



# THE JOURNAL OF INFORMATION WARFARE

---

Password Recovery and Data Retrieval in the Android Operating System

Author(s): D Hintea, R Bird and J Moss

Source: *Journal of Information Warfare*, Vol. 16, No. 4 (Fall 2017), pp. 14-25

Published by: Peregrine Technical Solutions

Stable URL: <https://www.jstor.org/stable/10.2307/26504115>

---

JSTOR is a not-for-profit service that helps scholars, researchers, and students discover, use, and build upon a wide range of content in a trusted digital archive. We use information technology and tools to increase productivity and facilitate new forms of scholarship. For more information about JSTOR, please contact [support@jstor.org](mailto:support@jstor.org).

Your use of the JSTOR archive indicates your acceptance of the Terms & Conditions of Use, available at <https://about.jstor.org/terms>



JSTOR

*Peregrine Technical Solutions* is collaborating with JSTOR to digitize, preserve and extend access to *Journal of Information Warfare*

# Password Recovery and Data Retrieval in the Android Operating System

D Hintea<sup>1</sup>, R Bird<sup>1</sup>, J Moss<sup>2</sup>

<sup>1</sup>*School of Computing, Electronics and Mathematics  
Coventry University  
Coventry, UK*

*E-mail: diana.hintea@coventry.ac.uk; robert.bird@coventry.ac.uk*

<sup>2</sup>*BlackBerry Ltd.  
Kidderminster, UK*

*E-mail: jamoss@blackberry.com*

**Abstract:** *This paper considers how data and passwords are recovered across different versions of the Android Operating System (OS). This solution has been achieved by forensically retrieving passwords that were set up in Android 4.4 to 5.0.2. XRY Extraction Wizard and XRY Reader were used as tools to conduct this investigation. The results obtained showed that encryption is the safest method of keeping data secure, while also showing a lack of backwards compatibility with security updates. Android uses a single-use authentication token to authenticate Google accounts, but fails to secure Chrome passwords in the most widely used versions of Android.*

**Keywords:** *Android, Forensic Analysis, Password Recovery, Data Retrieval, XRY*

## Introduction

The Android Operating System (OS) was first introduced into the market in 2008. It was based on the Linux kernel and was devised initially for mobile telecommunication devices. When Android 2.0 was released, encrypting a device was not an option. This changed in version 3.0; the Honeycomb OS was released in February 2011, with the option for encryption, which was AES-CBC (Advanced Encryption Standard Cipher Block Chaining) 128-bit encryption for the device (Android 2015a). Many users would have been unaware of this option and uncertain of what it meant. The same encryption type is used in today's Android systems. Until the announcement of Android 5.0 Lollipop OS in November 2014, there had been no major encryption overhauls in Android devices, and encryption was still an option (Android Twitter 2014). Google initially planned to have full Android OS encryption by default. The plan was later changed due to the way the devices were designed. To make sure that no compromise of personal data ever occurred, a collateral intrusion plan was put into place. This means that the effective measures were used to prevent personal data from reaching the public domain. The devices used for this paper were either purchased new or were previously used devices that were completely wiped using a standard process of encryption, factory resets, and dummy data as described in Graziano (2014). Data and

passwords used on the devices for the purpose of recovery and retrieval were accounts set up specifically for this research.

## Literature Review

This literature review presents current research into password recovery and retrieval in the Android OS. Lessard and Kessler (2010) used the HTC Hero smartphone running Android 2.2 in their research and highlighted the issues in Android 2.2. They provided an in-depth view of logical data extraction and passwords that have been recovered, suggesting that in Android version 2.2 Google did not address the security of plain-text passwords stored in Chrome. This suggestion led the authors of this paper to investigate whether this security flaw has now been patched or if it is yet to be recognised in the newest versions of Android. Hoog (2011) discussed all aspects of investigation into Android phones and the problems that can occur for a user or organisation, stating that

If an attacker was looking for an effective way to infiltrate a corporation, an employee's device (or better yet, several employees' devices) can provide many insights and avenues for an attack—not to mention recovering sensitive corporate information directly from the device.

