MASTER

Security analysis of SMS and related technologies

Chaudhari, A.S.

Award date:
2015

Disclaimer
This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

Technische Universiteit
Eindhoven
University of Technology

Department of Mathematics and Computer Science

Security Analysis of SMS and Related Technologies

Master's Thesis

Aniket.S.Chaudhari

Supervisor:
Dr. ir. L.A.M. (Berry) Schoenmakers

Eindhoven, November 2015

Abstract

This thesis analyzes the security of Short Message Service (SMS) which is a permanent service on mobile networks. Mobile networks have evolved from GSM Technology for more than 20 years. Security is a headline issue these days and use of SMS service has become an extension of our lives and plays a paramount role in daily chores since its inception with most immediate and efficient form of communication. Due to the available functionality of the mobile networks, SMS are exposed to different kinds of attacks. SMS is one of the fundamental features of the mobile phone and is considered to be a fascinating area for attackers. For the increasing demand for secure SMS, it is important to perform vulnerability analysis of SMS implementation and finding out additional security vulnerabilities within the network, and smart-phones. With the existence of the mobile phone over the years, SMS has been widely embraced as a standard for quick and easy communication. SMS has proceeded from normal message service to two-factor authentication (2FA) scheme for account login and registering. Ever since the growing mindshare and outsized new security valuations to the users for their accounts, SMS service provides best possible forms such as one-time password (OTP) and mobile-Transaction Authentication Number (mTAN) for 2FA. The most important and challenging part of mobile communication is SMS security as attackers illegally access the sensitive data through messages and sometimes compromising the device. If these themes are not addressed adequately, through security controls and measures, the underlying threats could compromise the confidentiality, integrity and availability of SMS service. A detailed study of the mobile networks, SMS protocol structure, and various attack methods were investigated to understand the different properties of authentication and encryption methods that can be applied to counteract the exploits for the applicability of SMS messages in near future.

Acknowledgement

I am grateful to numerous local and global peers who have contributed towards shaping this thesis. At the outset, I would like to express my sincere thanks to Berry Schoenmakers for his advice during my thesis work. As my supervisor, he has constantly encouraged me to remain focused on achieving my goal. His observations and comments helped me to establish the overall direction of the research and to move forward with some detailed investigations. He has helped me greatly and been a source of knowledge to me. I wish to extend my gratitude to Benne de Weger and Jerry den Hartog for accepting to be in the evaluation committee of my final examination of thesis. My sincere thanks to everyone who has provided me with kind words, a welcome ear, new ideas, useful criticism, or their invaluable time, I am truly indebted. My gratitude also goes out to my younger brother Sanket Chaudhari , my family and friends in Eindhoven and India who helped me all the way during my Master's and during the thesis with any technical problems on the way. I must acknowledge the academic resources that I have got from *Technical Univeristy of Eindhoven.* Last, but not the least, I would like to dedicate this thesis to my family, for their love, patience, and understanding.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

More than six billion people worldwide consider mobile phone as an integral part of our lives. There is increasing demand for usage of mobile phones globally. More than 90 percent of the mobile users do not leave home without their phones. The first Short Message Service (SMS) message was sent over the Vodafone GSM network in the United Kingdom on December 3, 1992, from Neil Papworth of SEMA group using a personal computer to Richard Jarvis of Vodafone using an Orbitel 901 handset[1]. The first commercially sold SMS service was offered to consumers, as a person-to-person text messaging service by Radiolinja in 1993. The SMS service is defined within the GSM digital mobile phone standard. SMS technology enables to send and receive messages on mobile devices and Internet-connected computers. Further improvements in wireless technologies were observed such as Code Division Multiple Access (CDMA) and Time Division Multiple Access (TDMA) which were developed in the United States for SMS service. ETSI initially developed the GSM and SMS standards, and they are laid down in ETSI TS 03.40 [19]. ETSI is an abbreviation of European Telecommunications Standards Institute. Now 3GPP (Third Generation Partnership Project) handles the development and maintenance of these standards.

The idea to develop the SMS service was to optimize the mobile phone system and make use of messaging service that works on signaling system that could control the voice traffic when no signaling voice traffic existed. The unused signaling traffic service with the system was utilized for SMS service to establish the communication between two parties with a low-cost service.

Mobile phones consist of both personal and private data. But short messages, nowadays are used for both personal and business communications that can consider as value added services. Some of the typical examples are listed below:

- Information services such as alerts and notifications to customers by stock brokers, bank, and stock transaction status, credit card transaction messages from banks for high-risk transactions, flight and train confirmation details, hotel reservation details and so on.

- One-time passwords (OTPs) and transaction authentication numbers (TANs) being sent to the customers of banks or organizations via SMS messages for authorizing and confirming high-risk online transactions. The OTP are sent to users to authenticate their account and proceed with the online transfer of funds. Using the mechanism such as OTP and mTAN for the authentication process, SMS service's are being used for mobile commerce applications.

- SMS Provides a smooth interaction with people using text messages through mobile phones. E-Marketers have also started to target their customers to deliver services through text messaging by advertising and providing discount coupon codes, promotional services, etc.

[1] See `http://theweek.com/articles/469869/text-message-turns-20-brief-history-sms`

The increasing use of SMS service in various areas and the growing number of exploits is one of the motivations for collecting, describing and evaluating the details for SMS security. The technical specifications of the network architecture are not made public instead they are preserved to keep it secret. Bearing this in mind, many academic researchers try to find out vulnerabilities and exploits in the system and targets to provide a unique solution. Whereas, hackers perform attacks on a large scale by stealing the confidential and sensitive information of the user.

SMS follows the mechanism of store and forward service which is similar to Simple Mail Transfer Protocol (SMTP) mail service, i.e., SMS are not sent directly from sender to recipient, but always via Short Message Service Centre (SMSC) with a low-cost service. SMSC are used to store messages before they are delivered to the destination mobile user's service provider or another SMSC. Every mobile cellular network that supports SMS service has one or more SMS Centers to handle and manage the short messages. SMS has a unique feature for confirmation of delivery of messages. The method is unlike paging request; users do not directly send an SMS and trust to hope that it gets delivered to the recipient. Instead, the sender of the message can receive a return message notifying whether the SMS has been delivered or not. Additionally the messages also have a maximum validity period if they are not delivered to the recipient, i.e., if the intended SMS recipient is not online, the SMSC will keep the SMS message in its memory until the validity of the message and delete it from storage when the validity is expired.

Short messages can be simultaneously sent and received with GSM voice, data and fax call as they use dedicated radio channel. Whereas, short messages are transmitted over the radio channel using the signaling path and Out-of-band service[2].

1.1 Research Question

The research of this thesis is to focus on a broad range of vulnerabilities related to SMS messages in different networks and further emphasis on the Two-Factor Authentication (2FA) scheme used by users to deliver single-use passphrase (one-time password (OTP) and mobile Transaction Number (mTAN)) which uses SMS service to authenticate the users.

To examine and detect the exploits in the system that uses SMS service for transmission of the message over the air interface between different components in the network. To what extent these methods are still a secure method for 2FA process with a smart-phone.

To answer the above central research questions the following sub-questions are defined to provide a concrete and concise information.

Overall Analysis

1. The current vulnerabilities in the SMS service.
2. The possible exploits on the SMS service in the telecommunication network.
3. Which security mechanisms are required to reduce the attacks on mobile applications to interfere the SMS service for attacking OTP and mTAN codes.
4. Which countermeasures can be introduced to make the systems more secure.

[2] See at http://www.activexperts.com/sms-messaging-server/sms/smsintro/

1.2 Structure of the thesis

The remainder of the thesis is organized as follows.
Chapter 2: Studies and elaborate an overview of GSM Family background which includes GSM, UMTS and LTE network architecture. This chapter also highlights how the technology evolved and developed over the years.
Chapter 3: Discusses the technical description of an SMS short message that is transmitted over the air (OTA) interface and the different protocol layers. It highlights the various transmission scenarios and the protocol hierarchy of SMS.
Chapter 4: Highlights the vulnerabilities in the GSM telecommunication network .
Chapter 5: Describes the attacks performed on SMS service in past and recent years by attackers and academic researchers to experiment and test the reliability of GSM network architecture.
Chapter 6: Explains various mobile Operating Systems security and testing of the third-party mobile applications.
Chapter 7: Explains SMS based Authentication methods such as OTP and mTAN used by online service providers, banking, and mobile applications. It also describes methods that are used to abuse these secure authentication methods.
Chapter 8: Provides a list of potential countermeasures suggested by academic researchers to develop a safe communication process for SMS based authentication schemes. The methods and approaches included in this chapter help to protect the SMS messages by eavesdropping, intercepting, modifying and lastly a general conclusion about the technology relying on SMS communication.

Chapter 2

GSM Family Structure

2.1 Global Service for Mobile Communication

Global System for Mobile Communications (GSM) is the first standard developed for mobile telephony systems in the world. In 1982, the European Conference of Postal and Telecommunications Administrations (CEPT) created the GSM to develop a standard for a mobile telephone system that adopted across Europe. GSM is presently recognized as the global default standard for mobile communications which has spread over 90% worldwide and is currently expanding to more than 200 countries worldwide. GSM established a low-cost implementation of the SMS, also called text messaging, which is supporting to all other mobile phone standards as well.

The connection establishment between Mobile Station (MS) and the GSM network can be observed in Figure 2.1. In the GSM, only the airway traffic between the MS and the Base Transceiver Station (BTS) is encrypted with a weak stream cipher (A5/1 or A5/2). There are three main types of cryptographic algorithms used in GSM. A5 is a stream cipher used for encryption of data and voice, A3 is used as authentication algorithm for authenticating the user and A8 is the session key agreement algorithm[1]. The description of these authentication and encryption algorithms can see seen in Table 4.1. GSM network authenticates the user to the network and not vice versa. Thus, the security model offers confidentiality and authentication; but limited authorization capabilities and no non-repudiation. Although it's not formally announced, its description was found by Briceno, Goldberg, and Wagner[8].

The BTS usually acts as a transmitter and receiver of the radio signals from mobile phones. The MS is also recognized as the cell phone that transmits the signals and data to BTS. The radio signals received from MS are converted to digital format and are forwarded to the Base Station Controller (BSC). BSC controls multiple BTSs within a small geographical area. The BSC then forwards the received signals from BTS to Mobile Switching Centre (MSC). The MSC has Equipment Identity Register (EIR) and Authentication Register (AUC) databases for equipment verification and user authentication. The message is transmitted to SMSC and is stored in the queue and will wait until it gets delivered. SMSC interrogates its databases (Home and Visitor Location Register (HLR and VLR)) for the location information about the destination mobile handset. By finding the location of the target MS, the message is sent to the respective MSC, and the same process follows until the message is delivered to the recipient. Analyzing the case for an SMS, the message contents are still available in the SMSC persistence database even after the SMS is transferred to the user.

In additional situation when an SMS is transmitted from an External Short Message Entity

[1] See at `http://www.iol.ie/~kooltek/a3a8.txt`

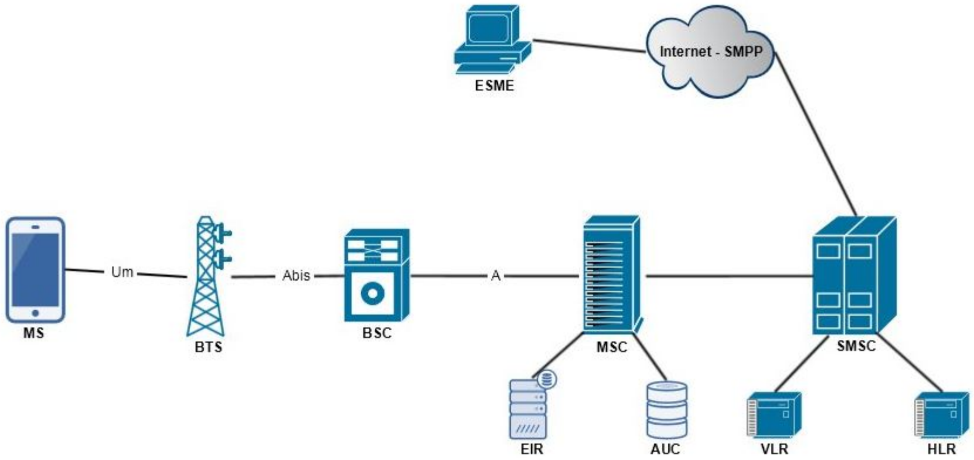


Figure 2.1: Architecture of GSM

(ESME)[2] example from a PC, it is sent via Internet using TCP/IP and Short Message Peer-to-Peer (SMPP) connections, and it makes use of SMS Gateway to transmit the message to SMSC and the further process continues in the same manner as explained above [37, 27]. The work-flow of this method can be observed in Figure 2.2.

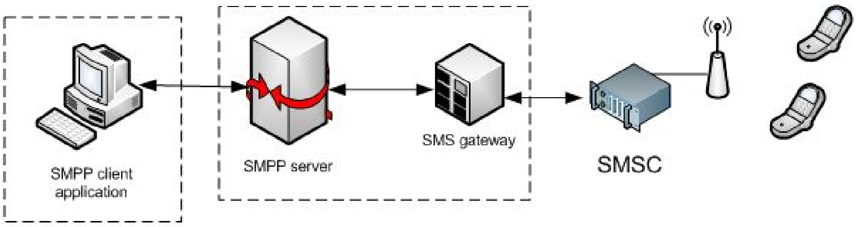


Figure 2.2: SMS send via ESME

Since SMPP is a standard protocol, designed to provide a flexible data communication interface for the transfer of short message data between ESME, SMSC and other routing entities. SMPP relies on TCP/IP connection to communicate with two objects and binds on port 2775. The ESME has to link the SMSC using this port number and a password. To maintain a virtual good quality of service for all different commercial service providers connected to SMSC, it uses asynchronous type of communication.

The SMSC transfers the message to the recipient in a point-to-point format like the serving system. The system uses the paging service to establish the connection with the device, and if it responds, the message is delivered. Upon successful transmission of the message to the end user, SMSC receives an acknowledgment from the user that categorizes as "sent" or "received" and will not attempt to send the same message again.

The different kind of air interfaces involved in the GSM architecture[3] is as follows [22]:

- The **Um** interface is the GSM specific air interface between MS and the BTS. It bares this name because it is a mobile analog to the U interface of Integrated Services Digital Network

[2] See at http://www.engineersgarage.com/mygarage/how-websites-send-sms-to-mobile-phones

[3] See at https://www.informit.com/library/content.aspx?b=Signaling_System_No_7&seqNum=111

(ISDN). Um also supports GPRS packet-oriented data. This is documented in GSM 04.01 (04.xx and 05.xx), Section 7 specifications. The Um interface are defined in the lower three layers of the GSM: The Physical Layer (L1), The Data Link Layer (L2) and The Network Layer (L3).

- The **Abis** interface in GSM handles transmitting and signaling information between BTS and BSC and is the first actual physical connection. The protocol used for sending signaling information on the Abis interface is Link Access Protocol on the D Channel (LAPD).
- The **A** interface in GSM handles transmitting signaling information between BSC and MSC. It provides the information about signaling and traffic between MSC and BSC.

2.2 Universal Mobile Telecommunication System

UMTS has evolved from General Packet Radio Service (GPRS) by replacing the radio access network. The architecture for UMTS is similar to that of GSM, but the names of the network entities have revised. A significant change can be witnessed is that GSM was circuit-switched, and UMTS is packet-switched. 3GPP is now responsible for the development and maintenance of UMTS service. This specification emerges from GSM. The architecture of UMTS compared with GSM can be noticed in Figure 2.3

The Serving GPRS Support Node (SGSN) was developed and introduced for GPRS; and its still used in UMTS network architecture which provides several functions such as:

1. Mobility Management: The UMTS network is a packet switched, the User Equipment (UE) connects to this packet switched domain and SGSN generates the MM information based on the mobile's current location. This functionality of the element is further explained in Section 3.2.3.
2. Session Mangement: The SGSN manages the session data with the required quality of service which is named as PDP (Packet Data Protocol) through which the data is transmitted to different layers.
3. Interaction with other area networks: SGSN manages its entities to communicate within the different network such as MSC and circuit switched network areas.

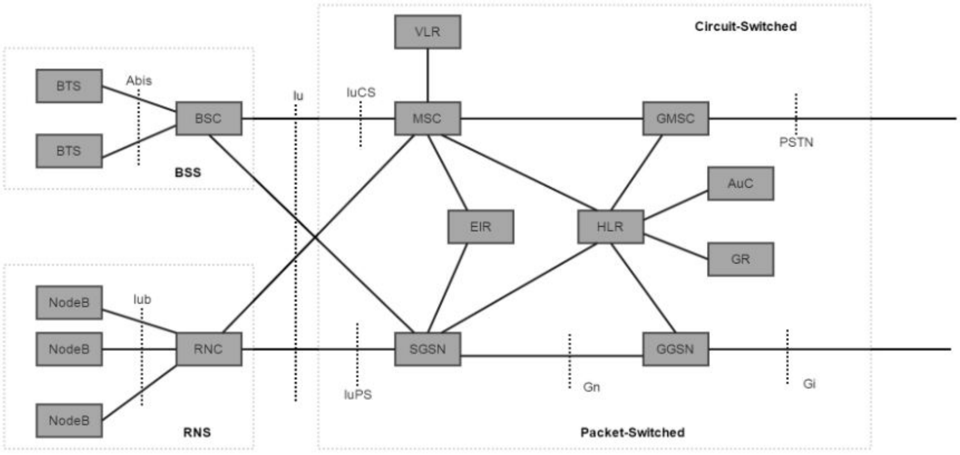


Figure 2.3: UMTS architecture

The changes in the UMTS architecture[4] are described in the Table 2.1 [41, 1]:

[4] See at http://www.rfwireless-world.com/Tutorials/UMTS-Network-Architecture.html

Table 2.1: UMTS architecture evolved from GSM

GSM Termiologies	**UMTS Terminologies**.
Mobile Station (MS)	User Equipment (UE)
Base Transceiver Station (BTS)	NodeB
Base Station Controller (BSC)	Radio Network Controller (RNC)
Mobile Switching Centre (MSC)	Serving GPRS Support Node (SGSN)
Gateway MSC (GMSC)	Gateway GPRS Support Node (GGSN)

The air interface involved in the UMTS architecture has also been replaced when compared with GSM architecture with improvements that are explained below:

- The UE connects with NodeBs through the radio interface **Uu** based on the Wideband CDMA (WCDMA) technology.

