AndroKit: A Toolkit for Forensics Analysis of Web Browsers on Android Platform

Muhammad Asim, Muhammad Faisal Amjad, Hammad Afzal, Mian Waseem Iqbal, Haider Abbas

***Abstract*—Due to the pervasive nature of smart phones and devices, users are becoming more and more dependent on such devices for accessing online information. In addition to custom applications, web browsers have become a primary means for accessing information provided on Internet or file systems in private networks. The basics of web browser forensics revolve around artifacts such as web sites visited, malicious URLs, time stamps, counts of access, search histories, cookies, downloaded activity and the potential to rebuild web pages from cached files. However, leveraging and locating this information can be challenging without the needed prerequisite information. Objective of this paper is to perform forensics analysis of data structures used by popular web browsers (Chrome, Opera, Mozilla Firefox, and Dolphin) on Android and how a forensics investigator can acquire forensics artifacts from web browsers (For analysis, CHCS (Cloud Health Care Service) on android web browsers were examined for forensic artifacts). To strengthen digital investigation, a toolkit named as AndroKit is proposed for Android web browsers forensics (Especially for mHealth forensic). The paper demonstrates that the AndroKit can successfully acquire and analyze forensics evidence such as Web History, Downloads, Cookies, Bookmarks, Chrome stored user credentials, decode base64 encoded images, Tabs information etc. Finally, a comparative analysis of AndroKit with standard forensics tool-kits such as Oxygen forensics, Andriller, MOBILedit and Belkasoft evidence center is presented.**

***Index Terms*—Forensic Investigation, Web Browsers Analysis, Android Operating System, Cloud Health Care Service, Mobile Health,**

I. Introduction

Since last few years, the usage of of smart-phones is increasing day by day. Smart phone overtook the laptop as the most popular device to get online in 2015 and its usage is increasing with every passing year. The latest report from Ofcom, i.e. The Communication Market report 2017, shows that 76% of adults (in UK) own a smart-phone whereas, the ownership of laptops is 64% and that of tablets is 58%. This results in smart-phones being the most popular device to get online as well. More than four in ten users (42%) use smart devices to get online for various activities including browsing internet, shop online and using social media etc. The increase in smart-phone surfing marks a clear change since 2014 [1]. Figure 1 shows that the usage and popularity of smart phones has clearly overtaken the usage of other devices during last few years.

The popular operating systems on smart phones include Apple's iOS, Google's Android phone, Microsoft's Windows, Symbian, RIM etc. During last few years, Android has been the most used operating system on smart devices (phones, tablets etc). Developed by Google, Android is based on a modified version of the Linux kernel and other open source software. The first release of Android was made in September 23, 2008 whereas, the most recent version is released in December 5, 2017. The graph in Figure 2 shows that the popularity of Android has been continuously increasing since its first release [2]. According to a recent report from StatCounter Global Stats [1], Android has overtaken Windows as the most popular operating system to access the Internet in 2017. This is also indicative of the fact that smart phones are getting popular as the primary device to gain access to Internet as compared to traditional devices such as laptops, desktops etc.

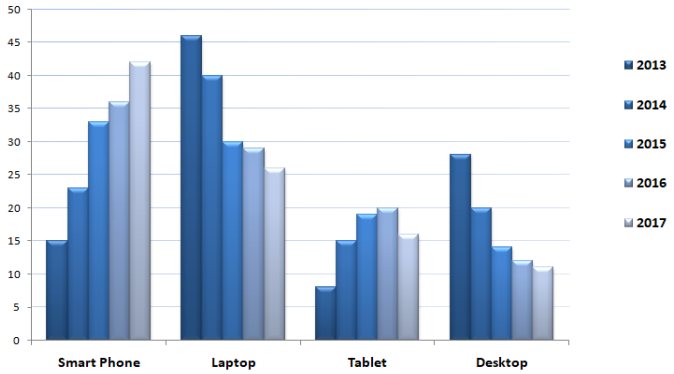

Fig. 1. The increased usage of Smart Phones since 2013 as shown in Ofcom Report 2017

In digital forensic investigations, the term *mobile forensics* refers to the recovery of digital evidence from mobile devices. Forensics data acquisition from mobile device is different from Desktop systems. The main difference is that some forensics tools require a communication vector, thus standard write protection does not work during data acquisition process. Forensic acquisition methods may involve removing a chip or installing a boot-loader on the mobile device prior to extracting the data in forensics sound manner for analysis. Some challenges, specifically related to Mobile forensics, are *hardware difference*, *mobile operating system*, *mobile platform security features*, *lack of resources*, *dynamic nature of evidence*, *lack of availability of tools* and *legal issues* etc. In this paper, we are focusing on getting information regarding CHCS (Cloud

Muhammad Asim, Muhammad Faisal Amjad, Hammad Afzal, Mian Waseem Iqbal and Haider Abbas are with the National University of Sciences and Technology (NUST), Islamabad, Pakistan e-mail: haider@mcs.edu.pk

Manuscript received

[1] http://gs.statcounter.com/press/android-overtakes-windows-for-first-time

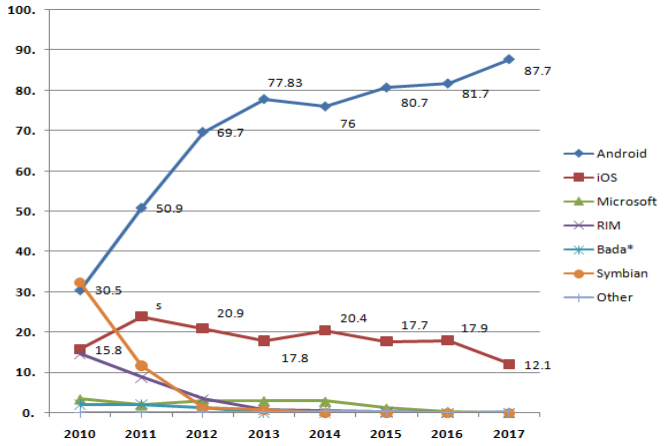

Fig. 2. Mobile Operating System Market Share - 2010 to 2017

Health Care Service) from Web browsers on Android operating system.

The objective of this paper is to find that how popular browsers in Android OS store data in memory that can be utilized in forensics process. The secondary objective is that, how a forensics investigator can acquire and analyze web browsers forensic artifacts (i-e mHealth care forensic information) on Android devices. To automate evidence extraction and analysis process, a forensic toolkit (AndroKit) is designed and proposed for android web browsers forensics.

A. Motivation

On Desktop platforms such as Windows, Linux and Mac-OS, several forensics tool-kits and published research literature is available for web browsers forensics. However, very little research has been published related to web browsers forensics on Android platform. Standard forensics tool-kits provide limited support on Android OS for web browsers forensics (discussed in comparative analysis). Since the popularity of smart phones as the most widely used device to access Internet, a forensic toolkit for web browsers on Android platform is required to facilitate the digital forensics investigators.

B. Contributions:

This paper presents the process and data structures involved in forensic evidence acquisition and analysis for web browsers on Android platform. The contributions are enlisted below:

- The proposed system will help digital forensic industries in strengthening there technology in performing Android web browsers forensics such as acquisition of Web History, Bookmarks, Cookies, Tabs information, Chrome stored user credentials, Decoding of Base64 Encoded images etc.
- Web browsers vendors may also get benefits from this research in strengthening user data privacy in their technologies.

The paper is arranged in VII sections. Section II presents the related work in android web browsers forensics and evidence extraction methods from android device. Section III discusses mobile health care systems (mhealth). Section IV discusses data structure used by popular Android web browsers. AndroKit Design and Implementation is discussed in Section V. Comparative analysis of AndroKit with other standard forensic tool-kits (Oxygen forensics, Andriller, MOBILedit and Belkasoft evidence center) is performed in Section VI, and finally, Section VII concludes the paper.

II. Mobile Health Care

The use of smartphones and other communication devices to monitor patient health and educate consumers about health services are known as ”mHealth” (mobile Health applications). mHealth applications is increasing popularity due to their ability to store, access and transmit user health information. Sensitive and personally identifiable information (e.g personal information, geolocation data etc) are potentially an important source of digital evidence. mHealth and its privacy are rapidly increasing an effective influence on the healthcare systems. The most visible element of mHealth is the profusion of the smartphone’s application. One of the barriers most cited by patients and medical professionals is lack of adequate privacy, security policies and regulation for mHealth apps.

MCC (Mobile Cloud Computing) is an integration of CC (Cloud Computing) into mobile devices. Emerging development in CC and smartphone technologies have inspired different patterns of CHCS (Cloud Health Care Service) and devices. The aim of MCC in healthcare apps reduces the limitations of traditional medical applications (e.g. small storage, medical errors, security [3] [4]). In the cloud systems, the health-related information is stored and transmit to medical caregivers through web services. These web services may be accessible on the device via web browsers and other applications. On Client device, Web browsers also store internet history, cookies, bookmarks etc., and other useful information which help in digital investigations. This web browser storage data may be potential damage to user data privacy and security. However, a forensics investigator may get benefits from this available information.

