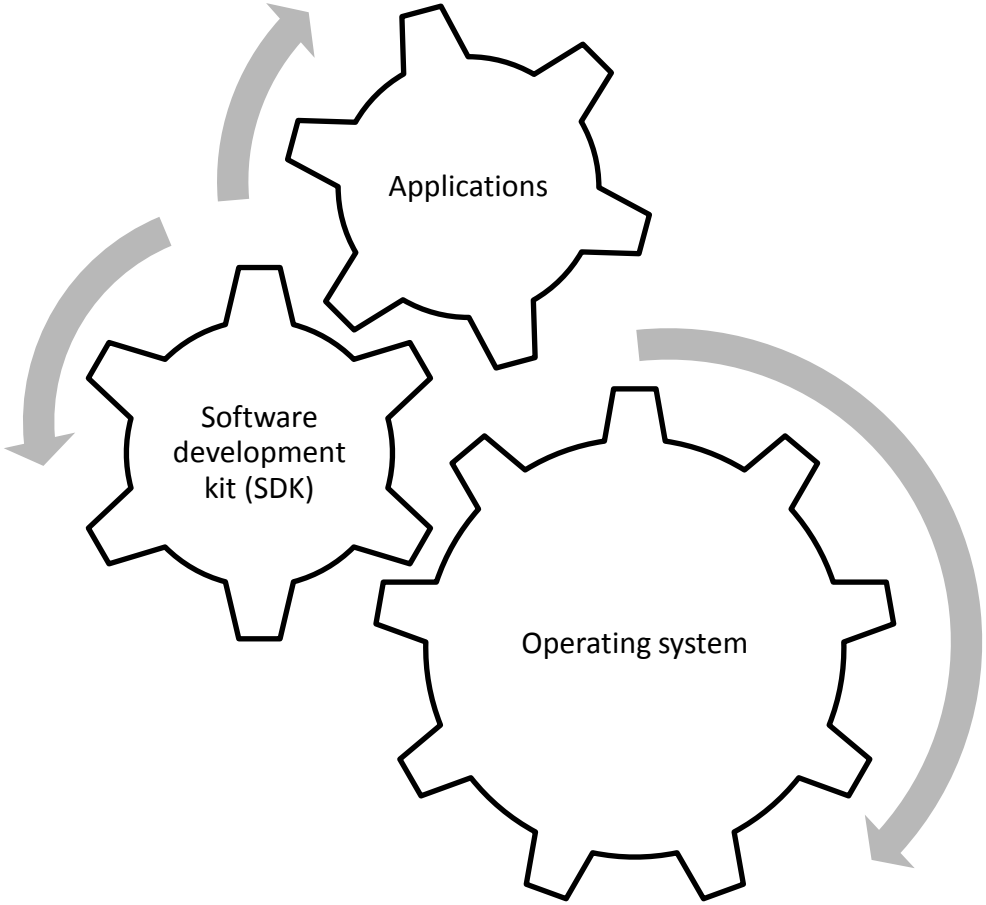
Brief History of Mobile phones



Android Platform

- Android is an open operating system designed for use on mobile devices.
- Bought by Google Inc. in 2005,
- On November 5th 2007, the (OHA) a consortium of over 80 companies was founded and contributed immensely t o the development of Android.

…Android Platform



…Android Platform

Current versions of Android

- Paper does not cover 3.x and 4.x versions because
- 3.x version is dedicated for tablets
- 4.0 was just released when the research was made

Version	Codename	API	Distribution
1.6	Donut	4	0.2%
2.1	Eclair	7	1.9%
2.2	Froyo	8	7.5%
2.3 - 2.3.2	Gingerbread	9	0.2%
2.3.3 - 2.3.7		10	43.9%
3.1	Honeycomb	12	0.3%
3.2		13	0.9%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	28.6%
4.1	Jelly Bean	16	14.9%
4.2		17	1.6%

From: http://developer.android.com/about/dashboards/index.html

Components of the Android OS

- The software stack is divided into four layers, including five different groups
- Application layer
  - Basic sets of applications: web browsers, e-mail client, SMS program ,calendar etc.
- Application framework
  - Provides an open and standardized development environment
  - API is available for application development

..Components of the Android OS

- Libraries
  - Written in C/C++, and invoked through a JAVA interface
  - Typical libraries are the ones that manage windows (surface manager), 2D and 3D media (codecs), SQ Lite database to the web browser web kit

