NTNU
Norwegian University of
Science and Technology

Proposal and Implementation of An IDS for Potential SMS Spam Signaling Messages on SS7

Paul Ntim Yeboah

Master of Telematics - Communication Networks and Networked Services
Submission date: August 2016
Supervisor: Van Thanh Do, ITEM
Co-supervisor: Thanh Nguyen Hai, Telenor-Trondheim

Norwegian University of Science and Technology
Department of Telematics

NTNU
Norwegian University of
Science and Technology

Paul Ntim Yeboah

Proposal and Implementation of An IDS for Potential SMS Spam Signaling Messages on SS7

Master of Telematics - Communication Networks and Networked Services.
Submission Date: 5 August, 2016.
Supervisors: Prof. Van Thanh Do, ITEM.
Dr. Hai Thanh Nguyen, Telenor-Trondheim.

Thesis Project of 30 Credit Points.
Norwegian University of Science and Technology.
Department of Telematics.

Proposal and Implementation of An IDS for Potential SMS Spam Signaling messages on SS7

Paul Ntim Yeboah

Problem Description

Reports on spam activities reaching telecom operators reveal flaws in routing for SMS on the Signaling System 7 network. This thesis consists of the following tasks:

- A comprehensive Study of vulnerabilities on the SS7 network.
- Proposal and implementation of an intrusion detection system to mitigate signaling for SMS spam on the SS7 network.
- Evaluation of the proposed detection method.

Assignment given: 11th March, 2016

Supervisor: Professor Van Thanh Do, ITEM.

ABSTRACT

The signaling system no. 7 (SS7) network has been the driving force of the telecommunication network. SS7 relieves the mobile network from its ever growing advanced services through the provision of robust and sophisticated signaling services. Being a closed network, the SS7 network has been in castle walls as very few telecom operators had access to the network. However, current alleviations in laws and regulations governing the market of the SS7 network and attempts of merging the network for the appropriate interoperability with other networks (such as IP) has led to a wall-less state of the network's castle. Likewise, telephone mobility and advanced telecom services have added a number of threats to the network's security. Without an initial cryptographic security mechanism to authenticate signaling nodes, the SS7 network is currently vulnerable to abuse from signaling messages which were created to facilitate subscriber mobility. Such signaling messages meant to maintain mobility on the SS7 network are exploited to track mobile subscribers, intercept calls, send unsolicited SMS text messages and deny services to legitimate subscribers. Advanced services including short message service (SMS) and intelligent services provided by SS7's customized application for mobile networks enhanced logic (CAMEL) have added additional vulnerabilities to the network. In this study, a comprehensive discussion of the threats facing the various SS7 network elements is given. For the three fundamental SS7 network nodes, a description of the various corresponding entry points as well as the kind of vulnerabilities they breed on the network are discussed. An intrinsic vulnerability posed by SMS and CAMEL's signaling architectures is also described in this study.

SMS is a widely exploited data application on the telecommunication mobile network. Recently, SMS has been a resort for two-factor authentication for many online business services. However, reports of fraudulent activities reaching telecom operators reveal deficiencies in the SMS's architecture for signaling. One focus of this thesis aims at proposing an intrusion detection system (IDS) to mitigate signaling for SMS spam on the SS7 network. The proposed detection method provides a near duplicate detection method on similar volumes of signaling messages sent simultaneously over a short period of time in request for SMS service. The IDS deploys a space efficient data structure algorithm called Counting Bloom Filters to record the appearance of SMS signaling message features. Test results shows that by using counting bloom filters with the proposed detection method, detection rate on SMS spam signaling messages can reach 100%.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to Prof. Van Thanh Do for the help and supervision of this thesis work. Am also grateful to Dr. Hai Thanh Nguyen and Kritoffer Jensen for the support and the provision of the necessary tools for the thesis. My final appreciation goes to my family and friends for their motivations and blessings.