In [5], a systematic review of healthcare applications for smartphones is performed. Total 83 applications were examined, in which 57 applications focused on disease diagnosis (21 on drug reference, 8 on medial calculators, 6 on the literature search, 3 on clinical communication, 4 on hospital information system (HIS), 2 on medical training and 7 were categorized as general healthcare application), 15 applications focus on disease management (6 for chronic illness, 4 for ENT related issues, 3 for fall-related and 2 for other conditions), and 11 applications focus on medical education for students and nursing. In [6], cloud-oriented middleware is implemented for reliable communication. The author proposed SOPHRA for high information availability. Currently, SOPHRA supports both RESTful and SOAP web services protocols and facilitate wireless communication over Wi-Fi using HTTP. For SOPHRA implementation and development, embedded browser pattern was employed to write single code which

can be deployed on multiple platforms (Enables cross-platform mobile application). In [7], the author presented pros and cons of native apps versus web apps for healthcare applications. In [8], the author investigates the security level in web-based application versus native application for mHealth. The author proposed Mina Vardkontakter framework for mHealth apps (web-based). In [9], the author describes security concerns and data privacy issues in healthcare systems. In [10], an overview of mHealth apps (Android and IOS) is presented with special focus on potential damage to users information, security, and privacy. During analysis, it was observed that; Most mHealth apps require access to more sensitive personal information which increases the potential damage to user data privacy and security. In [11], the author proposed a HealthCloud based on MCC and Android OS for mHealth care information management. In 2015 [12], hybrid cloud architecture based on cryptographic technique is presented for secure mobile health applications. [13] Proposed MCMAS (Medical Cloud Multi-Agent System) based commanding capability to cope with problems of traditional applications. Further, MCMAS performance is compared with the traditional systems. In [14], 40 popular mHeatlh application were examined on Android device for digital forensics artifacts. The researcher had successfully recovered user details, geolocation data and user credentials (e.g passwords and PIN numbers). In [15], the current state of privacy and security is investigated in mHealth application by forensic analysis. The author found that most of the applications didn't provide security and privacy for healthcare information. Based on his results, policy framework (Security and privacy) is proposed for mHealth applications.

The smartphone technology offers an innovative approach to address complex health concerns. Currently, in Android market (Playstore) many mHealth care applications are available to facilitate patients, doctors, and health service providers. However, a very little research has been conducted to address digital forensic artifacts in mHealth apps and services. The purpose of this study is to investigate Android web browsers for mHealth artifacts. Major findings of this study include; mHealth sessions on android web browsers, bookmarks, cookies etc. for forensics artifacts.

III. Related Work

Web browsing activity is a major source of information in digital forensics investigation [16]. Forensic analysis depends on the architecture of the web browser and thus forensic toolkits need to adapt their code to new versions or new browsers [17] with different platforms (operating systems). The forensic analysis of web browsers on traditional platforms such as Windows and Linux Based systems have been reported in many research works [27] [21] [22] [23] [25]; however, very little research on web browsers forensics on Android platform has been carried out.

In [27] author discussed how to analyze data of web browsers on different OS (Windows, Linux, and Mac-OS) and showed the usage of tools to examine the records in the web browsers. However, they focused on Desktop based operating systems and not the Android. In (2009), Pereira [18] explained the change in the history system that occurred when Firefox 2 was restructured and Firefox 3 was released. The author proposed a new method of searching the deleted history information with the help of *unallocated fields*. The author suggested a technique of extracting history from Firefox 3 by examining structure analysis of the SQLite database. However, their work is limited to only desktop applications. In another paper on Firefox forensics [26], authors presented a survey of web browser forensics analysis tools and evaluated their performance. Each forensic tool has its own strengths and weaknesses. However, the author found that "FoxAnalysis" is best suited for Firefox forensics analysis on Windows system.

In [24], forensic analysis of Chrome was performed on the Windows, Linux and Mac OSX system. Chrome was chosen for analysis on the basis of popularity among internet users. In this analysis, cookies, user profiles, prefetch file, RAM and web history is analyzed for forensic artifacts. Final results showed that forensic investigator can collect valuable information about suspect activities from chrome sessions.

In 2013, N. A. Barghouthy [19] performed forensic investigation of Android Private Browsing Sessions using Orweb on two Samsung Galaxy S2 smart-phones. One of the devices is rooted, whereas other device is non-rooted. Their results showed that the browser history, and important corroborative digital evidence can be tracked and found on rooted device. This research only consider private browsing session on Android device.

In [20], author suggested a digital forensic process for digital devices using social network services (SNSs). The author proposed digital forensics investigation methodology applicable for social network services but this research only covers social networking applications.

In 2017, [21] presented forensic analysis of Epi Privacy Browser, performed on Windows Operating System (7 and 10). Epic Privacy Browser prides itself on protecting the user's privacy. Temporary files and folder get deleted at the end of the browsing session. However, this paper demonstrated that this data can be recovered using standard toolkits. Web browsers stores users data in different ways and locations, this depends on the operating system. In [22], the author analyzed how commonly used web browsers store user data and what forensic information can be recovered on different operating systems (such as Windows, Linux, and Mac OSx). They demonstrated the usage of standard toolkits such as Internet Evidence Finder, FTK, Browser History Examiner. Further, the author presented the working of major forensics tools such as WebHistorian 1.3, Index.dat, Analyzer 2.5, ChromeAnalysis Plus, NetAnalysis v1.52 and Web Browser Forensic Analyzer to analyze the web browsers forensics on the windows system. They performed analysis on Cookies, cache, bookmarks, history, search words and download lists. Forensic tools successfully recovered this forensic evidence.

In 2011 T. Vidas [28] discussed a general methodology for the examination and collection of evidence from Android devices. Similarly, forensic analysis of social media applications

such as WeChat [29], WhatsApp [30], Viber [31] and other IM applications [32] on Android system was performed. Standard tools also provide limited support.

A. Popular Forensics Tools

1) Oxygen Forensic Suite: Oxygen Forensic Suite (OFS) 2014, developed by *Oxygen Software* is a mobile forensic software. It can be used to analyze not only the cell phones but PDAs as well. OFS can extract contacts, calendar events, SMS messages, files, event logs, device information and metadata related to these artifacts. According to vendor's claim, the suite support more than 8,400 devices. The range of devices spans all popular brands including Nokia, Vertu, Sony Ericsson, Samsung, Motorola, Blackberry, Apple iPhone series, Apple iPod Touch, Apple iPad, Panasonic, Siemens, HTC etc. In terms of operating systems, OFS supports all popular operating systems including Symbian OS, Android OS devices, iOS and Windows Mobile 5/6, Phone 8 etc. OFS uses advanced proprietary protocols to access devices.

2) Andriller: Andriller provides a number of forensic tools for smartphones. It can be used with Android devices. The features supported by Andriller include Lockscreen cracking for Pattern, PIN code, or Password; custom decoders for Apps data from Android databases as well as some Apple iOS Windows for decoding communications. Tool produces extraction and decoding reports in HTML and Excel formats.

3) MOBILedit: MOBILedit Forensic (MF) is a product of Compelson Labs. MF can be used for a number of forensics activities such as searches, examines and report on data from GSM/CDMA/PCS cell phone devices. MF can be connected to smart phone device using a number of interfaces including an Infrared (IR) port, Wi-Fi, cable interface or a Bluetooth link. MF stores data in the .med file format after accessing it from devices. Following fields are populated with data: subscriber information, device specifics, Phonebook, SIM Phonebook, Missed Calls, Last Numbers Dialed, Received Calls, Inbox, Sent Items, Drafts, Files folder. Items present in the Files folder, ranging from Graphics files to Camera Photos and Tones, depend on the phones capabilities.

4) Belkasoft Evidence Center: Belkasoft Evidence Center Toolkit (BECT) extracts digital evidence from multiple sources by analyzing volatile memory dumps, hard drives, backups of iOS, Blackberry and Android backups, UFED, JTAG and chip-off dumps. BECT can be used to quickly locate and analyze information found in social network remnants, instant messenger logs, internet browser histories, mailboxes of popular email clients, peer-to-peer data, multi-player game chats, office documents, pictures, videos, encrypted files, mobile backups, system and registry files.

The summary of the literature review is that, there has been substantial work reported for web browser forensics on desktop based operating systems, however, very little work has been reported for Android. Moreover, most of the papers focused only on a single browser. In this paper, we have presented a detailed analysis on four popular web browsers on Android. Based on analysis results, a toolkit named as AndroKit is designed and presented. The performance of Androkit is compared with standard toolkits. We have also presented a brief overview of some standard toolkits with which, the performance of Androkit is compared. Further, CHCS (Cloud Health Care Service) were examined for digital artifacts.

IV. Analysis: Data Structure used by popular Android Web Browsers

We have performed the analysis of data structures and forensics analysis of popular android web browsers following devices:

- Samsung Alpha - Android OS version 4.4
- Samsung Grand Prime - Android OS Version 5.0
- Android Emulator SDK V21

All popular web browsers available on Android are installed on above mentioned devices and Emulator. In particular, following browsers are installed:

- Google Chrome Version 56.0.2924.87
- Opera Version 42.3.2246.113338
- Mozilla Firefox Version 53.0.2
- Dolphin Version 11.5.19

In Android OS, application data is stored in *"/data/data"* directory. This directory is accessible only with root privileges. During our analysis, we have used Flash Custom Recovery technique to gain root access on the device [33]. After gaining root access via stock recovery mode, the directories shown in Table I are pulled via *adb* from the device for forensic and data structure analysis. In this analysis, all popular web browsers' application data and users data is examined. This data would help digital forensics investigators in getting artifacts from web browsers session on Android device.