- In UMTS every NodeB is connected to RNC through **Iub** interface.

- Every RNC is connected to Serving GPRS Support Node (SGSN) through the **IuPS** interface.

- The UMTS Terrestrial Radio Access Network (UTRAN) consists of NodeB and Radio Network Controller (RNC) connected to an Asynchronous Transfer Mode (ATM) network.

- The RNC and NodeB serve as MS so called as Serving Radio Network System (SRNS).

GGSN was also introduced in GPRS and still used in UMTS network. The GGSN works as a central processing elements within UMTS which handle the inter-working state between the UMTS-packet switched network and external packet switched network. The functionality of the GGSN can be compared to a router. It receives the data addressed to a particular user, it then checks if the user is active and forwards the data to the serving SGSN serving the specific UE.

2.3 Long Term Evolution

UMTS further developed to advanced technology to form Long Term Evolution (LTE - 4G). It is further introduced as Evolved Universal Terrestrial Radio Access (E-UTRA). E-UTRAN is the combination of E-UTRA, UEs, and eNodeBs. It is also phrased as the next generation of cellular wireless technology beyond 3G. The architecture of LTE is similar to UMTS but with a few changes[5]. The architecture of LTE evolved from UMTS can be seen in Figure 2.4

The changes in the LTE architecture are described below [12, 14]:

Table 2.2: LTE architecture evolved from UMTS

UMTS Terminologies	**LTE Terminologies**.
User Equipment (UE)	User Equipment (UE)
NodeB	eNodeB (evolved NodeB)

The other features of the LTE network architecture compared with GSM and UMTS are explained below:

- LTE-Uu interface is the air interface between UE and eNodeB.

[5] See at `http://www.rcrwireless.com/20140513/network-infrastructure/lte/lte-network-architecture-diagram`

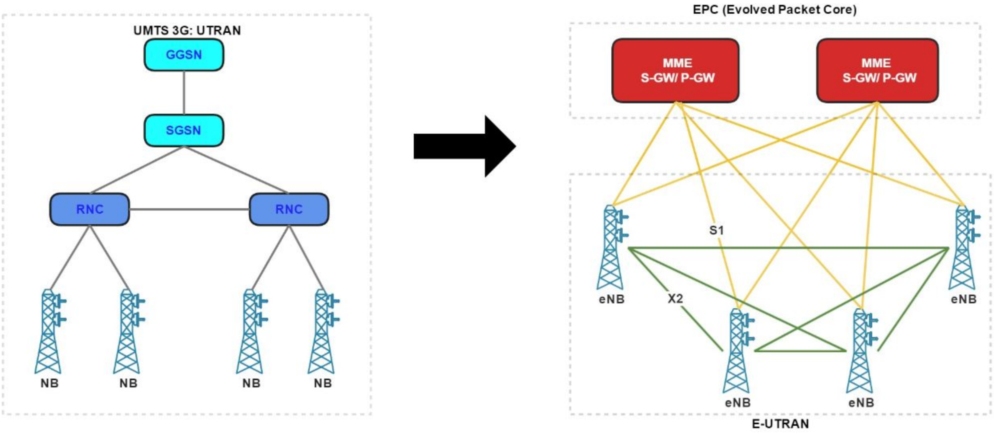


Figure 2.4: LTE evolved from UMTS

- eNodeB is handled as a Base Station, which manages radio resources and mobility. The **X2** interface is the air interface between the eNodeBs and **S1** interface between Mobility Management Entity (MME) and eNodeB. It also provides radio resource management functions. MME manages mobility and provides security. It can operate in the control plane and provides authentication.
- Serving Gateway (S-GW) also provides mobility and is typically responsible for routing and forwarding the voice signals and data messages.
- LTE has a flat architecture, so it does not contain RNC.

The telecom industry is acquiring more advanced technology and are working to implement the 5G that is titled as 5th Generation Mobile Network/ Wireless Systems [25]. The research of this technology is still in progress, and it provides more advanced wireless connections than 4G can offer[6].

[6] See at http://www.techrepublic.com/article/does-the-world-really-need-5g/

Chapter 3

SMS Technical Specifications

Short Message Service (SMS) is a connectionless service that allows the exchange of messages from sender to receiver. A single SMS message contains at most 140 bytes (1120 bits) of data, so one SMS message can contain up to 160 characters if 7-bit character encoding is used. (7-bit character encoding is suitable for encoding Latin characters like English alphabets). The basic structure of a message can be observed in Figure 3.1. The structure of the SMS was developed as a part of GSM according to ETSI technical specifications. But now it is being carried out in the scope of 3GPP activities. The messages can be sent from mobile devices using the network (GSM/UMTS/LTE) and also from ESME such as Internet hosts, telex, etc. explained in Figure 2.2. Messages can travel in both directions which communicates on the signaling links and allows messages to be received while the user is calling. Since the payload of a single message is limited; multiple text messages can be sent to the user, and the built-in software will accordingly concatenate it in most of the smart-phones. SMS uses OOB channel (out-of-band) for the data transferred through this stream that is independent of the primary in-band data stream (voice calls). It provides an independent channel, which allows data to send through this mechanism such as MMS (Multi-Media Message Service) and EMS (Enhanced Message Service) to be kept separate from in-band data. Since this thesis is focusing mostly on SMS security and 2FA methods that comprise of OTP and mTAN send through SMS, we will describe the details involved in the messaging system.

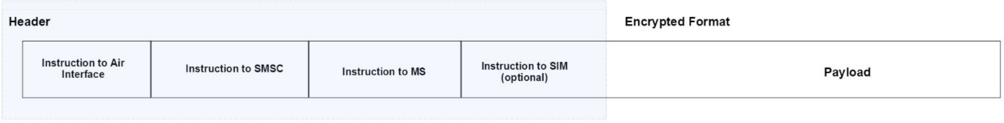


Figure 3.1: SMS Packet Structure

3.1 Flow of the Message

We will consider a standard message delivery i.e. point to point messaging. The message containing the data has to pass through various network entities before it is delivered to the respective user. The Figure 3.2 will give a brief overview of the message transfer path.

1. The message is sent from MS or ESME of the sender, such as mobile phone or computer to SMSC through MSC. The MSC is often referred to Gateway MSC (GMSC). The delivery of the messages is always done through BTS. The BTS delivers the message through wireless (air interface), from BTS to MS.

2. The MSC consist of EIR and AUC, which handles mobile authentication devices and validates that the messaging is originating from a valid device. MSC is connected to SMSC, where

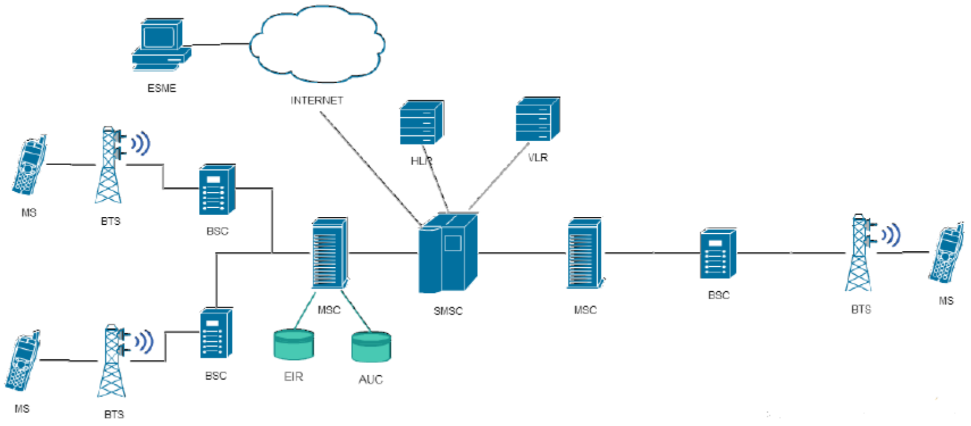


Figure 3.2: SMS Workflow Achitecture

the messages are stored in the queue, and waits to get delivered to the respective user.

3. The SMSC is the telecommunication core network (GSM/UMTS/LTE). SMSC determines the presence of the receiver of the message to be delivered by querying to HLR/ VLR. Once the location of the user is discovered the message is forwarded to the MSC through roaming protocol.

4. The role of the MSC is to determine the location of the user to deliver the message by querying the VLR and sends the message to BSC. The BSC then delivers the message to the MS through BTS. When the user receives the message on his mobile phone, it is usually stored on the device. The storing of the messages is performed by SIM card (Subscriber Identity Module) of the MS.

3.2 The Protocol Hierarchy

During the transmission of several messages, different protocol layers interact with each other. The various protocol layers of the hierarchy involved in the transmission of the message with the Data Units are explained in this section.

3.2.1 The Short Message Transfer Layer

The Short Message Transfer Layer (SM-TL) is used to carry the messages that originate at the application layer with the delivery reports. The SM-TL of the MS and SMSC communicate with each other using Short Message Service Transfer Protocol (SMS-TP). This protocol consists of four types of Transfer Protocol Data Units (TPDU). These Data Units are also known as the fundamental features in the SMS Technology, and every message is supported by these characteristics [30]

SMS-SUBMISSION REPORT

When a user sends the message from his mobile phone, it is transmitted to the SMS Center. After reaching the message to SMSC, it will send back a message submission report to the sender's mobile phone to acknowledge if there were no errors or failures (e.g. incorrect SMS message format, busy SMS center, etc.). If no error or failure is detected, the SMSC will send back a positive submission report to the mobile phone else it sends a negative submission report. In rare scenarios when the mobile phone do not receive the message submission report after a period, it concludes that the message submission report has lost. The user may re-send the same SMS message. Observing at the same SMS message content by the SMSC, a flag will be assigned to the same message as the same message was sent before. The SMSC will ignore the message and sends back a message submission report to the mobile phone. This method helps to prevent sending the same message to the recipient multiple times. There are two types of submission reports namely Positive and Negative Submission Report, which can be noticed in Table 3.2.

1. The parameters included in Positive Submission Report are [30]:
   - Message type (SMS-SUBMIT-REPORT)
   - Parameter indicator (presence of protocol identifier, data coding scheme, user data length)
   - Protocol identifier
   - Data coding scheme
   - Service center time stamp (time at which the SMS center received the associated message)
   - User data header
   - User data (with associated length)
2. The parameters included in Negative Submission Report are [30]:
   - Message type (SMS-SUBMIT-REPORT)
   - Parameter indicator (presence of protocol identifier, data coding scheme, user data length)
   - Protocol identifier
   - Data coding scheme
   - Failure cause
   - Service center time stamp (time at which the SMS center received the associated message)
   - User data header
   - User data (with associated length)

SMS-DELIVERY REPORT

In this situation, the message is delivered to the recipient's mobile phone from SMSC. Same as SMS-SUBMIT, the recipient mobile phone sends a delivery report to the SMSC positive or negative depending on the successful message delivery or failed message delivery during the transmission process.

Depending upon the request of the sender, the SMSC will send a status report when it receives the message delivery report from the recipient. Sometimes the message delivery report mechanism is not used, and the acknowledgment of message delivery is done in the lower layers. The parameters included in SMS-DELIVER are mentioned below, and the description of these parameters can be noticed in Table 3.3 [30]:

- Message type (SMS-DELIVER)
- Indication that there are more messages to be received
- Request for reply path

- Request for a status report
- Address of the originator SME
- Protocol identifier
- Data coding scheme
- Service center time stamp (time at which the SMSC received the message)
- User data header
- User data(associated with length)

SMS-STATUS-REPORT

This feature is utilized when the sender has requested to know whether an SMS message has reached the recipient mobile phone successfully. To obtain the information, the sender is required to set a flag in the SMS message to notify the SMS center to deliver the status report about the delivery of the message. The functioning and description of this parameter can be noticed in Table 3.2, and the following values can be assigned to this 1-bit parameter:

- Value 0: no status report requested.
- Value 1: a status report is requested.

SMS-COMMAND

This feature is used during the transmission process to execute the commands from MS to SMSC and vice versa. The different types of commands used are setting new messages remind, deletion of a message, message storage settings, read a message, message module mode and send a message. The execution of the command is usually requested by an application server and not by the MS. A brief overview of command identifiers is listed in Table 3.1

Table 3.1: Command Identifiers adapted from [30]

Command Identifier (hex)	**Description**.
0x00	Enquiry relating to a previously submitted message. With this command, a request is made for the generation of a status report.
0x01	Cancel status report request relating to a previously submitted message. No status report is to be generated for this command execution.
0x02	Delete a previously submitted message. No status report is to be generated for this command execution.
0x03	Enable status report request relating to a previously submitted message. No status report is to be generated for this command execution.
0xE0....0xFF	Values specific for each SMSC.

3.2.2 The Short Message Relay Layer

The Short Message Relay Layer (SM-RL) provides the connection between the Short Message Transport Layer and the Link Layer to transfer the short message traffic. The messages within the Relay Layer communicating from MS and SMSC, and vice versa is identified as Short Message Identifier (SMI) that is generated once when a message is registered with the layer. The SMI is generated at every layer and is not always the same, so a mapping is done at every layer between the SMIs [20]. The functions of the Relay Layer are as follows [20]:

- Accepting Transport Layer messages and delivering them to the next indicated relay point or end point.
- Providing error indications to the Transport Layer when messages cannot be delivered to the next relay point or end point.
- Receiving messages and forwarding them to the Transport Layer.
- Interfacing to and controlling the Link Layer used for message relay.
- Formatting messages according to the SMS standards and/ or other message standards, as required by the Link Layer and/ or peer SMS layers.

The SM-RL comprises of six types of protocol elements. But we will discuss only four protocols that are most important [20]:

RP-DATA

This protocol consists of RP-MO-DATA and RP-MT-DATA (Mobile Originated and Mobile Terminated) which is used to transfer a TPDU from MS to SMS center. The elements included in this protocol are the originating address of the SMS, the destination address of the SMSC and the User Data Parameter containing the TPDU.

RP-ACK

This protocol includes the RP-DATA acknowledgments with SMS-SUBMIT-REPORT TPDU and SMS-DELIVER-REPORT TPDU, which is a part of positive acknowledgments to SMS-SUBMIT or SMS-COMMAND and SMS-DELIVER or SMS-STATUS-REPORT respectively.

RP-ERROR

This protocol contains the RP-DATA acknowledgments with SMS-SUBMIT-REPORT TPDU and SMS-DELIVER-REPORT TPDU, which is a part of negative responses to SMS-SUBMIT or SMS-COMMAND and SMS-DELIVER or SMS-STATUS-REPORT respectively.

RP-SM-MEMORY-AVAILABLE

This protocol contains the information i.e. International Mobile Subscriber Identity (IMSI) of MS to the network to receive one or more messages.

3.2.3 The Connection Management Sub-Layer

This Connection Management Sublayer (CM-Sub) protocol provides services to the upper layers can be observed in Figure 3.3. The MS, which contains Short Message Control (SMC), communicates with MSC using Short Message Control Protocol (SM-CP). The MS usually contains two SMC entities, one for Mobile Originated (MO) short message service, and the other for Mobile Terminated (MT) short message service. The SMC entity cannot simultaneously perform messaging in the same direction. Before the message sending procedure starts an MM-connection (Mobility Management) must be established between MS and MSC. The SM-CP comprises of following protocol elements [2].

CP-DATA

This protocol is invoked by SM-CP service primitives MNSMS-ESTablish or MNSMS-DATA to establish an MM-connection and transfer RPDU on the established connection between MS and MSC.

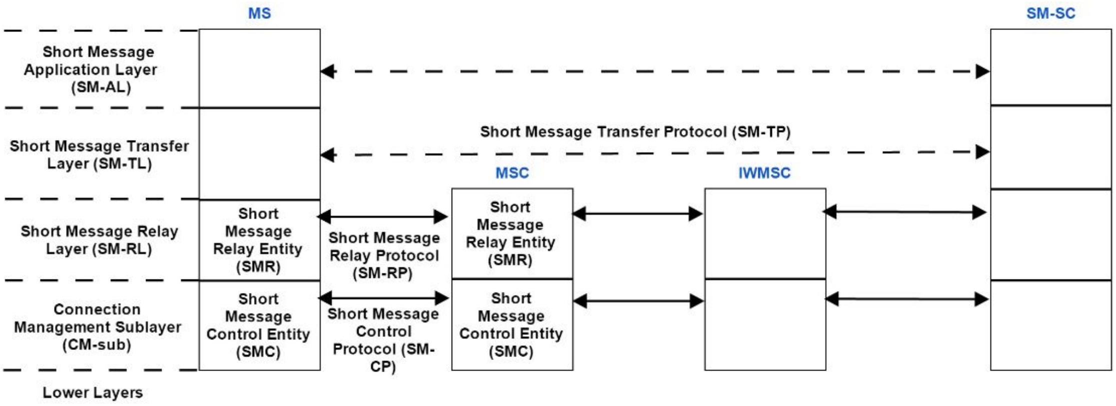

MS
SM-SC
Short Message Application Layer (SM-AL)
Short Message Transfer Layer (SM-TL)
Short Message Transfer Protocol (SM-TP)
MSC
IWMSC
Short Message Relay Layer (SM-RL)
Short Message Relay Entity (SMR)
Short Message Relay Protocol (SM-RP)
Short Message Relay Entity (SMR)
Connection Management Sublayer (CM-sub)
Short Message Control Entity (SMC)
Short Message Control Protocol (SM-CP)
Short Message Control Entity (SMC)
Lower Layers

Figure 3.3: SMS protocol hierarchy in GSM

CP-ACK

This protocol acknowledges the corresponding CP-DATA. It does not contain any specific information element. When the short message delivery is done, the MM-connection is released by SMC using the MNSMS-RELease service primitive.

CP-ERROR

This protocol is invoked by SM-CP service primitives MNSMS-ABORT or MNSMS-ERROR, providing the details for error occurred during the transmission of the CP-DATA protocol.