This observation puts into perspective the importance of mobile-device data security in large businesses. The authors of this paper use this knowledge to show that Android stores a large amount of sensitive data insecurely on its devices, which could be used as an attack vector in a large business if a device were stolen and forensically analysed.

*The XRY Extract Data Manual* (Micro Systemation AB 2015) fails to state anything regarding Wi-Fi password recovery. Therefore, this omission provides the opportunity to add to the academic body of knowledge regarding how Wi-Fi passwords could be recovered in the Android versions such as those discussed in this paper. In the process of conducting the current study, the authors have discovered that the current Android OS market share statistics provide critical support for the scope of the current project. This allows the focus to be shifted to the most common Android OSs that are currently used and the most recent that have not previously been tested for forensic data recovery using the methods described in this paper. **Table 1**, below, shows the usage distribution of Android versions at the time this paper was written.

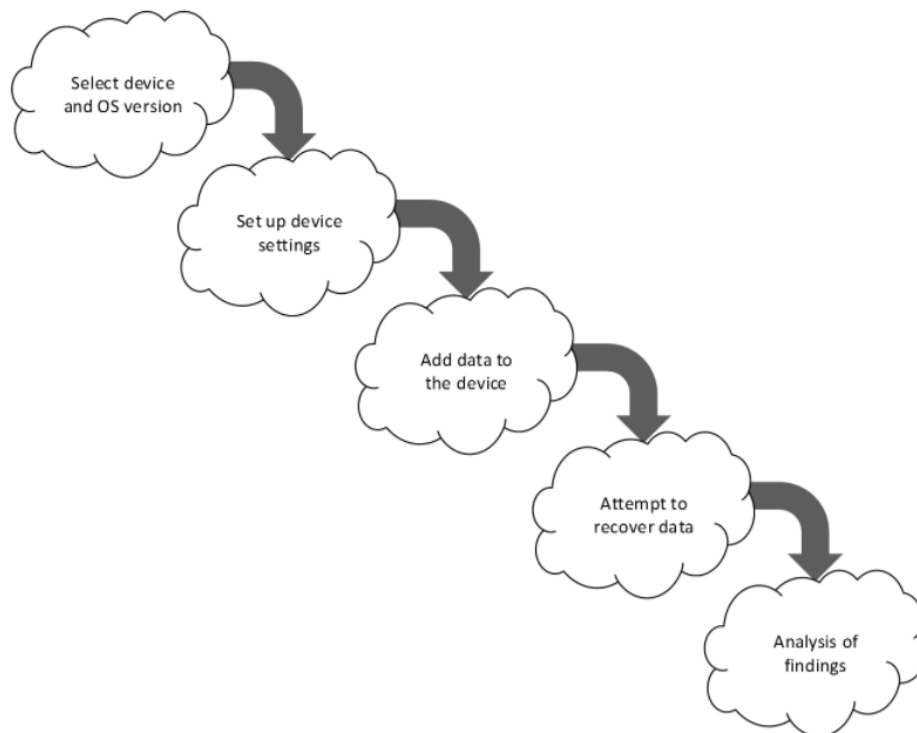
New versions of Android are being released continually. It is difficult to find research regarding the newest versions, a fact that limits the ability of the authors (and others) to add knowledge to the area based upon tests and analyses of the most recent versions of Android.