TABLE I
The Location of Popular Web Browsers Application and User Data

Web Browser	Path
Google Chrome	*/data/data/com.android.chrome/*
Opera	*/data/data/com.opera.browser/* */sdcard/Android/data/com.opera.browser*
Mozilla Firefox	*/data/data/org.mozilla.firefox/* */sdcard/Android/data/org.mozilla.firefox*
Dolphin	*/data/data/mobi.mgeek.TunnyBrowser/* */sdcard/TunnyBrowser/* */sdcard/Android/data/mobi.mgeek.TunnyBrowser*

A. Cookies

Cookies are used for a variety of purposes such as *tracking the identity of users*, *recording user preferences* and *preserving session information between multiple page requests* etc. Cookies stored information can be very helpful in digital forensics investigations. *"HTTP Set-Cookie" headers* is used to pass cookies from server to a web browser. Android based web browsers store Cookies in SQLite database file. Table II shows on device cookies storage path of popular android web browsers. Cookies are stored in Cookies table inside Cookies SQLite database file. Structure of Cookies table is presented in Table III.

TABLE II
THE PATH OF COOKIES ON POPULAR WEB BROWSERS APPLICATION ON ANDROID

Web Browser	Path
Google Chrome	*/com.android.chrome/app_chrome/Default/cookies*
Opera	*/com.opera.browser/app_opers/cookies*
Mozilla Firefox	*/org.mozilla.firefox/files/mozilla/xxxx.default/&cookies.sqlite*
Dolphin	*/mobi.mgeek.TunnyBrowser/app_webview/cookies*

TABLE III
THE STRUCTURE OF COOKIES SQLITE3 FILE: FIELDS OF COOKIES TABLE

Item	Description
creation_utc	data/time creation
host_key	identify host
Name	Name of cookies.
Value	store value of cookie such as password value, login value etc.
Path	Path of cookies. If path is *"/*, this means cookies will accessible via all pages in domain. If path is set to /subfolder, then cookies will accessible to only subfolder webpages.
expire_utc	Cookies Expire time in UTC.
secure	Support Boolean value 1 or 0. 1 indicates that cookies can be sent only over encrypted (HTTPS) requests.
Httponly	Support Boolean Values 1 or 0. 1 tells the web browser that cookies should be only accessed by server, Restrict client side requests.
lastaccess_utc	Last access time of Cookies.
hasexpires	Support Boolean values, 1 shows cookies have expired.
Persistent	Support Boolean value 1 or 0. Persistent cookies expires at a specific date/time, if value is set to 1.
encrypted value	Encrypted Value of cookies

B. Bookmarks

Bookmark is a URI (Uniform Resource Identifier) that is stored in application data on device for later retrieval. Google Chrome and Opera store Bookmarks (Table IV) in plain text file using Bracket delimiters format [34] [35]. Bookmark file Bracket delimiters format structure is discussed in Table V. Mozilla Firefox and Dolphin store Bookmarks data in SQLite database file in (Table IV). We have included the capability of parsing both Bracket delimiters and SQLite files for retrieving forensics information in proposed AndroKit. As per our knowledge, the other standard forensics tools only parse SQLite files for bookmarks.

TABLE IV
THE PATH OF BOOKMARKS ON POPULAR WEB BROWSES ON ANDROID

Web Browser	Path
Google Chrome	*/com.android.chrome/app_chrome/Default/bookmarks*
Opera	*/com.opera.browser/app_opera/bookmark*
Mozilla Firefox	*/org.mozilla.firefox/files/mozilla/xxxxxxxx.default/ browser.db*
Dolphin	*/mobi.mgeek.TunnyBrowser/databases/browser.db*

C. Web History

Web history is another source of large amounts of forensics information during digital investigation. Web browsers use

TABLE V
THE STRUCTURE OF BOOKMARK FILE WITH BRACKET DELIMITERS

Item	Description
Checksum	bookmark file hash(for integrity).
Id	Unique ID of each record.
Date Added	Date and Time in UNIX format.
Name	Title of web page.
Type	Show bookmark type (normally it is URL)
URL	URL Address

different SQLite formats and locations for storing visited web history on the device (Table VI). Google Chrome and Opera store web history in "*History.db*" SQLite file. The main difference between Chrome and Opera "*History.db*" structure is that Opera stores downloads information in separate Bracket delimiters plain text file *Figure 3* and most visited websites details in "*databases/mostvisited.db*" SQLite file. History file has *meta*, *urls*, *visits*, *visit_source*, *keyword_search_terms*, *downloads*, *downloads_urls_chains*, *segments* and *segments_usage* tables. Description of download table is shown in (Table X) and visited URLs table in (Table VII). Table VIII shows the structure of URLs table, Table IX shows the information of searched keywords.

TABLE VI
THE PATH OF WEB HISTORY STORED BY POPULAR WEB BROWSERS ON ANDROID

Web Browser	Path
Google Chrome	"/com.android.chrome/app_chrome/Default/History"
Opera	"/com.opera.browser/app_opera/history.db"
Mozilla Firefox	"/org.mozilla.firefox/files/mozilla/xxxxxxxx.default/ browser.db"
Dolphin	"/mobi.mgeek.TunnyBrowser/databases/browser.db"

TABLE VII
THE STRUCTURE OF URLS TABLE (VISITED URLS)

Item	Description
Id	Unique ID of each record in table
url	Visited URL address.
Title	Title of web page.
visit_count	Total visits counts.
last_visit_time	Last Visit Time in UTC
favicon_id	Website favicon ID. A foreign key to the favicon table which stores the favicon for each URL
hidden	Indicates if the URL will be displayed by the auto complete function. A value of 1 will keep it hidden and 0 will display it.

TABLE VIII
THE STRUCTURE OF KEYWORD SEARCH TERM TABLE

Keyword_id	Unique ID of each record in table
url_id	Foreign key of url.
lower_term	store searched keyword

Mozilla Firefox and Dolphin web browser store web history in *History* table, Recent open tabs information in *recent_tabs* table, Most visited websites information in *top_sites* table and Searches keywords history in searches table of "*browser.db*"

Fig. 3. Opera Downloads Bracket delimiters file.

SQLite file. AndroKit analyzes all popular browsers web history and shows results in separate tab for each browser.

TABLE IX
THE STRUCTURE OF VISITS TABLE

Id	Unique ID of each record in table
url	Foreign key of urls table
visit_time	Webpage visit time (UTC).
from_visit	Stores the id from where the URL came from originally. If the URL does not have a referring URL this value is 0
transition	Value describes how URL was LOADED in Web browser [36].
segment_id	Stores the segment id. It is not clear what segments are. There are tables called segments and segment_usage in history. It stores the domain names of accessed URLs along with a total visit count
visit_duration	Total website visit duration (UTC).

D. User Credentials

Chrome, Mozilla and Dolphin uses SQLite file while opera uses "*.json*" file for storing user credentials. Chrome (version 56.0.2924.87) stores user credentials in plain text, while other browsers store this information in encrypted/encoded form. In Chrome, *Original_url* and *url_action* fields in Login table store website address with actionable page address. *User_element* stored element name of web page and its value is stored in user *name_value*. *Password_element* field store password element name of website and its value is store in *Password_value* field in BLOB format. To find this BLOB Value; Open *login_data* file in any Hex editor (Figure 4).

Mozilla Firefox stores passwords in encrypted form. The files used to encrypt/decrypt the passwords are *cert9.db*, *key4.db* and *pkcs11.txt*, which are also stored in "*org.mozilla.firefox/files/mozilla/xxxxxxxx.default/signons.sqlite*" directory. Dolphin store user credentials in encrypted form in "*mobi.mgeek.TunnyBrowser/databases/passowrd.db*"SQLite file. Opera store Username/Passwords in encoded form in "*/app_opera/prefs.json*" file. AndroKit has ability to extract chrome stored user credentials in plain text.

E. Cache

Web browsers cache store valuable information such as images, strings, visited websites, searches history etc. This data provides a lot of forensics information. Chrome

TABLE X
THE STRUCTURE OF DOWNLOAD TABLE

Item	Description
Id	Unique ID Field
current_path	save current downloaded image path
target_path	if user move downloaded file to another location then path of file stored in target_path
start_time	downloading/downloaded file starting time in UNIX format
recived_bytes	field shows Downloading/downloaded file total bytes received
total_bytes	show total file size in bytes.
State	status of downloading file, this is 1 for completed and 0 for incomplete file
interrupt_reason	if file is interrupt in downloading, interrupt_reason filed is used to stored interruption reason.
end_time	End time show time stamp of file downloading completion.
Opened	User had open file after downloading or not, in opened file if value in 1 this indicates that file has opened by user and if value is 0 this indicate user didn't open this file via chrome. If user open this file from any other application such as file manager etc then chrome will not change value in opened field of download table.
Referrer	Referrer URL of downloaded file.
last_modified	store information about if user has made modification to the file the time stamp when this modification occurs.
Mimetype	store downloaded file type information such as if file is jpeg image it will store image/jpeg value in mimetype field.
tab_url	contains url of current tab.
site_url	path from where file is downloaded.