Trondheim, Norway, August 2016
Paul Ntim Yeboah

Preface

This thesis report is submitted in fulfillment of a requirement for Master's degree in Telematics at the Norwegian University of Science and Technology. Many thanks goes to the department of Telematics (ITEM) for giving me the chance to partake in this thesis work.

Trondheim, Norway, August 2016
Paul Ntim Yeboah

Table of Contents

List of Figures

Chapter 1

Introduction

The early telecommunication network provided very few services to subscribers, mostly call services. End users were compelled to be stationary since the network consisted of telephones with fixed location residing in places likes offices. Today, wireless technologies have overridden the previous stationary state of telephones, transitioning telephone devices to a mobile state. Mobile devices introduce additional services including location management and subscriber authentication. The former provides a means to enable the mobile device update the network with its present position, whereas the latter enables the network confirm the identity of the subscriber. Beside mobility services, there are other advanced services such as Short Message Services (SMS), prepaid billing and no prefix dialing (intelligent service) which the telecommunication network provides. Subscriber mobility and the newly developed advanced services provided by telecommunication network have added complexity to the network's operations.

Technically, the telecommunication network comprises of two separate networks. One part of the network handles voice and data traffic whiles the other part serves as a control network. Even though a layperson views the network as a single network controlling all activities, intrinsically, there exists another network, namely the core or signaling network. This core network is perceived to be the driving force of the telecommunication network.

To control such a complex network, telecommunication networks employ a standard signaling system known as the Signaling System No. 7 (SS7). The SS7 network is widely deployed on telecommunication networks to relief the network from its ever growing services. SS7 enables nodes on the telecommunication network to exchange vital information pertaining to location management, subscriber authentication, SMS, and other intelligent services. However, like every other network, performance efficiency is a concern for both the network service providers and subscribers. One area of performance which has caught attention in recent times is the network's security. Issues of security on the SS7 network have been of keen interest to researchers these days, since the network has been a carrier of sensitive information for the telecommunication network. One part of this thesis is concerned with shedding light at the vulnerabilities and attacks on the SS7

network. In view of that vulnerabilities and threats arising from both the network's architecture and newer services are thoroughly elaborated in this dissertation.

The messaging service which is believed to be the most widely exploited data application on the telecommunication network is the SMS. Currently the SMS application has been the resort for two-factor authentication for many online shopping and banking services. However, recent reports of fraudulent activities reaching telecom operators reveal deficiencies in the SMS architecture. This deficiency is described in section 3.2.3 as "an intrinsic Vulnerability from SMS Routing". The routing flaw in sending SMS enables attackers to flood the SS7 network with illegitimate signaling messages resulting in SMS spamming. In this study, we propose a network based intrusion detection system (NIDS) to raise alarm of such unsolicited signaling messages on the SS7 network, see Figure 1.1 below.