Version	Codename	Distribution
2.2	Froyo	0.4%
2.3.3-2.3.7	Gingerbread	6.4%
4.0.3-4.0.4	Ice Cream Sandwich	5.7%
4.1.x	Jelly Bean	16.5%
4.2.x	Jelly Bean	18.6%
4.3	Jelly Bean	5.6%
4.4	KitKat	41.4%
5.0	Lollipop	5.0%
5.1	Lollipop	0.4%

**Table 1:** Data collected from Android (Google 2015)

## Research Methodology

**Figure 1**, below, illustrates the methodology used for the research discussed in this paper.



**Figure 1:** Research methodology

The authors considered three types of research: quantitative, qualitative, and physical testing. Each method has its own advantages and disadvantages. Quantitative research is often used for large data sets and numbers. This would be useful for the datasets and numbers related to the Android market share. Qualitative research can be based on opinions, such as surveying people about their Android knowledge. Physical testing is what the research in this paper is mostly based on and consists of performing data extraction techniques and forensic information retrieval on the data.

For this project, the authors used both primary and secondary research. The primary research consisted of recovering data from the devices running different versions of Android, while the secondary research involved finding existing research to help support the project's claims.

The following subsections provide detailed information on the tools, devices, OSs, and settings used.

## **Tools**

The tools used for this research are well known in industry and widely regarded as forensically sound and admissible in a court of law:

1. XRY Extraction Wizard – Version 6.13.1
2. XRY Reader – Version 6.13.1

The authors chose to use XRY as it is used by numerous private companies and government organisations for data recovery. The software is also designed as a bundle so that the data extracted by the XRY wizard can be used in the reader.

## **Devices**

Google Nexus 7 was used as a device for this research. With 52% of the market share of Android covered over 4.3 to 5.0 (Android 2015b), this device is expected to give a wide range of results.

## **Operating Systems**

The OSs tested were as follows:

1. Android 4.3 - out of the box
2. Android 4.4.4
3. Android 4.4.4 - encrypted
4. Android 5.0.2
5. Android 5.0.2 - encrypted

The OS versions currently supported on Nexus 7 range from 4.3 to 5.1. With technology being continually released and updated, the authors could not justify testing Android 2.0 - 3.0 because the number of devices currently running the seven-year-old OS is only 6.8% of the market share. The authors of this paper believe that the results would be irrelevant since Micro Systemation AB (2015) states full recovery is viable in these versions and there is no option to encrypt the device in 2.0.

## **Settings**

There are numerous settings that were selected on the device to allow the recovery of data. A screen lock password on the device is not used as the device would have had to be rooted using XRY or manually. This would have voided the warranty on a borrowed device. This meant that, when the recovery was being performed, the authors had to comply with ACPO Principle 1: “No action taken by law enforcement agencies or their agents should change data held on a computer or storage media which may subsequently be relied upon in court” (Association of Chief Police Officers 2015). In addition, to ensure that the research in this paper is forensically sound, the authors made sure that any access being made complies with ACPO Principle 2: “In exceptional circumstances, where a person finds it necessary to access original data held on a computer or on storage media, that person must be competent to do so and be able to give evidence explaining the relevance and the implications of their actions” (Association of Chief Police Officers 2015).

Below are the settings used for the different OSs:

### **Android 4.3-5.0.2**

- Enable Developer Mode - to allow access
- Enable MTP (Media Transfer Protocol)
- Enable USB Debugging
- Encrypt the Device

### **Android 4.4.4 & 5.0.2 encrypted**

- Enable Developer Mode - to allow access to other options
- Enable MTP (Media Transfer Protocol)
- Enable USB Debugging
- Encrypt the Device

## **Setup procedure**

The device used for this research was brand new. All settings required for the acquisition were enabled. Then, the authors proceeded to enter the default information the researchers were attempting to recover. It was decided that the information that was being saved onto the device would be all OS-based applications from Google and not third-party applications. The artefacts tested were:

- Gmail
- YouTube
- Google Drive - a document called “secretdocument.docx”, which contained random text
- Google Chrome
- Wi-Fi
- Pictures

Various pieces of data were saved, as **Table 2**, below, presents: a Gmail username and password into Gmail, YouTube and Google Drive; the university login into the Chrome browser; and the home Wi-Fi password to provide an Internet connection. An image was also added into the gallery

for further results. Once this was completed, the screen was locked and no other data on the device was changed.

