CELLULAR SECURITY
Abstract— SMS has become an indispensable communication tool used by banking, government and other agencies. However, the SMS is not a secure service to be used to transport sensitive data, as its current security mechanism cannot guarantee protection from modification, eavesdropping and man-in-the-middle attacks. This paper examines existing security mechanisms; vulnerabilities and contributions put forward by researchers in enhancing SMS security features. Their proposals are criticized with a view to finding gaps and developing a comprehensive solution.

Index Terms—GSM, SMS Security, PKI, UMTS.

I. INTRODUCTION

The past two decades have seen an exponential increase in mobile communication usage. One of the reasons for this has been because of the increased functionality and services offered. Some of these include international roaming capability, cost effectiveness and new services (call waiting, call forwarding) and the Short Message Service (SMS). Of all services provided by mobile telephony, the SMS has become extremely popular.

The SMS is a service that provides text-messaging capability over a mobile network. The service is unique as it has a maximum message length of 160 characters, requires low bandwidth & provides assurance in message delivery within a specified period. This popularity of the SMS has made entities such as commercial, government take leverage of this platform to communicate with individuals and businesses. For example it is not uncommon for bank transaction or voting to be made via SMS, prompting acronyms such as m-banking and e-voting. In as much as SMS provides such a wide reach and convenience, is its platform secure? Is there assurance on the origin of the message or mechanism to protect data from eavesdroppers?

This paper aims to provide an overview of SMS security and provide a framework for which a comprehensive solution may be adopted. Section II describes the GSM network architecture that the SMS operates on. Message flow within the network is also described. Section III describes current security mechanisms service provides adopted with respect to protecting the SMS from modification, eavesdropping and unauthorized access to content. Vulnerabilities associated with the mechanisms are also discussed. Section III describes research work done to enhance SMS security and finally Section IV looks at the different elements that will be studied in view of enhancing SMS security. It should however be noted that this is in its design phase and results will be published subsequently.

II. SMS NETWORK ARCHITECTURE

This section discusses SMS traversing a GSM platform (figure 1.0). For delivery, messages must undergo three processes[1]: submission, routing and transportation medium.

Initiation of a text message is achieved through a mobile devices or External Short Messaging Entities (ESMEs). For ESMSs to deliver SMS, they must be connected to the mobile platforms through the SMS Interworking Mobile Switching Centre (SMS-IMSC)[1]. Operators offering SMS service usually have one or more servers known as the Short Messaging Service Center (SMSC). These servers are designed to operate in a store and forward fashion [2]. This is to enforce correct SMS message format and provide storage facility if the recipient is not available at time of delivery. Once the message is in SMS format, it is ready for routing to its destination.

Before routing can occur, the SMSC queries the Home Location Register (HLR), a permanent repository that stores information about subscribers, about the routing information of the recipient. Other information the HLR stores are the visitor location register number, service restrictions (e.g. roaming limitations), subscription data[3]. Depending on the result of the query, the SMSC either stores the message for later delivery or receives information about the Mobile Switching Center (MSC) where the intending recipient is being served. The MSC handles traffic routing, authentication, location updates and handover procedure on a mobile network and as such can route messages to its destination[3]. With the aid of the Visitor Location Register (VLR), a database for temporary storage of information about the subscribers being serviced by an MSC, the MSC obtains device information if its HLR is not local. With knowledge of device location, the MSC forwards the message to the Base Station (BS) in communication with the device for transmission over the air medium to the device.

The BS on receiving the message from the MSC needs to forward the content to the recipient (mobile equipment) via
Authentication is deemed successful if SRES from the challenge and compares it with that of the ME.

Signed Response (SRES).

The network also computes this onds to that sent to the Network

The l

the air. The medium comprises of the Control Channels (CCH) and traffic Channels (TCH)[3]. The base station uses the CCH to deliver SMS to mobile devices as mobile devices are designed to listen to this channel for SMS traffic. CCH is subdivided into the Random Access Channel (RACH) and Paging Channel (PCH). The BS uses the PCH to send a message with the Temporary Mobile Subscriber ID (TMSI) alerting the ME of an intending SMS. When the ME hears its TMSI, it contacts the BS via the RACH and confirms availability to receive SMS. The BS then instructs the ME to listen to a particular Standalone Dedicated Control Channel (SDCCH). With the SDCCH, the BS is able to perform encryption, authentication procedures and deliver the SMS message[3].

A. SMS SECURITY MECHANISMS

The resilience of SMS messages from alteration, replay and SMS spoofing attacks is subject to the platform the SMS transverses. As described the previous section, SMS can be transmitted via mobile networks (GSM, GPRS, UMTS and CDMA) and ESMEs. In this section a case study of security mechanisms deployed by two different network platforms will be examined.

CASE I: Global System for Mobile Communication (GSM)

GSM incorporates security mechanisms to ensure confidentiality and integrity of services. These mechanisms are embedded within the ME and the operator network. In order to ensure secrecy, GSM operators deplores the A5 algorithm. The ME initiates encryption at the request of the network; it uses an algorithm A8 to generate a ciphering key Kc with RAND gotten from the HLR. The key Kc is then used with the A5 algorithm to encrypt data[4]. Figure 2.0 depicts procedure.

The legitimacy of the user is determined by the network issuing a challenge to the intending user (ME). The HLR is assigned to send a 128bit random number (RAND), generated by AUC, to the ME. The ME uses an authentication algorithm (A3) and key Ki to compute a Signed Response (SRES). The network also computes this challenge and compares it with that of the ME. Authentication is deemed successful if SRES from the Network corresponds to that sent to the Network [4].

VULNERABILITY ANALYSIS

In ensuring non-disclosure of data GSM utilizes the A5 algorithm; however it has been demonstrated to be weak and liable to cryptanalysis as demonstrated in [5] who found the secret key in minutes. Although higher variants of this algorithm have being developed it is still not resistant to attacks.

ESMEs communicate with the SMSC is done through Short Message Peer to Peer protocol (SMPP)[3]. The SMPP protocol does not encrypt its content making SMS originating via ESMEs susceptible to man in the middle attack. Besides the weak encryption the GSM network utilizes, the network does not provide end-to-end security. This exposes plaintext messages at certain points within the infrastructure thereby making those areas vulnerable to man-in-the-middle attack.

Replay attacks are possible with the authentication mechanism the GSM operates. GSM infrastructure lacks mutual authentication [6]. As a result there is a possibility where an attacker depletes an authentication request or response to be replayed. Impersonation and false transaction could be executed thereby compromising existing authentication mechanisms. Moreover the GSM does not provide non-repudiation mechanisms. This greatly compromises the SMS in its ability to provide information assurance.
CASE II: Universal Mobile Telecommunication System (UMTS)

UMTS security mechanism was designed to eliminate the security flaws of its predecessor, the GSM. The security mechanisms can be classified into five distinct features; network domain security, network access security, application domain security, user domain security and visibility and configurability of security [7, 8]. Of the above mentioned security features, the network access security is a primary component as it is tasked with providing confidentiality, integrity protection of signaling data and authentication [7].

Unlike its predecessor, the UMTS incorporates mutual authentication between Serving Network (SE) and ME. In order to achieve mutual authentication, authentication vectors consisting of Encryption Key (CK), user challenge (RAND), expected user response (XRES), the integrity key (IK) and the authentication token for network authentication (AUTN) are used [9].

Authentication is initiated in the VLR and mutual authentication is successful when VLRs XRES matches the received RES. Figure 3.0 depicts the process involved in mutual authentication. Exchange of CK and IK to their entities are then permissible to perform integrity protection.

When authentication is complete, Message Authentication Code (MAC) function is applied on the link with CK and IK derived from the authentication process to preserve integrity.

VULNERABILITY ANALYSIS

UMTS design allows for interoperability with the GSM network. Interoperability between UMTS and GSM was necessary for easy transition to UMTS networks[8]. Moreover GSM had a wider coverage, it was necessary for UMTS subscribers to roam in areas with GSM only coverage and vice-versa. But lack of mutual authentication in GSM means a man-in-the-middle attack could still occur as shown in [10].

The failure of a mobile station to authenticate the serving network means that a variant of false base station attack could still occur. This was demonstrated in [11] where a redirection of traffic to a different network with weak encryption algorithm was possible.

In all, vulnerabilities in underlying networks could enable attacks on SMS possible. Contributions on eliminating these vulnerabilities are examined in the following section.