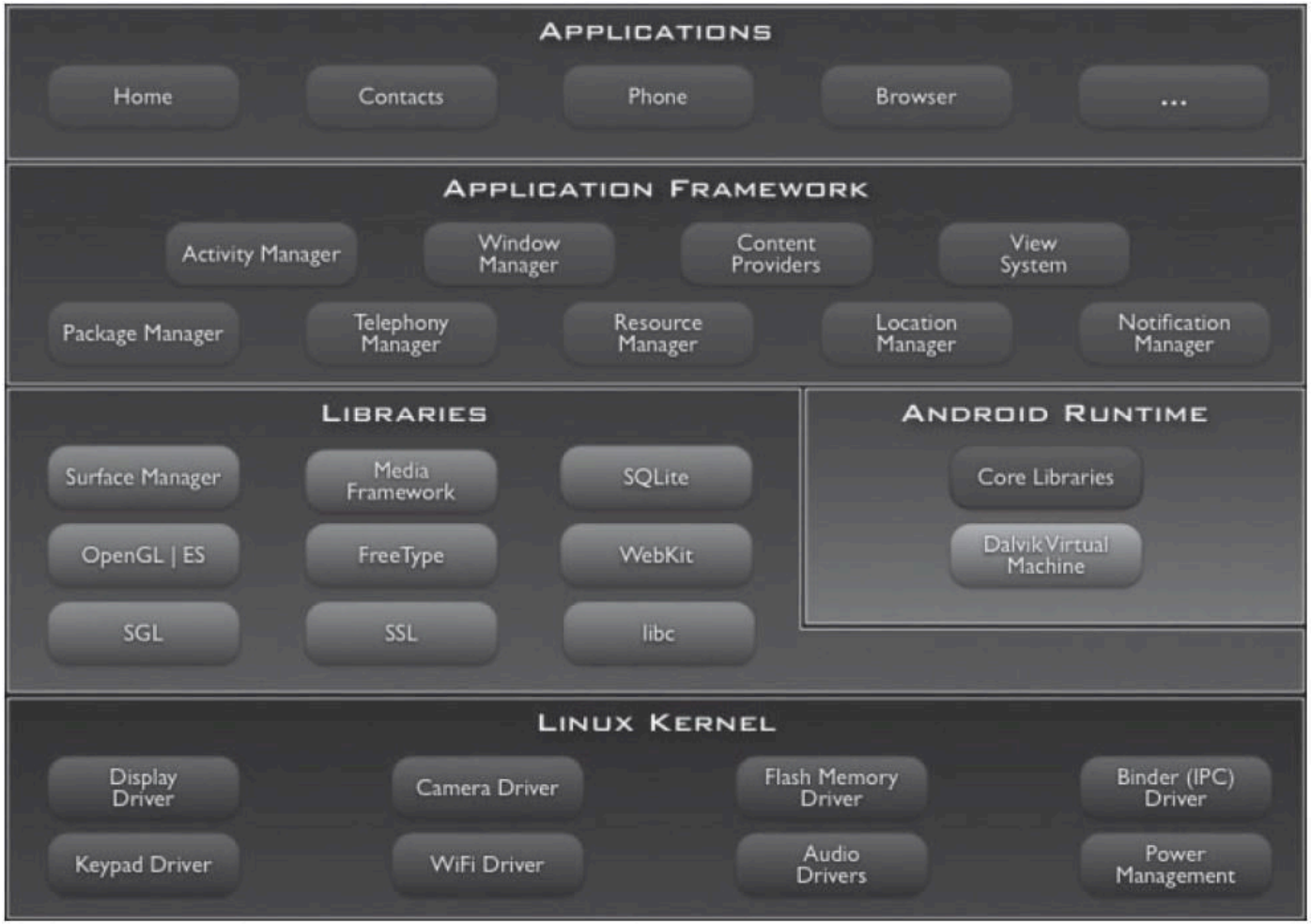
..Components of the Android OS

- Runtime environment
  - Consists of sets of libraries that provide all the features available in JAVA libraries in the OS
  - The DVM works by interpreting and translating Java code into a language understood by the OS.

..Components of the Android OS

- The Linux kernel
  - Acts as an abstraction layer between the hardware and software stack.
  - Responsible for device process management, memory management, network management and system security.