Application	Username/Data	Password
Gmail	dissertationforensicrecovery@gmail.com	dissertation123
YouTube	dissertationforensicrecovery@gmail.com	dissertation123
Google Drive	dissertationforensicrecovery@gmail.com	dissertation123
Google Drive	secretdocument.docx	dissertation123
Google Chrome	Mossj5	*****
Wi-Fi	Deathstar-2G	*****
Wi-Fi	PHOENIX-NET-SECURE	*****
Pictures	Image of USB Stick	N/A

**Table 2:** Summary of the artefacts tested

### Investigation procedure (without encryption)

The device was connected via the micro USB cable to the workstation. The device was identified by the computer, but this was ignored as the authors did not want to access the file system via Windows Explorer. XRY Extraction Wizard was then booted, which immediately identified the device as a Nexus 7. The device was selected; XRY then requested that the device be accessed and that USB debugging be allowed on the device. The device was accessed; the ‘remember this computer’ option was selected; and the device was given debugging access. XRY then requested the system dump type and the ‘Logical (full read)’ option was selected.

The ‘Backup’ method of data extraction was chosen. This mimics backing up the device to a computer, although, in fact, XRY retrieves all the information that is transferred. Using this method allows all data to be transferred from the device, rather than just core system applications. The device must be accessed again to confirm the backup; once this is done the extraction process begins. As soon as this process is completed, the extracted file system can be opened for viewing in XRY Reader. Reader provides detailed information on the device, including the Android OS version. Once the OS version was confirmed, the authors attempted to find the data stored into the device previously. There is a ‘Find’ option in XRY Reader, which helps avoid wasting time looking for information manually. After attempting to collect all the data, the authors recorded the results. This procedure was repeated for all Android versions tested, except Android 5.0.2, because its data is encrypted.

### Investigation procedure (with encryption)

For the encrypted versions of Android tested (4.4.4 and 5.0.2), the authors attempted to replicate a real-world scenario. The same investigation procedure as the one for non-encrypted versions was followed; however, a different type of data extraction was performed—an ‘agent’ logical recovery, which focuses on the device’s system applications. A backup extraction was undesirable, as it would not show the results of the encrypted device. The authors also attempted to get results from the encrypted device, with USB debugging disabled and using a complex password. This attempt made the device completely useless in a forensic sense, and the authors had no way to access it.

### Results and Analysis

This section presents the results obtained from this password-recovery research based on the different Android versions tested. The results produced from the primary research were surprising. The largest market share in Android was running the 4.4 KitKat OS at the time this paper was written. It is as secure as the version 4.1, which was released on July 10, 2012 (Christensen 2014). XRY supports a partial recovery of the system in the 4.1 and 4.4 Android versions. Although there was a time difference of one and a half years between the two releases, there were no security updates to prevent data recovery on the device.

### Android 4.3 to 4.4.4

**Table 3** summarises the results of this research. XRY managed to recover a large amount of the data saved on the devices.

Application	Username/Data	Password	Results
Gmail	dissertationforensicrecovery@gmail.com	dissertation123	Not recovered
Youtube	dissertationforensicrecovery@gmail.com	dissertation123	Not recovered
Google Drive	dissertationforensicrecovery@gmail.com	dissertation123	Not recovered
Google Drive	secretdocument.docx	dissertation123	Partially recovered
Google Chrome	Mossj5	*****	Recovered
Wi-Fi	Deathstar-2G	*****	Recovered
Wi-Fi	PHOENIX-NET-SECURE	*****	Partially recovered
Pictures	Image of USB Stick	N/A	Recovered