III. RELATED WORK

Several proposals have been put forward towards securing SMS services; contributions have being centered on either modifying network infrastructure or providing end-to-end security at the application layer. Hessain et al. in [12] suggested modifying the GSM infrastructure at the transport layer to enhance confidentiality. His proposal could strengthen confidentiality however the encryption process proposed is subject to operators’ encryption scheme and therefore requires their consent for implementation.

Most researchers don’t concentrate effort at the transport layer due to the limitation mentioned.

The second category of research is concentrated on is the application layer. It appeals to researchers and developers because implementation can be achieved without the service providers’ consent. However, providing security platforms at this layer is subject to the capability and capacity of the mobile equipment[13]. Therefore in developing security
mechanisms, encryption schemes, consideration of key management, energy consumption and speed must be considered.

Due to the limitation of mobile equipment, symmetric encryption techniques would be ideal as its encryption process is fast and uses shorter keys when compared to asymmetric encryption. However, key distribution is a problem when deploying symmetric encryption and the number of keys needed for communication is approximate to the square of number of senders [14]. Another drawback is its limitation to ensure confidentiality of data.

Ratshinanga introduced a hybrid scheme by proposing the use of asymmetric key encryption for key distribution process [15]. This ensured key protection and integrity of SMS service. However, the computational process is too high due to incorporation of asymmetric encryption scheme in encrypting the SMS distribution and key agreement. Researchers tend to combine asymmetric cryptography and symmetric cryptography to achieve fast processing speed and safe key agreement. The schemes described above can be does not incorporate authentication mechanisms.

The Public Key Infrastructure (PKI) provides a framework of providing assurance that the parties communicating are authentic. It involves obtaining certificate and extracting public key before encryption can be applied. The PKI scheme is high computationally intensive due to large key generations; it cannot be directly imbedded on a mobile device [16]. Many researchers in adopting this framework attempt to segment the high computational part to a dedicated server or third party. Others adopt the symmetric key cryptography to reduce the asymmetric cryptography overhead. Table I shows different schemes of PKI framework.

Although the PKI framework can be used to enhance SMS security, it is beneficial to large organizations such as banks or governmental usage. This is due to the large cost of maintaining servers or third party license cost. The PKI framework is also fraught with lots of complexities such as Service provider approval, certificate standardization and will require technical support.

IV. PROPOSAL FOR ENHANCING SMS SECURITY

Providing efficient end-to-end cryptographic solution is the most effective method of enhancing SMS security. This approach provides a single point of decryption (at the receiving node) thus ending numerous cycles of encryption–decryption processes currently experienced at different nodes of the mobile network. This approach will also reduce the number of attack nodes. End-to-End security can be applied at the application layer on the mobile equipment device and is software driven subject to:

a) Processing capabilities of mobile phone.
b) Memory capacity of the mobile phone or SIM card.
c) Processing capabilities of a crypto-processor that is embedded in the ME.

Due to the limitations of mobile phones, a lightweight infrastructure is proposed. The cryptographic technique deployed would be hybrid (both symmetric and asymmetric encryption). Using both modes of encryption takes advantage of their cryptographic strength and enhances the overall security of the SMS service. The infrastructure will incorporate

1) Using Symmetric Encryption for encryption and decryption.
2) Using asymmetric cryptography for key distribution. The Elliptic Curve Cryptographic (ECC) technique will be deployed as it is efficient and uses lesser number of keys as opposed to other schemes.
3) Deploying a pair of private-public keys to each subscriber.

![Diagram of Key Agreement Using Asymmetric Algorithm](attachment:Diagram.png)

**Figure 4.6: Proposed Solution Architecture**

- **1) Symmetric Encryption to provide confidentiality**
- **2) Hash Message to ensure integrity**
- **3) Sign with Private key**
- **1) Hash Message**
- **2) verify Signed message**
- **3) Decrypt with Secret key**
Integrity of the message will be achieved by hashing the message before it is sent to the recipient. Figure 4.0 shows the steps taken in achieving this process. On the receiver end, the message is first hashed and verified to ensure the integrity of the message.

The asymmetric encryption in the proposed architecture performs two tasks: it ensures safe key exchange between the sender and the recipient, it also prevents the sender from denying not sending the message as the private key will be used sign the message.

Validation of the proposed solution will be done in two phases:

- Testing different encryption techniques using a suitable emulator to determine suitable option for mobile systems.
- Installing developed solution on multiple mobile phone devices to determine their performance on different processing capabilities.

V. CONCLUSION

SMS is one communication tool that will be used for a long time due its simplicity and low cost. Current security mechanisms cannot curb modification, replay and man-in-the-middle attacks. End-to-End security is seen as a permanent solution in preserving user data. Challenges that need addressing include providing efficient yet lightweight cryptographic mechanisms, reliable key management and distribution system. This paper highlights the challenges with a view to get a framework where a solution can be obtained.

REFERENCES

When the Droid became the Bot
Trends, Threats and Investigation of a Mobile Botnet

Stephen Wilson, Kashif Kifayat
School of Computing and Mathematical Sciences
Liverpool John Moores University
Liverpool, U.K.
S.Wilson1@2008.ljmu.ac.uk, K.Kifayat@ljmu.ac.uk

Abstract—The vast expansion of cellular and Internet architecture has allowed for more capable and powerful smartphone devices which, in turn, allow a user to be connected continuously. However, the popularity and capability of these devices have fuelled the development of mobile malware, particularly over the last two years. More recently, researchers have begun to see elements of PC botnets creep into mobile malware, indicating the potential construction of mobile botnets in the future.

This research introduces the new threats posed by mobile botnets and examines the difficulties for examiners investigating a mobile botnet. The aim of this work is to raise the awareness of examiners and researchers and also to provide a potential solution to the investigation shortfall.

Keywords—Mobile malware; Mobile botnet; Smart Phones; Mobile network; SMS; Forensics.

I. INTRODUCTION

One of the major security concerns in recent times has been the development of botnets. These networks of infected computers are able to take part in multiple different types of criminal activity without the owners of the infected computer being aware.

The malicious code required for bots will either be developed by the botmaster/botherder (the controller of the bots) or adapted from openly released code from other malware (malicious software) writers. Once a host has become infected with malicious bot code, it has become a bot or a zombie. When multiple systems are comprised and controlled by a third party, it becomes a botnet - a collection of compromised computers [1]. This collection of computers and processing power is then at the mercy of the botmaster. According to Panda Security, 50% of computers it scanned during early 2011 were infected with malware [2]. This highlights the widespread infection rates of home users and the huge potential for botmasters.

Many bots, once infected, will proactively search for other vulnerable hosts attached to a common network. Bots are able to traverse the Internet looking for unprotected systems. Once found, they will infect the host then report back to its botmaster. A number of different methods can also be employed to infect other machines such as, malicious email attachments, infected websites or the downloading of a Trojan [3].

Two of the major threats that botnets pose to the internet is spam [4] and Distributed Denial of Service (DDoS) [5]. In 2011, one of the largest botnets discovered was Rustock. This was successfully taken down by efforts from industry and law enforcement agencies. At its peak, this botnet was responsible for 47.5% of all spam, judged to be in the region of 44 billion spam messages per day [6].

The second major threat, DDoS, involves a large number of bots making connection requests to a single target, usually crippling the target. This type of attack will typically mimic the same traffic and connection methods of a legitimate web browsing user [7]. The infected victim may not even be aware that they are taking part in a large scale attack [8]. This makes this type of threat extremely difficult to mitigate or prevent.

Mobile malware has seen an exponential growth in the last two years and perhaps more concerning is the shortage of security measures available for these devices. Antivirus and firewalls are almost unheard of for mobile devices. The recent finding of mobile malware with botnet like features [9], potentially indicates a growing trend by malware authors to successfully design a mobile botnet. With little smartphone security measures in place, malware authors have a large vulnerable target audience creating ideal conditions for the propagation of a mobile botnet.

In this paper we use a proof of concept, SMS controlled mobile botnet to spam an unsuspecting user and show the difficulties forensic examiners face investigating a mobile botnet. This research will show that without examiner awareness, it is unlikely that an infected handset would be detected or that the mobile botmaster could be tracked down. Additionally, a proposed solution will also be presented in the form of an interceptor or a monitor placed on the infected handset. As far as the author is aware, this is the first attempted research into forensic artefacts left behind by a mobile botnet or investigation into tracking mobile botmasters. One of the earliest pieces of research by Flo and Josang [10], presents the question, "Can mobile botnet become as difficult to track and close down as botnets on the Internet?". This research aims to answer that question.