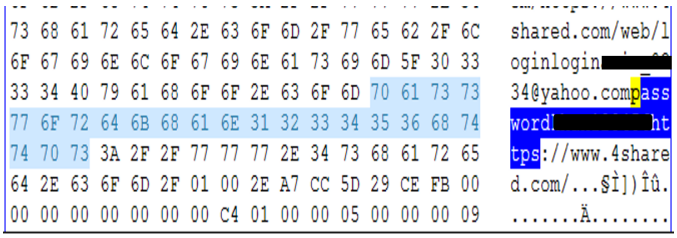

Fig. 4. User Credentials as stored by Chrome

and Opera store cache data in "*/cache/*" directory inside application data directory. Mozilla Firefox stores cache data in "*org.mozilla.firefox/cache/xxxxxxxx.default*". Inside cache directory, there are sub directories (i.e. news, cache, okhttp etc) which store cache data in random files (such as file begins with "caticon_xxxx" store PNG Images). Weppy Images are also stored in cache. Weppy images are created by Google, to improve performance of web pages and make website faster by reducing image size [37]. In analysis, we found images in "*cache/okhttp*" (Figure 5), Base64Encoded Images (Figure 6), HTML pages with URL (Figure 7)and plain text data. Dolphin store cache data in "*mobi.mgeek.TunnyBrowser/databases/dolphin_webviewCache.db*" SQLite file and cache images in *"sdcard/Android/data/mobi.mgeek.TunnyBrowser/cache" directory*. Andro-Kit successfully extracted cache data (including images, URL, Web pages etc) from all popular Android web browsers.

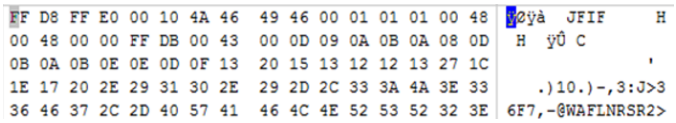

Fig. 5. Cache Image

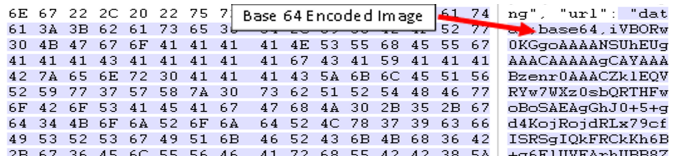

Fig. 6. Cache Base64 Encoded Image

F. Stored Sessions

Each session of activity in web browsers is recorded by the web browsers. In event of browser crashing, session can be restored. Sessions records store information about closed/open tabs, cookies etc. Therefore, it is possible to recover multiple evidences in order to show a user's web activity. Chrome store session data in "*Session Storage*" directory, Opera and Dolphin store this data in SQLite file which Mozilla Firefox utilize "*sessionstore.js*" file for storing session data. Paths on which popular android web browsers store session data are shown in Table XII. Chrome store session data in *"com.android.chrome/app_chrome/Default/Session Storage/"* directory. Opera store this information in *"databases/appboy.db"* SQLite file. Mozilla store session data in *"org.mozilla.firefox/files/mozilla/yybtc8zi.default/sessionstore.js"* file. Dolphin store session information in *"mobi.mgeek.TunnyBrowser/databases"* SQLite file. AndroKit has ability to recovery sessions data from all popular android web browsers.

G. Save Pages

Web browsers often save web pages for offline use. For forensics investigators, interesting information may be stored in such files. Chrome and Opera store these pages in MHTML format. MHTML is a file extension for web pages archive format. MHTML save webpage contents and external resources such as applets, images etc. into HTML Document [38]. Mozilla saves pages in portable document "PDF" format. Dolphin browser saves web pages in "*.htm*" format. Dolphin

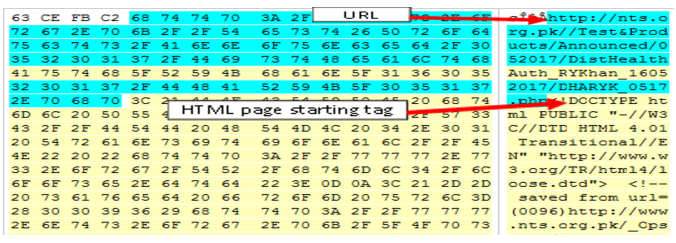

Fig. 7. Cache URLs and HTML Page

Fig. 8. Cache URLs and HTML Page

browser has no direct support for saving web pages into Portable document. Add-ons such as "Web to PDF" may be used to save webpages for offline use. A summary of locations on which all web browsers store save pages are shown in Table XII.

H. Tabs information

Web browsers store information about current/close tabs in application data. This information will help forensics investigator in correlating their findings regarding web sessions such as visited pages. Browsers use different formats and locations for recording this information. Table XII lists format used by browsers for recording tabs information and locations on which web browsers' tabs related information is stored. AndroKit is the only toolkit which extract tabs information while other standard forensics tool-kits do not deal with open/closed tabs information.

I. Local Storage

Local storage files are saved in SQLite format. Each file contains items table. Item tables contains key-value pairs, the semantics of which depend on the individual website. The extension "*.localstorage*" indicates an application support file created by the web browsers using WebKit, such as Google Chrome and Apple Safari. These files store browser settings or local data for a browser extension, and enables extensions to store a local cache of user data saved in a SQLite database format. This information will help forensics investigator in acquiring local data and settings from extensions. Table XII shows the paths for all web browser on which Local storage data is stored.

J. Chrome .log and .ldb files

Chrome stores "*.log*" and "*.ldb*" files in "*com.android.chrome/app_chrome/Default/DeltaFileLevelDb/*". Sessions data are stored in "*.ldb*" files. Bookmarks and history are stored in "*.log*" files (Figure 8). This is sync data (Web history and bookmarks) of other devices on chrome. This information helps investigator in extracting web history from other devices where suspect had signed-in in chrome.

K. Other Forensics Data

Web browsers save some other forensics data related to browsing session, time, search engine keywords, frequency of access, user profile and web browsers setting in memory. Therefore, in investigating suspect's device, this evidence can

provide useful information. It is necessary to extract such significant data related to forensics investigations. Table XII shows the list of some other valuable forensics data with description, which can be extracted from web browsers.

V. AndroKit Design and Implementation

The major contribution of this paper is the toolkit that can assist in digital forensics regarding all the artifacts and data structures explained in Section II. The environments in which AndroKit works include Windows 7, 8 and 10, and the targeted Android web browsers for forensic analysis are Chrome, Opera , Mozilla Firefox and Dolphin. The basic structure of the tool is illustrated in Figure 9.

A. AndroKit Development Environment

Development environment for AndroKit is Windows Presentation Foundation (WPF) forms in C# using Visual Studio 2015. Following NuGet packages are used in AndroKit implementation;

mAdb and sharpAdbClient: *sharpAdbClient* and *mAdb* are free open source libraries that allow .net applications to communicate with android devices. *SharpAdbClient* is .net client for Android Debug Bridge. Android Debug Bridge is command line tool that assists in communication with connected android device or emulator. *Adb* provides access to UNIX shell. *Adb* is client server program that includes a *Client*, *daemon* and a *Server*. *Client* runs on development machine which sends commands, *Daemon* (adbd) is background process on device which run commands on device and *Server* is responsible for management of communication between the *daemon* and *Client*. *Server* is also on development machine as background process. Details about *adb* and its commands are discussed in [39]. *sharpAdbClient* and *mAdb* provide implementation of the *adb* protocol and give flexibility to the developer to launch *adb.exe* and parse console output. Following *sharpAdbClient* and *mAdb* methods are used in Andro-Kit.

- *SharpAdbClient.AdbClient.Instance* class provides methods which allow application to interact with android device. To communication directly with android device, *adb.exe* is intermediate process between android device and application.
- *AdbServer.StartServer* method is used for starting *adb shell*.
- *DeviceMonitor* is used to check device connection status (device connected/disconnected).
- *AdbClient.Instance.ExecuteRemoteCommand* method is used to run "*adb shell*" commands on android device.
- *IOutputReciver* object is used to read/receive executed "*adb*" command output.

SQLite Package *System.Data.SQLite* Package is used for forensics analysis of SQLite database files.

Regular Expressions *System.Regex* class is used for extracting forensics data from web browsers cache. Regular expression is a pattern that can be matched against an input text. A pattern may consists of character literals, constructs and operators. There are different categories of operators, constructs and characters that let you to define regular expressions [34] [35].