3.3 Short Message Service Packet Format

3.3.1 SMS-SUBMIT

The structure of an SMS message packet when a user sends (submits) a short message from his MS i.e. Mobile Originated can be observed in Figure 3.4, and the description of each parameter is explained in Table 3.2 [46]. Usually, the SMS messages are sent via AT-command interface explained in Section 3.5. There are two ways to submit a message on GSM network. First, the messages are sent in normal text mode (supported by some phones) and second, in Protocol Data Unit (PDU) format. A message submitted in text will look like:

```
at+cmgf=1
OK
at+cmgs="+41278965344"
>this is an sms in text mode
+CMGS: 359
OK
```

The AT-command sets the text mode to submit the message, and then the destination number with the text message that is to be send is entered. This method works only to send regular text messages. The SMS PDU format works in a similar way with a difference that the messages are not sent in text form. Instead, they are sent in bits.

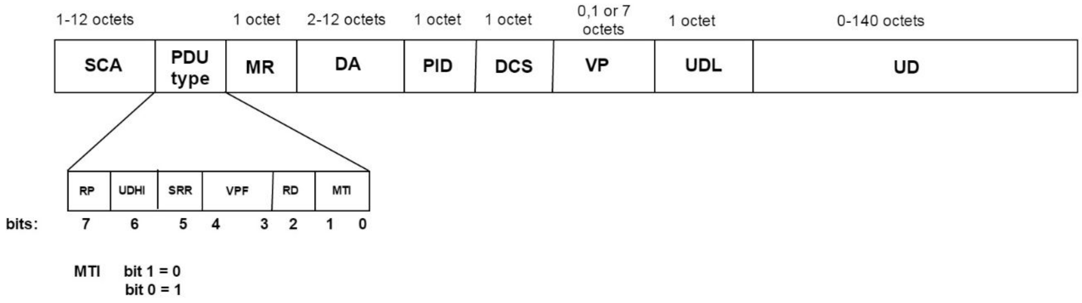


Figure 3.4: SMS-SUBMIT

Table 3.2: SMS-SUBMIT (MO)

Parameter	**Full-form**	**Description.**
SCA	Service Centre Address (Information Element)	Telephone number of the service centre
PDU type	Protocol Data Unit type	
MR	Message Reference	Successive number (0..255) of all SMS-SUBMIT Frames set by the M20
DA	Destination Address	Address of the destination SME
PID	Protocol Identifier	Parameter showing the SMSC how to process the SM (as FAX, Voice etc)
DCS	Data Coding Scheme	Parameter identifying the coding scheme within the User Data (UD)
VP	Validity Period	Parameter identifying the time from where the message is no longer valid in the SMSC
UDL	User Data Length	Parameter indicating the length of the UD-field
UD	User Data	Data of the SM
RP	Reply Path	Parameter indicating that Reply Path exists
UDHI	User Data Header Indicator	Parameter indicating that the UD field contains a header
SRR	Status Report Request	Parameter indicating if the MS has requested a status report
VPF	Validity Period Format	Parameter indicating whether or not the VP field is present
RD	Reject Duplicate	
MTI	Message Type Indicator	Parameter describing the message type 00 means SMS-DELIVER 01 means SMS-SUBMIT

3.3.2 SMS-DELIVER

The structure of an SMS packet when a user receives (deliver) a short message on his mobile station i.e. Mobile Terminated can be observed in Figure 3.3 and the corresponding parameter details in Table 3.3 [46]

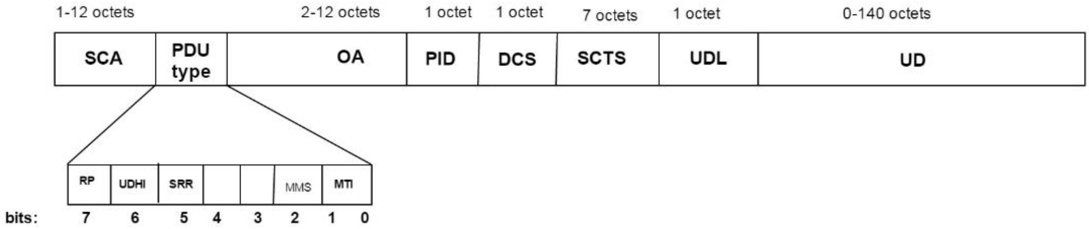


Figure 3.5: SMS-DELIVER

Table 3.3: SMS-DELIVER (MT)

Parameter	**Full-form**	**Description**.
SCTS	Service Centre Time Stamp	Parameter identifying time when the SMSC received the message
OA	Originator Address	Address of the originating MS
MMS	More Messages to Send	Parameter indicating whether or not there are more messages to send

3.4 Transmission of a Message

Since this thesis focuses on the SMS security, we will observe various scenarios of transmission of a short message over the telecommunication network. The scenario consists of two parts firstly when the message is sent from the MS of the sender to the SMSC and secondly when the ESME delivers the message to the MS of the receiver. The first case is also known as Mobile Originated Messaging and the second case is known as Mobile Terminated Messaging. The transmissions of the message in these scenarios are described below.

3.4.1 Mobile Originated Messaging

As stated above, Mobile Originated Short Message (MO-SM) refers to the transmission of the message from an MS to the SMSC. The successful transmission of an SMS can be seen in Figure 3.6. The following are the steps involved in the transmission of a short message[1].

1. The BTS authenticates MS and gets registered with the network.

2. The MS sends the SMS to the MSC.

3. The MSC verifies with the VLR that the message is arriving from a valid MS and does not violate the services and have a valid destination address.

4. The MSC forwards the SMS to SMSC using the *forwardShortMessage* operation.

5. The SMSC delivers the SMS to the terminating MS by querying the VLR (and optionally receives acknowledgement).

6. The SMSC sends the delivery report to the MSC on the successful delivery of the *forwardShortMessage* operation.

7. The MSC sends the status report to the Originating MS upon the successful delivery of the SMS to terminating MS.

[1] See at `http://wiki.yatebts.com/index.php/GSM_Concepts`

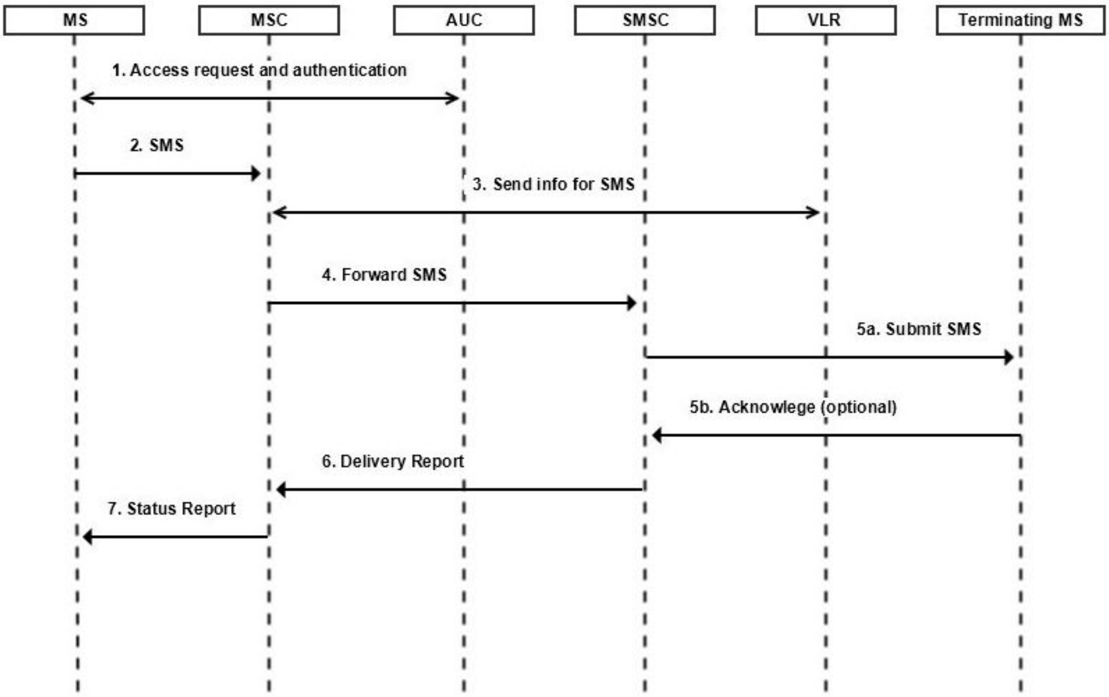


Figure 3.6: MO-SM

3.4.2 Mobile Terminated Messaging

Mobile Terminated Short Message (MT-SM) refers to the transmission of the message from an ESME such as the bank or commercial service provider to the terminating MS of the receiver. The successful transfer of message for MT-SM[2] can be observed in Figure 3.7.

1. Bank or commercial service provider application server sends the SMS upon request to its SMSC through SMS Gateway and using SMPP protocols.

2. SMSC will send the routing information for the SMS to HLR.

3. HLR locates the nearest MSC and discovers the information of terminating MS (active or inactive) for delivery of the SMS. SMSC will forward the SMS to that MSC.

4. With the help of AUC, MSC sends the SMS to nearest BSC. BTS will authenticate the MS through paging request (over the air transmission)for the delivery of the SMS and the authentication procedure starts.

5. AUC informs the MSC about the authentication of the MS.

6. If the authentication process is successfully performed then the MSC will forward the SMS to the MS.

7. MSC will send a response to SMSC for the successful delivery of SMS.

8. The SMSC will inform the bank or commercial service provider application server about the successful delivery of the SMS through MT-SM operation.

[2] See at `http://wiki.yatebts.com/index.php/GSM_Concepts`

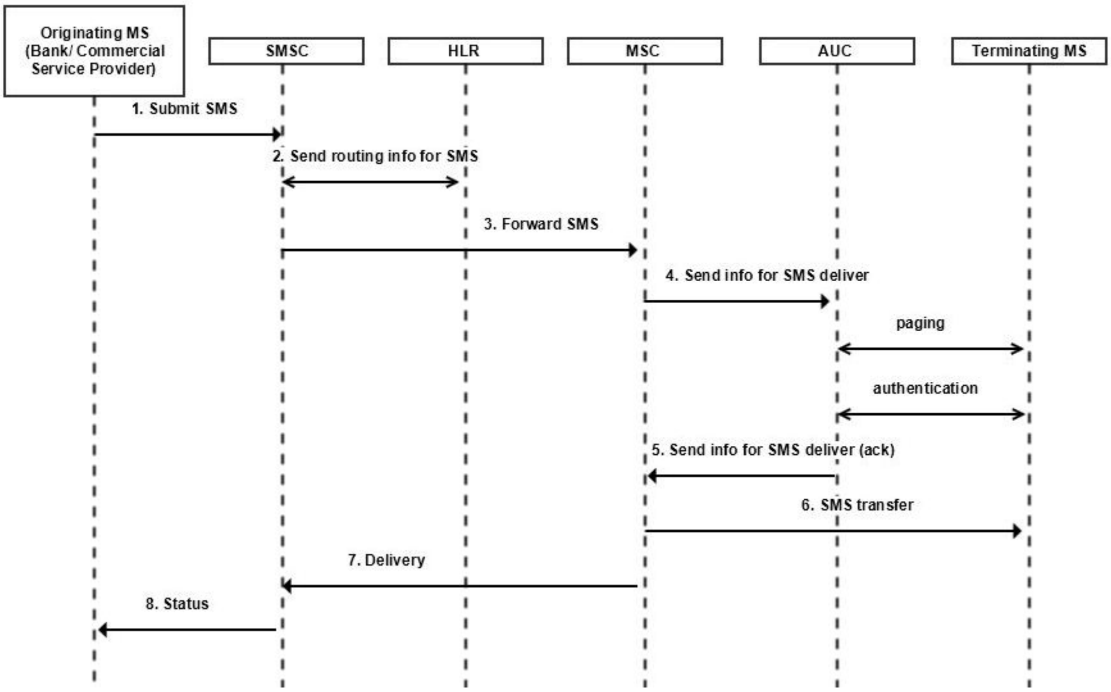

Originating MS (Bank/ Commercial Service Provider)
SMSC
HLR
MSC
AUC
Terminating MS
1. Submit SMS
2. Send routing info for SMS
3. Forward SMS
4. Send info for SMS deliver
paging
authentication
5. Send info for SMS deliver (ack)
6. SMS transfer
7. Delivery
8. Status

Figure 3.7: MT-SM

3.5 SMS PDU Format

SMS message technical specifications are specified by ETSI (GSM 03.40 and 03.38 standards). PDU (Protocol Description Unit) Format is used to send the message in binary mode either 7 bit or 8-bit format. It is usually used to send the data in a compressed format. For transmitting of an OTP or mTAN code through SMS service, it makes use of 7-bit text format. The codes are in the numeric format, and it is important to study the PDU mode as there are only a few commands used to set the numeric format that will change the setting or have appropriate application in the smart-phone operating system (OS) to convert the SMS message from PDU digital format to text format. Every smart-phone has a different type of text mode with an encoding bit scheme that represents the PDU size. These methods are implemented at OS level of the smart-phone and are called as AT commands. AT is the abbreviation for "*Attention*" commands to perform specific tasks such as read, send, check signal strength, etc. The AT commands that are used in SMS message service are listed in Table 3.4.

The PDU format message is composed of several elements regarding the message and also details about the user. Some of the elements are as follow [21]:

- Length of SMSC.
- Service Centre Timestamp.
- Originator Address: the phone number of the sender.
- Protocol Identifier.
- Data Coding Scheme.
- User Data Length: tells how long is the message.

Table 3.4: AT Command Description

AT Command	Description
AT+CMGF=1	This command puts the device into text mode so that the messages can be sent and received
AT+CSMS=0	Checks if the device supports SMS commands
AT+CMGS=<number><CR><message>	Sent a text message
AT+CMGR =<index>	Read SMS message at index number
AT+CMGL="ALL"	List all text messages that are on the device (or network)
AT+CIMI	Get the IMSI number of the module
AT+CMGD=2 <ENTER>	The device will delete a single message
AT+CPMS=<read>[,<send>,<receive>] <ENTER>	The first parameter sets the storage to read from, the second optional specifies the storage to send messages from and the last optional parameter tells the device where to store newly received messages
AT+CSCA	This command is used to get the service center address

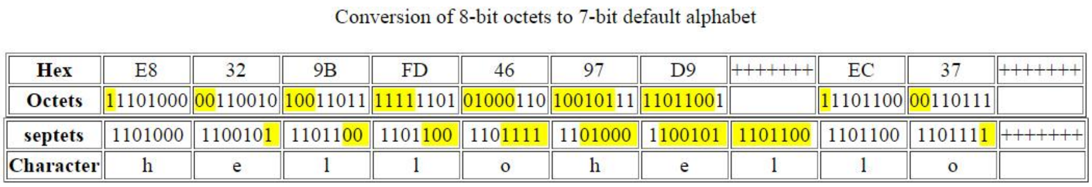
Below sample shows a short message with the detailed explanation of the SMS in PDU format[3].
SMS Text Message: *hellohello*

PDU format: *07911356131313F311000A9260214365870000AA0AE8329BFD4697D9EC37*

Table 3.5: PDU String Explanation

Octet(s)	Description.
07	Length of the SMSC information
91	Type-of-address of the SMSC. (91 means international format of the phone number)
13 56 13 13 13 F3	Service Center (SMSC) number. (indecimal semi-octets) The length of the phone number is odd (11), so a trailing F has been added to form proper octets. In this case "31653131313"
11	First octet of this SMS-SUBMIT message
00	TP-Message-Reference. The "00" value here lets the phone set the message reference number itself
0A	Length of the sender address. (0A hex= 10 dec)
92	Type of address of the sender number
60 21 43 56 87 00	Sender number (0612346578). Note that this has length 10
00	TP-PID. Protocol Identifier
AA	TP-DCS. Data encoding scheme
0A	TP-User-Data-Length. Length of message
E8329BFD4697D9EC37	TP-User-Data. These octets represent the message ***hellohello***

[3] See at http://www.smartposition.nl/resources/sms_pdu.html



Conversion of 8-bit octets to 7-bit default alphabet

Hex	E8	32	9B	FD	46	97	D9	+++++++	EC	37	+++++++
Octets	11101000	00110010	10011011	11111101	01000110	10010111	11011001		11101100	00110111	
septets	1101000	1100101	1101100	1101100	1101111	1101000	1100101	1101100	1101100	1101111	+++++++
Character	h	e	l	l	o	h	e	l	l	o	

Figure 3.8: PDU String Description

3.6 Wireless Delivery of SMS

The air interface between the MS to BSC in GSM technology is insecure due to various vulnerabilities in authentication and encryption algorithms that are discussed in Chapter 4. The coverage area of a BSC in a wireless network is called a cell. The cell is described as a hexagonal shape that can cover the particular area by a single BTS. The air interface is typically divided into two class of Um logical channels as established in GSM 04.03 specifications [3] – the Control Channels (CCHs) and Traffic Channels (TCH). TCHs are used to carry voice traffic after the call setup has established. CCHs give the information about the network and are used to assist in call setup/ SMS delivery. Many multiple BTS broadcast the paging request for the recipient user to determine quickly the area in which the device is located. Upon hearing the request for a temporary identifier on the PCH, the recipient device responds to the request and is ready to accept the incoming communications using the slotted ALOHA-based Random Access Channel (RACH) uplink. The recipient device is then assigned a Standalone Dedicated Control Channel (SDCCH) by listening to the Access Grant Channel (AGCH) [24]. Depending upon the communication, if the recipient device has an SMS message to be delivered, the BTS delivers the message over the SDCCH. Figure 3.9 depicts the overview of SMS message delivery on the air interface between MS and BTS [43].

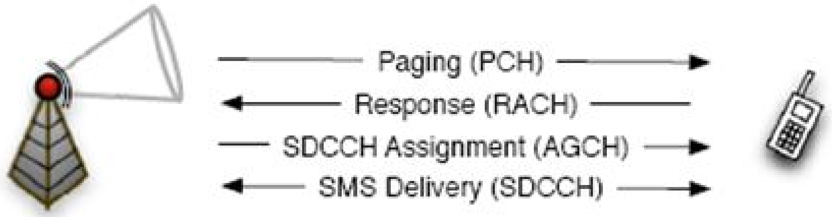


Figure 3.9: Wireless delivery of SMS

The transmission of the SMS message on the Um channel of GSM is explained below.

1. Layer 1 is present in Dedicated Control Channel also known as Dm channel that usually consist of SDCCH and SACCH. The functionality of this layer is within BSC.