The remainder of this paper is organised as follows: Section II introduces related work into the subject. Section III...
introduces the mobile botnet and potential capabilities. Section IV discusses our test environment and forensic artefacts. In section V we discuss the implementation of our proposed solution, the interceptor or monitor. Our paper then concludes in section VI.

II. RELATED WORK

Mobile botnets have only recently become a topic of interest to the research community. The primary research available focuses on the command and control methods and topologies and the threats posed from mobile botnets.

A. Flo et al. describes current mobile security mechanisms as inadequate for dealing with the latest breed of botnets [10]. C. Xiang et al also reports that, "Although advanced mobile botnets have not been witnessed in the main population of smartphones, we believe it is just a matter of time" [11]. The majority of research has been carried out on the most effective methods of command and control for mobile botnets. Singh et al. [12] presents a mobile botnet model that uses Bluetooth /SMS hybrid as a command and control method. This method would be difficult to manage and does not use smartphones to the best of their ability hence restricting some infected nodes to short range Bluetooth. Xiang et al [11], Zeng et al [13] and Mulliner et al [14] address these short comings by utilising both SMS messages and HTTP. Using HTTP restricts handsets that maybe only have 2G coverage. This, however, is probably the most effective way to propagate the malicious code rather than via MMS messages as discussed by Flo et al [10].

Traynor et al [15] discusses the potential impact of a mobile botnet on the cellular infrastructure and highlights one of the weaknesses that exist in local HLRs (Home Location Register). This is only one of the types of attacks that a mobile botnet could carry out.

III. MOBILE BOTNET AND POTENTIAL CAPABILITIES

Mobile botnets have only recently become a topic of interest to the research community. The primary research available focuses on the command and control methods and topologies and the threats posed from mobile botnets.

As 3G coverage spreads and allows smartphones to have a growing presence on the Internet, the threat from these devices will also increase. Not only are these devices connected to the Internet, they are also connected to the cellular infrastructure. This grants these devices a huge amount of flexibility, not only to the users of the device, but also to attackers.

The author of [16] describes mobile bots as an emerging trend in mobile malware and describes some of the commands that may be available to botmasters:

- Send SMS messages.
- Copy SMS messages stored on the device to a server.
- Copy the contact list stored on the device to a server.
- Install an application.
- Remove an application.

- Dial a phone number.
- Open a webpage.
- Change the list of C&C servers to connect to.

Access to the cellular network gives a mobile botnet a range of new attack vectors. Although these threats have not yet come to fruition, they are likely to happen in the future. Some of the types of attacks are detailed in the following sections.

A. Surveillance /Spyware

The motivation for this type of attack is fuelled by the wealth of personal information that is stored on users smartphones today such as, names, addresses and credit card data. A bot may also be able to carry out these additional functions:

- Listen in to calls.
- Monitor all texts and emails.
- Monitor the location of the handset.
- Monitor any Internet usage, in particular, banking websites/applications.

Using this information an attacker could then formulate different types of attacks such as, social engineering, phishing, spear phising and whaling which, as a result, would be more effective.

B. Cellular DDoS

As discussed earlier, smartphones also expose the cellular network to threats typically associated with computer network attacks. A cellular DDoS has the same aim as a computer based DDoS, to cause as much disruption to legitimate users as possible. It primarily targets call centres, automated or manned and can cripple these types of services, much like a computer DDoS would cripple a web server. The diagram below shows how it would work.

![Example of a Cellular DDoS attack.](image)

Figure 1: Example of a Cellular DDoS attack.

As can be seen above, the mobile botnet would be ordered to call a specified number at a specified time. This scenario presented above against the Emergency Services call centres could potentially end in the loss of life. In the future, this could
be regarded as a significant cyber weapon used to increase the effects of a terrorist act.

**D. Cellular Network Core Attack**

A cellular core network attack targets the infrastructure that manages the cellular network, potentially crippling the cellular network. Elements of this can be seen during public holidays such as New Years Eve [17] when users are unable to call or send SMS messages to friends and family. Research conducted by [15] demonstrates that a botnet that comprises of as few as 11,750 mobile bots can degrade area code sized regions by 93%.

The diagram displayed above show a mobile botnet flooding the cellular network core with traffic. The areas in red indicate the immediate strains on the network. Depending on the number of bots and requests, any of these elements could see their available resources tied up. However, the particular bottleneck discovered in [15] is the HLR (Home Location Register). The function of this component is described as, "When someone attempts to call a user, the caller's switch queries the appropriate HLR with a request for the targeted user's current location." The HLR is also a key component in the billing of customers and authentication methods.

Figure 2: Flooding of the cellular network core.

Traynor et al demonstrates that this component can be crippled with relatively little traffic and that it might even lead to its failure [15]. This would cause a large cellular blackout for the users requiring use of that particular HLR.

**IV. IMPLEMENTATION AND INVESTIGATION**

**A. Test Environment**

Our test environment has been set up using three Android handsets, two HTC Wildfires, the botmaster and spam victim, and one HTC G1 Dream, the infected handset. These three handsets were chosen as they all run on Android. These are well tested, stable handsets capable of being rooted.

The proof of concept bot code was presented at a hacker conference in the U.S. [18] and has been implemented in the rooted HTC G1 handset. The bot code currently only has the capability to SPAM using the infected handset. This would be carried out by sending a specially constructed SMS message to the infected handset that is pointed at the victim. The user of the infected handset never sees the forwarding of the constructed SMS message.

![Figure 3: SMS command and control.](image)

As can be seen above, the constructed SMS message can be broken down into:

**Bot: spam 079**** spam message**

or

**Identifier: function – target – malicious data**

Each of the elements of the constructed SMS message are customisable by the botmaster. They are able to change the identifier and function by manipulating the bot source code. The target and message are obviously specified by the botmaster depending on the type of attack he wishes to carry out.

The command and control method implemented by this bot code is SMS messages. This provides an extremely effective mechanism for sending commands to the infected handsets as it provides an almost guaranteed mechanism for delivering commands.

**B. Forensic Examination**

The aim of the investigation was to trace the source of the spam SMS messages and to identify the botmaster. The first step was to view the received spam SMS messages on the victim's HTC Wildfire. The circumstances of the investigation did not require the use of forensic tools on the victims handset. As such, only a manual examination was carried out to view some of the SMS messages received.

In this case, the owner was receiving spam messages from their friend's infected handset. The infected handset did not display any sent SMS messages in its sentbox, nor did it reside
anywhere in the SMS folder. However, billing information for the network provider displayed the SMS messages being sent to the victims handset.

Forensic analysis was conducted on the infected handset, the HTC G1 Dream. This was carried out using industry standard mobile forensic tools, XRY and CelleBrite. Both pieces of forensic equipment supported a logical examination however none supported a physical examination - typically done to recover deleted SMS messages.

The SIM card, memory card and handset were examined, but no evidence of the SMS messages were found. A number of small programs were found on the memory card with no immediate relevance. Potentially, an examiner may conclude the investigation, stating that the user of the infected handset has been sending out spam messages then deleting the evidence.

C. Further Investigation

As the infected handset was rooted, allowing access to other areas of the handset, we were able to follow the procedure outlined in [19] [20]. This procedure would allow us to image the user data areas of the handset, allowing us to view the deleted SMS messages on the device. This was carried out using the Android SDK (Software Development Kit). We then shelled into the handset and navigated to /dev/mtd. Within this directory six key files exist:

- mtd0 - Miscellaneous tasks.
- mtd1 - Recovery image.
- mtd2 - Boot partition.
- mtd3 - System files.
- mtd4 - Cache.
- mtd5 - User data.

Each of these files was then imaged for further analysis using the DD command:

```
dd if=/dev/mtd/mtd# of=/sdcard/mtd#.dd bs 4096
```

Unfortunately, searches were conducted over each of the files using message content. This still yielded no results, indicating that the SMS messages were not deleted and appear to never have existed.

D. Outcome

As can be seen from the investigation into this particular mobile botnet, very few artefacts can be found. Analysis into the workings of the botcode is provided below and gives us a possible explanation as to why this might be.

This code in particular was displayed at Blackhat [18] and displayed how the initial research conducted by [21] could be used to create a mobile botnet. [21] was able to successfully insert a layer between other layers involved in the receiving and sending of SMS messages. This was tested successfully on iPhone, Android and Windows handsets.