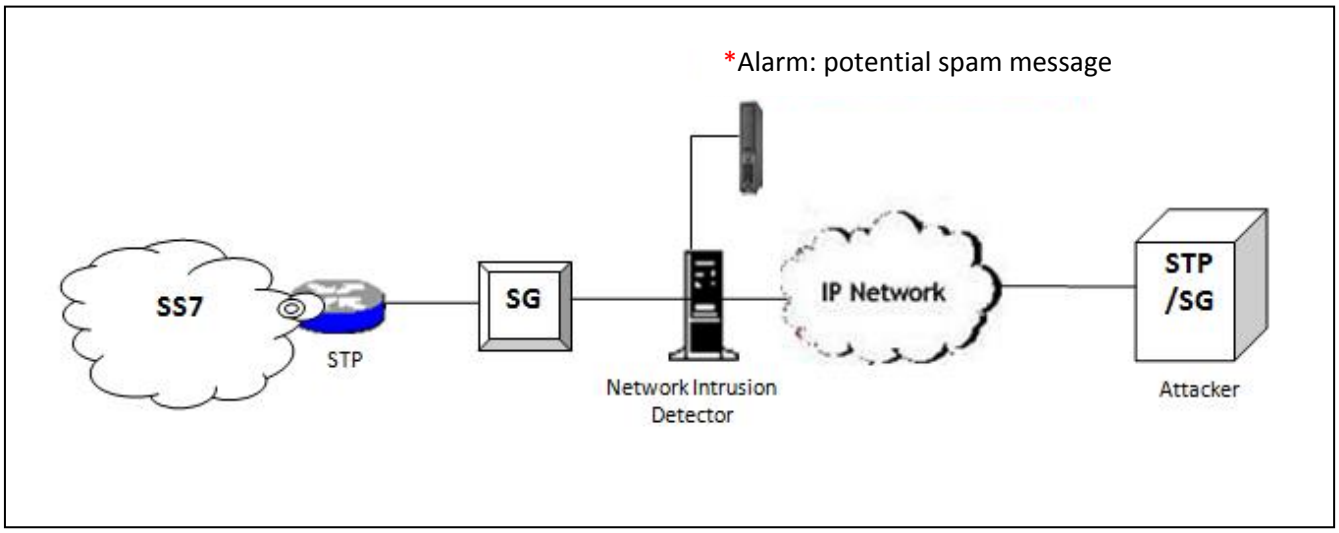


Figure 1.1: A NIDS for SMS spam on SS7-over-IP network.

The IDS can be deployed on both traditional SS7 network and SS7-over-IP network (as shown in Fig 1.1). This thesis focuses on proposing and implementing a mechanism to detect similar signaling messages that are repeatedly sent in high volume in a short period of time over the SS7 network in request for short message service .If such signaling messages are encountered on the network, the IDS report as potentially spam messages. A further step in the thesis shows the evaluation of the efficiency of the proposed IDS to detecting spam signaling messages on the SS7 network.

1.1 Aim of Thesis

This thesis is a supplement to other recent research made on the security of the signaling system no.7 network. The project consists of two major tasks:

- Investigate on the vulnerabilities and threats arising on the SS7 network and their effect on the telecommunication network. This consists of a thorough analysis of vulnerabilities sourcing from the SS7 network architecture itself and their threats to the telecommunication network. Other perceived sources of vulnerability on the SS7 network elaborated in this thesis are those vulnerabilities arising from the newly created advanced telecommunication services.

- The second task consists of a proposal and implementation of a network based intrusion detection method for SMS spam signaling messages. An evaluation of the efficiency of the proposed method to detecting SMS spam signaling messages is presented as well.

1.2 Contribution

The thesis gives a comprehensive study of the vulnerabilities and attacks on the SS7 network. In view of that, an explanation on how each signaling node on the SS7 network is exposed to threats is given. In addition, a discussion presented in this thesis shows how attackers abuse legitimate signaling messages pertaining to newly developed advanced mobile services to perform attacks including call interception, tracking, denial of service (DOS) and SMS spamming.

The Thesis further shows a proposal and implementation of a NIDS for SMS spamming on the SS7 network. The proposed scheme detects near-duplicate SMS signaling messages repetitively sent on the SS7 network. One requirement for the NIDS is that a high volume of signaling message for SMS should be stored over a period of time. We show how such bulky information (Big data) is stored without any hindrance. Many other projects on SMS spam detection use only the SMS text as input parameter for their machine learning, however, in this project, we describe how additional parameters such as the address of the originating SMS center (SMSC) can be added to the input parameters to improve detection accuracy. And latter an evaluation of performance for the IDS to detecting SMS spam signaling messages is shown.

Mostly, telecommunication operators rely on SMS spam reports from subscribers to detect spamming activities on their network. The proposed IDS in this thesis is a proactive approach to detecting SMS spamming activities on the SS7 network, thus detecting and reporting potential SMS spam messages before they actually occur.

1.3 Simulation and Development Environment

To have samples of SMS signaling messages, a simulation with Mobicent JSS7 stack was performed. Mobicent JSS7 stack deploys the SS7 protocol stack in Java. Simulation was done in a virtual machine (VMware workstation) on Ubuntu version 12.04. Additionally, the proposed IDS was developed and tested in Java Eclipse Mars.2 IDE.