2. Layer 2 comprises of Link Access Procedure Channel Dm (LAPDm) which is used in devices that operates on GPRS and use Logical Link Control Channel if required. The functionality of this layer is valid in BTS.

3. Layer 3 is used as a Connection layer and operates in MSC.

4. Layer 4 is present in the Relay layer and also operates in MSC.

5. Layer 5 is present in the Transfer layer, and its functionality terminates in SMSC.

6. The L1, L2, L3, L4 and L5 are the signaling system no.7 protocol layers that can be observed in Figure 3.10

7. Generally, when a message is transferred to L(n) layer from L(n-1) it requires a transfer and acknowledgement of the same. L1 to L4 layers are visible on Um channel.

3.7 SS7 Protocol Stack

When an SMS is sent from a sender to recipient, it has to pass through various entities involved in the telecom network architecture as described in Figure 2.1. The transmission of a message between these entities is performed by over the air (OTA) transmission and signaling system no. 7 (SS7). SS7 is a set of international telecommunications standard protocol that defines the network elements in a Public Switched Telephone Network (PSTN) for the exchange of information in a digital format. These are the different layers in the protocol that provide connections and services to the upper layers in the protocol stack [32, 40]. These SS7 protocol stack can be compared to OSI layers in computer architecture when two computers are connected to different or same networks to perform an inter-communication between them. The comparison of both the layers can be observed in Figure 3.10

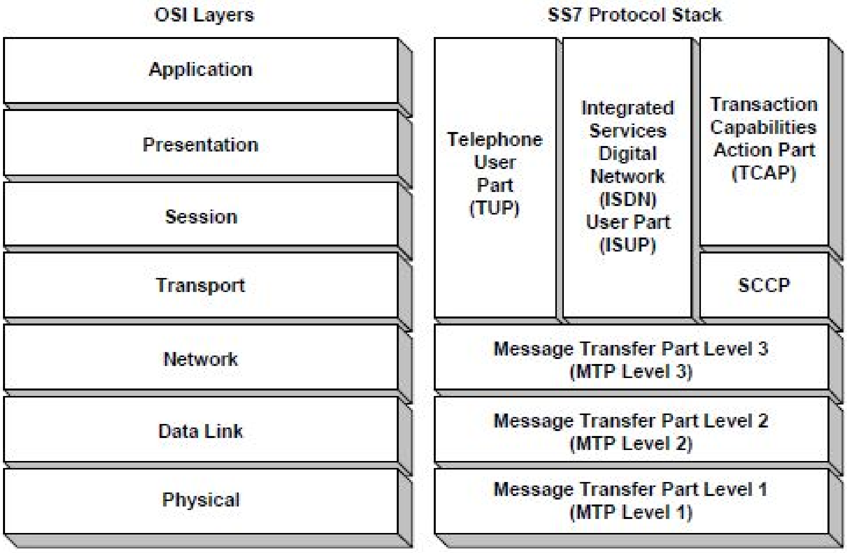

OSI Layers
Application
Presentation
Session
Transport
Network
Data Link
Physical
SS7 Protocol Stack
Telephone User Part (TUP)
Integrated Services Digital Network (ISDN) User Part (ISUP)
Transaction Capabilities Action Part (TCAP)
SCCP
Message Transfer Part Level 3 (MTP Level 3)
Message Transfer Part Level 2 (MTP Level 2)
Message Transfer Part Level 1 (MTP Level 1)

Figure 3.10: Comparision between OSI Layer and SS7 Protocol Stack

SS7 uses out-of-band signaling, which signifies that signaling information is carried on separate dedicated control channels rather than the same channel as a voice call. Since these signaling protocols were developed during the GSM establishment in 1975, the protocols have been less secured (weak), and some security researchers from Germany demonstrated that the attackers could exploit security loopholes in SS7 to track cell phone users' movements, communication and eavesdrop on conversations. Same scenarios can be seen in the recent years how SS7 eavesdropping are used to locate the movement of the user[4]. We will discuss some of the network vulnerabilities [33, 6] in next Chapter.

[4] See at http://blog.ptsecurity.com/2015/01/mobile-eavesdropping-via-ss7-and-first.html

Chapter 4

Vulnerabilities within Network

Continuous advancements are done in the telecommunication network to provide the best possible service to the users. But at the same time the GSM network has led to a myriad of vulnerabilities. These vulnerabilities are present at the network level that is eventually retaining the confidentiality and integrity of the private data through SMS messages transmitted within different network entities. Below sections describe and analyze the vulnerabilities in GSM architecture.

4.1 Lack of mutual authentication between MS and BTS

GSM was the first telecom network to establish and provide communication through mobile devices. It is preceded by GPRS and Enhanced Data Rates for GSM Evolution (EDGE), which uses the same authentication and encryption standards of GSM. As stated before in Chapter 3 the authentication procedure is a one-way process, which means only BTS can authenticate the MS and not vice versa. The authentication process requires the IMSI or Temporary Mobile Subscriber Identity (TMSI) of the device to authentication it. It then calculates the SRES (Signed RESponse) and RAND (RANDom number) explained in Table 4.1 for authentication and key agreement process. The attacker set up a fake BTS that broadcasts and authenticates the MS with GSM standards, and the legitimate MS will try to connect to the fake BTS while the user thinks the MS is attempting to connect with real BTS. The attacks performed when a fake BTS attempts to link MS to intercept the data communication for voice and SMS in GSM has been observed in recent years [1] [2].

4.2 Fallback to lower technology

When an MS attempts to connect to the appropriate network connection (UMTS or LTE) and is not successful to establish the connection, it eventually downgrades to connect the lower network i.e. GSM network. At this point, the attacker attempts to be a fake BTS and establishes a connection with the MS. By performing such fake connection, the attacker can choose not to use the encryption for the data communication. Therefore, the attacker can capture and impersonate the data. This type of vulnerability is observed in UMTS network that also holds the GSM/ GPRS/ EDGE network capabilities and are configured to connect forcefully to GSM network whenever the appropriate network is not available.

[1] See at `https://www.defcon.org/html/defcon-18/dc-18-speakers.html`

[2] See at `http://www.ibtimes.co.uk/19-fake-mobile-base-stations-found-across-us-are-they-spying-crime-1464008`

4.3 Vulnerability related to BSC

The authentication procedure is a one-way process, the attacker can place a fake BTS and authenticate the MS and capture all the network traffic. The fake BTS acts like the standard BTS operator but since the attacker is not aware of the shared secret key and encryption algorithm; it would not use encryption. Perez et al. [36] have mentioned in their experiment how they attacked the mobile data communication with a fake BTS device connected through a PC.

4.4 Vulnerability related to SMSC

The SMSC is the core network entity where the messages are stored and forwarded to the recipient. Humans handle the network components for the maintenance and stuff related to network congestion problems. If an attacker gets access to SMSC, then there is the possibility to read or modify the messages in the queue or also read the SMS traffic passing through the system. The SMSC components lie with the SS7 networks and how these systems are protected is not published. The protection of these systems varies between countries and between different operators who have their own SMSCs. Some attacks on SMSC that have performed in late 2000 such as obtaining the personal (confidential) information. One of the high-profile victim of such an attack in past was England football captain David Beckham, whose SMS exchange with his personal assistant Rebecca Loos was intercepted and published in a tabloid[3]. Also, two employees from European phone operator *mmO2* were dismissed for helping their friend to obtain copies of his girlfriend's SMS messages[4].

4.5 Vulnerability related to ESME

ESME uses SMPP protocol that depends on TCP/IP connection to connect the SMSCs discussed in Chapter 2. TCP/IP lacks security properties such as authenticity and privacy of the data passed over the network. In general, there is no security mechanism defined for SMPP and the data exchanged between two entities is in plaintext and may be intercepted over the Internet. Due to such lack of security the possible form of attacks on SMPP may consist of interception of messages, DoS attacks and SMS Spoofing. These type of attacks are explained in Chapter 5. To avoid such kind of attacks and solve the problem of authenticity and confidentiality, is to make use of Secure Socket Layer SSL/TLS connections. Also, TCP/IP is less secure and to enhance the security, *IPSec* protocols were developed. Another method can be used to setup a VPN connection between the two entities. The data packets over the network are encrypted, and an insecure channel gets a secure channel. Using VPN, neither ESME nor SMSC requires encryption methodology.

4.6 Capturing and Modifying the data during OTA Transmission

SMS messages are sent over the air interface between the MS and BTS explained in Chapter 2. All the GSM traffic; voice and data communication are performed in an encrypted format. HLR/VLR and MSC/AUC are used to perform authentication and encryption of the data under the code names A3 and A8 for authentication. The data which is sent across the radio link is encrypted with the A5 algorithm and the key derived from A3/A8 and the shared secret. It is the BTS who initiates the authentication procedure with the MS and to encrypt the data in the network. The general authentication and encryption algorithms used are explained in Table 4.1.

[3] See at `http://www.tmcnet.com/usubmit/2006/10/17/1985881.htm`

[4] See at `https://www.gartner.com/doc/379178`

Table 4.1: Authentication and Encryption Algorithms

A3	Takes the 128-bit Subscriber Authentication Key (Ki) that is stored both in the SIM and HLR and produces a 32-bit Signed Response (SRES) as an output to a random 128-bit number (RAND) challenge which is send by the (HLR).
A8	Produces 64-bit Session Key (Kc) from the 128-bit random number (RAND) and the 128-bit Ki.
A5	Uses Kc and the sequence number of the transmitted frame to encrypt the speech and data communication. A5 is implemented into the phone.

Different flavours of the A5 algorithms are:

- A5/0 means “no encryption”. Data is sent unencrypted. In some countries, this is the only allowed mode.
- A5/1 is the old “strong” algorithm with 64-bit key encryption, used in Europe and North America.
- A5/2 is the old “weak” algorithm with 16-bit key.
- A5/3 designed by 3GPP based on Kasumi cipher (semi-open process).
- A5/4 introduced in 2009 which uses 128 bit key.

The A5 algorithm is a symmetric cipher. Since the technology has been developed to LTE the A5/1 and A5/2 used in GSM and susceptible to cryptanalysis attacks. Shamir et al. [5] in 1999 described in their research that the secret key can be cracked in a minute, rendering A5/1 only to counter casual eavesdropper and A5/2 completely insecure. The attacker place a fake BTS and use weak encryption, by authenticating the MS and capturing the data. Furthermore, an attacker can intercept the messages and modify the data send to and fro from the MS.

4.7 Loss of SMS messages during transmission

The SMS messages are delivered to the recipient in the best possible way. In rare cases, the SMS messages do not reach the destination. One could consider that the messages routed between MS and SMSC can be lost or dropped during the transmission. The effect of such loss of messages results in the data loss of the user. This type of vulnerability appears to be at the lower level unless the attacker has access to any of the internal network entities of the message handling system.

4.8 SMS Spoofing at Authentication Server

SMS Spoofing is also known as Forging Originators Address, and it exists when an attacker manages to inject the SMS messages into the network with a *spoofed* originator address. The attack is performed when the details of the signaling system are known, and the attacker can change the header fields in the SMS message. The detailed SMS Spoofing attack is explained in Section 5.7. The attacker can impersonate the AUC for a typical MS or impersonate the MS for an authorized AUC. In the first case, spoofing at a great extent is possible by sending an SMS message from ESME with correct headers without the recipient being able to detect that it comes from the Internet. In the second case, spoofing is possible but the attacker is required to know the authenticating information of the user.

4.9 Lack of protection for message passing through SS7

The SMS messages transmitted between different network entities depends on the SS7 protocol stack explained in Section 3.7, and it has different levels of security included in the system. The network operators handle the security of the SS7 systems. The vulnerability to the SS7 network would lead to loss of privacy of the data and extracting the sensitive data of the users and a possibly to promote a DoS attack. In the entire network if the attacker detects the weakest part, he will either try to capture the data, inject false data or modify the data. The GSM uses algorithms that are not highly encrypted as mentioned in Section 4.6 and has the weakest security provided by the operators. These proven circumstances and the knowledge of the attacker is enough to break the protection in SS7 protocol stack[5].

4.10 IMSI Catcher

IMSI Catcher is a type of vulnerability that is used to eavesdrop the mobile communication of the user. It is used as a mobile interceptor to capture the mobile traffic and to track the movement of a user. It is a *fake BTS* that is placed between the targeted MS and the service provider's existing BTS tower. It is a type of Man-In-The-Middle (MITM) attack and is one of the possible forms of an exploit. Usage of IMSI Catcher is illegal in general, but it is used in some countries by law enforcement and intelligence agencies for spying purposes and mostly utilized by the United States [10]. To counteract for finding the IMSI catcher applications are available on Android smart-phones to detect them[6]. Applications for Android such as SnoopSnitch and Android IMSI-Catcher Detector (AIMSICD) are used to identify IMSI Catchers in the network. Whereas Apple smart-phones do not require any application to detect IMSI Catcher as the essential details of the device are in encrypted format and is not easily traceable by the IMSI Catcher as mentioned in report[7]. Joeri de Ruiter with his colleagues has performed some methods to defeat from the IMSI catchers [45]. In general the color codes that are used to distinguish between various IMSI catcher activities are explained below:

1. Green: No indicators of an IMSI Catcher attack found.
2. Yellow: Some indicators that show the anomalies. These hints are not sufficient to postulate an IMSI Catcher attack. The user should avoid critical details in calls.
3. Red: Indicators strongly suggest an IMSI Catcher attack.
4. Grey: Not enough data available.

[5] See at `http://www.di.unisa.it/~ads/corso-security/www/CORSO-0304/SMS/SMSoverSS7.htm`

[6] See at `https://secupwn.github.io/Android-IMSI-Catcher-Detector/`

[7] See at `http://time.com/3437222/iphone-data-encryption/`

Chapter 5

Attacks on SMS

The central idea to develop SMS service was to send non-sensitive data across the GSM network. The security properties for SMS transmission such as mutual authentication, encryption, non-repudiation, end-to-end security of messages were unexamined during the design of GSM network architecture [5]. Due to undefined security properties, the attackers discovered to perform attacks on SMS messaging. Most ongoing attacks on SMS messaging are explained in this Chapter.

5.1 Denial of Service (DoS) Attack

DoS attack is also termed as Flooding of messages. This type of attack focuses on the network that offer services to the users. It temporarily suspends or interrupt the services of the user's connected to the particular network. But in our case it is flooding the user's mobile phone by sending repeated messages by making it inaccessible. Researchers from South Africa [9] have explained how a DoS SMS attack on GSM is performed and also pointing out the security vulnerabilities in the GSM network architecture. DoS threats can target the underlying server architecture or sometimes exploits the vulnerabilities in the application and communication protocol. DDoS is a type of DoS attack and is termed as Distributed Denial of Service attack. The difference between DoS and DDoS attack is that; DoS uses a single Internet entry point connection to flood the targeted mobile phone for repeated messages. Whereas, DDoS uses multiple connected devices that are distributed across the Internet to perform an attack.

This type of attack could be carried out at SMS centers, where the SMS messages are queued in the buffer and have the mechanism for storing and forwarding the SMS messages. The attacker could flood the buffer queue with irrelevant messages and target to specific mobile number [35]. Due to the flooding of messages it could cause SMS center to reject the incoming messages from the user as it has particular buffer queue. These type of attacks are mostly performed in financial sectors[1].

5.2 SMS Spamming

Online marketing service providers perform spamming of SMS messages. These providers make use of marketing channel to send bulk SMS messages to users for discount coupons, free application installation link, and many such related messages. The attacker makes use of this service and replaces the content of these messages by an invalid context that is not valid and affects the mobile phone by either reboot the device or performing a strange behavior. These attacks are most commonly carried out on big banner e-Marketing websites such as Amazon, eBay, Flipkart, etc. which uses the facility to send SMS messages to users for promotional activities. A new SMS spamming attack named *Gazon* was observed in North America for Amazon which contained fake

[1] See at `http://blog.mazebolt.com/?p=131`

Amazon gift card offers and eventually leaked the contact lists of the user and send the same message to all the available contacts on the user's mobile device[2]. The SMS Spamming has now further moved from regular SMS messages to WhatsApp message spamming. The internet based applications such as WhatsApp, Viber, IM+ instant messaging and many more provides a free exchange of messages. The attacks on mobile applications are observed in Asia[3].

5.3 SMS Phone Crashing

Receiving of SMS messages in a malformed manner could make mobile phone crash or unusable. The attacker sends a malformed SMS message to the user's mobile phone which triggers the phone to stop functioning [31]. After the phone crash, to work properly again the phone is required to reboot or battery removal to restart. A recent attack on mobile devices was observed on Apple iPhones users using WhatsApp application. As soon as the malformed message was delivered to the user the mobile phone automatically restarts or gets rebooted[4]. The description of this attack is explained in Section 6.2. A few years ago Nokia mobile phones running with old firmware (Symbian) were vulnerable to SMS phone crash when a malformed message was received on those mobile phones.

5.4 SMS Virus

SMS Virus is closely related to SMS Phone crash type of attack. Phone Crashing attack is performed when a malformed message is delivered to the user. In the same sense, the SMS message can have a virus attached in the form of a link to download an application or redirects to an webpage. It can also affect the mobile phones by corrupting the operating system or creating bad sectors in the file system. An example of SMS Virus named *Selfmite* was detected which made use of SMS service by screening the first 99 contacts and sending infectious SMS to the chosen list of contacts[5].

5.5 SMS Phishing

SMS Phishing is also known as SMiShing. It is a technique used similarly to Internet phishing emails. The attacker attempts to fool the mobile phone users by sending fake SMS messages with the help of social engineering practices. User assumes that the SMS message received is genuine, and the contents of the message looks interesting. Messages received by a legitimate user attempts to open the contents of the messages by clicking on the link provided that can be tricked to download the untrusted application on the mobile phones. A simple example of phishing is like asking a password for an email account to your friend in a direct or an indirect manner. A recent SMiShing attack was noticed in banking sectors where messages sent to legitimate bank customers saying that their credit cards were temporarily suspended or blocked and to unblock the card click on the link mentioned in the message[6]. E. Kirda et al. discuss techniques that a user must be aware of this type of attack. in his research [28].