The author of [21] also describes how the injector is implemented, "The injection framework is installed in three steps. First, the actual serial line device is renamed from /dev/smd0 to /dev/smd0real. Second, the daemon is started, opens /dev/smd0real and creates the emulated serial device by creating a TTY named /dev/smd0. In the third step, the RIL process (/system/bin/rild), is restarted by sending it the TERM signal. Upon restart, rild opens the emulated serial line and from there on will talk to our daemon instead of the modem."

The author of the bot code, SMSbot [18], has altered the injector layer above to include bot functionality.

In figure 6 above, the implementation of SMSbot [18] has shown with the injector layer replaced by a bot layer. The green arrows represent a normal SMS message being received. However, the SMS messages that are preceded with the identifier shown earlier are identified by the bot layer and then forwarded on (red arrows). This explains why no data is seen by the user and potentially why the data is also hidden from
forensic examination. This is confirmed by the author of [14], "the control SMS messages never touch the SMS application or SMS database and stays hidden".

This presents two major issues with current forensic methods:

Firstly, an examiner may not even realise that a handset is infected or part of a botnet. They rely on antivirus to detect malicious code running on the device - which in this scenario was not effective. It was easy to see how the owner of the infected handset could be accused of the crime unless the examiner was aware of the malicious code.

Secondly, if they are aware that the handset is being controlled, the lack of artefacts present a major investigation and evidential issue. There is no data to suggest that anyone is using the handset as a platform to forward on spam or commands to other handsets.

This analysis is applicable for this type of handset only (HTC G1). As with mobile forensics, there are many types of handsets that can be analysed in many different ways. However, as displayed in figure 6, it is likely that any handset that has this bot code running on it, will not display any of the sent SMS messages to an examiner during forensic analysis.

Key to the investigation is the awareness of examiners. If little is known about this type of threat, it would be very easy for any examiner to look at the evidence and accuse the infected handset owner of spamming a large number of other users and deleting the evidence. There are clues on the infected handset, such as the program files found on the memory card, that if examined further could indicate the running of malicious code. However, it must be noted that the botcode did not show up in the process list or was detected by antivirus. [22] reports that this is not unusual, many antivirus applications for mobile devices do not provide sufficient malware protection.

V. IMPLEMENTATION

Our investigation has proven how dangerous a mobile botnet can be. Considering the challenges shown in the investigation, we propose a solution that may prove effective in providing examiners with additional data on the device issuing the commands.

Once investigators tracked down the bot, they could introduce a further layer into the SMS stack, either replacing the 'bot layer' with another, or by adding a new layer below the 'bot layer'. This monitoring layer would intercept and produce a log file of all SMS data and commands that was received by the handset.

![Figure 6: Implemented bot layer](image)

Figure 6 shows the implemented bot layer.

![Figure 7: The proposed monitor layer](image)

Figure 7 above shows where the monitor layer would sit in the stack. It has to either replace the 'bot layer' or be put into place below it i.e. closer to the modem. This layer would then be able to intercept the commands being sent from the botmaster. This will give the examiner not only the full SMS command content but will provide the MSISDN (phone number) of the botmaster / servant bot.

If an examiner is able to get the MSISDN of the source of the commands, they then have a chance of locating the botmaster. This task can be made even more difficult if the botmaster has incorporated any evasion methods such as, prepaid SIM cards.

VI. CONCLUSION

As the capabilities of smartphones continue to grow, so does the amount of malware available for these devices. The aim of this paper was to highlight the potential new threats of a mobile botnet and to establish the challenges that may meet forensic examiners attempting to investigate this problem in the future.

Previously we presented the question posed in [10], "Can mobile botnet become as difficult to track and close down as botnets on the internet?". Evidence put forward in this paper suggests that it is just as difficult, if not more difficult. Current forensic methods were used in an attempt to track down the source of spam SMS messages. The infected handset was tracked down successfully using its MSISDN. However, the device was forwarding the SMS messages from an unknown third party (botmaster).

Unfortunately, very few artefacts were found on the infected handset to allow the botmaster to be traced. However, it was established that the received SMS messages containing the bot commands were not even reaching the SMS inbox or sent folder because the bot code that was running on the device was intercepting the messages.

Our proposed solution would require awareness from examiners and a level of technical knowledge to implement the 'monitor layer' correctly. The 'monitor layer' would then provide the examiner with not only the evidence that shows the handset is being controlled but also the MSISDN of the device sending the commands.
The work carried out in this paper confirms that mobile botnets are a very effective means for carrying out attacks with very little risk to the botmasters issuing the attack commands. The proposed solution will provide examiners with an effective method that can be used in future investigations.

Future work should focus on ways to mitigate some of the attack vectors outlined in this paper and also to explore the role in which network providers can play in identifying malicious SMS traffic or commands.

REFERENCES

Analysis and Design of Security for Next Generation 4G Cellular Networks

Ginés Escudero-Andreu, Raphael C.-W. Phan and David J. Parish
School of Electronic, Electrical and Systems Engineering,
Loughborough University, LE11 3TU Loughborough, U.K.
g.escudero-andreu@lboro.ac.uk, r.phan@lboro.ac.uk or d.j.parish@lboro.ac.uk

Abstract—The 4th Generation of mobile communications (4G) has been designed to fulfill strict security requirements. However, design shortcomings may open the door to threats, casting doubt on the reliability of the system. Recent publications found critical vulnerabilities, such as breach of user identity privacy, user location tracking or inability to guarantee privacy and integrity of communications, proposing amendments without considering fully backward compatibility and inflicting further menaces. This paper intends to give an in-depth insight into this issue, analysing the latest release of Long Term Evolution (LTE-Advanced) specifications in order to identify strengths and deficiencies. An evaluation of the alternative solutions is made, proposing future research on this matter with diverse approaches to design an enhanced solution more efficient, respectful with backward systems and similar traffic load.


I. INTRODUCTION

Mobile communications have been developed during the last decade, reaching unexpected figures of active users and making a reality services which nobody would ever imagine before. Social networks, real-time navigation systems along video-streaming services such You-tube or live music comprise a huge list of on-line services whose use requires high data throughput rates. In contrast with old tendency, now the information flows in both sides, demanding vast amounts of data traffic not only into the downlink, but also into the uplink to share multimedia content on Internet.

Currently, most of the mobile services are based on Global System for Mobile Communications (GSM) or Universal Mobile Telecommunications System (UMTS) access networks with High-Speed Downlink Packet Access (HSDPA), technologies linked with 3/3.5G which use is reaching the edge of their capabilities. Users claim for an unique mobile solution to provide broadband services without being affected by mobility factors, conditioned to urban coverage areas or lower peak rates than landlines Internet connections.

On the other hand, other services such as mobile banking, on-line payment systems and social networking deal with private information which protection is mandatory. Security becomes a priority issue for mobile operators. In the early stages, release 8 was named Long Term Evolution (LTE) and proposed as 4G candidate by the 3rd Generation Partnership Project (3GPP). Thereafter being selected as one of the two candidates for the International Mobile Telecommunications competition (IMT) by the International Telecommunication Union (ITU), it was improved to reach higher peak rates increasing user density per cell, calling it as LTE-Advance after release 10.

At this moment, 4G deployments are planned all over the world and billions of potential users await to gain access to high-speed and seamless communications. LTE technology, with its latest release of LTE Advanced, is the best candidate to be elected as 4G standard. Core and radio-access network are based on flat all-ip networking, making interoperability extremely easy and fast, but increasing the requirements of higher security capabilities. Being supported by most of the mobile operators providers and manufacturers, a victory is envisaged into the ITU competition to choose it as standard for IMT-Advanced.

This paper analyses the security aspects of LTE-Advanced technology, reviewing all the alternative solutions proposed by the research community and guiding future research. Section II depicts a brief description of the Evolved Packet System, protocols and policy used to provide safe communications. Section III completes the analysis pointing out the identified threats and weaknesses, linking them with alternative solutions on the Section IV. Finally, Section V and VI conclude with a summary and personal opinion, introducing future research on this matter.

II. EVOLVED PACKET SYSTEM (EPS)

EPS is an evolution of UMTS system in order to satisfy a huge demand of high-speed data rates and provide support for a constantly increased number of cellular users. In contrast with previous generations, LTE has been designed considering all the services as IP data traffic, voice calls included, facilitating integration of user data traffic into operator core network but fostering new well-known attacks for IP networks.

3GPP has developed a robust system, adapted to provide services with high requirements of data bandwidth and able to cope the growth of mobile-phone users. Their effort is also focused on security issues to consider the previous UMTS
threats in order to prevent future menaces in LTE.