1.4 Outline

The structure for this thesis is as follows. Chapter 2 gives background information about the Signaling System 7. An overview of the SS7 network architecture and the protocol stack is presented. Chapter 3 presents the vulnerabilities of the SS7 network. The various SS7 entry points are listed with their associated vulnerabilities. In addition, vulnerabilities arising from mobility and advanced mobile services are shown. Chapter 4 contains a proposal of a detection method for potential SMS spam signaling messages. In chapter 5, results and evaluation of detection rate on the proposed detection method is presented. The final chapter gives a summary of the entire thesis work with the obtained results.

Chapter 2

Background Information

2.1 Telecommunication and Signaling

Antonio Meucci is perceived to have developed the first device which enabled voice transmission over electric lines in 1849 [1]. Meucci's device which was intended to connect His bedroom to His office was revealed in 1860 by the italienskspråkig newspaper in the United State [1, 2]. Due to financial shortcomings, Meucci could not meet the application fee to patent His device. Two others, Elisha Gray and Alexander Graham Bell are believed to have developed a similar voice transmission device in the same era [1, 2]. Whiles Gray improved on His device, Bell succeeded to patent the conventional telephone in the year 1876 [2]. Telephony has undergone constant development ever since.

The early telecommunication system required that the communicating devices are connected directly to each other with cables. Interconnecting telephones with every other telephone creates a full mesh topology network [3]. Mesh topology network is expensive and tedious to maintain, hence a more sophisticated approach to connecting telephones was demanded. The demand led to the use of switches for interconnecting telephones. In 1878 at New Haven Connecticut, the first manual switch was demonstrated to exchange voice message [3]. The introduction of switches made it possible to connect a single cable to a telephone and initiate communication to other telephone users.

In the former days, signaling in telecommunication was manually implemented. The subscriber sends a message signal which lightens up a bulb on the operator's side. The operator connects to the calling party's line to ascertain the called party number. Operator alerts the called party of an incoming call through a ringing signal. If the call is answered, both parties are connected via an operator's patch cord.

Today, telecommunication network has grown wide and telephones have become mobile. [4] shows that telecommunication or the Public Land Mobile Network (PLMN) in general is a network system embodied with three distinct subsystems. The *Mobile Station Subsystem* identifies the subscriber's mobile equipment to the network with a unique number *International Mobile Subscriber Identity* (IMSI). The *Base Station Subsystem* puts the subscriber on the network via a radio link. And finally a *Network Subsystem* which mainly comprises of switches and databases. For switching services, the Network subsystem employs a *Mobile Switching Centre* (MSC) to connect the network to other Public Switch Telephone Networks (PSTN). Four distinct databases are maintained in the Network subsystem: a *Home Location Register* (HLR) which keeps information of all subscribers registered on a particular operator network, a *Visitor Location Register* (VLR) which records data of users currently on a coverage area, an *Authentication Centre* (AUC) for keeping authentication algorithms and user keys and an *Equipment Identity Register* (EIR) which maintains a list of valid and blocked mobile devices. Additionally the network subsystem is strengthened by *Intelligent Networks* (IN) which adds more sophisticated services (such as call transfer, call forwarding and voice mail) to the standard telecommunication services.

To ensure proper interoperability of subsystems within and outside a PSTN, nodes are required to exchange control information between each other to enable subsystems share the necessary information required to facilitating telecommunication connections. In the telecommunication network, the exchange of control information is perceived as an act of signaling. Signaling to exchange control information is usually initiated from the Mobile Station subsystem to the Base Station Subsystem when a user request for a telecommunication service. The request is passed onto the Network Subsystem which triggers signaling between nodes usually with intentions to retrieve control data from the databases or update them. A typical example of signaling in the telecommunication network subsystem occurs in subscriber's movement; a mobile device which transitions to a new coverage area (MSC) is required to announce its new location to the network, this call for signaling between subsystems to perform a location update in the respective databases.