B. AndroKit Implementation

AndroKit's implementation consists of following modules;
Information Extraction Module: Information extraction module employed in AndroKit gets device information and extracts forensics data from device. This module utilizes *adb* libraries (*sharpAdbClient* and *mAdb*). *AdbClient.Instance.GetDevices()* function gets connected device status. If device is connected, then *adb get prop* command is utilized via *AdbClient.Instance.ExecuteRemoteCommand* to get connected device information such as Device Serial No, Root access etc. Root access is required for performing Web browsers forensics data acquisition. If device has not root access, integrated third party module is activated for flashing stock recovery. We have discussed third-party module integration in later sections. In forensic data acquisition, evidence integrity is the major concern. To maintain evidence integrity *adb shell md5sum* method is utilized to calculate *MD5* hash before extracting forensic data. This hash is compared with extracted forensics data after pulling from device (*adb pull* method is used for pulling forensics data from device).
Integrated Third-Party Module: Third-Party Module consists of *Odin* and *fast boot*. *Odin* is the ROM Flashing tool for Samsung Android devices [40]. *Fast Boot* firmware flashing tool are used for MTK Chip based Android Smart-phones. These applications are integrated with AndroKit for flashing Stock/Custom recoveries. Flashing recovery is method for getting root access on android device. Custom recoveries are composed of mini OS which perform various system tasks. This is like small piece of OS running independent of system image and can control various system function and settings. In market, there are two popular stock recoveries. TWRP (Team Win Recovery Project) and CWM (Clockwork Mod); both provide fantastic features such as backup device, provide root access, wipe device, mount partitions, install applications, calculate md5sum, execute basic UNIX commands and much more. *adb* commands are also supported by these recoveries [41]. On Some Android device custom images are flashed with over the air (OTA) updates [42].
Forensics Data Analysis Module: AndroKit uses automated and manual forensics data analysis. SQLite Package *System.Data.SQLite* and Regular Expressions *System.Regex* classes are used for automated forensics data analysis. For manual analysis, SQLite Package is deployed. Evidence tampering protection is one of the major concerns in forensics data protection. For users query executions in manual analysis, queries are restricted to only *SELECT* query. This protects web browsers forensic data from tampering. Forensics data extraction from cache uses *Regex expressions*.
Regex expressions are also used for Base64 Encoded images extraction from cache files, However for decoding these images, *Convert.FromBase64String* functions is used.

TABLE XI
MISCELLANEOUS DATA FOR WEB BROWSERS FORENSICS

Artifact	Web Browser	Path	Description
Local Storage	Chrome	*"com.android.chrome/app_chrome/Default/Local Storage"*	These files store browser settings/local data for browser extensions and enable extensions to store a local cache of user data in a SQLite file.
	Opera	*"com.opera.browser/app_opera/Local Storage"*	
	Mozilla Firefox	*"org.mozilla.firefox/files/mozilla/yybtc8zi.default/ storage/"*	
	Dolphin	*"mobi.mgeek.TunnyBrowser/app_databases/ localstorage_jetpack"*	
Sync Data	Chrome	*"com.android.chrome/app_chrome/Default/ SyncData"*	Most modern web browsers support data sync between multiple devices. This data is stored in Sync data.
	Opera	*"com.opera.browser/app_opera/SyncData"*	
	Firefox		
	Dolphin	*"mobi.mgeek.TunnyBrowser/databases/Sync.db"*	
Saved Pages	Chrome	*"com.android.chrome/app_chrome/Default/Offline Pages/metadata/offlinepages.db"*	SQLite file "offlinepages.db" stores information about Saved Pages and MHTML file location.
		"com.android.chrome/app_chrome/Default/Offline Pages/archives/"	Saved/Offline Pages location.
	Opera	*"com.opera.browser/databases/reading.db"*	SQLite file "reading.db" stores information about Saved Pages.
		"com.opera.browser/app_opera/saved_pages/"	
	Firefox	*"sdard/Android/data/org.mozilla.firefox/files/Download"*	Firefox stores Saved pages in portable document format.
	Dolphin	*"sdcard/Downloads/"*	Dolphin stores Saved Pages in ".htm" format.
Tabs Info	Chrome	*"com.android.chrome/app_tabs/0"*	This folder contains a file "tab_state" that contains a list of all current tabs.
	Opera	*"com.opera.browser/shared_prefs/recently_ closed_tabs.xml"*	Recently Closed tabs title/URLs
		"com.opera.browser/shared_prefs/opera_osp_ local_attributes.xml"	Max Opened Private Tabs and Max Opened Tabs.
	Firefox	*"org.mozilla.firefox/files/mozilla/xxxxxxxx. default/sessionstore.js"*	In Firefox, sessionstore.bak store backup copy and previous.js store previous session
	Dolphin	*"mobi.mgeek.TunnyBrowser/app_tabstate/"*	
		"mobi.mgeek.TunnyBrowser/databases/browser.db"	SQLite file where information on recent tabs is stored in "recent_tab" table.
Favicons	Chrome	*com.android.chrome/app_chrome/Default/favicons"*	
	Opera	*"com.opera.browser/app_opera/favicons"*	
	Firefox	*"org.mozilla.firefox/cache/icons"*	
	Dolphin	"mobi.mgeek.TunnyBrowser/app_icons/Webpage IconsJetpack.db" & *"mobi.mgeek.TunnyBrowser /databases/browser.db"*	
Most Visited Websites	Chrome	*"com.android.chrome/app_chrome/Default/Top Sites.db"*	SQLite file stores information on Most Visited Sites
	Opera	*"com.opera.browser/databases/mostvisited.db"*	SQLite file stores information on Most Visited Sites
	Firefox		SQLite file stores information on Most Visited Sites
	Dolphin	*"mobi.mgeek.TunnyBrowser/databases/most_visited.db"*	SQLite file stores information on Most Visited Sites
Search Keywords	Chrome	*"com.android.chrome/app_chrome/Default/History"*	Search keywords are stored in *"Keyword_Search_terms"* table.
	Opera	*"com.opera.browser/app_opera/History.db"*	
	Firefox		
	Dolphin	*"mobi.mgeek.TunnyBrowser/databases/browser.db"*	Search keywords are stored in "searches" table.

Forensics Data Reporting Module: Data Reporting module is used to export forensics results to *pdf* or *html* file.

C. AndroKit Design

AndroKit GUI Design consista of following interfaces/Tabs (Figure 10.

Main Interface: AndroKit Main Interface GUI contains following controls;

- *Extract Data:* Extract data control is used for extraction of web browsers forensics data (Forensics data Extraction module is integrated in this control).
- *Select Extracted:* This control is used for loading already acquired web browsers forensics evidence.
- *Odin/Fast boot:* This control integrates Third-Party firmware flashing tools Odin (for Samsung devices) and Fstboot (for MTK chipset based Android Phones).

Information Tab: Information Tab shows the status of connected device, Mode (Online, Recovery, Downloading) and settings. This control integrates forensics information extraction module.

Web Browsers Tab: After Extracting/loading forensics data in AndroKit, this control shows each web browser forensics data in separate tab (Figure 11).

Cache Analysis Tab: After Extracting/loading forensics data in AndroKit, This control shows each web browsers cache data such as images (JPG, PNG, base 64 decoded images etc).