..Components of the Android OS



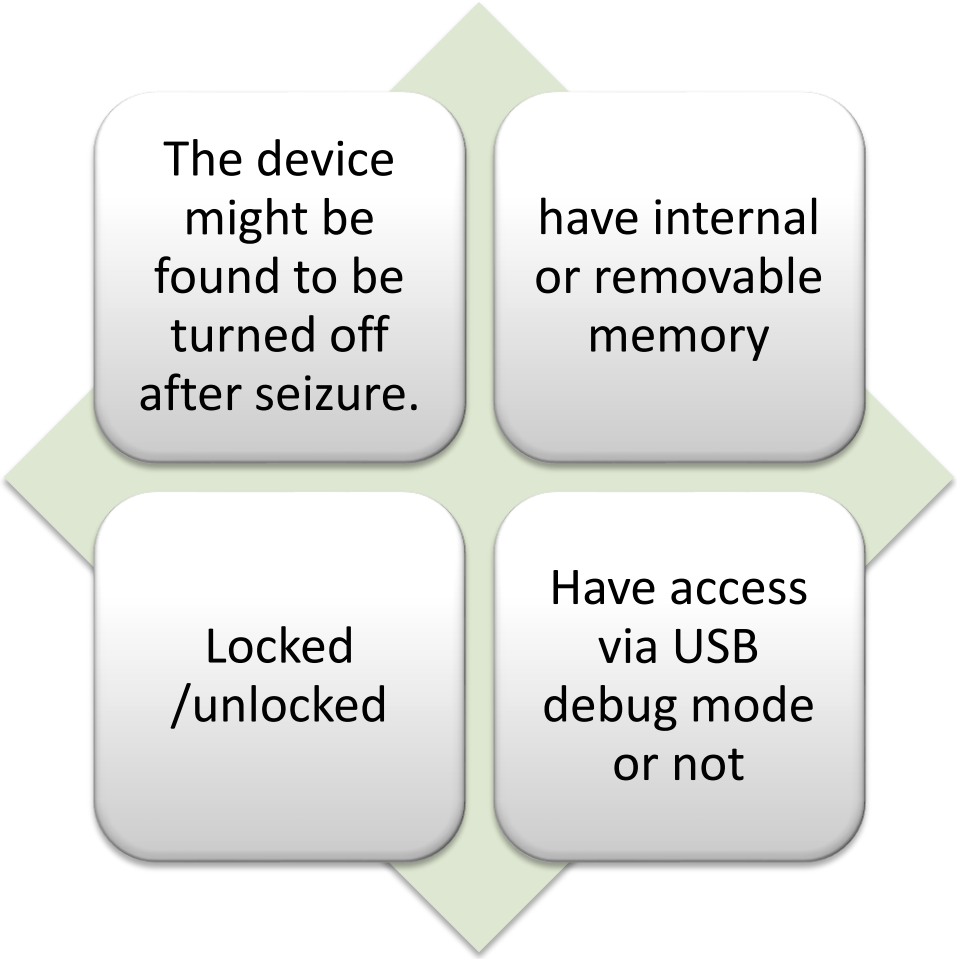
Challenges of Android Forensics

- Most Android devices adopts yet another flash file system 2 (YAFFS2), which is mostly not compatible to major forensic tools available.
- Android uses sandbox concept.
- Makes use of SQLite database
- Android debug bridge provides interface to an android phone using a computer

..Challenges of Android Forensics

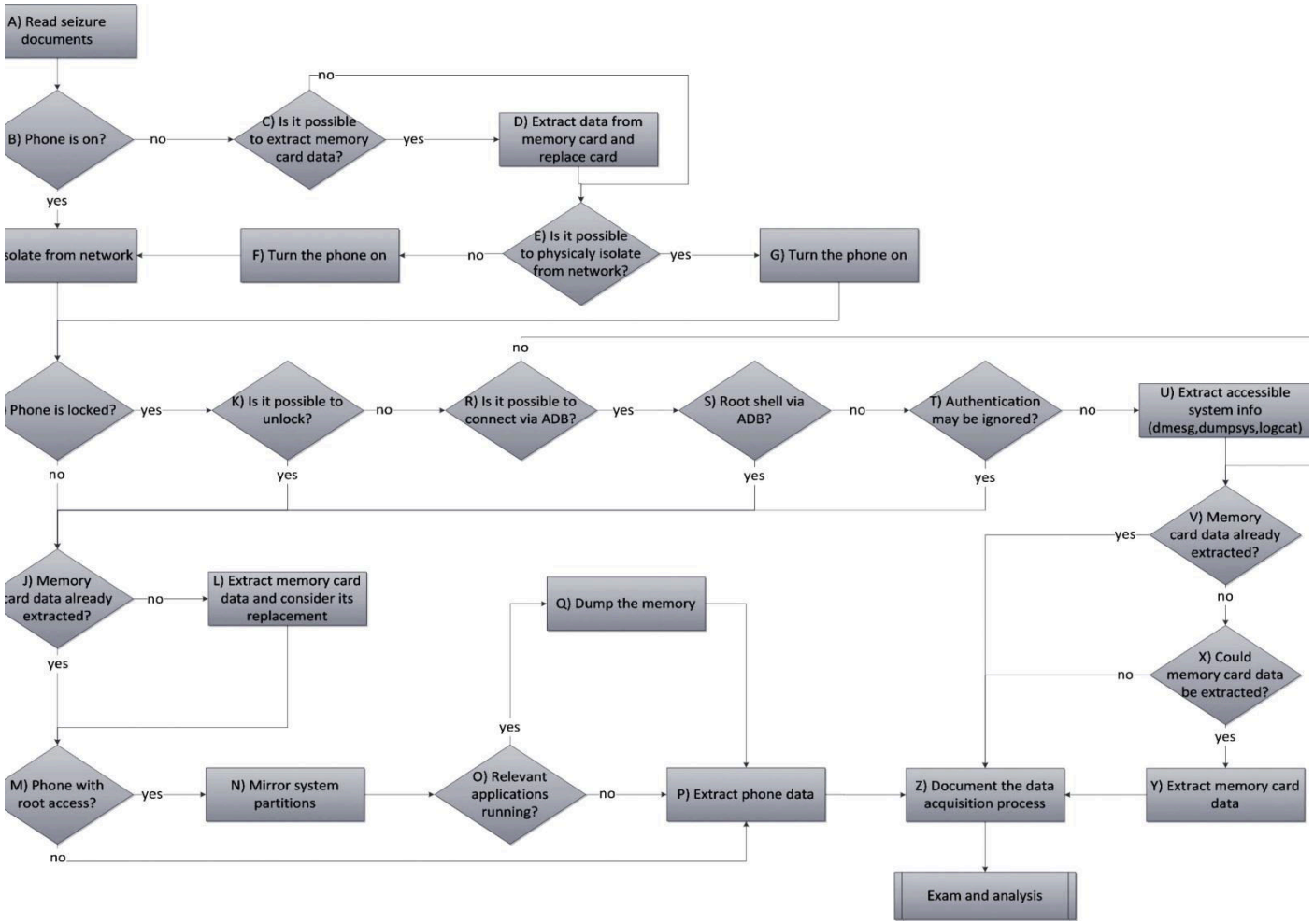
- Access to system partitions is restricted to the Android OS.
- Techniques for obtaining root privilege differ depending on Android version, device manufacturer and model.
- The OS has Authentication mechanisms that uses passwords, tactile patterns or biometric information

scenarios


The device might be found to be turned off after seizure.
have internal or removable memory
Locked /unlocked
Have access via USB debug mode or not