The exchange of control information between telecommunication subsystems as described earlier and the mobile nature of modern telephones and the additional services provided by intelligent networks places a demand for a robust signaling system to be used on the telecommunication network. One approach suggests that the signaling path should be separated from the data path, an approach known as the Common Channel Signaling (CCS) [5]. Based on the CCS protocol, the International Telecommunication Union (ITU) has developed a signaling system known as the SS7 [3, 6], which is presented in the next section.

2.2 Signaling System 7 Network Architecture

In 1957 AT&T developed a signaling protocol which is an implementation of the CCS standard to enable signals to be carried on a separate path from the voice channel [3, 7]. ITU in 1980 standardized AT&T's signaling protocol which today is deployed globally on the telecommunication network. ITU's signaling protocol, namely the Signaling System no.7 (SS7) protocol, was geared toward improving the efficiency of the network and ensuring proper utilization of resources. Being an Out of Band signaling protocol, the SS7 network allows telecommunication network nodes to exchange control information on a network channel other than the voice channel.

Initially, ITU's SS7 signaling network was intended to serve as a call associated signaling network, for setting up, maintaining and tearing down telephone calls on the PSTN network. This led to the development of some SS7 application protocols including Telephone User Part (TUP) and Integrated Services Digital Network User Part (ISUP) which are deployed in Europe and North America respectively [6]. Currently the network provides capabilities to carry non-call-associated and Non-facility Associated Signaling (NFAS) [6, 7]. The latter enables signaling whiles a call is still in progress. Non-call-associated services include SMS service, mobility management service and intelligent network service.

The SS7 network constitutes three essential signaling nodes as shown in Figure 2.1. These SS7 nodes which are referred to as signaling points (SP) are connected to each other by SS7 links. The signaling component that puts the mobile station on the SS7 network is the *Service Switching Point* (SSP). SSPs are switches which are loaded with software capable of originating, switching and terminating calls. Signaling messages which originate from the SSP contain routing information obtained from the *Service Control Point (SCP)* signaling point. SCPs are equipped with software capable of relaying routing information from telecommunication databases to other signaling nodes on the network. Usually SCPs receives database queries from SSPs requesting for information needed to facilitate the call processing.

The next signaling point is the *Signaling Transfer Point* (STP). STPs somewhat acts as intermediary between the SSP and SCP. They are sometimes referred to as routers on the SS7 network. STPs neither originates nor terminates SS7 signaling messages, instead routes incoming messages towards the specified destination. STPs are either deployed as standalone or integrated STPs [3]. Integrated STPs combines the role of both SSP and STP; generates database queries when acting as SSP and routes query messages to the SCP when acting as STP.