Between the characteristics of the latest release of LTE Advanced, a few innovations are highlighted:

- Carrier aggregation, providing a flexible and more efficient use of the spectrum.
- Enhanced downlink Multiple Input/Multiple Output (MIMO), increasing the cell throughput.
- Enhanced Inter-Cell Interference Coordination (ICIC), enhancing interference management between cells.
- Relay Nodes (RNs), used to cover wider areas.
- Home E-UTRAN Node-B’s (HeNBs), improving indoor data rates.

LTE-Advanced system should target peak rates of 1 Gbps/500 Mbps for the downlink/uplink respectively, according to the theoretical expectations [1], exceeding the 300 Mbps/75 Mbps rates of LTE standard and supporting mobile speeds up to 350 km/h.

A. Architecture

The whole system is divided in two parts: Evolved Packet Core (EPC) in figure 1 and Evolved UMTS Terrestrial Radio Access Network (E-UTRAN) in figure 2. Compatibility with old network infrastructures is provided, based on 3GPP or non-3GPP technologies, being them classified as trusted or non-trusted accessing networks [2]. This classification describes the way to authenticate the users, defines how to migrate their security contexts during a handover process and establishes the behaviour of the core network towards the user.

![Fig. 1. The EPC architecture](image1)

As specifications describe [3][4], core network is composed of:

- Mobility Management Entity
  MME is the process unit to manage communications between UE and e-NodeBs, identify and authenticate the user, assign bearers, establishing downlink/uplink and NAS security, storing key material and providing services.
- Home Subscriber Service
  HSS coordinates the functions of the authentication centre, stores the subscription-related information to support the network entities handling calls and sessions. It generates fresh key material over demand.

Serving Gateway
S-GW is the local mobility anchor, controlling the packet routing and forwarding tasks, beside of other transport level functions. It implements quality-of-service (QoS) functions and lawful interception features.

Packet Data Network Gateway
P-GW acts as intermediate between the operator network and Internet, providing IP address for each UE and recording service charges.

Serving GPRS Support Node
SGSN provides compatibility with packet switching services of old access networks, such GERAN (GSM system) or UTRAN (UMTS system).

On the other hand, the radio access network acts as a link between the User Equipment (UE) and the core network. EUTRAN Node B (eNB) entities provide services to the end user, by means of the radio channels. After release 10, support for Relay Nodes (RNs) and Home EUTRAN Node B (H-eNB) or femtocell are added to provide a wide coverage area without minimising QoS. A group of H-eNBs may be gathered and addressed to a Home EUTRAN Node-B Gateway, reducing the number of interfaces linked directly with the core network.

![Fig. 2. E-UTRAN architecture](image2)

B. Security Features

Security architecture groups the security features into five categories [5], establishing their scopes:

- Network access security, to provide a secure access to the services by the user.
- Network domain security, to protect the network elements and secure the signalling and user data exchange.
- User domain security, to control the secure access to mobile stations.
- Application domain security, to establish secure communications over the application layer.
- Visibility and configuration of security, bring the opportunity for the user to check if the security features are in operation.

Although, 3GPP enumerates the security and privacy requirements in which the system bases its reliability [6]. In term of security, the EPS should provide a high level of security equal or better than the current systems, protection against threats and attacks including Internet menaces and the assurance of any user is able to gain access to the serving network without being authorized previously.

Moreover, the requirements remark a basic principle for the user privacy: protection of communication confidentiality and keep the user location and its identity secret. This principle may be intentionally breached for lawful interception of communications.

C. User identification

Identification of the user is the first step before gain access to the network. UE establishes contact with the nearest e-NodeB triggering the registration process. During the first attempt, MME is not able to identify the UE by means of a temporary identity (Globally Unique Temporary Identity - GUTI), thus, serving network will send an IDENTITY REQUEST message.

A reply message is made by the UE with its International Mobile Subscriber Identity (IMSI) transferred in clear text, because no security context has been established before. It breaks the requirements of user identity confidentiality mentioned in the security requirements [6], exposing the user identity to eavesdropping attacks over the radio interface.

Once the MME receives the IDENTITY RESPONSE, GUTI is allocated and paired with the original IMSI. The temporary identity may change for different reasons, being no necessary to transfer the unique subscriber identifier unless serving network can not retrieve it from the GUTI.

D. Authentication and Key Agreement (AKA) protocol

Before the establishment of a security context, serving network needs to verify the authenticity of the mobile equipment (ME), as well as the UE has to verify the serving network. Mutual authentication is achieved by means of an AKA protocol, called EPS-AKA. The process is triggered by the serving network, after the identification success. Figure 3 shows the sequence diagram.

Initially, MME checks the stored key material and its freshness. If there is any authentication vector (AV) available it will use it to start the authentication process. If there are no vectors, or the existing ones are no fresh enough, MME will request new AV to the HSS. Each AV is associated with an IMSI, being only used once per AKA instance. 3GPP recommends to send only one vector on each reply message, but it is not mandatory.

Each AV is composed according to the equation

\[
AV := RAND \parallel AUTN \parallel XRES \parallel K_{ASME} \quad (1)
\]

where each parameter is:

- **RAND** is the challenge to proof the user authenticity.
- **AUTN** is the parameter to proof freshness of authentication vector and serving network authenticity.
- **XRES** is the expected response to the challenge.
- **K_{ASME}** is an identifier to derives the same key hierarchy in both sides.

The process start with an AUTHENTICATION REQUEST message, composed by an authentication challenge RAND, AUTN parameter to verify the freshness of the key material besides of serving network authenticity, and **KSI_{ASME}**, a value used by the mobile equipment to generate the same key value for **K_{ASME}**.

Once the ME receives the message, it retrieves the **KSI_{ASME}** parameter and passes the other ones to the Universal Subscriber Identity Module (USIM). USIM verifies the freshness of the authentication vector, deriving the sequence number from the AUTN parameter. If the derived value match with the expected sequence, a challenge response RES is computed and send back to the UE. Then, two keys are derived from the master key K, one for integrity (IK) and another for confidentiality (CK).

An AUTHENTICATION RESPONSE is sent back to the MME, generating on it the same key pairs CK/IK and completing the AKA process. Now, both extremes are able to generate the same key material, following the scheme of figure 4.

Each time an AKA process is called, key material is regenerated based on the new value of **K_{ASME}**. Master key K is securely stored in the HSS and IMSI, without being
transmitted or used directly. It is only used to derive the entire key hierarchy.

E. Access-Stratum and Non-Access-Stratum Security (AS/NAS)

In contrast with UMTS, LTE incorporates a secondary security layer with an extra encryption and hashing level. These two layers are AS and NAS security. AS security protects the signalling and data traffic between the UE and the e-NodeB, where all signalling messages are confidentially protected by $K_{RRC\text{enc}}$ and integrity protected by $K_{RRC\text{int}}$. User plane traffic is only encrypted using the key $K_{\text{UP\text{enc}}}$, leaving integrity protection as free choice of the operator.

On the other hand, NAS security duplicates the robustness of the system against attacks, incorporating another integrity and confidentiality protection for both data flows (signalling and user plane) between UE and MME. The keys used to encrypt and calculate hash codes are $K_{NAS\text{enc}}$ and $K_{NAS\text{int}}$ respectively.

III. SECURITY THREATS

A. User Identity

User-identity transference in clear-text during the initial attach procedure compromises the entire system, creating a security gap which exploitation allows an eavesdropper to track the user cell-location or perform a man-in-the-middle attack between the UE and the eNB by means of user IMSI impersonation and relay of user messages.

B. Femtocells

Femtocells becomes a great risk, as they are the only core-network devices out of the operator control. These devices are located inside the user home installations, connected to a local e-NodeB to provide wide coverage extension, but being physically accessible by the customer.

Tampering the device to avoid software updates, or modifying their usual behaviour may create an important breach inside the core network [7]. Once the device software is modified without the operator consciousness, UE may be driven to use clear-text communications during the AKA process, faking the security capabilities to force a null encryption.

C. Interoperability

3GPP establishes the conditions to migrate security contexts associated to an user, considering the classification of the access network. Direct context transference is only possible when the two technologies belong to 3GPP trusted access networks, but for a limited period of time. UE must be re-authenticated once the migration process is completed without interrupt the service.

Nevertheless, multi-access and seamless mobility requires a strong collaboration between the different operators and a restricted interoperability policy to avoid unauthorized access to the system. Furthermore, compatibility with legacy systems is also an acceptance of their vulnerabilities. GSM is a clear example of a vulnerable system, with remarkable weaknesses identified, such as a man-in-the-middle attack (MITM) [8].