Data acquisition Method for android Smart phones

- The aim is to obtain maximum information from the mobile device, and the evidence is:
  - Well documented
  - Preserved
  - Processed in the safest and least intrusive manner

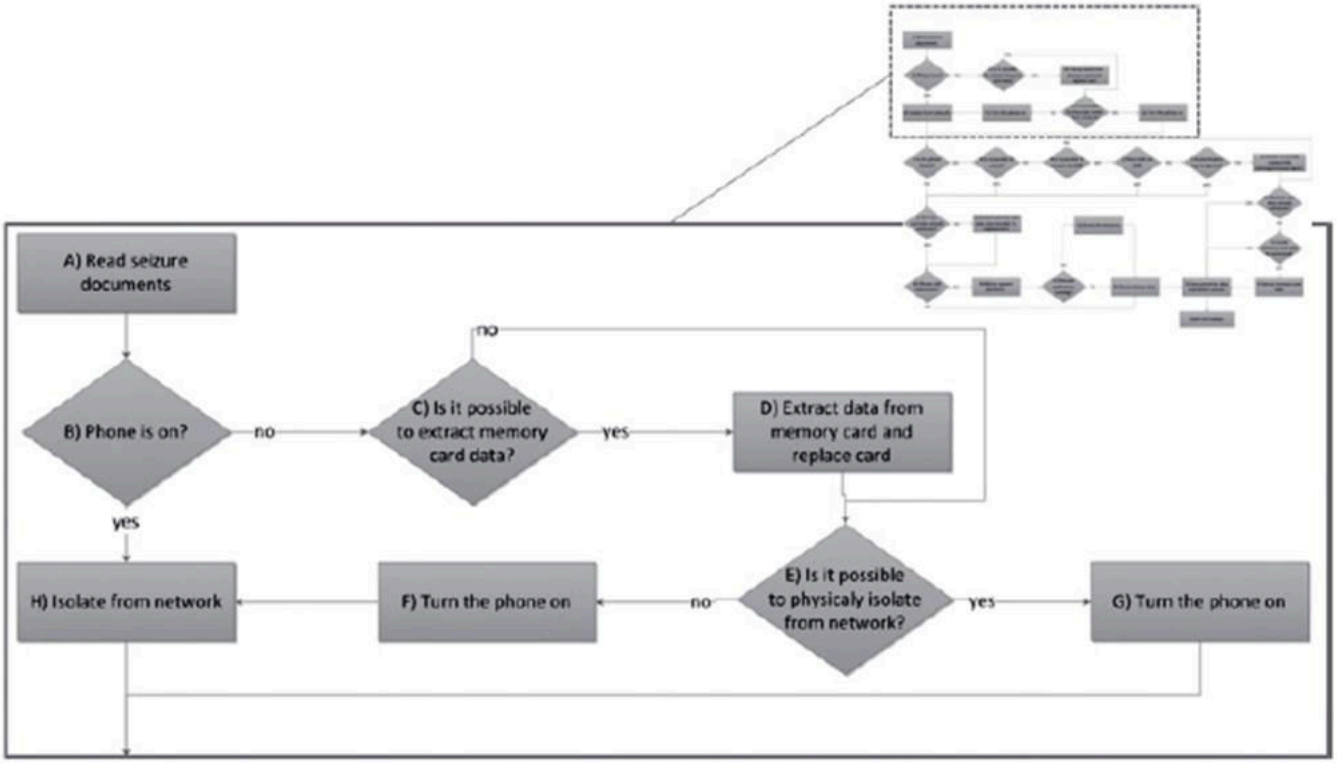


Workflow process

Initial procedure for data preservation in a Smartphone

- Main steps
  - Check if phone is ON/OFF
  - Possibility of extracting data from its memory card
  - Does it have a removable MMC?
  - Isolation from network
    - Either using a room with physical isolation from EM signals
    - Or switch the phone to offline mode

..Initial procedure for data preservation in a Smartphone



Smartphone without access control

- Least complex situation
- Make sure memory card data has been extracted and replaced.
- In the case of the memory card is not replaceable, removable data should still be mirrored prior to system mirroring and copying.
- Runtime information should be documented as well