5.6 SMS Spoofing

SMS Spoofing is performed by replacing the originator's ID with an alphanumeric text. SMS spoofing at the network level is explained in Section 4.8, and the technical details how to replace

[2] See at http://www.tripwire.com/state-of-security/latest-security-news/gazon-malware-spreads-via-sms-using-fake-amazon-gift-card-offers/

[3] See at http://www.adaptivemobile.com/blog/headsup-for-whatsapp

[4] See at http://www.theguardian.com/technology/2015/may/27/iphone-crash-bug-text-imessage-ios

[5] See at https://antivirus.comodo.com/blog/computer-safety/android-sms-virus-selfmite

[6] See at http://blog.easydns.org/2015/07/11/phishing-attacks-using-sms-text-messages/

the originator address can be seen in Table 3.2. Alphanumeric text can consist of impersonating another person mobile number, company name or a product name. Apple iPhone iOS 6.0 was vulnerable to SMS Spoofing flaw. pod2g (@pod2g - Twitter), a security researcher for Apple devices, discovered that iOS was displaying the reply to address rather than the senders address. SMS messages send from various devices such as a PC or a jailbreak application that allows to configure a fake reply to a message by the sender and trick the user to believe that it came from a trusted person. An attacker makes the phishing message more plausible by changing the display name to a name familiar to the recipient. The attacker then attempts to trick the victim to provide confidential information or to open a link provided by the attacker. PDUSpy was the tool used by the researcher to perform the necessary steps involved in the attack.

5.7 SMS Fuzzing

Fuzzing is an automated or semi-automated process that involves transmitting invalid inputs to the targeted mobile phones to trigger security problems and unexpected behavior. The unexpected behavior is typically something like program crashes that is not expected under normal test conditions. It is used for testing purposes to check the reliability of the operating system to handle the weird behaviour of the device. Fuzzing is used to inject fuzzed SMS messages to the mobile phones and monitor the application behaviour under stress conditions. In fuzzing, the data included in the SMS packet is malformed which performs unacceptable situation to mobile phones. Section 3.3 explains the SMS packet structure during submit and deliver where the attacker can make the necessary changes and perform the attack. Many academic researchers have presented fuzzing on GSM protocols which includes SMS messages and mobile phones to observe the strange behavior of the mobile phone and applications [23, 34, 44].

5.8 Replay Attack

A replay attack is performed if the attacker manages to arrange authentication request/ response messages to be replayed. A replay of messages on authentication request does not appear to be obvious, but for authentication response could be a more serious vulnerability. If replaying for authentication response messages is possible, the attacker could impersonate a legitimate user and authenticate a fake transmission. If the authentication request messages consist of anti-replay mechanism such as time-stamps in response messages, then it is hard to perform a replay attack.

Chapter 6

SmartPhone OS Security

Till now we have discussed the security vulnerabilities and possible exploits on GSM network and SMS messaging service. In this chapter, we will study detailed analysis of the mobile Operating Systems (OS) and applications that maliciously use SMS service on different platforms such as Android, Apple, and Windows. The mobile industry has rapidly moved from traditional low-end phone to multi-functional smart-phones. Smart-phones provides the same functions as traditional phones such as voice and data communication, but it also includes various features such as Internet services, advanced multi-processing applications such as location services and powerful Operating Systems.

The high-end smart-phones have their OS designed and written in specific programming languages to perform the tasks. The primary function of OS is to control the security, performance and extra features of the smart-phones. Due to the rapid increase in the selling of the smart-phones that can perform multi-tasking activities, there is competition between different manufacturers to provide best OS for smart-phones. Hence, there are varieties of mobile OSs available on the market. Various OSs supports a broad range of services and features but focusing on features in some cases might neglect security. The OS of smart-phones have challenges such as reduced storage capacity, minimal user API interface and power constraint.

Most of the time users using smart-phones expect the same level of security from their phones as their laptops and PC. The reason is that the user prefers to use their smart-phone to perform daily Internet activities such as sending emails, banking operations and apparently surfing social networking webpages. Android smart-phone captures more than 50% of the market share, 14% by Apple smart-phones, and the rest is for others which include Windows and Blackberry RIM. In previous years before the smart-phones were introduced, the use of mobile phones for security purposes were efficient and simple to use. It could provide end-to-end security of the SMS messages for OTP and mTAN. In the remainder of this chapter we will discuss the basic aspects of smart-phone OS security and attacks related to it.

6.1 Android Mobile OS

Google develops Android mobile OS. Android OS depends on Linux Kernel 2.6, programmed in Java. Android OS is open source and is designed to be secure [14]. But the protection domain of the Android OS application is limited by default but to grant more permission for other activities the user has to provide it during the installation process. A large number of mobile attacks are performed on Android smart-phones with fewer computation tricks. Before installing an application on the smart-phone the user has to accept the *Terms and Conditions* to proceed further with the installation procedure. The contents described during the instllation process are critical as it gives a brief description of the application and the services used within the operating system of the device. This is the point where the application looks legitimate and often ask for more permissions than required. It's the responsibility of the user to decide if the application is safe to install. The services are nothing but the information policies of the application. M.D. Ernst et al.[18] describes the information flow policies of the applications and proposes a verification model that could detect a malicious information flow. The results generated by using the proposed verification model were amazing as out of 72 applications; 52 were malicious applications. A tempting attacking vector for the attackers is that they try to find out bug doors that grant access to undesired functionality with the application. The attacker can insert his malicious code into the weakest part of the application. Others researchers who studied the Android malware application attacks have laid down their observations and findings in [12, 17]

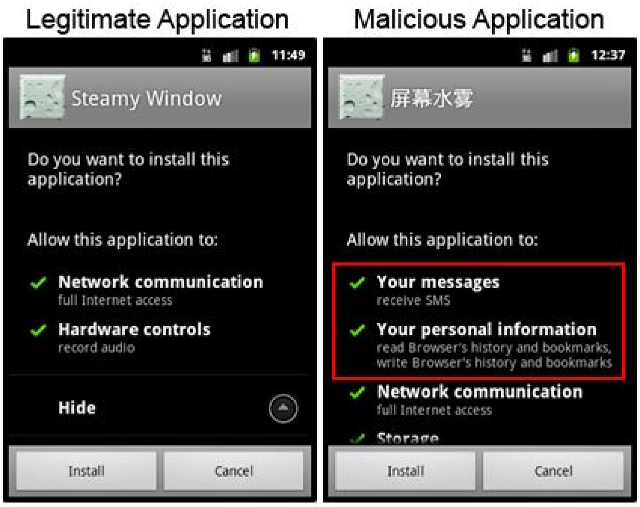
The threats from malicious Android applications during the installation process are:

1. The application tricks the user to give permissions to various system resources including APIs.

2. The malicious code is hidden behind the correct application permissions.

3. Sometimes the application tricks the user to enter confidential data such as credit-card number.

Andriod malware is always on the rise, a typical example of a compromised Android application can be observed in Figure 6.1. Once the application is installed, it is hard to differentiate between the legitimate and malicious version. It is important to identify the malicious version of an application during installation by observing the excessive permission requests. Joany Boutet studied various Android applications that perform malicious activities and laid his findings in his scientific paper [7].



Do you want to install this application?

Allow this application to:

- ✓ Network communication
  full Internet access
- ✓ Hardware controls
  record audio

Do you want to install this application?

Allow this application to:

- ✓ Your messages
  receive SMS
- ✓ Your personal information
  read Browser's history and bookmarks, write Browser's history and bookmarks
- ✓ Network communication
  full Internet access
- ✓ Storage

Figure 6.1: Legitimate and Malicious Version of Android application

Among many of the malicious Android mobile applications worldwide, one of the recent example in 2013 of SMS Trojan, which is mobile malware infected many smart-phones in South Korea. The application pretends to be a famous coffee shop coupon application, but it was a Trojan that could capture the SMS messages from the device. When a user makes an attempt to open the application, initially it would display an error message of "*server is overloaded*". Further with the help of social engineering techniques the user believed that the application is legitimate, and the user quits the application in typical fashion. But actually when the first error message displayed it would start its services in the background and send the user's mobile number to the requested URL, and thus it gets registered in the attackers database. After this process, the attacker could monitor all the incoming SMS messages with the messages stored on the smart-phone. The attacker could divert the direction or block the incoming messages notification to the user. In this way, the attacker could intercept the SMS messages thereby compromising the security and privacy of the user[1].

Another famous attack on European banks related to online banking in 2012 was termed as *Eurograbber* that stole more than 36 Million Euros through a mobile malware. The attack was mainly focused on infecting the PC and mobile phones of the users by a variation of Trojan named "ZITMO" (Zeus-In-The-Mobile)[2]. The devices infected made by the attackers would monitor their sessions and manipulate the online transactions performed by the user. The 2FA used by the banks was the sole purpose of compromising by the attackers. Zitmo was traditionally designed for Android devices but later it was developed for BlackBerry as business executives usually use these smart-phones. With the help of the phishing techniques, the attackers managed to install the Zeus Trojan unknowingly on the users PC by surfing through various websites on the Internet. The Trojans keeps a close watch on the sessions when the user logs into his online bank account. As soon as users logs into his account the Trojan intercepts the session with a malicious Java script informing the user about the *security upgrade* instructions. The Trojan requested the user to provide the mobile phones information such as the operating system (Android, Apple or Windows) with the mobile number. The upgrade security information looked legitimate to the user and all the user's provided the required information upon request. After registering the mobile number, the attackers send an SMS with a malicious link to download and upgrade the application on mobile phones. After installing the malicious application by the user, a verification code was sent to the mobile phones to confirm the up-gradation process. Performing all the installation steps, a message was displayed on the mobile phones of the user acknowledging successful completion of a system upgrade. The malicious application installed on the mobile phones would intercept the SMS messages from the mobile phones. The SMS messages are nothing but the TAN codes sent by the Bank to authenticate the online transaction. The detailed case study of this report was prepared by "Check Point Software Technologies" and "Verasafe", which can be explored in [26].

6.2 Apple Mobile OS

Apple iOS is the Apple's smart-phone OS. Darwin is the name of kernel iOS, which Apple released a UNIX foundation open source OS, programmed in *objective C* in 2007. *Swift* is new Apple's programming language that works similar to *objective C*. The applications on Apple smart-phones must be download from Apple App Store. It is considered to be the most secure App Store because Apple developers perform security testing of the application, and if the application detects no malicious activity, then it is released on App Store to download for the users.

Apple iOS provide various layers of security features to ensure that the applications run in a proper format. Every application that is listed on the App Store has to be approved, signed and verified by Apple and provide an Apple-issued certificate. The certificate is to verify that the applications come from a known source and does not affect the system integrity, thus providing iOS users to use these applications without any fear of malware or virus. In simple terms, the certificate is termed as *Certificate of Conduct*. It is a mandatory step to sign the code and prevent any third-party applications to modify the code. To develop or install Apple applications either a person or an organization; the developers have to be registered with Apple and join the iOS Developer Program. Also, these entities are verified by the Apple before the certificate is issued.

[1] See `https://blogs.mcafee.com/mcafee-labs/sms-trojan-targets-south-korean-android-devices/`

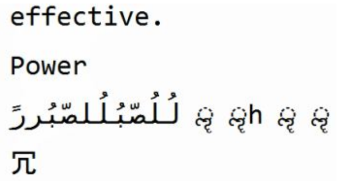
[2] See `https://threatpost.com/zitmo-trojan-variant-eurograbber-beats-two-factor-authentication-steal-millions-120612/77287/`

Before the applications are launched in the App Store, the applications are tested, and sandboxing methods are used to ensure that they do not contain any malicious data or any unwanted behavior [4].

An application that is developed by a third-party is tested thoroughly using sandboxing techniques to check that it does not access any other files stored on the smart-phone. This restricts the applications to modify or access sensitive information on the device. Following all the guidelines and security checks provided by Apple for its applications, it is not possible for an attacker to launch the application quickly in App Store. To protect all the data accessed by the applications Apple iOS provides Software Development Kit (SDK) for all the third-party developers to provide the highest level of security for their applications.

Discussing all the security parameters and data protection checks for the applications, first large scale attack was performed on App Store in China. This incident occurred on September 21, 2015. The name of the Malware was *XcodeGhost* which was first detected by a Chinese e-commerce firm Alibaba and further detailed analysis was carried by Palo Alto Networks. The Malware was named as *XcodeGhost* because it hacked the Apple's Software Development Suite named *Xcode*. Most traditional affected applications in China includes popular WeChat app, music downloading app, and a car-hailing app. Apart from the applications mentioned more than 25 apps were infected. The primary goal of the Malware attack was to send fake alerts to the infected devices to trick their owners to reveal their confidential information. The confidential information captured by the attackers was the iCloud (Apple ID) credentials and many other passwords[3].

Another recent example of SMS message bug was detected in May 2015. The SMS bug containing the keyword *effective power* with Arabic characters once received on iPhones would lead to reboot or crash the device. The sample of the SMS message can be seen in Figure 6.2. This type of message was either received through iMessage or standard SMS messages. The primary cause of the trouble was not with the Arabic characters but in fact the uni-code representing them as a Core Text, a library software that helps to display regular text on screens. The bug could access the invalid memory parameters within the operating system to kill the current running program that was the text message application at that times[4].



effective.

Power

لُلُصّبُلُلصّبُررً ॣ ॣh ॣ ॣ

冗

Figure 6.2: iPhone SMS bug

6.3 Windows Mobile OS

Windows Mobile is Microsoft's smart-phone OS. This OS provides a similar platform to desktop OS than other smart-phones. Windows Mobile OS is programmed in C# and C++ with .NET Framework. Windows Mobile 7 was the first smart-phone launched in late 2010. Windows Mobile and Apple iOS which has C/C++/C# programming nature is vulnerable to traditional C programming language weakness and bugs. To mitigate these vulnerabilities, Windows Mobile allows its developers to write secure code by using libraries such as *Stack Cookie Protection*, *StrSafe.h* and *IntSafe.h* [14]. Windows Phone Store is the location from where the users can download

[3] See http://www.databreachtoday.com/apple-faces-app-Malware-outbreak-a-8538

[4] See http://www.theregister.co.uk/2015/05/27/text_message_unicode_ios_osx_vulnerability/

the applications on their smart-phones. For Windows mobile application security, to launch the application in the store it has to be certified and authenticated by the Microsoft developers. Just like Apple application certificates, Windows Phone applications also uses certificates that grant permission to run with privileged permissions.

Not many mobile attacks are recorded on Windows Phones compared to Android and Apple because due to its low popularity of the mobile OS. A year ago in 2014 a serious security loophole was detected by XDA Developer hackers in OS of Windows Phone 8.1 which could make the OS easy to hack. The vulnerability allowed the attacker to run the malicious application with high privileges, and can edit the registry entry[5]. Since this thesis is focused on the SMS security, no hacks or malicious applications were developed that could capture the SMS messages of the user and leak the data.

6.4 Cross-Platform Smartphone Attacks

Cross-platform attacks are performed when a smart-phone and PC are connected to each other in some ways. The goal of these attacks is to corrupt the OS of the smart-phone or install a Trojan in PC without the user's permission. Cross-platform can be performed in two different ways in the same manner as MITM attack. The two methods mentioned below has proved to implement a systematic attack and is explained by A. Dmitrienko et al. in his findings [13].

PC-to-Mobile is a type of attack when malware infects PC or a laptop and is transferred to mobile through a USB connection. Users always connected their PC/ laptop either to a LAN or WLAN connection. Also, WLAN connections include the free WiFi hotspots that are untrusted connections and provides free connectivity. Surfing through various untrusted websites, the user may come across a malicious website from where malware could automatically installed in the system. This malware can infect the OS, applications or peripherals of the system. Furthermore, users usually connect their smart-phones to PC/laptop using USB connection to synchronize. During the data transfer, the infected files or data get transferred to the smart-phone, and it gets installed on smart-phone performing the malicious activity without the user interaction.

Mobile-to-PC is a type of attack that occurs when sharing the Internet connection from smart-phone. It is known as tethering. As already explained before smart-phones gets easily connected to the WiFi hotspots available in the surrounding areas. A situation arises where a user wants to perform a bank transaction, and the user's laptop is not able to connect to the Internet. In this case, the user shares his mobile Internet/ WiFi connection to perform the desired activity. If a malware already infects the smart-phone and sharing the Internet connection with the laptop, the malware may propagate from smart-phone to laptop thereby affecting both the devices and the attacker compromise the confidential information of the user.

[5] See `http://thehackernews.com/2014/11/windows-phone-81-hacked.html`

Chapter 7

SMS based Authentication Schemes

Apart from sending and receiving regular text messages, SMS is also used as 2FA scheme that provides a higher level of security. 2FA is a further extension of standard authentication procedure containing username and password. It is a second level for authentication process to login to your account. 2FA requires the user to have two out of three credentials before accessing the account. The three types are:

- *Something the user knows*, such as personal identification number (PIN), password or a pattern.
- *Something the user has*, such as ATM card, smart-phone.
- *Something the user is*, such as biometric like a fingerprint or face detection.

We are interested in the case, "Something the user has" i.e. smart-phone where the user can receive an SMS message containing OTP. Before logging into your account, the application vendor will send an OTP through SMS to the user's registered mobile number. When the user receives an SMS message containing OTP, the OTP must be entered on the other device i.e. PC to access the account. Identical methods are used for online banking services while performing online transaction payment. The banks use mobile Transaction Authentication Number (mTAN) to verify and authenticate the transaction being completed. The bank generates an mTAN that is sent to the user through SMS. The user has to confirm the transaction by entering mTAN code to complete the transaction successfully.

2FA proves better security level because even if one of the device (e.g. PC) is compromised, a typical scenario nowadays – the possibility of malware to gain control over the second device (e.g. smart-phone) simultaneously is limited. The primary goal of 2FA is to mitigate the account abuse even if a Trojan compromises the user's login credentials. 2FA is supported by almost all the online applications providers like Google, Microsoft, ING Bank in The Netherlands, DigiD, WhatsApp, Viber and so on. The user has to make use of his credentials and OTP to complete the login procedure. DigiD is a type of identity management system used by the Dutch government. It includes online services such as health insurance, tax benefits, rent benefits and custom administration services that are used to verify the identity of Dutch residents. The basic criteria to use DigiD is that one should already have a burgerservicenummer (BSN) number. DigiD also uses 2FA to defeat against cyber frauds. DigiD had suffered a DDoS attack as 10 million Dutch citizens were affected in 2013[1]. The DDoS attack was not performed on SMS based authentication service but a typical DDoS attack by making the service unavailable to the users. As the reports

[1] See http://www.nltimes.nl/2013/04/25/digid-cyber-attacked-again/

and a spokesperson for the Ministry of Interior Affairs stated that the personal details of the users were not compromised. To avoid confusion in our thesis, we will consider random numbers as OTP.