D. RRC signalling

3GPP specifications files include a few RRC signalling messages whose transmission happens in clear-text, before the security domain is established. These messages can be easily sniffed, and replied towards the eNB several times to collapse the system with a traffic-injection attack. The serving network is unable to detect the attack and process all the petitions, reducing the resources to attend real service request. The attacker simulates a usual behaviour of a legitimate UE with a direct consequence, a deny-of-service (DoS) attack [9].

E. Other threats

Being an all-IP networks makes the system vulnerable against IP attacks, such Deny of Service (DoS) over the public IP addresses of the core network interfaces, traffic eavesdropping and injection attacks. Selective flooding attacks may reduce the QoS or even cut the service of the legitimate users.

Furthermore, physical protection over the access network infrastructure must be considered, besides of the core-network elements and wired connections. Specially for HeNBs, as we pointed out above, where the devices are easily accessible by bad-intentioned users.

IV. ALTERNATIVE SOLUTIONS

Numerous alternatives have been designed all over the world, in order to increase the security and mitigate threats. Most of them are focused on the AKA protocol, proposing alternative authentication schemes to accomplish similar functionality of EPS-AKA. Other papers re-design part of the core-network architecture, adding new entities to manage authentication functions or act as trusted third parties.

A solution is focused on the seamless mobility scheme, identifying threats of the AKA protocol used for 3G/WLAN interworking that can be extrapolated to 4G systems, as EAP-AKA is the designed protocol in 4G interoperability. They proposed an amendment of the EAP-AKA specified
by 3GPP, which accomplish perfect forward secrecy, protect the privacy of user identity and avoids a man-in-the-middle attack [10].

It is based on Elliptic Curve Diffie-Hellman (ECDH), using a symmetric cryptographic system rather than using the centralised system of EAP-AKA, reducing the computational overhead at the same time. Further research should be conducted to circumvent shortcomings dealing with interoperability among 3GPP/non-3GPP technologies and backward compatibility.

Between the AKA improvements, paper [11] introduces J-PAKE mechanism to strengthen EPS-AKA adding perfect secret privacy property. In addition, they spot a threat, lack of identification privacy, without citing any solution to tackle it.

Another solution is proposed on paper [12], called SE-EPS AKA, incorporating an wireless public-key infrastructure (WPKI) to safeguard all the communications before the security context is established. Whilst, the public-key system not only provides mutual authentication between UE-MME, but also for the MME-HSS connection inside the core network.

Consequently, user identification is privacy protected and core-network links communications are safer, but conditioned to the secrecy of the private keys which should be stored in the USIM/MME/HSS at the point of manufacturing.

Likewise, paper [13] introduces EAKA in combination with a new concept of USIM, called ESIM, to achieve mutual authentication among UE-MME-HSS. Furthermore, this scheme amends the AV structure with additional challenge parameters to obtain a proof of authenticity of core-network elements. Now, USIM/ESIM assumes the entire authentication functions inside the UE, without delegating any function in the mobile equipment.

Obviously, this solution requires strong variations of the current mobile system to be compatible with the re-defined USIM card and AV, but it also demands the extended study of inter-operability between different accessing technologies for backwards compatibility. Operators may be concerned about the reduced benefit of accomplishing a strong refinement into the system.

Building on the former approach, a new AKA protocol is suggested together with an architecture improvement on papers [14], [15], considering all the threats that may compromise the mobile equipment. Using a Trusted Mobile Platform (TMP) and biometric identity, mutual authentication is achieved in a safe environment through the HAKA protocol. It is based on two phases: local authentication for Mobile Equipment and USIM, and remote authentication between UE and serving network.

However, this new scheme implicates several undesirable features to the system, increasing the required authentication traffic load to complete an AKA process and forcing the operators to acquire biometric parameters for each user and store them on the core network.

Moreover, paper [16] recommends an alternative communication framework, called Y-comm, which offer a common scheme to interconnect dissimilar accessing technologies such WiMAX, Wireless or Ethernet local networks, but sharing the same security environment.

This framework is designed to provide an impenetrable system, where each level is protected enough to evade potential attacks. Unfortunately, a reformation of the entire mobile system without considering the previous schemes may result in relevant cost expenses and extra effort, making this idea less interesting for the operators.

V. Conclusion

Mass effort has been made in order to protect and guarantee secure mobile communications towards LTE infrastructure since the earlier releases of 3GPP specifications. However, future improvement is required to neutralize the actual attacks and vulnerabilities that may compromise the user identity, data privacy, signalling traffic as well as access and core-network equipment.

Mobile services have become an integral part of our life, using them as payment method or storing private information. Secure communications are mandatory, in order to reduce the impact of eavesdropping attacks or impersonations of legitimate network equipment. Future research is required in order to tackle improvement into the EPS, counteracting the actual threats and to solve the identified weaknesses.

VI. Future Research

There are many issues regarding to the threats and vulnerabilities mentioned above that should be investigated widely to find alternative solutions that accomplish all the security requirements proposed on 3GPP specifications. Our research is focused on the evaluation of all the AKA alternatives proposed by research groups using Casper/FDR, a testing tool to evaluate security protocols modeled by functional programming. Identification of further weaknesses is essential to design better security environments for the 4G
Further on, using the gathered information from the simulations, the design of an on-line Intruder Detection System (IDS) tool will be conducted, completing our contribution to make the new mobile communication era safer and bringing an additional feature to deal with attacks in real time, reduce their scope or isolate them in order to reduce the effects on the QoS of the legitimate users.

REFERENCES


[10] Mun, H., Han, K., & Kim, K. 5G-WLAN interworking: security analysis and new authentication and key agreement based on EAP-AKA. Wireless Communications Symposium, WTS2009 (pp. 18). IEEE. (2009)


Abstract—There is a new demand on research into smartphones as they become more apparent in the court of law. We discuss the lack of security which is utilised by the user on an iPhone, and more importantly the lack of forensics tools that are available to extract the data which is needed in an investigation, whilst maintaining the integrity of the iPhone in question. We critically analyse forensics tools which are already available in an effort to discovering how we can access the data on an iPhone using the backup of the iPhone and display this recovered information into a format which will be easy to use and read for an investigator. We investigate how we can interact with the data in SQLite databases of the iPhone and learn how we can extract this data. Our solution for this problem is to decide on a technique that can be used in practice by all authorities, such as forensic investigators, working for law enforcement. This solution is an excellent idea in how we can achieve the investigation whilst holding the integrity of the original evidence item, and also making the investigation as error free as possible. Based on this underlying foundation of our research, and the solution we have identifies, we are able to propose a forensics tool which will be able to support a forensics investigator in not only the extraction of evidence from an iPhone, but other SmartPhones alike.

Keywords—iPhone Security; iPhone forensics; forensics analysis; forensics investigation;

I. INTRODUCTION

Basic mobile phones in the 90’s had very few functionalities. [3] As mobile devices revolutionised and they become far more advanced in their capabilities, we started seeing computer like features for example. This new generation of mobile phone introduced us to the SmartPhone. Perhaps the main example which we focus on is the highly popular iPhone introduced by Apple in 2007. [7] Forensic investigators follow the ACPO Guidelines [1], which when addressed the iPhone becomes a massive challenge when it comes to an investigation of the data it contains. Aside from extracting data from the iPhone another challenge is apparent in a way that devices are always connected to the internet, and using Apps the iPhone can be remotely wiped by the owner who can be logged onto the App on a computer. This challenge also brings about the question of security in iPhones, as if the user can access the iPhone remotely, who else can? People are storing more data on the iPhones everyday, including businesses, if it was lost or stolen the person who finds the iPhone may be able to retrieve sensitive data, or even details about where the user lives or where they go and who they are. This is a security threat that looms over this new evolution.

iiPhones can be added to digital forensics investigations, just as computers. This can be seen in the Dr. Conrad Murray trial, in which his iPhone contained enough evidence regarding the Michael Jackson death for prosecutors to make the case [6]. Cases such as these caused a demand for an extraction method of the evidence that could be found on an iPhone, as Apple had continued to stall in providing law enforcement with a method to do this, or access to the operating system on the iPhone in order to develop a tool themselves.