..Smartphone without access control

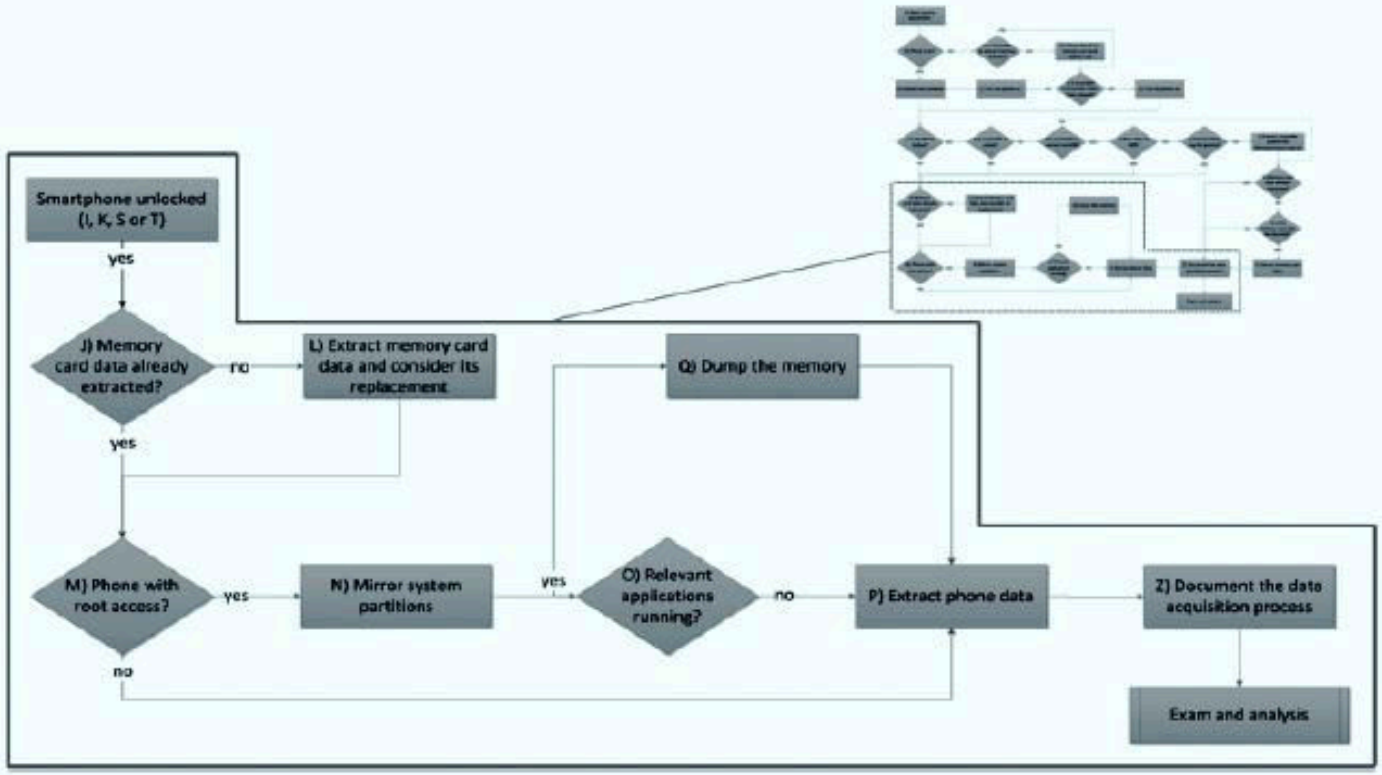


Figure 5. teps of data acquisition of an Android smartphone without access control.

..command to list connected devices, display partition information, and generate the partitions dumps.

```
C:\Android\android-sdk\platform-tools>adb devices
List of devices attached
040140611301E014        device
C:\Android\android-sdk\platform-tools>adb -s 040140611301E014
shell
$ su -
su -
# mount | grep mtd
mount | grep mtd
/dev/block/mtdblock6 /system yaffs2 ro,relatime 0 0
/dev/block/mtdblock8 /data yaffs2 rw,nosuid,nodev,relatime 0 0
/dev/block/mtdblock7 /cache yaffs2 rw,nosuid,nodev,relatime 0 0
/dev/block/mtdblock5 /cdrom yaffs2 rw,relatime 0 0
/dev/block/mtdblock0 /pds yaffs2 rw,nosuid,nodev,relatime 0 0
# cat /proc/mtd
cat /proc/mtd
dev:    size   erasesize  name
mtd0: 00180000 00020000 "pds"
mtd1: 00060000 00020000 "cid"
mtd2: 00060000 00020000 "misc"
mtd3: 00380000 00020000 "boot"
mtd4: 00480000 00020000 "recovery"
mtd5: 008c0000 00020000 "cdrom"
mtd6: 0afa0000 00020000 "system"
mtd7: 06a00000 00020000 "cache"
mtd8: 0c520000 00020000 "userdata"
mtd9: 00180000 00020000 "cust"
mtd10: 00200000 00020000 "kpanic"
# ls /dev/mtd/mtd*
ls /dev/mtd/mtd*
…
/dev/mtd/mtd6
/dev/mtd/mtd6ro
/dev/mtd/mtd7
/dev/mtd/mtd7ro
```

```
/dev/mtd/mtd8
/dev/mtd/mtd8ro
…
# dd if=/dev/mtd/mtd6ro of=/mnt/sdcard/mtd6ro_system.dd bs=4096
dd if=/dev/mtd/mtd6ro of=/mnt/sdcard/mtd6ro_system.dd bs=4096
44960+0 records in
44960+0 records out
184156160 bytes transferred in 73.803 secs (2495239 bytes/sec)
# dd if=/dev/mtd/mtd7ro of=/mnt/sdcard/mtd7ro_cache.dd bs=4096
dd if=/dev/mtd/mtd7ro of=/mnt/sdcard/mtd7ro_cache.dd bs=4096
27136+0 records in
27136+0 records out
111149056 bytes transferred in 41.924 secs (2651203 bytes/sec)
# dd if=/dev/mtd/mtd8ro of=/mnt/sdcard/mtd8ro_userdata.dd
bs=4096
dd if=/dev/mtd/mtd8ro of=/mnt/sdcard/mtd8ro_userdata.dd bs=4096
50464+0 records in
50464+0 records out
206700544 bytes transferred in 74.452 secs (2776292 bytes/sec)
# ls /mnt/sdcard/*.dd
ls /mnt/sdcard/*.dd
mtd6ro_system.dd
mtd7ro_cache.dd

mtd8ro_userdata.dd
```