TABLE XII
A COMPLETE SUMMARY OF VARIOUS ARTIFACTS OF WEB BROWSERS ON ANDROID

	Forensics Data	**Path**
Chrome	Web History	*/com.android.chrome/app_chrome/Default/History*
	Bookmarks	*/com.android.chrome/app_chrome/Default/ bookmarks*
	Cookies	*/com.android.chrome/app_chrome/Default/cookies*
	Local Storage	*/com.android.chrome/app_chrome/Default/Local Storage*
	User Credentials	*com.android.chrome/app_chrome/Default/Login_Data*
	Sync Data	*com.android.chrome/app_chrome/Default/ SyncData*
	Saved Pages	*com.android.chrome/app_chrome/Default/Offline Pages/metadata/offlinepages.db*
		com.android.chrome/app_chrome/Default/Offline Pages/archives/
	Recent Tabs	*com.android.chrome/app_tabs/0*
	Favourite icons	*com.android.chrome/app_chrome/Default/favicons*
	Most Visted Websites	*com.android.chrome/app_chrome/Default/Top Sites.db*
	Search Keywords	*com.android.chrome/app_chrome/Default/History*
	User Preferences	*com.android.chrome/app_chrome/Default/Preferences*
	Auto-Complete	*com.android.chrome/app_chrome/Default/Web Data*
	Security Certificate Settings	*com.android.chrome/app_chrome/Default/Origin Bound Certs*
	Cache	*com.android.chrome/cache/*
	Session	*com.android.chrome/app_chrome/Default/Session Storage/*
Opera	Web History	*/com.opera.browser/app_opera/history.db*
	Bookmarks	*/com.opera.browser/app_opera/bookmark*
	Cookies	*/com.opera.browser/app_opers/cookies*
	Local Storage	*com.opera.browser/app_opera/Local Storage*
	User Credentials	*/app_opera/prefs.json*
	Sync Data	*com.opera.browser/app_opera/Sync Data*
	Saved Pages	*com.opera.browser/databases/reading.db*
		com.opera.browser/app_opera/saved_pages/
	Recent Tabs	*com.opera.browser/shared_prefs/recently_ closed_tabs.xml*
		com.opera.browser/shared_prefs/opera_osp_ local_attributes.xml
	Fav Icons	*com.opera.browser/app_opera/favicons*
	Most Visited Websites	*com.opera.browser/databases/mostvisited.db*
	Search Keywords	*com.opera.browser/app_opera/History.db*
	User Preferences	*com.opera.browser/app_opera/pref.json*
	Security Certificate Settings	*com.opera.browser/files/keychain/0*
	IMEI	*com.opera.browser/shared_prefs/appsflyerdata.xml*
	Cache	*com.opera.browser/cache/*
	Session	*com.opera.browser/databases/appboy.db*
Mozilla	Web History	*/org.mozilla.firefox/files/mozilla/xxxxxxxx.default/ browser.db*
	Bookmarks	*/org.mozilla.firefox/files/mozilla/xxxxxxxx.default/ browser.db*
	Cookies	*/org.mozilla.firefox/files/mozilla/xxxx.default/cookies.sqlite*
	Local Storage	*org.mozilla.firefox/files/mozilla/yybtc8zi.default/ storage/*
	User Credentials	*org.mozilla.firefox/files/mozilla/xxxxxxxx.default/signons.sqlite*
	Sync Data	
	Saved Pages	*sdcard/Android/data/org.mozilla.firefox/files/Download*
	Tabs Info	textitorg.mozilla.firefox/files/mozilla/xxxxxxxx. default/sessionstore.js
	Favicons	*org.mozilla.firefox/cache/icons*
	User Preference	*org.mozilla.firefox/files/mozilla/xxxxxxx.default/Prefs.js*
	Search engines	*org.mozilla.firefox/files/mozilla/xxxxxxx.default/search.json.mozlz4*
	Auto-complete history	*org.mozilla.firefox/files/mozilla/xxxxxxx.default/formhistory.sqlite*
	DOM storage	*org.mozilla.firefox/files/mozilla/xxxxxxx.default/webappsstore.sqlite*
	Security certificate settings	*org.mozilla.firefox/files/mozilla/xxxxxxx.default/cert9.db*
	Cache	*org.mozilla.firefox/cache/xxxxxxxx.default*
	Session	*org.mozilla.firefox/files/mozilla/yybtc8zi.default/sessionstore.js*
Dolphin	Web History	*/mobi.mgeek.TunnyBrowser/databases/browser.db*
	Bookmarks	*/mobi.mgeek.TunnyBrowser/databases/browser.db*
	Cookies	*/mobi.mgeek.TunnyBrowser/app_webview/cookies*
	Local Storage	*/mobi.mgeek.TunnyBrowser/app_databases/ localstorage_jetpack*
	User Credentials	*mobi.mgeek.TunnyBrowser/databases/passowrd.db*
	Sync Data	*/mobi.mgeek.TunnyBrowser/databases/Sync.db*
	Save Pages	*/sdcard/Downloads/*
	Tabs Info	*/mobi.mgeek.TunnyBrowser/app_tabstate/*
		/mobi.mgeek.TunnyBrowser/databases/browser.db
	Favicons	*mobi.mgeek.TunnyBrowser/app_icons/Webpage IconsJetpack.db* & *mobi.mgeek.TunnyBrowser /databases/browser.db*
	Most Visited Websites	*mobi.mgeek.TunnyBrowser/databases/most_visited.db*
	Search Keywords	*mobi.mgeek.TunnyBrowser/databases/browser.db*
	Cache	*mobi.mgeek.TunnyBrowser/databases/dolphin_webviewCache.db*
		sdcard/Android/data/mobi.mgeek.TunnyBrowser/cache
	Session	*mobi.mgeek.TunnyBrowser/databases*

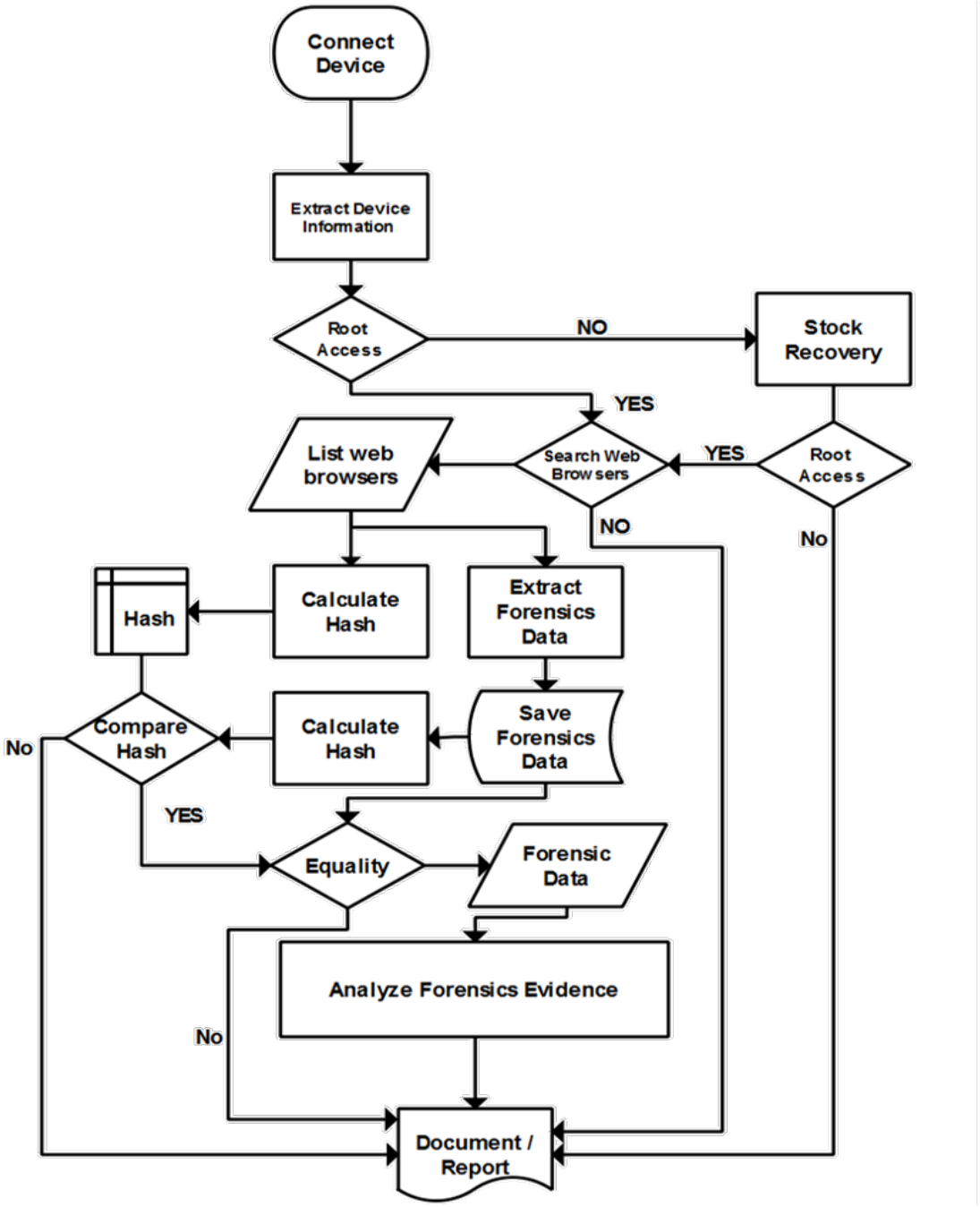

Fig. 9. AndroKit Flow Chart

Analysis: Analysis Tab shows all acquire SQLite files for manual analysis. User can execute SQL queries manually (Figure 12). Queries are restricted to only *SELECT* command. Forensics Data Analysis Module is integrated in this control.

All Files Tab: All files tab shows all extracted files, forensics investigators can use this option to display Binary, HEX and ASCII of all extracted files. This will help in searching any text, binary or hex values in extracted files.

VI. Comparative Analysis of AndroKit with Other Forensics Tool-Kits

For comparative analysis of AndroKit with standard forensics Tool-Kits; Chrome (Version 56.0.2924.87), Opera (Version 42.3.2246.113338), Mozilla Version (53.0.2) and Dolphin Version (11.5.19) are installed on Samsung Grand Prime (Android OS version 4.0 and 5.0), Samsung S3 (Android OS version 5.0) and Samsung Alpha (Android OS version 4.2 and 5.0). After installation of web browsers, Standard Mobile Forensics Tool-Kits; Oxygen Forensics, Andriller, MOBILedit