**Table 3:** Results for Android 4.3-4.4.4



## Google account

The Google account password was not recovered. The search function in XRY failed to find the password 'dissertation123'. The authors wanted to further investigate the security behind how Google stores the password on the device. Using the strings function and piping the file using the 'grep' command line utility, as illustrated in **Figure 2**, below, the authors discovered how the password was stored. Once a user enters his or her Google password the first time, Google uses an authentication token on the device rather than the account password that was first entered. In the past, it had been possible to push through usernames and authentication tokens to gain unauthorised access to accounts.

```
root@kali:~# strings /root/Desktop/Asus\ Google\ Nexus\ 7\ -\ 2\ WiFi
\ \ (K008\)\ Backup\ 3.xry | grep dissertationforensicrecovery@gmail.c
om | grep authToken
    <string name="dissertationforensicrecovery@gmail.com_authToken">D
QAAANwAAABp2cXQv252VaEe_1oTiBrd04HY6gM8sYmz01R4p1JTjhj026uisLARMJo0U0
kWBAbycWy-6-kvG-FEB6oVBm5ZlCzfzI2Dizea6z4FomeqDCk5wWmySYs7WjQeVBZY35r
lkoNDK0VsAziu6tEjPDWKjqCP5TQB0LZ4tmAeBDA-GBolk7L3u0N2WlglSwNnw3Cp1c6
bIgPUhVANT0GTlyvfiKr3_mhvoG6wU0Rv_o3oZS1kmHpM0euZYqfpTnqpt03XTSZx4oXe
g8bdLUxCd7pb638KTPWY0RnrmCro01q2w</string>
```

**Figure 2:** Recovery of Google account information

The authors further tested the security of the authentication token that Google uses with the developer playground (Google Developers 2015). The researchers found that Google is using a one-time-use token for authentication, as illustrated in **Figure 3**, below. If the Google account holder logs out and then logs back in, the system will produce a different one-time-use authentication code. This is an extremely secure method of authentication forensically, as there is no possibility of discovering the password for any of the Google applications.

```
HTTP/1.1 400 Bad Request
Content-length: 82
X-xss-protection: 1; mode=block
X-content-type-options: nosniff
Expires: Sun, 19 Apr 2015 23:24:39 GMT
Vary: Origin,X-Origin
Server: GSE
Cache-control: private, max-age=0
Date: Sun, 19 Apr 2015 23:24:39 GMT
X-frame-options: SAMEORIGIN
Content-type: application/json; charset=UTF-8

{
  "error_description": "Code was already redeemed.",
  "error": "invalid_grant"
}
```

**Figure 3:** Token and exchange authorisation testing

## Google Drive

XRY managed to identify the name of the document saved to the Google Drive. The only information recovered was the document name. It was inaccessible (as it is cloud-based so it does not store a copy on the device) and contained no data.

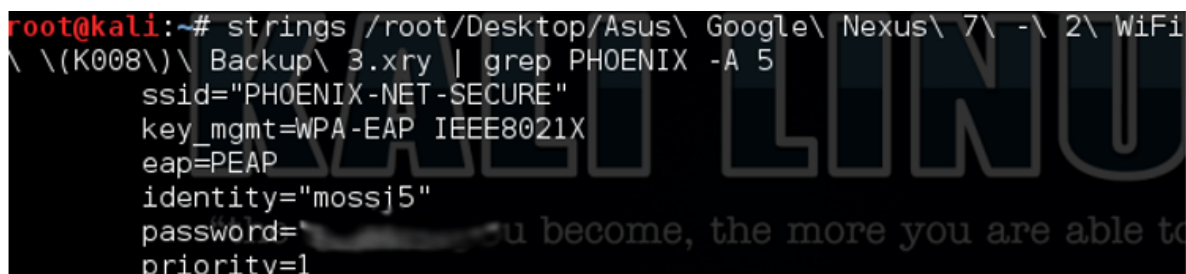
## Google Chrome

The username and password saved into Google Chrome on the device was fully recovered by XRY.