This research provided us with the incentive to solve the problem of how we could access and extract data from the backup of an iPhone and display it into an easy to read and use format for a forensics investigation, whilst ensuring we keep the integrity of the data which may be used in a forensics investigation. This aim is an important one to overcome, as with users putting more data on their iPhone than on a computer, a secure method of extraction is necessary for the iPhone which will enable the Computer Investigators to extract data they need. In this research paper we identified a solution which allowed us to meet this aim, first of all we created an Online Backup Tool which enabled us to extract text messages using an online application, in order to support this we created a website. Finally we create a proposal for an iPhone Forensics Suite which is our final solution to the problems we identified with tools and techniques which are already available.

The results of this paper provide an outline for a tool that showcases our solution and therefore verifies this research as a success. This tool is too large and far too complicated for us to implement the complete solution, our proposal therefore is the initial building blocks of the complete tool. This research paper and it’s results is a contribution to the start of a complete forensics suite that can be used in by investigators for the investigation of smart phone devices in the future. Using this paper, work on this tool can begin as we have proven our theory of data extraction works.

In the structure of this paper we will look at the background for our research and identify how we have arrived in this specific area. A tool analysis will then be created on some already existing tools in which we can identify their strengths and weaknesses. A design for our solution will then be built using the analysis from the tools. Following the design process we will add our implementation as to how we created the final outcomes of this research paper, and we will then present our final solution in this
research paper. An evaluation and conclusion of the research will then follow.

II. BACKGROUND

Mobile phones have always challenged an investigator when it comes to performing forensic acquisitions on them. With volatile and non volatile being held on them, operations which were not as simple as image the device and investigate the image in order to maintain the evidence were not as easy to complete. Looking at earlier mobile devices, we learn the methods used to extract data from them are via looking at data which can be extracted from the SIM card, data extracted from the memory of the phone and finally accessing the data which is stored by the service provider of the mobile phone. Each of these methods had advantages and disadvantages, for example the SIM card can only be used to access information such as the contacts stored on the SIM and also some text messages and recently dialed numbers at a push. Although containing great evidential values, this solution missed the results which could be obtained from other applications such as the internet evidence, and also evidential items which can be present in other aspects of the phones features such as calendar dates and notes. Gaining information about a users mobile phone can be hard when contacting the service provider. Sometimes they will not be willing to give out client data. [8] Phone memory data can also be hard to extract from, especially seeing as some mobile devices do not have an external memory card, they are built internally. [4]

These problems with the evidence extraction from older mobile devices has broadened and increased with the smartphones we are seeing today, such as the iPhone. As we have already seen the current acquisition processes do not allow for finding other data which may be available on the iPhone. This data can include information from a range of inputs on the iPhone, such as call logs, SMS messages, GPS data, notes, contacts, app data, social networking etc.

The sheer amount of this data availability leads us to ask how secure is the iPhone as a device which is used in many businesses to handle and communicate sensitive data. As more and more data is stored on the iPhone, more people will attempt to find this information. As the iPhone is a closed system and will not let users download apps from anywhere but the App Store, many users have continued to hack their iPhone by jailbreaking it. This jailbreak reduces the security which is left on the iPhone, and we start to get iPhones which are targeted because of this weakness. We have already seen this to some extent with iPhone hackers, in one case for example a worm named ‘iKee.A’ was released onto jailbroken iPhones. The worm rendered the iPhone as useless until the hacker was payed a ransom by the targeted iphone’s owner. Within two weeks, nearly 21,000 iPhone were infected with this worm. [9] Another addition to this worm then developed ‘iKee.B’, which was an entire BotNet of iPhones. [11] As we have already discussed, the user inputs a lot of data onto their iPhone, making it something similar to a copy of their life, this makes it very dangerous if this iPhone can be hacked, making it a serious privacy issue for the owner. Availability of GPS on the iPhone and apps such as ‘Find Friends’ on which a user can see other users with iPhones and where their location is in real time also adds to the impact of privacy threats, especially to users who do not know how to properly secure their iPhone. [5]

The advanced and sophisticated technology on the iPhone makes it harder to extract the relevant data from it. Currently there are three known ways to access evidential data on the iPhone, this is to view the iPhone backup on it’s paired computer, hack the iPhone or disassemble it. [12] Critically though, we know that the latter technique will damage the evidence and is therefore unsuitable, this can also be said for the hacking technique, it is therefore the iPhone Backup which we see an opportunity for access. [2]

III. TOOL ANALYSIS

From our research we learnt the problem is with extracting data from the iPhone. In order to provide a solution for this, we decided to test several pieces of already existing software in order for us to derive an answer which would both be better then these software examples, and also give us an idea of what we want to achieve and how we can do this. We tested MDHelper, iPhone Backup Extractor and also FTK (Forensics Toolkit) to help us do this. If we look at the two tools we reviewed and gave us a result we were looking for, we can see that the iPhone Backup Extractor tool was much more user friendly than the MDHelper tool. It was easier to install for a user who did not have knowledge of how to use terminal or access to a Mac for that matter. This was a downside for both of the tools regarding the restricted Operating Systems, as each of them could not run on the other. MDHelper is a free tool, whereas the Backup Extractor must be paid for, although a trial is available. In answer to our first question then, “How easy is the software to install and who is it available to?” we can define that the Backup Extractor tool is the simplest to install and use, and also has a better user interface, however the MDHelper tool is free of cost. This also answers our next question “How user friendly was the software to use?” as we have already decided that the Backup Extractor was the easiest of the two to use. In terms of content, our final question “What content was available to be recovered and how was this displayed?” both of the programs delivered the same data, although in our opinion this data was difficult to parse through to find data which was of interest to the examiners due to the way the results were displayed. We were able to extract nearly everything from the iPhone Backup using these tools, this includes the messages,contacts, images and other necessary aspects in the backup. Using these analyse results, we can determine that our aim is how we can access the data and extract it from the backup to display it into an easy to read and use format for a forensics investigation whilst keeping the integrity of this data.

IV. DESIGN

We are aware this is an ongoing problem to find a solution for as iPhone are continuously improving and the forensics
methods are lagging behind and Apple are refusing to release a forensics suite themselves, therefore we have divided the progress into three steps. First of all we have decided to implement an Online Backup Extractor Tool which will also require a supporting website which will be the second step. This tool will extract the relevant data from an iPhone backup, which is a useful tool to have running on the internet without the need for downloads, and also will allow us to learn what we can and can not do when it comes to extracting data from the iPhone. We can then use this knowledge to then start to implement the final step in this research, which is a proposal for a forensics tool that will extract the data from an iPhone backup whilst with holding the integrity of the device, and also presenting the examiner with useful data after the process.

V. IMPLEMENTATION

In this section we will now start to develop our designs. We will first look at the Online Backup Tool, followed by the supporting website. The results we find in this section will be the foundation for forensics suite proposal.

A. Online Backup Tool Implementation

The online backup will be running from the website, in which the user will be able to upload a file, convert it and then download it again. The file we will be uploading is a messages SQLite file taken from the backup of a used iPhone.

Uploading the file is the first step we need to look into. This will be done through the use of a PHP script which will be run from the form shown below in Figure 1.

Figure 1: Upload form

We have used a local server called ‘MAMP’ in order to complete this stage. The upload section works and we are able to continue.

In order to use this online backup tool, we need to be able to get the data required, which is in the sqlite databases, and put it into a format in which we can read without the use of other tools. We first tested this by using the Terminal window on the Mac in order to learn the commands which would allow us to interact with the SQLite databases. Once this was achieved and we were able to retrieve data in the terminal window, we moved to PHP coding. After a number of SQLite variation problems between databases, we were able to create a PHP code which read the SQLite database on the local server.

By inserting loops and other expressions such as ‘PDO’ we were able to run a PHP which would take an iPhone Backup of the SMS data, and export it into a format which was shown on http as a table, with the option of downloading the file as a csv document. The data returned included the address, physical text, ID and time of the SMS message sent. This proved our theory of using the SQLite databases as a success and something we were able to carry on looking into. An exported example can be seen below in Figure 2.

Figure 2: Export from backup

A supporting website for this PHP code was also created, in a way that it could be expanded and added to in the future.

In summary to this implementation section, we know we can retrieve the data from the iPhone backup and we also know how we can do this. If this tool we to be put together completely and then moved onto the supporting website, we would have a complete tool which works. We have not done this ourselves as this is out of the scope of the research and now we know we can achieve this, we can move onto the final aim for this paper.

B. Supporting Website

This website is not the main feature of this research although it does still have a role to play. First of all the website is now providing a platform on which the Online Backup Tool we have already created the implementation for can be situated onto. This is the main purpose of the website. However it is also giving us an opportunity to make a resource website as an addition to this, as iPhone users will be able to read through this website in order to determine if their iPhone is secure or not and take the necessary actions to secure it iPhone forensics investigators will be able to find a resource base for iPhone Forensics information, as well as learning about the tool which we will finally propose in our next section.