..command to copy runtime data

- This is the data used by running applications
- Could be helpful in obtaining passwords and cryptographic keys
- Processes have to be 'killed' before they are copied.

```
# chmod 777 /data/misc
chmod 777 /data/misc
# kill -10 6440
kill -10 6440
# kill -10 6379
kill -10 6379
# kill -10 6199
kill -10 6199
# kill -10 5797
kill -10 5797
# ls /data/misc | grep dump
ls /data/misc | grep dump
heap-dump-tm1303909649-pid5797.hprof
heap-dump-tm1303909632-pid6199.hprof
heap-dump-tm1303909626-pid6379.hprof
heap-dump-tm1303909585-pid6440.hprof
#
…
C:\android-sdk\platform-tools>adb -s 040140611301E014 pull /
data/misc/heap-dump-tm1303909649-pid5797.hprof
2206 KB/s (2773648 bytes in 1.227s)
C:\android-sdk\platform-tools>adb -s 040140611301E014 pull /
data/misc/heap-dump-tm1303909632-pid6199.hprof
2236 KB/s (3548142 bytes in 1.549s)
C:\android-sdk\platform-tools>adb -s 040140611301E014 pull /
data/misc/heap-dump-tm1303909626-pid6379.hprof
1973 KB/s (3596506 bytes in 1.779s)
C:\android-sdk\platform-tools>adb -s 040140611301E014 pull /
data/misc/heap-dump-tm1303909585-pid6440.hprof
1968 KB/s (2892848 bytes in 1.435s)
```

…

- Analysis is done with tools that are able to mount images having the device's file system
- The logical data should also be backed-up
- When 'Super user' privileges are not enabled, data is extracted from its internal memory by visually inspecting the GUI

```
C:\android-sdk\platform-tools> adb pull /data pericia/
Pull: building file list…
…
684 files pulled. 0 files skipped
857 KB/s (194876514 bytes in 226.941s)
```

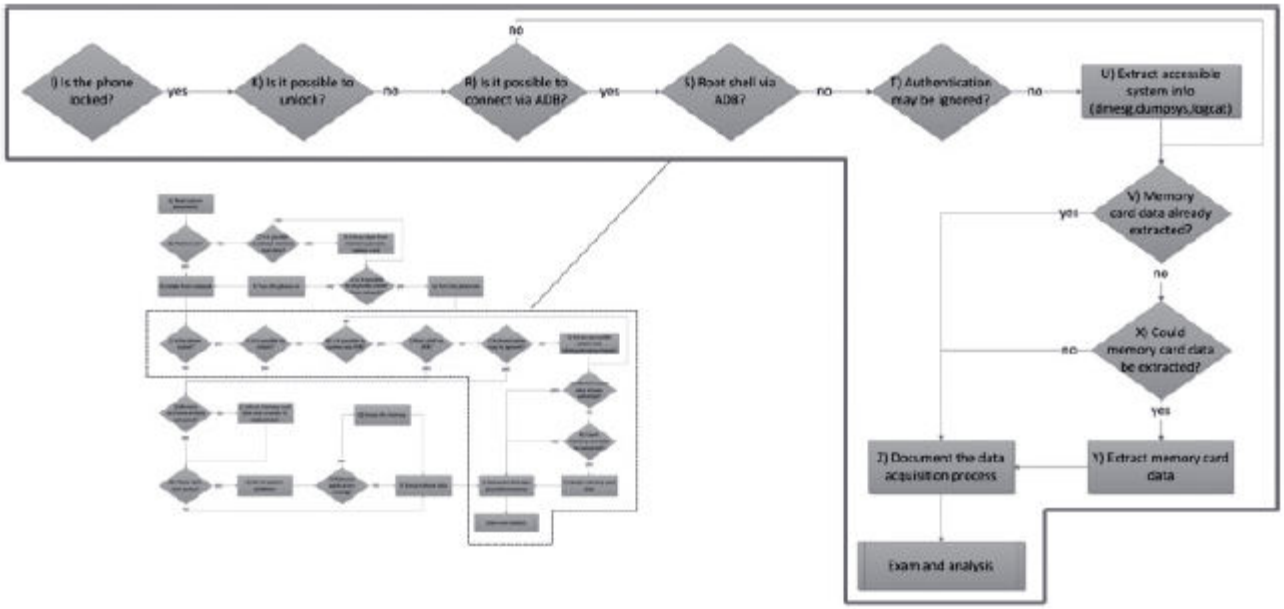
Smartphone with access control

- Lock could be by password or tactile pattern
- Three ways are suggested by NIST
  - Investigative method
  - Hardware access
  - Software access (easiest)
- It should be done in a less intrusive manner
- Smudge left on the phone screen/keypad can give clues

..Smartphone with access control

- If bypass does not succeed then check for Android is configured to accept USB debugging using ADB
- Super user privilege is then obtained and acquisition method is resumed.
- If no super user privilege, then we use the ADB to install a screen unlock software.