## Wi-Fi

XRY managed to recover both Wi-Fi network SSIDs, but only one of the Wi-Fi passwords. It seemed unusual that only one of the Wi-Fi passwords was recovered; however, this can be explained by the fact that PHOENIX-NET-SECURE is a WPA-Enterprise standard using an authentication method called Remote Authentication Dial in User Server (RADIUS). Authentication is achieved on a RADIUS Wi-Fi network by the device sending user credentials through a Wireless Access Point (WAP) to a RADIUS server to check active directory. Once this has been checked, the RADIUS server replies to the WAP and either grants or denies access to the device that attempted to connect. This process of authentication is entirely encrypted. This process was researched in more detail as the user credentials were saved onto the device for the PHOENIX-NET-SECURE Wi-Fi, even though it was not recovered in XRY. Applying the same technique used for the retrieval of the Google Account attention token in Linux, the authors found that the user credentials were stored in the raw file that XRY examined, as illustrated in **Figure 4**, below; it just was not picked up by XRY.

The PHOENIX-NET-SECURE SSID and password are stored inside the XRY file, even though it is not shown in XRY Reader. On the other hand, Deathstar-2G uses a localised pre-shared key for authentication. Everyone who accesses the Wi-Fi network is using the same key; and, if the device has previously accessed the network, it allows for automatic connection once the device is in range. It makes sense that XRY recovered this password due to how common this method of Wi-Fi connection is.

A terminal window on a Kali Linux system showing the output of a command. The command is: `strings /root/Desktop/Asus\ Google\ Nexus\ 7\ -\ 2\ WiFi\ \ (K008\)\ Backup\ 3.xry | grep PHOENIX -A 5`. The output shows several lines of text: `ssid="PHOENIX-NET-SECURE"`, `key_mgmt=WPA-EAP IEEE8021X`, `eap=PEAP`, `identity="mossj5"`, `password=`, and `priority=1`. A large, semi-transparent watermark with the word "LINUX" is visible in the background of the terminal window.

**Figure 4:** Wi-Fi password storage

## Images

The picture stored on the device was recovered by XRY. This is not surprising, as XRY states that it has full support for image recovery in the non-encrypted versions of Android 4.\*.

## **Android 5.0.2**

The results collected from 4.3 up to 4.4.4 were mirrored in the Android 5.0.2 data recovery and retrieval. The single difference discovered was that the information from the Google Chrome application was not recovered. The saved password on the device and any history or information about browsing were unavailable to view in XRY. The authors checked the raw data to see if any information pertaining to the browser could be retrieved, but the attempt was unsuccessful. The authors are unsure if this is a limitation of the XRY software itself or if Google has secured the Chrome browser in Android. However, as the information was recovered from the device using the 'backup' technique, it would have been expected that the information was stored somewhere. Therefore, if Google has indeed secured its built-in browser from forensic investigation, there is no reason why it cannot roll out this feature as an update for older versions of Android (pre-5.0), which account for 94.6% of the current market share.

## **Android 4.4.4 and Android 5.0.2 encrypted**

During testing it was discovered that if a device has a secure password and is encrypted it provided no useful data adding to the body of knowledge for this research. There are other methods of brute force that could be used to get the device's password, but this is not directly available through XRY. It must be kept in mind that, in circumstances where criminal activity has taken place, it is possible to request a court order for an encrypted device. The Regulation of Investigatory Powers Act 2000, Part 3, gives UK authorities the power to compel the disclosure of encryption keys or decryption of encrypted data by way of a Section 49 Notice. Suspects instructed to disclose keys can be prevented from telling anyone else about it, outside of their legal representatives. Refusal to comply can result in a maximum sentence of two years imprisonment, or five years in cases involving national security or child indecency (Open Rights Group 2015).

The authors attempted to view the raw recovered data through Linux, but could not retrieve anything relevant to the information initially stored on the device.