Usually users' like to use the services that are free of cost for day-to-day communication applications. Also, the users depend on the digital data, and it is important to protect the data both online and on mobile devices. Users do not see any problems while using these services. But when a security exploit is detected by an attacker and gets published in the media; grabbing the attention of users towards it. The usage of the application downgrades eventually. It is obvious that security is the biggest issue related to communication over the Internet. The usage of smart-phones have increased rapidly, and user's try to connect their smart-phones to unsecured or untrusted network environments without knowing the consequences of it. Mobile applications usually automatically login over untrusted networks. Due to the un-noticed login of applications increases the security risks and confidential data stored on the smart-phone by the user is compromised leaving behind no traces. In the remainder of this Chapter, we discuss the mechanism of OTP and mTAN used for authentication and online banking services. Also, we present three case scenarios namely Google Authentication, ING Bank and instant messaging applications where 2FA are used and explains various attack vectors used to compromise the 2FA mechanism.

7.1 Two Factor Authentication

As mentioned above 2FA methods such as OTP and mTAN are widely spread and deployed for Internet-based web-applications login procedures and online banking respectively. It usually makes use of two methods "*Something the user has*" and "*Something the user knows*" which is already mentioned in the above section. A classic example to support this situation is to withdraw money from ATM. The user has a bank card, and the user also knows the PIN for his card. It provides a better mechanism for security and privacy of the users. In 2FA scheme, mostly user's do not have to carry hardware security tokens. Instead, it makes use of SMS. For this purpose, the user has to register his mobile number during the registration procedure, so that in future during login the service provider will send the OTP code to the registered mobile number and this code is used as a second step for the authentication process. After opting for 2FA, the question arises how the service will be delivered to the user?. Service providers that offer 2FA must make sure that they meet the requirements of the user to expand the business globally. The use of 2FA methods is used to make a trade-off between security, cost and simple usability for the users to protect their login credentials. 2FA uses various technologies such as security tokens (e.g. RSA SecurID); codes received through SMS and codes generated by dedicated mobile application.
The SMS 2F verification is used for:

- Fraud prevention
- Identity verification
- User verification
- Strong password
- Secured account access

To overcome this situation, we will investigate how 2FA works in different scenarios and evaluate the security schemes for mobile 2FA that are utilized by the users currently. The basic idea to perform an attack on these authentication schemes is to access the original code inside the SMS and replace the original code with a fake code using SMS spoofing method. The SMS Spoofing attack is explained in Section 5.6. The fake SMS message includes an invalid activation code that looks like the original SMS message sent from the service provider where the user is trying to register.

A methodology required to break 2FA contains the following main points.

1. Obtaining the user's username and password by brute force attack or other methods such as phishing, keylogger, social engineering practices.

2. The user's mobile number linked to the account by observing the account settings.

3. A spoofing mobile voice call service.

4. Voicemail number of the mobile networks for remote access.

Above explained method to break 2FA for an account is not difficult to perform. Obtaining the account password can be done using traditional methods described in Section 7.5.1 and also mobile number attached to the account. Using a mobile voice spoofing service such as *Spoofcard* which only costs $10 for 60-minute usage. Instead of using Spoofcard application one could also use VoIP service which also allows CallerID spoofing. Also, obtaining the voicemail number of the mobile network operators are a few Google searches away.

The first method of the attacker to perform an exploit is that the attacker logs into the victim's account. The attacker then engages a call on victims mobile number (for 1-minute maximum). At the same moment, the attacker opts for 2FA option to receive the code via phone call. Engaging the victim in a call made by the attacker, the 2FA calling service will deliver the code into the victim's voicemail inbox. This is the flaw and can be considered as a reasonable amount of risk involved when 2FA code is delivered in the voicemail. There are various voicemail hacks performed over the past many years, by sending the OTP codes to voicemail, as a possible way to bypass 2FA.

The second method is involved with the mobile networks that are vulnerable. Usually, Automatic Number Identification (ANI) or CallerID is used to determine whether or not the caller is the owner of the voicemail account. If the ANI matches with the account holder, the system does not prompt for the pin code to access the voicemail account.

7.2 One-Time Password

OTPs mostly contain numeric characters but sometimes alphanumeric characters that are used during the authentication process and used for a single session. The OTP code that are generated by the application server is time-synchronized, which are valid only for a short time. Most of the time OTPs uses a secure way to register a particular mobile number for mobile applications such as WhatsApp, Viber, Facebook Messenger or to log into web account. OTPs are the latest tool used by banks and service providers in the fight against cyber fraud. OTPs are based on the algorithm HMAC SHA-based One-time Password (HOTP) which is described in IETF RFC 4226 article. The HMAC SHA algorithm is used to perform the authentication process using challenge-response mechanism. It is not an encryption algorithm but uses a hashing algorithm that transforms a set of bytes to another set of bytes. The number generated by this algorithm is not reversible which means by using the result you can go back to the source.

An HMAC SHA initially uses a key to transforming the input into an array of bytes. The key is the secret key that should never be accessible to the attacker, and the input is the challenge. This states that OTP is based on challenge-response mechanism. The secret key must be at least 20 bytes long and usually a counter of 4,5 or 6 bytes to create a 4,5 or 6 digit OTP. OTP can be generated at the client side and also on the server side. To create OTP at the client side, the user has to make use of the hardware token or mobile application provided by the particular vendor that generates OTP. The OTPs generated at the server side are delivered to the user in the form of SMS through out-of-band (OOB) service. To receive OTPs from application server the user has first to register the mobile number where the OTP has to be received.

7.3 Transaction Authentication Number

TAN is a type of OTP which is used for online banking services to authorize online transactions. The mTAN code sent by the bank to the user's mobile phone through SMS may also contain transaction data details with the code that is generated for every transaction made by the user. Various forms of authentication schemes have TAN codes such as classic TAN, indexed TAN, indexed TAN with CAPTCHA, mobile TAN, chip TAN and photo TAN. The security and convenience level of using such different type of TAN methods by the user are mentioned in the article[2]. Since this thesis is focused on SMS insecurities we will consider the mobile TAN, which is currently used by ING Bank in The Netherlands. An mTAN is a type of TAN which uses SMS message service to deliver the code to the user when a transaction is initiated. A typical scenario how OTP and mTAN are used can be observed in Figure 7.1

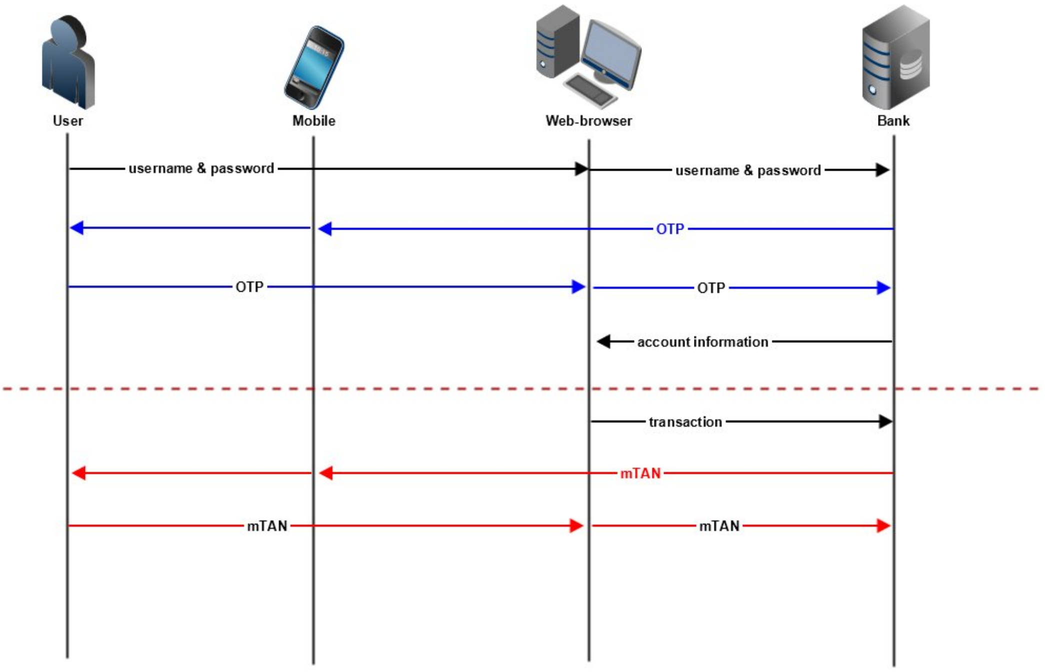


Figure 7.1: Authentication Procedure (OTP+mTAN)

1. The user logs into his account with username and password.
2. The web browser forwards the first step of the authentication request to the server and sends OTP with the registered mobile number of the user.
3. The server sends the OTP to the mobile number linked to the account via SMS message service
4. The user receives the OTP on his smart-phone and enters the OTP into the web browser and confirms the second step of the authentication procedure. The user submits the OTP and the user can access his account. These are the steps followed for 2FA.
5. After logging into the account to carry out a financial transaction such as money transfer to another bank account. The user generates the online transaction data and submits it to the bank.

[2] See at http://www.ghacks.net/2014/05/08/secure-different-online-banking-payment-authorization-methods/

6. To authorize and authenticate the transaction that is performed by the user the bank server creates the mTAN and sends it to the users mobile number registered with the account.

7. The user receives the mTAN on his smart-phone verifies the details and enters the mTAN to authenticate the transaction and submits it.

8. The bank confirms the transaction details and the mTAN code and successfully processes the transaction.

7.4 Google Authentication

Google is one of the biggest Internet service providers worldwide and was an early adopter for 2FA. Google offer many services such as Gmail, YouTube, Google+ and many more. The competitors of Google are Hotmail, Yahoo, Facebook, etc. To manage the security and privacy issues of the users; Google has introduced various 2FA methods such as Google Authenticator[3], Security Key[4] and SMS based OTP to enhance the security level of the authentication procedure.

One of the best things to make sure that the user accounts are not compromised is to introduce 2FA. Only having a good password does not solve the problem of security; if someone gets your password, then they can access your account and change the security settings. But to a certain extent 2FA solves the problem. Google Authentication is one of the applications that uses 2FA. There are various ways one can choose to apply a second step verification to his Google account. The standard method of using 2FA is to register the mobile number of the user to the account and every time to login to your account an OTP is sent to the registered mobile number.

The Google Authenticator mobile application generates OTPs at the client side. Instead of providing the mobile number during the setup process or while activating 2FA one can just download the application from the internet on the smart-phones. This application is available on Android, iOS, and Windows mobile operating systems. The OTP generated in this application is valid only for 30 seconds. The OTPs are generated in time-based fashion.

Apart from the security mechanism explained above for 2FA Google also makes use of backup codes. Typically these codes are used when the user can't receive the SMS code on the registered phone, or the phone is stolen. There are ten pre-defined backup codes for every account. One can use these codes in emergency situations when the user's smart-phone is misplaced. One can make a note of these codes or either keep it in a safe place where no one can access it except the user.

Implementation of 2FA initially for the security purposes was not up to the merit level. But as the time passed some attackers found the vulnerabilities in the system and hacked the accounts of the users'. Since then the security and privacy of the users' data are considered, and the application is developed and maintained in a secured manner.

A famous incident how hackers hacked the Google account through an Instagram application which had 2FA. Blakeman, the victim describes in his quotes as *"The attack actually started with my cell phone provider, which somehow allowed some level of access or social engineering into my Google account, which then allowed the hackers to receive a password reset email from Instagram, giving them control of the account"*[5]. A method that can be used to break the 2FA in general scenario is explained in Section 7.1.

[3] See at `https://en.wikipedia.org/wiki/Google_Authenticator`

[4] See at `https://support.google.com/accounts/answer/6103523?hl=en`

[5] See at `http://gizmodo.com/how-hackers-reportedly-side-stepped-gmails-two-factor-a-1653631338`

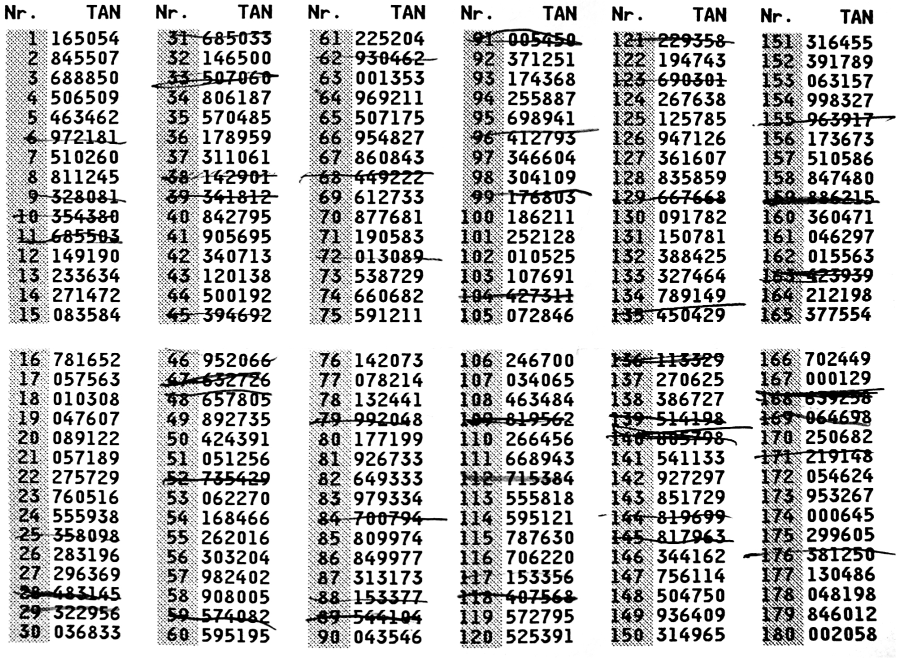
7.5 ING Bank, The Netherlands

The ING Bank in The Netherlands is one of the multi-national financial institutions that provide online banking services to its customers. There are various online services offered by the ING Bank are listed in Table 7.1

Table 7.1: Online service offered by ING Bank

Service	**Authentication Methods used**
Money transfer to other bank account	Username and Password + mTAN
Check account balance	Username and Password
Amount transfer for trading of stocks	Username and Password + mTAN
Amount transfer from savings to current account	Username and Password + mTAN

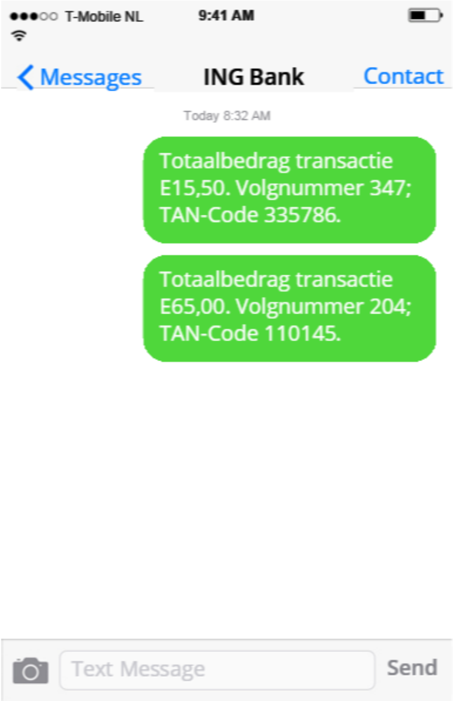
The user does not choose the username and password. Instead, it is sent by the bank to the user, and it cannot be customized. To perform certain services such as amount transfer mTANs are used. For this purpose, the user has to register his mobile number with the bank. The typical scenario how ING Bank uses mTAN for performing an online transaction can be seen in Figure 7.1. The bank provides two types of TAN codes. Paper-based TAN code and mobile TAN code. Paper-based TAN codes are a list of 50 codes printed on paper and sent to the user in advance, and it is not transaction dependent as the user has to use each code for per transaction. Whereas mobile based TAN code is sent via SMS message service to the user's mobile number registered with the account and contain transaction details. As an example of paper based TAN list and SMS based mTAN received on mobile can be seen in Figure 7.2 and Figure 7.3 respectively.



Nr.	TAN	Nr.	TAN	Nr.	TAN	Nr.	TAN	Nr.	TAN	Nr.	TAN
1	165054	~~31~~	~~685033~~	61	225204	~~91~~	~~005450~~	~~121~~	~~229358~~	151	316455
2	845507	32	146500	~~62~~	~~930462~~	92	371251	122	194743	152	391789
3	688850	~~33~~	~~507060~~	63	001353	93	174368	~~123~~	~~690301~~	153	063157
4	506509	34	806187	64	969211	94	255887	124	267638	154	998327
5	463462	35	570485	65	507175	95	698941	125	125785	~~155~~	~~963917~~
~~6~~	~~972181~~	36	178959	66	954827	~~96~~	~~412793~~	126	947126	156	173673
7	510260	37	311061	67	860843	97	346604	127	361607	157	510586
8	811245	~~38~~	~~142901~~	~~68~~	~~449222~~	98	304109	128	835859	158	847480
~~9~~	~~328081~~	~~39~~	~~341812~~	69	612733	~~99~~	~~176803~~	~~129~~	~~667668~~	~~159~~	~~886215~~
~~10~~	~~354380~~	40	842795	70	877681	100	186211	130	091782	160	360471
~~11~~	~~685503~~	41	905695	71	190583	101	252128	131	150781	161	046297
12	149190	42	340713	~~72~~	~~013089~~	102	010525	132	388425	162	015563
13	233634	43	120138	73	538729	103	107691	133	327464	~~163~~	~~423939~~
14	271472	44	500192	74	660682	~~104~~	~~427311~~	134	789149	164	212198
15	083584	~~45~~	~~394692~~	75	591211	105	072846	~~135~~	~~450429~~	165	377554
16	781652	46	952066	76	142073	106	246700	~~136~~	~~113329~~	166	702449
17	057563	~~47~~	~~632726~~	77	078214	107	034065	137	270625	167	000129
18	010308	~~48~~	~~657805~~	78	132441	108	463484	138	386727	~~168~~	~~639258~~
19	047607	49	892735	~~79~~	~~992048~~	~~109~~	~~819562~~	~~139~~	~~514198~~	~~169~~	~~064698~~
20	089122	50	424391	80	177199	110	266456	~~140~~	~~805798~~	170	250682
21	057189	51	051256	81	926733	111	668943	141	541133	~~171~~	~~219148~~
22	275729	~~52~~	~~735429~~	82	649333	~~112~~	~~715384~~	142	927297	172	054624
23	760516	53	062270	83	979334	113	555818	143	851729	173	953267
24	555938	54	168466	~~84~~	~~700794~~	114	595121	~~144~~	~~819699~~	174	000645
~~25~~	~~358098~~	55	262016	85	809974	115	787630	~~145~~	~~817963~~	175	299605
26	283196	56	303204	86	849977	116	706220	146	344162	~~176~~	~~381250~~
27	296369	57	982402	87	313173	117	153356	147	756114	177	130486
~~28~~	~~483145~~	58	908005	~~88~~	~~153377~~	~~118~~	~~407568~~	148	504750	178	048198
~~29~~	~~322956~~	~~59~~	~~574082~~	~~89~~	~~544104~~	119	572795	149	936409	179	846012
30	036833	60	595195	90	043546	120	525391	150	314965	180	002058

Figure 7.2: Paper TAN list

Compared to other banks in The Netherlands, Rabobank provides Random Reader, ABN-AMRO provides e-identifier, and SNS Bank provides Digipass, which all are the forms of hardware



Totaalbedrag transactie E15,50. Volgnummer 347; TAN-Code 335786.