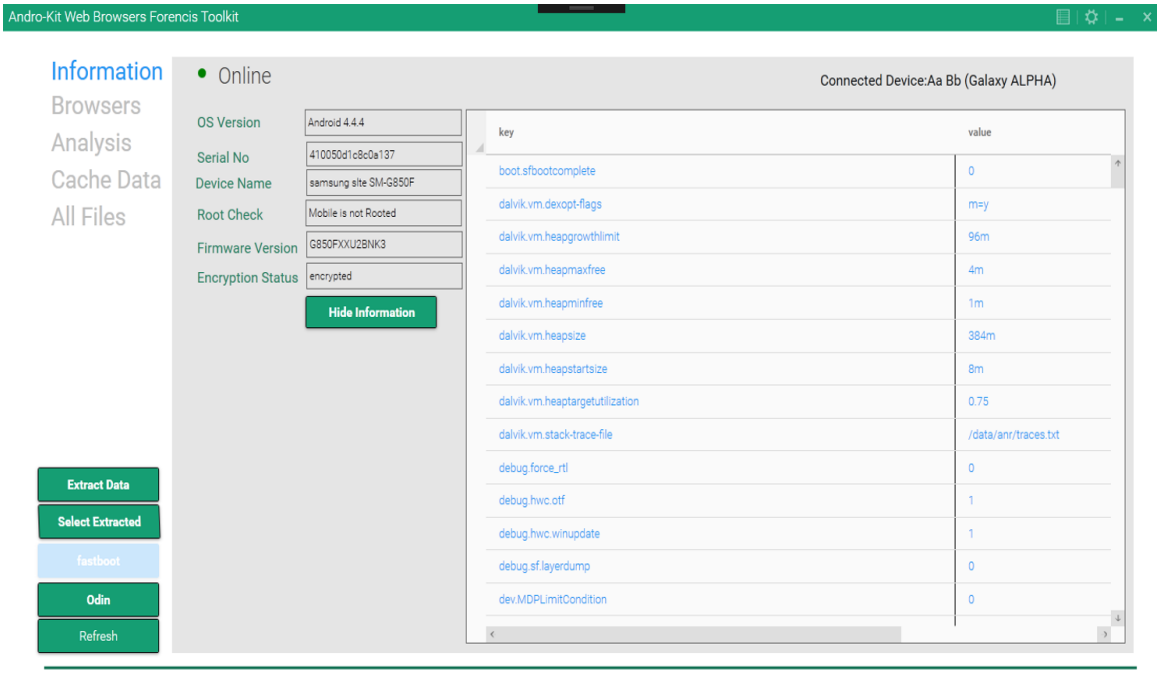

Fig. 10. AndroKit: User Interface

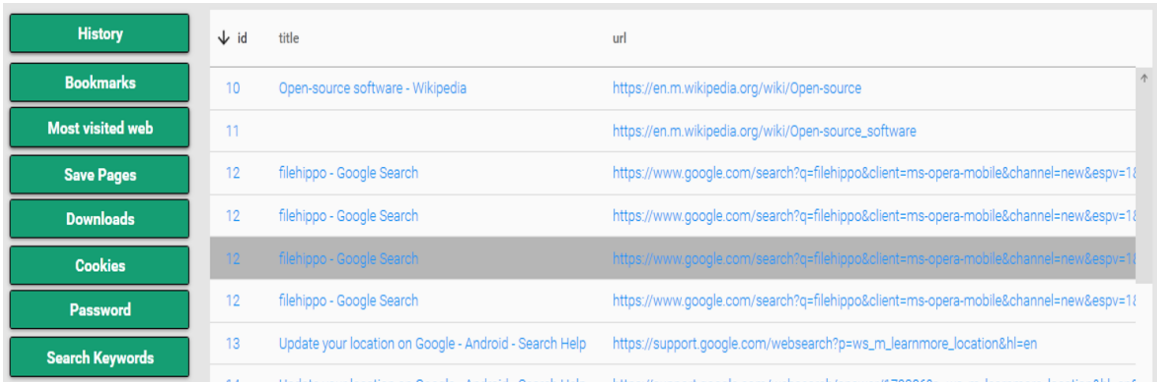

Fig. 11. AndroKit: Web History

and Belkasoft Evidence Center are installed on Windows 7 PC. Web browsers forensics analysis is performed on following scenario;

Non-Rooted Locked Android Device: In this scenario, Standard forensics toolkits are compared with AndroKit. All standard forensics tool-kits failed to extract forensics evidences from Non-Rooted locked Android devices shown in Table XIII. Standard tools like Oxygen Forensics provides lock screen bypass and device rooting capabilities only for MTK Chipset based devices. Only AndroKit is able to successfully get root access on device and bypass lock screen via flashing stock recovery mode. After getting root access, forensics evidence is acquired from Android devices.

Rooted Android Device: On Rooted devices, all forensics tools are able to acquire forensics evidence form android web browsers but analysis features vary in each forensics tool-

TABLE XIII
COMPARATIVE ANALYSIS OF ANDROKIT FEATURES WITH OTHER FORENSICS TOOL-KITS FEATURES

Forensics Tool-Kits	**AndroKit**	**Oxygen Forensics**	**Andriller**	**MOBILedit**	**Belkasoft evidence center**
Screen Lock bypass	✓	✓ (Only MTK)	×	×	×
Root device	✓	✓ (Only MTK)	×	×	×
Flash Stock Recovery	✓	×	×	×	×
Custom Query Execution	✓	×	×	×	×

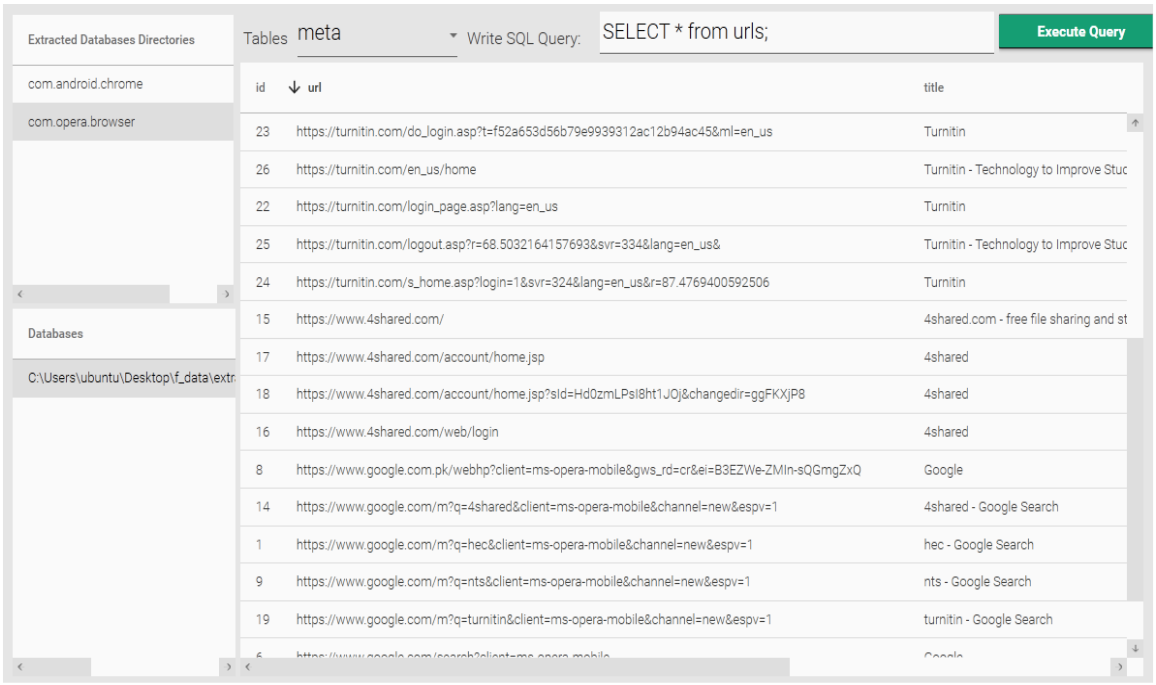

Fig. 12. AndroKit: User Query (Manual Analysis)

kit. AndroKit has more analysis features (such as custom queries, cache data etc) as compared to standard forensics tool-kits shown in Table XIII & Table XIV. Some Standard tool-kits, like *MOBILedit* need to install client app on device for forensics evidence association but this is not feasible forensics sound method for acquire evidence from device, because it violates user data partition integrity on installing client application. In analysis, we found Androkit has better features then other standard forensic tool-kits. Detailed comparative analysis results are presented in Table XIII and Table XIV.

VII. Conclusion

This paper discusses popular android based web browsers (Chrome, Opera, Mozilla Firefox and Dolphin) and mhealth care services on android web browsers; how these web browsers store forensics data on android devices; how investigator can acquire Web browsers forensics data from Android smart-phones (Especially from mHealth and CHCS). The design of AndroKit and its implementation is discussed. Comparative analysis of AndroKit with other tool-kits is performed with forensics in a sound manner. Comparative analysis results show that AndroKit provides maximum features as compared to other standard forensics tool-kits.

In smart phones, digital forensics industries is not aligned with smart phones technology development due to faster development in technology as compare to forensics industries. In future, AndroKit can be extented to support other android applications such as social media applications (Whatsapp, Viber, Messenger etc).

References

[1] "The Communications Market Report 2017: United Kingdom," [Online]. Available: $_{h}ttps$: $//www.ofcom.org.uk/_{data/assets/pdf_file}/0017/105074/cmr-2017-uk.pdf$. [Accessed 08 Jan 2018].

[2] "Global mobile OS market share in sales to end users from 1st quarter 2009 to 2nd quarter 2017" [Online]. Available: https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/. [Accessed 08 Jan 2018].

[3] Molla S Donaldson, Janet M Corrigan, Linda T Kohn, et al. To err is human: building a safer health system, volume 6. National Academies Press, 2000.

[4] D Kopec, MH Kabir, D Reinharth, O Rothschild, and JA Castiglione. Human errors in medical practice: systematic classification and reduction with automated information systems. Journal of medical systems, 27(4):297313, 2003.

[5] Abu Saleh Mohammad Mosa, Illhoi Yoo, and Lincoln Sheets. A systematic review of healthcare applications for smartphones. BMC medical informatics and decision making, 12(1):67, 2012.

[6] Richard K Lomotey, Shomoyita Jamal, and Ralph Deters. Sophra: a mobile web services hosting infrastructure in mhealth. In Mobile Services (MS), 2012 IEEE First International Conference on, pages 8895. IEEE, 2012.