VI. PROPOSED SOLUTION

Having completed the Online Backup Tool building blocks and also it’s supporting website, we are now able to move onto this final stage. Our proposed solution, and also how this project led us to it can be seen in the Figure 3 below.
This solution addresses the problem we have identified earlier on in this paper and uses our implementation to prove a way in which we can achieve it. In this solution we can extract the data, display it in easy to read formats and also formats which are of more use to a forensics investigator. The integrity of the evidence can be contained because the backup can be retrieved from an image of an already seized computer, rather than tampering with the iPhone device itself and risking loss of data or contamination of evidence. This solution allows this proposed solution to follow the ACPO guide which we have already mentioned in this paper, although further work will need to be continued before we can move onto this stage.

We can now look at the features which make our solution as it is. As a start, we will not be able to create the forensics suite in PHP, we have therefore chosen Objective-C as the language as it is the language which is used by the Apple devices anyway and will enable us to interact with the SQLite databases easier.

‘Investigation Case Details’: This is a component which has been implemented for the help of the forensic investigators. While this applications is running, this component will always be visible, which allows the examiner to check information which is available about the case and maybe required when reporting data and also keeping track of which case is being viewed. Options available to the examiner include ‘edit’ using which they can change investigation case details, ‘save case’ in which they can save the progress they have made and ‘Visit Log’ which will display a log which is automatically created by the application, although an examiner can add to it themselves.

Main Window: The investigator will be able to choose which device they want to investigate from a list here, this can include a physical drive which is connected to a computer, however an image which has been created of a suspects computer can also be selected, this is the method which is recommended.

Device information window: Once a device is selected this feature will be shown and is useful to an examiner at this
stage as they can learn the details of all the devices, and start with the one which they deem to be more useful, in this case the iPhone would contain more evidence for an examiner to find for example so they can start on this one.

When investigating a feature on the device, the examiner will be able to select from a number of different data types to retrieve. When this data type is selected, the data will be shown in a timeline, which will allow the investigator to see exactly what happened and when. An overview page will also be visible which will allow the investigator to see all the data that can be found on the iPhone in the order of the events that occurred.

Other Components: We have decided this tool must be able to interact with the various operating systems which are available on the iPhone, and new releases must be thought of instantly when a new update is brought out. This is an improvement on one of the tools we review in this paper, and we believe that it is important and professional to ensure the product is up to date for continued support.

Summary: We think this forensics suite is the right way to complete iPhone investigations using the backup of the iPhone. It will not damage the evidence which ensures the integrity remains intact for a start. The forensics suite itself allows the investigator to concentrate on what is needed in the investigation, which is the analysis of the data in front of them. We have completed everything we could which would help to provide an easy to use and simple environment. The tool will be updated when new features are needed, and the proposal we have created allows for this to happen. Any new components can simply be added on when they are needed, and this will not mean a new installation of the entire suite, only a plug-in or an update will need to be installed which will also ensure it does not affect any cases which are running on the software already. This is what makes out tool much better than other that already exist, the functionality of the forensics suite we are proposing is far more superior and advanced than other tools, and will move forward with the iPhone technology rather than fading into the past and being taken over by other forensics tools.

VII. Evaluation

We aimed to create a solution for how we could we access and extract data from the backup of an iPhone and display it into an easy to read and use format for a forensics investigation whilst ensuring we keep the integrity of the data which may be used in a forensics investigation. Our design was to create a proposal that would meet this aim. Based on the implementation we have completed in this paper and looking at what we have learnt form this and our final proposal for the iForensics Suite, we have met this aim and have developed a solution for the problem we identified earlier, using work which we implemented.

If we take a closer look at the Online Backup Tool, we can see from our problem analysis that it has met all of it requirements. We made the decision to output the results into a .csv file, which we were unsure of at the time because this is not directly editable and some users may not understand how they work, however we decided this was the best output as we can import it into many different formats from the csv file which makes it far more flexible to use and also allows the investigator to open it on different operating systems. This is apparent as we have used Macs to create and implement the majority of this research paper implementation and a csv file will open in Microsoft Excel the same as it would in Mac’s version, Numbers. This is a strong point of this research as it allows flexibility for the investigator in terms of how they want to look at their evidence. We limited the ability to search the exported data to using the search functions in Excel or Numbers or any other software the csv file is opened with. This we thought was weak at first because the file would have to be downloaded first and the html data view could not be used, however we soon realised there are the search functions on the webpage anyway. Also the addition of other search functions in the html page in the PHP script may slow down the processes on the website, which we do not want to happen as we want this Online Backup Tool to be as efficient and easy to use as possible, the way we have done this meets this aim and there are still options for the user to search the exported file so we are very happy with this.

We will now analyse our work in greater depth, We could have made a proposal for the final iForensics Suite without having to implement the Online Backup Tool and its supporting website. In hindsight this would have been possible. However, we couldn't have done this without the knowledge that using the SQLITE databases in the way we did worked. It would have been impractical if we had submitted a proposal for an application which wouldn't have worked in the first place and therefore it is this knowledge that allowed us to continue with the research proposal to a successful result. Creating the Online Backup Tool has also been productive in this research as it has allowed us to create another application on top of the iforensics Suite that can be used by anyone, and also without the need of downloading it. Another aspect of the research we would of changed was the interaction with the SQLlite databases via the Terminal application on the Mac Operating System. Leaving this section out would have saved us time, although if we had not have completed this step first we would not have known how we could interact with the SQLlite databases at all and we would have struggled with the implementation in PHP. So really, this step was also a necessary one in this research to undertake. The proposal of the forensics suite has provided us with an underpinning for a proposal of a much greater scale. In the start of this project we talked about the general smart phone population before we focused on the iPhone. This iPhone forensics suite is based on underpinning, so new features such as new iPhone updates to new phones such as the Blackberry and the HTC can be introduced into the tool as a new module. The complete code will not have to change, only the code in the module added in. Unlike other tools which are a very specific to one make or model of mobile phone, or in extracting one particular piece of data, this forensics suite gives this tool the room to continually improve. This is what makes this tool an excellent addition to any digital forensic examiners tool kit, as rather than going
out of date like other tools we have looked at in this research, this tool will remain as a necessity to the examiners and their everyday work while it’s technology and mobile operating support continues to increase.

VIII. CONCLUSION

This research paper was set to determine the iPhone forensics and security risks and implications, let's remind ourselves of what has been achieved. After identifying a problem of how we could access and extract data from the backup of an iPhone and display it into an easy to read and use format for a forensics investigation whilst ensuring we kept the integrity of the data which may be used in a forensics investigation, we split it up into smaller tasks. In order to achieve this we decided that the three sub tasks would be to: 1) Create an Online Backup Tool, B) Create a website that could support the online Backup Tool and C) Create a proposal for an iForensics Suite using what we had learnt in order to create the Online Backup Tool. The iPhone Forensics Suite proposal was to be our final solution for the problem we have identified.

These findings are extremely significant in terms of the current climate. iPhones bring not only new technology but also the risk of contaminating data on the device if it is investigated physically. It is because of this we have decided the best way to complete an investigation of the iPhone is on the backup it has created on its paired computer.

In our investigation, we have discovered the implication that not only can the technology we have developed for our proposal works on the iPhone, but it can also be applied to the vast majority of smart phones which are today using SQLite databases to store their information. This opens the doors in terms of the audience and provision of this proposal. The findings we have discovered in this research are an excellent foundation for the continued creation of this proposed forensics suite. The Online Backup Extractor Tool plus the forensics suite proposal we have implemented make this research a fully worthwhile investigation, which has now answered a number of questions in order for this research to continue.

This research paper has created a number of areas of future research that could be completed. Although we have proposed an solution, we need to validate this process as fully solving our problem, and also a process that will follow the ACPO guidelines and be accepted as a law enforcement means of retrieving data from the iPhone, and other smartphones alike.

VIII. FURTHER WORKS

We are aware of the amount of research which will be needed to follow this paper up in order to validate this solution as an iPhone Forensics process. We believe the best way we can start to do this is to start conducting tests on certain areas of the solution in order to fully identify it’s own success. In future stages of this research, it has been proposed that we can run tests on the solution we have proposed, to other forensic acquisition methods such as the Jailbreak method. By completing these tests we can determine to what extent the integrity of the data is held and also test it on other aspects of the problem solution.

REFERENCES