## **Conclusions and Future Work**

For the everyday user, it is vital to encrypt one's device. As smartphones and tablets have become more popular, people tend to store more and more important and personal information on them. The results obtained in this research show that, when people save their banking details onto their Android 4.4.4 device in Google Chrome, they are extremely vulnerable to someone attempting to retrieve their private data and bank details using data-recovery software. However, when running a device with full-disk encryption by default, all their data is completely secure from a normal forensic recovery attempt. Therefore, it is disappointing that Google backed out of full-disk encryption in Android 5.0 by default (Cunningham 2015), as an everyday user is often unaware of the consequences of a lost device. Users who have saved personal data and have not encrypted their devices are susceptible to full data recovery on their handset or tablet using traditional forensic software, such as XRY. The discovery made that RADIUS Wi-Fi authentication passwords fail to show in XRY adds to the body of academic knowledge and was the most important finding of this paper.

Using just one piece of software kept the results consistent. As much as this was convenient, the authors would have liked to have used other pieces of software, such as Cellebrite, to compare the two products. This would help in concluding and verifying these findings, such as the sudden loss

of access to saved passwords in Chrome in Android 5.0.2. The authors would also like to carry out this research on older devices that support Android 2.1 up to Android 4.2. This would help to discover and conclude whether Google supports backwards compatibility with security in its older OSs. It would also be interesting to perform some qualitative research by surveying groups of people regarding the version of Android they are running and their understanding of Android security to discover how secure people think their phones are and whether they believe that secure updates are rolled out onto older devices.

## References

Association of Chief Police Officers 2015, *ACPO good practice guide for digital evidence*, viewed 20 January 2016, <[http://www.digital-detective.net/digital-forensics-documents/ACPO\\_Good\\_Practice\\_Guide\\_for\\_Digital\\_Evidence\\_v5.pdf](http://www.digital-detective.net/digital-forensics-documents/ACPO_Good_Practice_Guide_for_Digital_Evidence_v5.pdf)>.

Android Twitter 2014, *Twitter feed*, viewed 5 February 2016, <<http://t.co/XVFX2tEbHv>>.

Android 2015a, *Encryption*, viewed 20 February 2016, <<https://source.android.com/devices/tech/security/encryption/>>.

Android 2015b, *Dashboards*, viewed 20 January 2016, <[https://developer.android.com/about/dashboards/index.html?utm\\_source=suzunone](https://developer.android.com/about/dashboards/index.html?utm_source=suzunone)>.

Christensen, D 2014, *CyanogenMod 99 success secrets*, Emereo Publishing, Aspley, Australia.

Cunningham, A 2015, *Google quietly backs away from encrypting new Lollipop devices by default*, ArsTECHNICA, viewed 21 January 2016, <<http://arstechnica.com/gadgets/2015/03/google-quietly-backs-away-from-encrypting-new-lollipop-devices-by-default>>.

Google Developers 2015, OAuth 2.0 Playground, viewed 29 January 2016, <<https://developers.google.com/oauthplayground/>>.

Graziano, D 2014, *The best way to completely wipe your Android device*, CNET, viewed 30 January 2016, <<http://www.cnet.com/uk/how-to/the-best-way-to-completely-wipe-your-android-device/>>.

Hoog, A 2011, *Android forensics - Investigation, analysis and mobile security for Google Android*, Syngress, Burlington, MA, U.S.A.

Lessard, J & Kessler, G & 2010, *Android forensics: Simplifying cell phone examinations*, Edith Cowan University Research Online, viewed 10 January 2016, <<http://ro.ecu.edu.au/cgi/viewcontent.cgi?article=7480&context=ecuworks>>.

Micro Systemation AB (MSAB) 2015, *Extract data manual*, viewed 19 January 2016, <[https://www.msab.com/download/product\\_sheets/en/XRY\\_Product\\_Sheet\\_EN\\_Digital.pdf](https://www.msab.com/download/product_sheets/en/XRY_Product_Sheet_EN_Digital.pdf)>.

Open Rights Group 2015, *Regulation of Investigatory Powers Act 2000/Part II*, viewed 25 February 2016, <[https://wiki.openrightsgroup.org/wiki/Regulation\\_of\\_Investigatory\\_Powers\\_Act\\_2000/Part\\_II](https://wiki.openrightsgroup.org/wiki/Regulation_of_Investigatory_Powers_Act_2000/Part_II)>.