[7] Kirusnapillai Selvarajah, Michael P Craven, Adam Massey, John Crowe, Kavita Vedhara, and Nicholas Raine-Fenning. Native apps versus web apps: which is best for healthcare applications? In International Conference on Human-Computer Interaction, pages 189196. Springer, 2013.

[8] Andreas Dahl and Kristofer Nylander. Differences in security between native applications and web based applications in the field of health care, 2015.

[9] Barbara L Filkins, Ju Young Kim, Bruce Roberts, Winston Armstrong, Mark A Miller, Michael L Hultner, Anthony P Castillo, Jean-Christophe Ducom, Eric J Topol, and Steven R Steinhubl. Privacy and security in the era of digital health: what should translational researchers know and do about it? American journal of translational research, 8(3):1560, 2016.

[10] Tobias Dehling, Fangjian Gao, Stephan Schneider, and Ali Sunyaev. Exploring the far side of mobile health: information security and privacy of mobile health apps on ios and android. JMIR mHealth and uHealth, 3(1), 2015.

TABLE XIV
COMPARATIVE ANALYSIS OF ANDROKIT WEB BROWSERS FORENSICS WITH OTHER FORENSICS TOOL-KITS

Web Browser	**Forensics Evidence**	**AndroKit**	**Oxygen Forensics**	**Andriller**	**MOBILedit**	**Belkasoft** (evidence center)
Chrome	Web Browser History	✓	✓	✓	✓	✓
	Bookmarks	✓	✓	✓	✓	✓
	Save pages	✓	✓	×	×	✓
	Downloads	✓	✓	×	×	✓
	Cookies	✓	✓	×	×	×
	Passwords	✓	✓	✓	×	✓
	Most Visited Websites	✓	×	×	×	×
	Search Keywords	✓	×	×	×	✓
	Cache Data	✓	×	×	×	✓
	Encoded Images Decoding	✓	×	×	×	×
Opera	Web Browser History	✓	✓	✓	✓	✓
	Bookmarks	✓	✓	✓	✓	✓
	Save pages	✓	✓	×	×	✓
	Downloads	✓	✓	×	×	✓
	Cookies	✓	✓	×	×	×
	Passwords	✓ (encrypted)	×	×	×	×
	Most Visited Websites	✓	×	×	×	×
	Search Keywords	✓	×	×	×	✓
	Cache Data	✓	×	×	×	✓
	Encoded Images Decoding	✓	×	×	×	×
Firefox	Web Browser History	✓	✓	✓	✓	✓
	Bookmarks	✓	✓	✓	✓	✓
	Save pages	✓	✓	×	×	✓
	Downloads	✓	✓	×	×	✓
	Cookies	✓	✓	×	×	×
	Passwords	✓ (encrypted)	×	×	×	×
	Most Visited Websites	✓	×	×	×	×
	Search Keywords	✓	×	×	×	✓
	Cache Data	✓	×	×	×	✓
	Encoded Images Decoding	✓	×	×	×	×
Dolphin	Web Browser History	✓	✓	✓	✓	✓
	Bookmarks	✓	✓	✓	✓	✓
	Save pages	✓	✓	×	×	✓
	Downloads	✓	✓	×	×	✓
	Cookies	✓	✓	×	×	×
	Passwords	✓ (encrypted)	×	×	×	×
	Most Visited Websites	✓	×	×	×	×
	Search Keywords	✓	×	×	×	✓
	Cache Data	✓	×	×	×	✓
	Encoded Images Decoding	✓	×	×	×	×

[11] Charalampos Doukas, Thomas Pliakas, and Ilias Maglogiannis. Mobile healthcare information management utilizing cloud computing and android os. In Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE, pages 10371040. IEEE, 2010.

[12] Khaled A Nagaty. Mobile health care on a secured hybrid cloud. J Sel Areas Health Inform, 4(2):19, 2014.

[13] Jemal Hanen, Zied Kechaou, and Mounir Ben Ayed. An enhanced healthcare system in mobile cloud computing environment. Vietnam Journal of Computer Science, 3 (4):267277, 2016.

[14] Abdullah Azfar, Kim-Kwang Raymond Choo, and Lin Liu. Forensic taxonomy of popular android mhealth apps. arXiv preprint arXiv:1505.02905, 2015.

[15] Stacy Mitchell, Scott Ridley, Christy Tharenos, Upkar Varshney, Ron Vetter, and Ulku Yaylacicegi. Investigating privacy and security challenges of mhealth applications. 2013.

[16] Oh, J., Lee, S. and Lee, S., 2011. Advanced evidence collection and analysis of web browser activity. digital investigation, 8, pp.S62-S70.

[17] Gratchoff, J. and Kroon, G., 2015. Project Spartan Forensics. Amsterdam University.

[18] Pereira Murilo Tito, "Forensic analysis of the Firefox3 internet history and recovery of deleted SQLite records," Digital Investigation, vol. 5, pp. 93-103, 2009.

[19] E. A. Barghouthy, A. Marrington and I. Baggil, "The Forensic Investigation of Android Private Browsing Sessions using Orweb," in 5th International Conference on Computer Science and Information Technology (CSIT), Dubai, UAE 2013, 2013.

[20] Jang, Y.J. and Kwak, J., 2015. Digital forensics investigation methodology applicable for social network services. Multimedia Tools and Applications, 74(14), pp.5029-5040.

[21] Reed, A., Scanlon, M. and Le-Khac, N, 2017. "Forensic Analysis of Epic Privacy Browser on Windows Operating Systems". Proceedings of the 16th European Conference on Cyber Warfare and Security (ECCWS 2017), 1, pp.341-350

[22] Akbal, E., Günes, F., Akbal, A, 2016. "Digital Forensic Analyses of Web Browser Records.". JSW, 11(7), pp.631-637

[23] Nalawade, A., Bharne, S. and Mane, V., 2016. "Forensic analysis and evidence collection for web browser activity". Automatic Control and Dynamic Optimization Techniques (ICACDOT), International Conference on, pp.518-522

[24] Rathod, D., 2017. "Web Browser Forensics: Google Chrome". International Journal of Advanced Research in Computer Science, 8(7), pp.518-522

[25] Varol, A. Sonmez, Y. U., 2017. "The importance of web activities for computer forensics". Computer Science and Engineering (UBMK), 2017 International Conference on, pp.66-71

[26] Mahaju, S. and Atkison, T., 2017. "Evaluation of Firefox Browser Forensics Tools".Proceedings of the SouthEast Conference (ACM), pp.5-12

[27] Akbal, E., Gnes, F. and Akbal, A., 2016. Digital Forensic Analyses of Web Browser Records. JSW, 11(7), pp.631-637.

[28] Vidas, T., Zhang, C. and Christin, N., 2011. Toward a general collection methodology for Android devices. digital investigation, 8, pp.S14-S24.

[29] S. Wu, Y. Zhang, X. Wang and X. Xiong, "Forensic analysis of WeChat

on Android smartphones," in Proceedings of the 16th Annual USA Digital Forensics Research Conference, DFRWS, USA, 2016.
[30] A. Shortall and M. A. H. B. Azhar, "Forensic Acquisitions of WhatsApp Data on Popular Mobile Platforms," in Sixth International Conference on Emerging Security Technologies (EST), 2015.
[31] A. Hamid, F. Ahmad, K. Ram and A. Khalique, "Implementation of Forensic Analysis Procedures for WhatsApp and Viber Android Applications," International Journal of Computer Applications,, vol. 128, no. 12, pp. 26-33, 2015.
[32] S. C, K. MT and L.-K. NA, "Forensics Acquisition and Analysis of Instant Messaging and VoIP Applications," Computational Forensics. Lecture Notes in Computer Science, vol 8915. Springer, Cham, 27 June 2015.
[33] P. E. King, "Using TWRPs new ADB interface," [Online]. Available: http://www.pocketables.com/2014/10/using-twrps-new-adb-interface.html [Accessed 08 FEB 2018]
[34] Friedl and J. E. F, Mastering Regular Expressions: Powerful Techniques for Perl and Other Tools. O'Reilly, ISBN 0-596-00289-0, 2002, p. 319.
[35] M. L. Scott, Programming Language Pragmatics. Morgan Kaufmann, ISBN 1-55860-442-1, 1999.
[36] [Online]. Available: http://kb.digital-detective.net/display/BF/Page+Transitions. [Accessed 07 FEB 2018].
[37] "Webp," [Online]. Available: https://developers.google.com/speed/webp/docs/riff_container?csw=1.
[38] "What is MHTML? What Opens a MHTML? File Format List from WhatIs.com", [Online]. Available: http://whatis.techtarget.com/fileformat/MHTML-MHTML-document-MIME. [Accessed 08 FEB 2018].
[39] "Android Debug Bridge," [Online]. Available: https://developer.android.com/studio/commandline/adb.html#howadbworks. [Accessed 25 JAN 2018].
[40] "Samsung Odin," [Online]. Available: http://odindownload.com. [Accessed 08 FEB 2018].
[41] P. E. King, "Using TWRPs new ADB interface," [Online]. Available: http://www.pocketables.com/2014/10/using-twrps-new-adb-interface.html. [Accessed 08 FEB 2018].
[42] "OTA Updates," [Online]. Available: https://source.android.com/devices/tech/ota/. [Accessed 08 Feb 2018].