Totaalbedrag transactie E65,00. Volgnummer 204; TAN-Code 110145.

Figure 7.3: mTAN received via SMS service

security tokens that generates OTP (random numbers) and TAN codes at client side and works on challenge-response procedure. The banks do not charge any extra money for the reader, and it is provided for every new account opened by the customer. This increases the cost to provide the security token to every customer, so to reduce the production cost of hardware tokens ING Bank uses SMS service to deliver the mTAN codes to its customers. We are interested in learning about SMS security and for that reason we will focus on mTAN that is used by ING Bank. This technology has proven better for a couple of years and still working fine but there are risks and security issues related to this SMS service in the present world and how they can be affected will be discussed in the following sections.

7.5.1 Threats to SMS in Online Banking

We have already summarized the sophisticated attacks and network vulnerabilities on the cellular GSM network in Chapters 4 and 5. In this section, we will discuss more specific risk factors depending on SMS message service that can be exploited in Financial sector.

Usually, banks use two types of messages *Push* and *Pull messages*. The description of these messages is explained below.
Push Messages consist of:

1. Bank chooses to send out messages to customer smart-phones.
2. Reporting of salary and other credits to the bank account.
3. Large amount withdrawals and payments using bank cards (Debit and Credit cards).
4. OTP for authentication.

Pull Messages consist of:

1. These messages are initiated by the user using a smart-phone obtaining information or performing a transaction in a bank account.
2. Account balance and Mini statement request.
3. Requesting or De-activating ATM cards to suspend when it is lost.
4. Electronic bill payments.

There are various effective ways for banks to offer services to their customers. In general, we have banks from different continents such as first the banks from Asia, second the banks from Europe and third the banks from the United States. For example, the banks from Asia and the United States do not use the method of mTAN for transaction authentication. Instead, it works just on the online transaction password that is alphanumeric to perform an individual transaction from a bank account. Whereas, banks in Europe uses random readers and mTAN for performing online transactions. Also, banks in Asia makes use of *Push* and *Pull* messages as a value added service to keep the users alerted whenever an activity is performed on their bank accounts whereas banks in Europe do provide such services but they have to be activated by the user.

Today, the telecom industries use Fourth Generation (4G) service known as LTE and utilizes most secure authentication and encryption standards that are not known to public as compared to GSM. Attackers are constantly trying to use new methodologies to find a vulnerability or break into the systems. As already mentioned to perform an attack on both the systems i.e. PC and smart-phone at the same time is not practically possible but by combining certain attack vectors it is possible and is experienced in past, *Eurograbber* case in Europe; explained in Section 6.1. But we will still discuss the matter how both the things can be simultaneously attacked. A tree diagram in Figure 7.4 will demonstrate how an attacker could perform his attack on both the systems simultaneously.

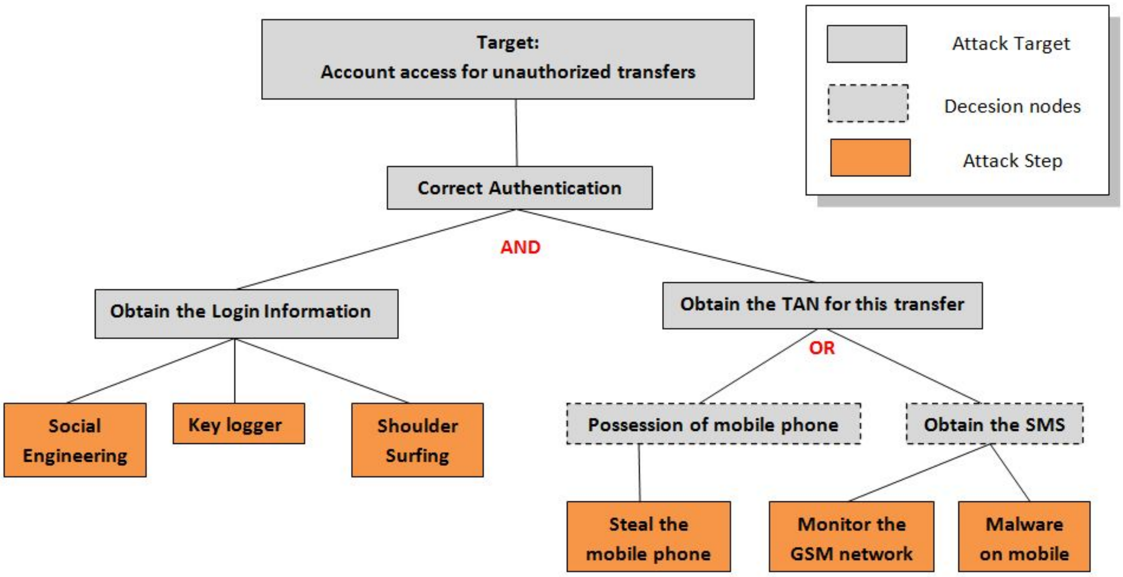


Figure 7.4: mTAN Threat Tree Model

The different types of attack vectors involved to steal the login information mentioned in Figure 7.4 are described below. The login information also includes of online service providers like Google, Hotmail, Yahoo, etc. Online banking Trojans like ZeuS/ZitMo and SpyEye/Spitmo were the first mobile malware introduced to trick the users and steal the confidential information from them with the help of phishing techniques. Researchers from Germany have performed some attacks to break the online 2FA for mTAN that is explained in their research paper [11]. Also, P.Schartner et al. describes the attacks conducted on banking application that uses mTAN are outlined in the journal [38].

Phishing and Social Engineering techniques are the most common practices performed by attackers to steal the login credentials of the users. These are nothing but the type of spam emails send to the users with fake information and gaining the sympathy of the users to click on the link mentioned in the email and provide their login credentials for the account. The regular SMS Phishing attack is explained in Section 5.5. Also, it can also ask the user to provide the smart-phone number to provide a URL for the application link to download on the smart-phones. The mobile application may contain malware or trojan that steals the rest of the confidential information from the smart-phone. Some of the attacking steps are explained below.

1. **Social engineering**: It is a non-technical practice used by the hackers that depend on tricking the users to reveal their confidential information. This kind of attacks is usually seen when a user receives a spam or junk email. It can also be practiced by talking to the users directly on the smart-phone and asking them to give the authentication information for security purposes. It is still an underestimated risk factor and for this purpose many campaigns and posters are displayed on the bank websites to warn the users not to provide any details to other persons as they are the private and confidential information of the user.

2. **Key Logger**: This type of attack can be distinguished between two types mainly hardware and software based. It usually captures and create logs for all of the keystrokes the user types either through a regular keyboard or virtual keyboard. The advanced version of key logger is an automatic process that sends a file to the attacker periodically of the keystrokes used by the user.

3. **Screen Capturing**: This is often used in combination with the Keylogger. In this case, both the keystrokes and the visual items such as the virtual keyboard are captured by the attacker. Sometimes these attacks can also view or alter the confidential information stored on the system. As the name suggests of this attack, it can also take the screenshots or capture the whole screen.

4. **Shoulder surfing**: It is a method to disclose the sensitive information such as login credentials of the user by looking at the screen or keyboard while the user is performing an online transaction. An example of this kind of attack would describe by placing a camera at the ATM centers and capturing the sensitive information of users by looking at their ATM PINs as this is an automated process of obtaining data.

From the attacker tree model in Figure 7.4 other attack vectors to obtain the mTAN code from mobile device are as follows:

1. Steal the smart-phone of the targeted user.
2. Clone the SIM card[6].
3. Intercepting the network signals OTA, which can also be termed as network sniffing.
4. Run malicious code on the targeted user smart-phone to intercept the SMS messages.

Various combinations of attack vectors are still possible for an attacker to execute an attack on both the devices simultaneously.

[6] See http://www.iol.co.za/news/south-africa/victim-s-sim-swop-fraud-nightmare-1.385531#.VgPlmBGqpBd

7.6 SmartPhone Instant Messaging Applications

In recent years, while downloading an application on smart-phones the user must provide login credentials (username and password) as a first step to download the application and later to use this application the user has to register his mobile number as a second step of authentication. OTP is sent as a regular SMS message to confirm the mobile number. On confirming the mobile number the user can access the application. Previously SMS messaging was the only service used as a medium of communication but now it has moved on to internet based mobile applications such as WhatsApp, Viber, Facebook Messenger, Google Hangouts, IMO, WeChat and many more. Our primary concern is the security of the mobile applications that uses OTP via SMS message service to confirm 2FA process of the users' mobile number for a particular application. Authentication of a user by various methods is a current field of research in information security for web-based services. The properties and security features with information flow policies for applications of Android [18, 12, 17] and iOS [15] are already studied by researchers.

One interesting feature in regards to user's mobile number for identification during authentication is that Apple applications do not allow to access the mobile number, instead user have to insert their mobile number manually for confirmation. Whereas, Android application grants permission to access the mobile number of the user directly.

During the initial stages, the applications we are going to discuss in this section were not secure. The reason for not being secure at the first phase would be that it mainly focused on the communication services as this was the primary concern of these applications. But as the time elapsed more users started using the applications and security became the main issue. Many researchers have examined and performed security evaluations of these applications, and the later version includes a better form of encryption standards. The basic secure authentication scheme used by mobile communication application can be observed in Figure 7.5.

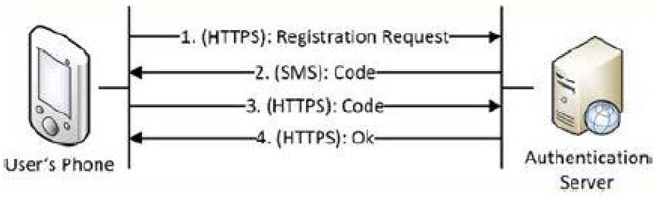


Figure 7.5: Secure Authentication scheme used by WhatsApp, Viber, Facebook Messenger

7.6.1 WhatsApp

WhatsApp is the most popular instant messaging application available for various mobile platforms. Till now the application vendor has not released any information regarding the application and is trusted to be a secure instant messaging application. The installation procedure of the application is discussed in above section. The user identification is performed by registering the mobile number of the user, and the application server sends an OTP to the mobile number. There might arise some problems such as server has to verify the user input for the mobile number to check if it is not a malicious user and could enter someone else mobile number and hijack or create the account with fake credentials. The mobile numbers used acts as user ID for authentication procedure, and OTP, which is sent through an SMS message, acts as 2FA to verify the account. In the earlier version of the application, it had access to SMS messages that was revealed in a security event that stated it would use smartphone's part of IMEI number as the password.

The basic idea of authentication procedure is that short messages are continuously transmitted in the background as HTTPS request that can be observed in Figure 7.5, and WhatsApp authentication server is listening for all incoming messages. In case if no SIM card or mobile number is entered, WhatsApp will try several attempts to ask the user to enter a mobile number. As the mobile number is entered a regular SMS message is sent to the mobile number provided which contains the OTP. The user enters the received OTP in the application; verifies it and the account is registered. WhatsApp application can be used on both devices PC and smart-phone simultaneously. To use the service simultaneously, the mobile internet should be activated and connected. Over past many years, researchers have studied the security mechanism of this applications and found that it was not a trusted application to be used. The messages send through this application were not encrypted and can be easily readable by the humans at the server side[7]. But in recent years they are using secure encryption standards, and the messages are encrypted during the transmission process. Still in some countries like India, encryption standards are not accustomed to a great extent and fights for *Net Neutrality* but, if used the government wants to read the private messages of the users[8]. The type of attacks can be conducted while authentication or registration process are explained in Section 7.6.4.

7.6.2 Viber

Viber is also one of the popular instant messaging application that provides video and voice calling over the internet connection that acts as a VOIP application. It does not require the telecoms network for using its services. The authentication procedure is same as WhatsApp, but it includes a feature of automatic callback also known as speech synthesizer to receive the code (OTP) by a telephone call. Viber application can be used simultaneously on various systems such as PC, smart-phones, and tablets, unlike WhatsApp. Since the Viber application vendor has not disclosed the source code and claims that it uses encryption standards as mentioned by the authors in the Viber Communication Security report[9]. But initially it did not have high encryption standards. The attacks performed during authentication process are explained in Section 7.6.4.

7.6.3 Facebook Messenger

Facebook is one of the biggest social networking website used by everyone worldwide. As years passed, many developments were noticed by Facebook. Most recent significant development was to develop the instant Facebook Messenger for chat, voice, and video call function. It can be used in both the systems simultaneously PC and smart-phones. To use these functions the user has to register his mobile number with the application to us on smart-phone, that has the same procedure of the above two applications discussed. It also uses the 2FA mechanism to authenticate the user identity. This is the standard procedure followed by every application vendor nowadays to authenticate the user by registering his mobile number and following the mechanism of OTP as discussed earlier. Another interesting security issue was that previous version of application would demand grant permission to read the incoming SMS messages. If the user does not grant the permission, the application will not proceed to install.

A serious side-effect of this applications is that it access the address book of smart-phone with Facebook friends to verify the users using the same application. It allows to import the friends list from Facebook profile and contacts from smart-phones to store them on the server. This can be fatal when a user installs a malicious application that can access more permissions than accessing address book from the smart-phone. The privacy of the user is compromised. The fraudulent application could install a Malware. Since the world is moving towards a digital age where all the things are to be performed by smart-phones and if the smart-phone is hacked through a malware

[7] See http://thehackernews.com/2014/12/hackers-can-read-your-private-sms-and.html

[8] See http://thehackernews.com/2015/09/national-encryption-policy.html

[9] See http://www.academia.edu/5717224/Viber_Communication_Security_unscramble_the_scrambled_Contents

then there would be consequences like the confidential and private information of the user stored on the smart-phone is compromised.

7.6.4 Threats to SMS in Instant Messaging Applications

General ways of performing attacks on SMS messages are discussed in Chapter 5, but during the registration of the account on smart-phones for application there are various ways for an attacker to hack the account to capture the OTP through SMS messages that are described below.

1. **Account Hijacking**: It is the method to hijack a user's account when the OTP send from the server through an SMS message to verify the mobile number provided by the user. During this process, the attacker can hijack the user account by-passing a fake mobile number instead of the legitimate user. This attack was possible in the early implementation of WhatsApp application as the OTP, which was supposed to be generated by the server and send to the user was generated at the user side itself [39]. Now, the OTPs are generated at the server and send an SMS message to the user to verify the mobile number provided by the user.

2. **ID Spoofing**: This type of attack is possible with account hijacking. Once the account is hijacked the attacker can spoof the sender ID. The attacker can also send malicious messages by manipulating the message. This attack is performed after the registering the account during the regular send and receive message transmission.

3. **Enumeration**: Most of the mobile messaging applications once registered access the address book and location services of the user. It compares all the entries in the list with the server and returns the list of users who are using the same application [39]. By providing access to the address book, the main problem is that the attacker gets access to the all the mobile numbers and send out phishing messages or carry out a DoS attack.

7.7 Invalidated OTPs

An interesting topic to be discussed related to the invalidation of OTPs. While setting up the process for 2FA on smart-phones, the user request to register the mobile number to use the application. After receiving the OTP on the smart-phone, the user does not wish to proceed to finish the account setup and quits the process. Again after few days the user tries to complete the same process, requesting again for OTP. We noticed that the same OTP is generated again for the same mobile number. This weakness of the OTP could be exploited by an attacker to capture the generated OTP and prevent the user from submitting it back to the service provider. It means that the captured OTP remains validated unless it's not used. This attack is a type of MITM attack. This vulnerability was noticed in Facebook Messenger mobile application.

7.8 Bypass Mobile 2FA

In this scenario, there is a possibility to bypass the 2FA from mobile devices, when the OTPs are generated on the server side and use OOB channel to deliver the code through SMS message. The prototype attack performed by A. Dmitrienko et al. [13] against SMS based OTP and bypass 2FA for current service providers is explained below.

Direct OOB channel is the standard OOB channel used to deliver the OTP through SMS message. This type of service is used by banks and online service providers to provide mTAN and OTP to its users during authentication and verification process. To bypass mobile 2FA; first the malware has to capture the login credentials entered by the user on the web site and intercept these details before they are submitted to the server. Second, the malware will also intercept the

SMS messages from the mobile phone number registered to the account. The attack is performed by A. Dmitrienko et al. [13] which uses DLL injection[10] to inject a library function call to intercept the data on the HTTPS protocol. The malicious library function is added that acts a man-in-the-browser attack to the web-browser to redirect the internet traffic and capture the data by the attacker. Also, a mobile malware that acts as an MITM between GSM and smart-phone that intercepts the SMS messages and forwards it to the attacker.