..Smartphone with access control



Installing application via ADB

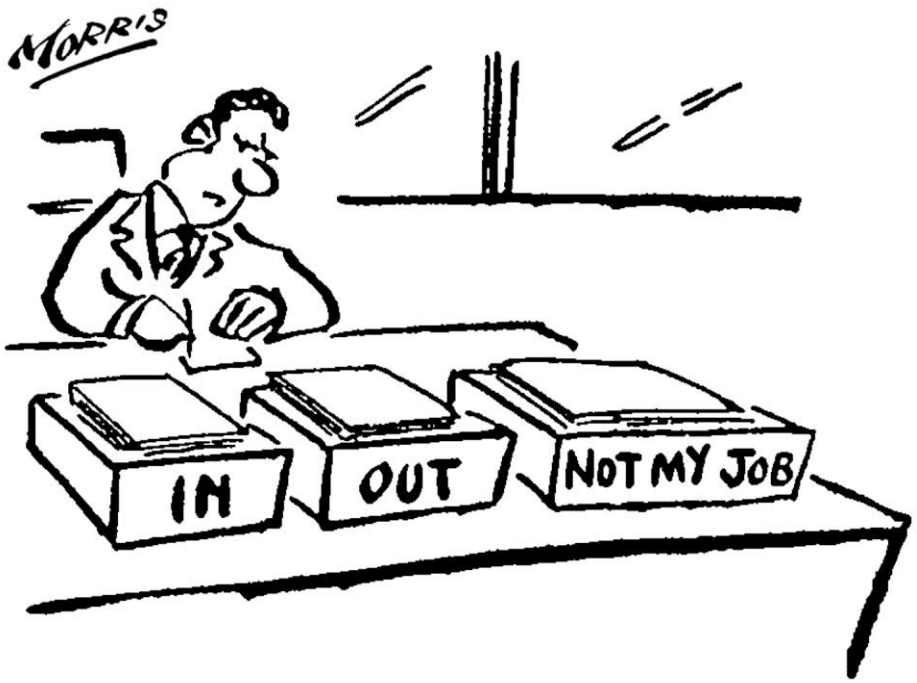
```
C:\android-sdk\platform-tools>adb -s 040140611301E014 shell
$ su -
su -
Permission denied
$ exit
...

C:\android-sdk\platform-tools>adb -s 040140611301E014 install
screenlockbypass.apk
224 KB/s (22797 bytes in 0.100s)
        pkg: /data/local/tmp/screenlockbypass.apk
Success

C:\android-sdk\platform-tools>adb -s 040140611301E014 install
AndroidForensics.apk
716 KB/s (31558 bytes in 0.046s)
        pkg: /data/local/tmp/AndroidForensics.apk
Success
```

Figure 10. Connection via ADB, root access check and application installation in order to ignore access control.

Acquisition Documentation



..Acquisition Documentation

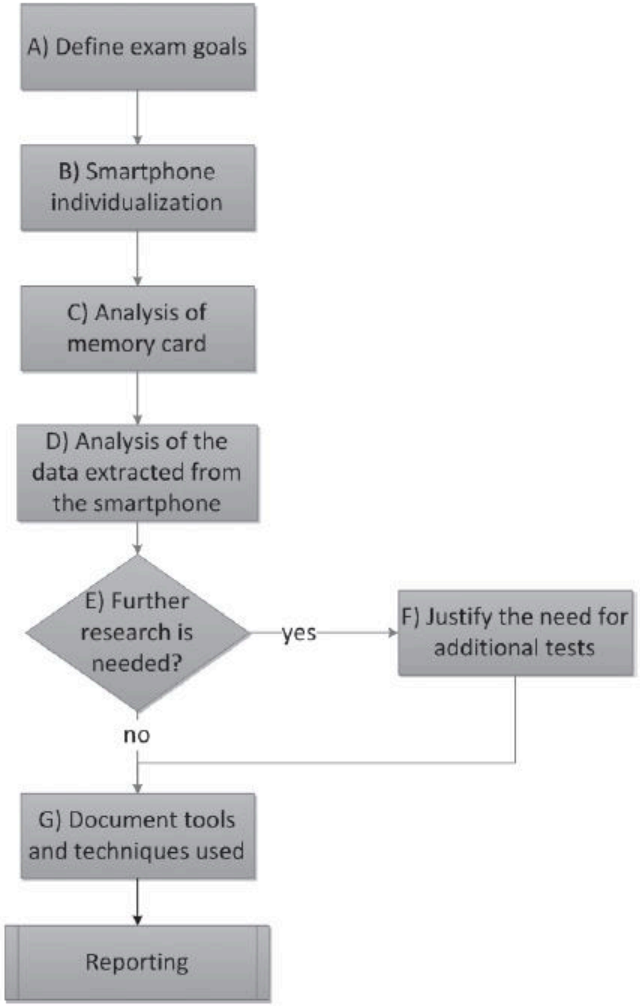
- Documentation should be done for all techniques and procedures carried out
- Enables auditabilty and reliability.
- State any caveats
- Register HASH codes
- Documentation is very important!!!!!!

Examination and analysis

..Examination and analysis

- Define Goals of the investigation
- Smartphone individualization
- Device data analysis
  - Use tools such as SQLite database, Hex editors to study the data

..Examination and analysis



jure 11. Workflow of the examination and analysis process.

The proposed method Validation

- Six Android phones were used, among these handsets, four different scenarios were identified

..validation

Scenario	Turned on	Removable card	Locked	Unlockable	Super user
1st Scenario (Motorola Milestone II A953)	No	Yes	Yes	Yes	No
2nd Scenario (Sony Ericson Xperia X10 miniPro)	Yes	No	No	Does not apply	No
3rd Scenario (Motorola Defy) (Samsung Galaxy S 9000[a])	No	Yes	No	Does not apply	Yes
4th Scenario (Motorola I1) (Motorola Milestone A853)	No	Yes	No	Does not apply	No

[a] In addition to the removable microSD card, that phone has a built-in memory card which is not removable

1st scenario

Device is turned OFF

Its memory card is removed

A memory card holding the copy was inserted

Phone is then switched ON and set to flight mode

Cell phone is locked, but USB debugging access is enabled, using ADB tool shell was obtained

But no super user privilege-preventing mirroring system partitions

Form the ADB it was possible to install “screen lock bypass”, then the ‘logical android application forensics’ was used to extract data.

..1st scenario

- In the exam and analysis, the phone was individualized via its Google account.
- Images were obtained from the memory card
- Little SMS was received/sent, calendar entries are obtained
- Used tools and techniques were documented

2nd scenario

- Smartphone was not locked
- Was put into flight mode immediately
- MMC was not removable thus it was mirrored(copied entirely)
- And then its own memory was used to extract it's information using "logical Android Application Forensics"
- Data is extracted as in scenario 1

.. 2nd Scenario

- There were in the Bluetooth 60 ‘vcf’ files business cards- there was a file named ‘home.vcf’
- Two photographs were obtained which has metadata of geographical location.
- Several received and missed calls were obtained.
- Because the user is an average Smartphone user, no further investigation is carried out.
- The methods are then documented.

3rd Scenario

MMC is removed and replaced by a mirror

↓

device is then turned ON and immediately put in flight mode.

↓

Device is unlocked, and has a second MMC, it was also mirrored.

↓

The Smartphone has super user privileges, thus, system, user data and cache partitions mirrored

↓

Logical extraction was also done.

..3rd Scenario

- Then the Cellebrite UFED system 1.1.7 tool was used to extract forensic data from the phone.
- The system, cache, and user data partition mirrors were examined in FTK, with the data carving option.
- Limitation is there is no support for YAFFS2
- Logical analysis of the data copied in directory /data/system, the list of applications installed were obtained in file (package.list)

..3rd Scenario

The account set up for the Google phone with encrypted password (accounts.db) file.

In the /data/misc folder, Wi-Fi settings and WPA2 passphrases were found stored in clear text in the “wpa_supplicant.conf” file

Cache file retrieved from /data/data, payment and money transfer receipts, current account statements.

Credit card limits were found in the “br.com.bb.android” application data.

..3rd Scenario

The phone had the “seek Droid” (“org.gtmedia.seekdroid”) application, which allows location blocking and data deletion remotely via www.seekdroid.com website.

In the “prefs.xml” file was found, which contained its configuartion,username and password.

The “Gtalk” application provided in the “talk.db” file, chat history and friends list were obtained

Information about sent and received e-mails, along with date, times, sender and recipient were obtained from the “mailstore.<googleusername>@gmail.com.db” file of the "com.google.android.gm" application

3rd Scenario

SMS messages were stored in the "mmssms.db" file of the "com.android.providers.telephony" application.

Calendar events were found in the "calendar.db" file of the "com.android.providers.calendar" application.

From the "webview.db" file of the "com.android.browser" application we found the phone user had logged on websites such as Facebook (http://m.facebook.com), Yahoo (http://m.login.yahoo) etc.

..3rd Scenario

From the "DropboxAccountPrefs.xml" file of the "com.dropbox.android" application, the configured user name.

Also the "db.db" file had directories and files list, with their respective sizes.

The system configurations were found in the "settings.db" file of the "com.android.provider.settings" application.

4th Scenario

The memory card was removed, mirrored and replaced while the device was still turned off.

phone was turned on and immediately put into flight mode.

The phone was unlocked-the Cellebrite UFED System 1.1.7 tool was used to extract forensic data from the phone.

The data were examined and analyzed and the procedures were documented.

Conclusion/Future work

- Android Smartphone platform is already the most present among mobile communication devices.
- Existing approaches to forensic examine cell phones and computers are not adequate to the peculiarities of Android Phones.
- specific method was proposed to address data acquisition of devices that use the Android Platform

Conclusion/Future work

- Account was taken of the operating system characteristics, its most popular applications and hardware features.
- It was possible to foresee the difficulties forensic experts might face.
- method was proposed in a broad fashion, so that as technology progress they just fit into the framework.

Conclusion/Future work

- Proposed method was validated by its application onto the examination of six Android smart phones.
- Grouped into four scenarios, involving different situations that an analyst might encounter.
- For future work, it is suggested that the method be validated for the Android 3

Conclusion/Future work

- Evaluating its effectiveness in the Google system for tablet devices.
- And also android 4.x
- Thus making the adjustments that may be required
- Creation of a forensic tool that supports the YAFFS2 file system, focuses on NAND flash memory,
  - facilitating data extraction and access and also mounting images from those storage media.

Questions