[10] See `http://securityxploded.com/dll-injection-and-hooking.php`

Chapter 8

Enhancing the security for SMS based Authentication

In this chapter, we discuss the potential countermeasures to enhance the security for SMS based Authentication Schemes against the vulnerabilities and attacks mentioned earlier in Chapter 4 and 5 for SMS service. We strived to investigate and cluster the reasonable approaches and methods suggested by academic researchers working towards the development and security for SMS. The methods that describe below are categorized into two sections namely Basic and Advanced measures to improve the security and support the service levels for the network operators and smart-phone OS manufacturers.

8.1 Basic Measures to Enhance the SMS Security

8.1.1 Securing the backbone traffic

The basic measure to enhance the security of the SMS is to secure the network traffic. The transmission of the data between various protocol layers and network entities should be secure using better encryption methods. The attacker should not be able to eavesdrop, modify or insert fake data during the transmission process. Certain security specifications are also inclined towards the hardware system of the network entities to protect the data from the attackers and implementing better secure algorithms A3/A8 and ciphering techniques A5/3 to encrypt the data to avoid the fraudulent activities for SMS messaging [42].

8.1.2 Securing the OOB Channel

OOB is a dedicated channel used for transmission of the SMS messages currently apart from the primary in-band voice channel. To make a proper use of the SMS service for OTP and mTAN the service providers must make use of a secure OOB channel that is separate from the standard OOB channel. By making use of such a secure channel for mobile devices, the secure OOB channel will avoid third-party applications to access the secure channel thereby reducing the chances to get the mobile device compromised. As suggested by A. Dmitrienko et al. [13] the secure OOB channel is easier to use when the mobile devices have a fixed location. But since mobile devices are movable implementing a dedicated secure OOB channel for service providers is a bit challenging.

8.1.3 Early detection of Malicious Mobile Apps

A primary step to be considered for malicious mobile applications, as most of the time, they steal the private and confidential information of the user. As already mentioned in Chapter 6 about various smart-phone OSs signifies that the malicious applications depend on the permission granted by the user. Various methods should be used to test the application such as static and dynamic analysis to detect the malicious set of permissions included by the vendor. Static analysis is used to identify the malicious set of permissions during compile time, and dynamic analysis is used during the run-time testing process. Make use of tainted data methods to identify the information leakage within the application and correctly to define the information flow policies [16]. It also helps to check if there are any changes made to the kernel extensions. Also, behavioral analysis is used to detect SMS receivers that utilizes or forwards the SMS messages received on the smart-phone [13].

8.2 Advanced Measures to Enhance the SMS Security

C.Mulliner et al. has carried various research related to SMS OTP [33] and explained different advanced methods to defend the attacks in his investigation. The solutions that are provided requires the support of mobile cellular services, mobile OS manufacturers and network operators. In addition to these solutions, they proposed two design methodology that should work independently from the standard services. We will discuss all the methods and approaches in the following section.

8.2.1 SMS end-to-end encryption

This type of advanced measure is used to maintain the confidentiality and integrity of the SMS, which contains OTP and mTAN codes. The idea of this method is based on *application private storage* that is available on all mobile platforms today, described by the researcher. This method depends on encryption methods for OTP messages that are transmitted to users, and a dedicated application is required to decrypt the SMS messages using a secret key by the user. The researcher has provided a novel idea of encryption the OTP messages and to protect against SMS OTP Trojan that can access the standard SMS messages. But it would require a key to decrypt the OTP message. The researcher just provides an idea and not specifying the functionality of how the solution will be achieved as it requires the user must maintain the secret keys.

8.2.2 Virtual dedicated channel for handset

Trojans are the major threat to smart-phone SMS and quickly perform attacks on them. To protect against Trojan attacks on SMS messaging containing OTP and mTAN codes, a minimum support from mobile OS manufacturers is required. The method requires creating a dedicated virtual channel inside the smart-phone OS that will be used to deliver the SMS messages containing only OTP and mTAN. All the messages sent through this channel will be secured against local interception of messages. This channel will only be used to receive the OTP and mTAN messages on the smart-phones and will store these messages in separate mobile storage called *application private storage* which cannot be read by any third-party applications. The name suggested to the "application private storage" is *OtpMessages*. The solution just requires simple modification in the routing of the messages depending upon the content of the messages through changes in the mobile OS. Nowadays, the online service providers have adapted to use 2FA mechanism and the users receive OTPs from them during account login, and they have dedicated names or message title when we receive a message such as GOOGLE, FACEBOOK, MSFT for Microsoft and many more. For these dedicated names, a separate virtual dedicated channel must be provided.

8.2.3 SMS Port-Based Channel

The design approach suggested can be used to implement particular SMS ports for OTP messages. These ports are similar to TCP/UDP ports that are used such as SMTP:25, HTTP:80, HTTPS:443. In above section, we discussed dedicated virtual channel that is similar to SMS port number concept. The idea is to dedicate a specific port number for the SMS OTP messages. During the delivery of SMS OTP messages, the BTS will listen to *OtpMessage* application port, and if it responds, the messages will be delivered to this port. To apply such a secured process, the mobile OS must support the signed applications and SMS message routing based ports. Modification in the OS is required to support for SMS port implementation and online service providers that make use of 2FA must need to know if the particular mobile device supports the dedicated virtual channel. Making use of this approach, it will help to reduce the Trojan attacks on SMS messages as it cannot bind to this support because it will require OS assistance.

8.2.4 Message filtering on Channel

The second design approach suggested only require changes in smart-phone OS. As the name of this approach is message filtering, a filtering technique will be needed to filter the OTP messages from the regular SMS messages. The filter inspects every incoming SMS messages and decides if it is an OTP message or a standard SMS message. If it detects it is an SMS OTP message, it will filter and route on virtual dedicated channel receiver and forwarded to *OtpMessage* application, else default SMS path is used. For filtering purpose the researcher has developed two methods:(i)Keyword based filter that matches with a set of words against the contents or title of the message such as GOOGLE, FACEBOOK, MSFT and many more.(ii)Sender-based filter that matches the originator address of an SMS message.

Apart from the methods and approaches mentioned above various methodologies have been suggested by many researchers to enhance the security of SMS messages such as applying cryptographic techniques like Caesar Cipher with One-Time pad, Elliptic Curve, RSA at the application layer rather than modifying the smart-phone OS. L.Koot [29] have also performed risk analysis for SMS based mTAN attacks on smart-phones and his investigation states various classifications for SMS mTAN attacks. In the case of a mobile banking application, the precautions that must be considered by the user during the installation process such as granting or denying permissions and detecting vulnerabilities in the smart-phone OS are explained. Also after investigating various attack vectors, a proper classification of countermeasures to be considered by users, mobile OS manufacturer, and service providers are provided that will help to defend against the attacks. Since the size of the payload is not utilized fully as the message containing data of OTP or mTAN is not more than 10 byte, and the rest of the size remains unused. A more secure way to utilize the unused size is to include security parameters such as signatures and timestamps to the message and transmit to the user that will help to avoid Replay attacks explained in Section 5.8.

Chapter 9

Conclusion

In this thesis, we have first described the technical description of the SMS followed by all possible vulnerabilities and attacks related to SMS. The fact that many technologies emerge from GSM and to achieve an efficient and effective results we focused on SMS service. It was necessary to gather all the relevant information to understand the technology and describe it. Besides, the research also focused on 2FA nowadays used by financial sectors, smart-phone applications, and Internet-based service providers. This thesis provides a summarized findings of all the possible vulnerabilities and attacks in SMS technology and might help the other academic researchers as a starting point to perform further empirical investigations.

We have examined and discovered that the fundamental security issues related to authentication procedure within the GSM network, from BTS to MS which is a one-way process and uses a weaker form of encryption. Message transmission between different SS7 layers and vulnerabilities related to MS such as OS and its applications are addressed. Various threats such as modification, intercepting of messages during OTA transmission and spoofing of messages within the network entities are discussed that compromises the user's data.

Furthermore, we studied the security for 2FA schemes that have received significant attention and deployed for online banking services and login purpose for additional security. Explaining the importance and usefulness of providing additional security such as 2FA which uses SMS service to deliver OTP and mTAN on mobile devices; we have also pointed out some weaknesses in 2FA by compromising the security of the PC and mobile at the same time by providing an attack tree model. Cross-platform infection is also an another form of attack that could infect two things at the same time.

Also, we discussed various smart-phone OS security features for Android, iOS, and Windows Mobile and explained recent attacks performed at OS level. Different approaches are described that should be considered to protect the OS and information flow within the applications of the smart-phone. We further discussed three case scenarios - Google Authentication, ING Bank and most commonly used instant messaging applications like WhatsApp, Viber and Facebook Messenger where 2FA is utilized and to what extent the security is guaranteed. Various forms of attacks on 2FA methods for stealing the OTP and mTAN codes are explained which compromises the user's personal data in these scenarios. The attacks that are listed are still being performed but not on a large extent as nowadays security is hardened with strong encryption and authentication algorithms.

Finally, we summarize potential countermeasures for SMS based authentication schemes that are suggested by researchers that help to provide secure SMS transmission. Collin Mulliner does a significant contribution by providing advanced measures to enhance SMS security and performing various vulnerability analysis framework for SMS implementations on various smart-phones OS and GSM technology. All the information presented in this thesis can be a source of encouragement for further research in the area of SMS which is being adopted in every sector rapidly because of its low-cost and dynamic nature.

Bibliography

[1] 3GPP TS 23.040 version 4.2.0 Release 4. http://www.etsi.org/deliver/etsi_ts/123000_123099/123040/04.02.00_60/ts_123040v040200p.pdf. Accessed: 26-10-2015. 7

[2] Digital cellular telecommunication system (Phase 2+); Point-to-Point (PP) Short Message Service (SMS) support on mobile radio interface (GSM 04.11). http://www.etsi.org/deliver/etsi_gts/04/0411/05.01.00_60/gsmts_0411v050100p.pdf. Accessed: 15-10-2015. 15

[3] Digital cellular telecommunications system (phase 2+); Mobile Station - Base Station System (MS - BSS) interface; Channel structures and access capabilities (GSM 04.03). http://www.etsi.org/deliver/etsi_gts/04/0403/05.01.00_60/gsmts_0403v050100p.pdf. Accessed: 27-10-2015. 22

[4] iOS Security Guide, iOS 8.3 or later. http://www.apple.com/business/docs/iOS_Security_Guide.pdf. Accessed: 30-09-2015. 36

[5] Alex Biryukov, Adi Shamir, and David Wagner. Real time cryptanalysis of A5/1 on a PC. In *Fast Software Encryption*, pages 1–18. Springer, 2001. 27, 29

[6] Valer Bocan and Bocan Cretu. Threats and countermeasures in GSM networks. *Journal of Networks*, 1(6):18–27, 2006. 23

[7] Joany Boutet and Lori Homsher. Malicious android applications: Risks and exploitation. *SANS Institute*, 22, 2010. 34

[8] Marc Briceno, Ian Goldberg, and David Wagner. An implementation of the GSM A3A8 algorithm. *Unpublished report*, 1998. 5

[9] Neil J Croft and Martin S Olivier. A silent SMS denial of service (DoS) attack. *Information and Computer Security Architectures (ICSA) Research Group South Africa*, 2007. 29

[10] Adrian Dabrowski, Nicola Pianta, Thomas Klepp, Martin Mulazzani, and Edgar Weippl. IMSI-catch me if you can: IMSI-catcher-catchers. In *Proceedings of the 30th Annual Computer Security Applications Conference*, pages 246–255. ACM, 2014. 28

[11] Lucas Davi, Alexandra Dmitrienko, Christopher Liebchen, and Ahmad-Reza Sadeghi. Over-the-air cross-platform infection for breaking mTAN-based online banking authentication. *BlackHat Abu Dhabi*, 2012. 47

[12] Lucas Davi, Alexandra Dmitrienko, Ahmad-Reza Sadeghi, and Marcel Winandy. Privilege escalation attacks on android. In *Information Security*, pages 346–360. Springer, 2011. 34, 48

[13] Alexandra Dmitrienko, Christopher Liebchen, Christian Rossow, and Ahmad-Reza Sadeghi. On the (in) security of mobile two-factor authentication. In *Financial Cryptography and Data Security*, pages 365–383. Springer, 2014. 37, 50, 51, 53, 54

[14] Himanshu Dwivedi. *Mobile application security.* Tata McGraw-Hill Education, 2010. 33, 36

[15] Manuel Egele, Christopher Kruegel, Engin Kirda, and Giovanni Vigna. Pios: Detecting privacy leaks in iOS applications. In *NDSS*, 2011. 48

[16] William Enck, Peter Gilbert, Seungyeop Han, Vasant Tendulkar, Byung-Gon Chun, Landon P Cox, Jaeyeon Jung, Patrick McDaniel, and Anmol N Sheth. Taintdroid: an information-flow tracking system for realtime privacy monitoring on smartphones. *ACM Transactions on Computer Systems (TOCS)*, 32(2):5, 2014. 54

[17] William Enck, Machigar Ongtang, and Patrick McDaniel. Understanding android security. *IEEE security & privacy*, (1):50–57, 2009. 34, 48

[18] Michael D Ernst, René Just, Suzanne Millstein, Werner Dietl, Stuart Pernsteiner, Franziska Roesner, Karl Koscher, Paulo Barros Barros, Ravi Bhoraskar, Seungyeop Han, et al. Collaborative verification of information flow for a high-assurance app store. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 1092–1104. ACM, 2014. 34, 48

[19] GSM ETSI. 03.40. *Digital cellular telecommunications system (Phase 2+)*. 1

[20] TS ETSI. 123 040 3GPP ts 23.040 version 4.2. 0. 14, 15

[21] Michael Harrington. Understanding SMS: Practitioner's basics. *CFCE, EnCE*, 2008. 20

[22] Gunnar Heine. The air-interface of GSM. *GSM Networks: Protocols, Terminology, and Implementation,(MA: Artech House, Inc.)*, pages 89–100, 1999. 6

[23] Brinio Hond. Fuzzing the GSM protocol. *Master's thesis, Radboud University Nijmegen, Kerckhoff's Master, The Netherlands*, 2011. 31

[24] Ikechukwu Ibekwe and Salem Aljareh. SMS security: highlighting its vulnerabilities & techniques towards developing a solution. 2012. 22

[25] GSMA Intelligence. Understanding 5G: Perspectives on future technological advancements in mobile. *GSMA Intelligence Understanding 5G*, pages 3–15, 2014. 9

[26] Eran Kalige, Darrell Burkey, and IPS Director. A case study of eurograbber: How 36 million euros was stolen via malware. *Versafe (White paper)*, 2012. 35

[27] Veena K Katankar and VM Thakare. Short message service using SMS gateway. *International Journal on Computer Science and Engineering*, 2(04):1487–1491, 2010. 6

[28] Engin Kirda and Christopher Kruegel. Protecting users against phishing attacks. *The Computer Journal*, 49(5):554–561, 2006. 30

[29] Laurens Koot. Security of mobile TAN on smartphones. *A risk analysis for the iOS and Android smartphone platforms. Master's thesis, Radboud University Nijmegen*, 2012. 55

[30] Gwenaël Le Bodic. *Mobile Messaging technologies and services: SMS, EMS and MMS*. John Wiley & Sons, 2005. xixi, 12, 13, 14

[31] Steve Lord. Trouble at the telco: when GSM goes bad. *Network Security*, 2003(1):10–12, 2003. 30

[32] G Lorenz, T Moore, G Manes, J Hale, and S Shenoi. Securing SS7 telecommunications networks. In *Workshop on Information Assurance and Security*, volume 2, pages 273–278, 2001. 23

[33] Collin Mulliner, Ravishankar Borgaonkar, Patrick Stewin, and Jean-Pierre Seifert. SMS-based one-time passwords: attacks and defense. In *Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 150–159. Springer, 2013. 23, 54

[34] Collin Mulliner and Charlie Miller. Fuzzing the phone in your phone. *Black Hat USA*, 25, 2009. 31

[35] Collin Mulliner and Charlie Miller. Injecting SMS messages into smart phones for security analysis. In *USENIX Workshop on Offensive Technologies (WOOT)*, 2009. 29

[36] David Perez and Jose Pico. A practical attack against GPRS/EDGE/UMTS/HSPA mobile data communications. *Black Hat DC*, 2011. 26

[37] Saurabh Samanta, Radhesh Mohandas, and Alwyn R Pais. Secure short message peer-to-peer protocol. *International Journal of Electronic Commerce Studies*, 3(1):45–60, 2012. 6

[38] Peter Schartner and Stefan Bürger. Attacking mTAN-applications like e-banking and mobile signatures. Technical report, Technical report, University of Klagenfurt, 2011. 47

[39] Sebastian Schrittwieser, Peter Frühwirt, Peter Kieseberg, Manuel Leithner, Martin Mulazzani, Markus Huber, and Edgar R Weippl. Guess who's texting you? evaluating the security of smartphone messaging applications. In *NDSS*, 2012. 50

[40] Hemant Sengar, Ram Dantu, Duminda Wijesekera, and Sushil Jajodia. SS7 over ip: signaling interworking vulnerabilities. *Network, IEEE*, 20(6):32–41, 2006. 23

[41] Stefania Sesia, Issam Toufik, and Matthew Baker. LTE–the UMTS long term evolution, 2001. 7

[42] Mohsen Toorani and A Beheshti. Solutions to the gsm security weaknesses. In *Next Generation Mobile Applications, Services and Technologies, 2008. NGMAST'08. The Second International Conference on*, pages 576–581. IEEE, 2008. 53

[43] Patrick Traynor, William Enck, Patrick McDaniel, and Thomas La Porta. Mitigating attacks on open functionality in sms-capable cellular networks. *Networking, IEEE/ACM Transactions on*, 17(1):40–53, 2009. 22

[44] Fabian van den Broek, Brinio Hond, and Arturo Cedillo Torres. Security testing of GSM implementations. In *Engineering Secure Software and Systems*, pages 179–195. Springer, 2014. 31

[45] Fabian van den Broek, Roel Verdult, and Joeri de Ruiter. Defeating IMSI catchers. 28

[46] Rakesh Verma, Deepak Singh Tomar, and Shashi Kant Rathore. Extraction and verification of mobile message integrity. In *Communication Systems and Network Technologies (CSNT), 2011 International Conference on*, pages 49–53. IEEE, 2011. 16, 17