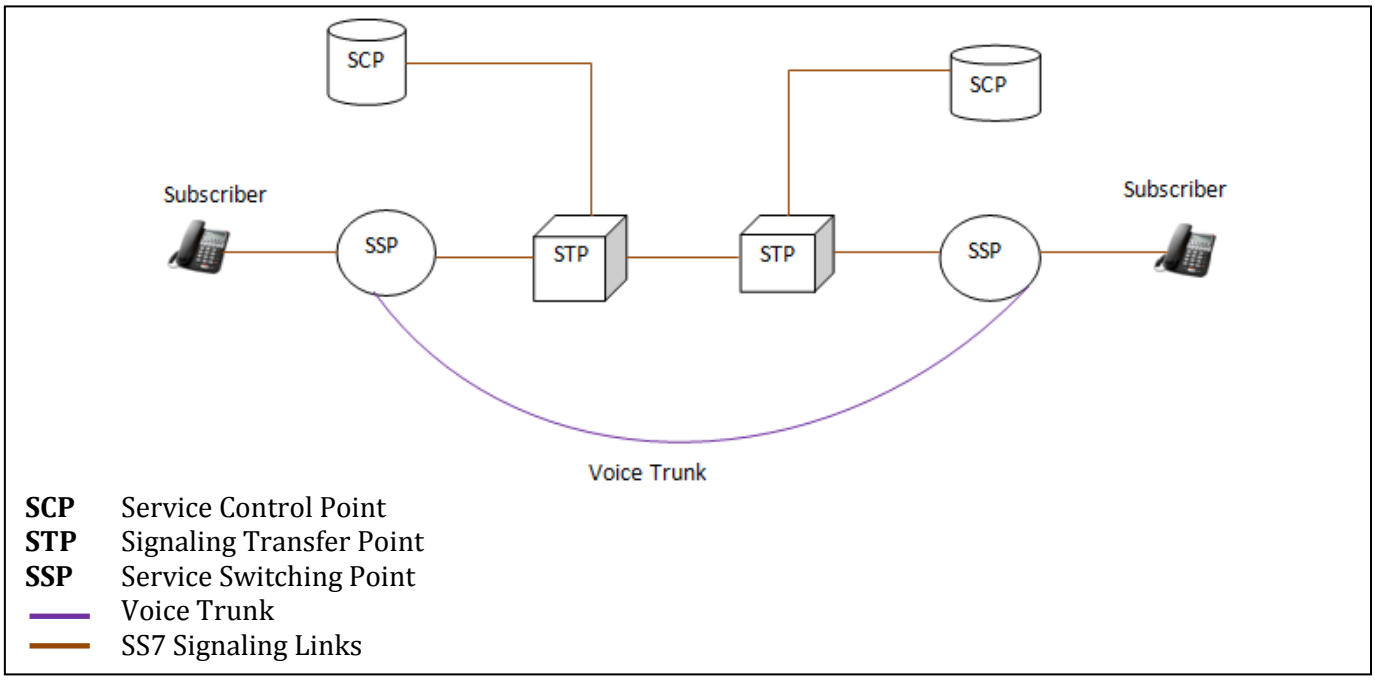


Figure 2.1: SS7 network component

2.2.1 Routing on SS7 Network

Each signaling node on the SS7 network is assigned a unique address called Signaling Point Code (SPC) to identify the node on the network. Two SPC address are found in every signaling message; address of the originating node and address of the destination node. These addresses are referred to as the Originating Point Code (OPC) and the Destination Point Code (DPC). DPC is the key element for routing messages on the SS7 network whiles OPC helps trace the origin of the signaling message.

The SS7 network imitates the addressing scheme deployed on the IP network. On the IP network, IP addresses are categorized into private and public addresses. Private addresses are common to every IP network service provider whiles public addresses are unique to each service provider. Similarly on the SS7 network, point codes are regarded as private addresses, allowing telecommunication network operators to assign similar point code addresses to signaling nodes residing on their SS7 network. However there are public addresses too on the SS7 network. These public addresses are referred to as Global Titles (GT). GT address uniquely identifies the SS7 network and exposes the SS7 network to the global SS7 network. Every signaling message contains a destination GT address which aids in routing the signaling message to the destination SS7 network and a DPC to identify the signaling nodes.

2.3 SS7 Protocol Stack

To ensure proper communication between signaling nodes on the SS7 network, there is the need to define systematic steps for every SS7 network to emulate. On the SS7 network, these steps are perceived as protocols which are layered on each other. This section introduces and elaborates the various protocols behind the functioning of the SS7 network. To understand the SS7 protocol stack, it will be expedient to first of all take a look at the Open System Interconnection (OSI) model which is a reference model for the SS7 protocol stack [3, 6, 8].

2.3.1 Open System Interconnection (OSI) Model

The International Standard Organization (ISO) in the 1970s initiated a project which aimed at defining a standard for communication between networked systems [9]. The outcome of their project produced a standard known as the OSI Reference model. OSI ever since has served as a reference model for data communication on the IP and telecommunication network. The OSI model is made up of two building blocks: abstraction layers and network protocols.

As shown in Figure 2.2, the OSI's protocol stack comprises of seven layers. Each of the layers on the OSI model is responsible for providing services to the layers above them whiles maintaining its own layer's services.

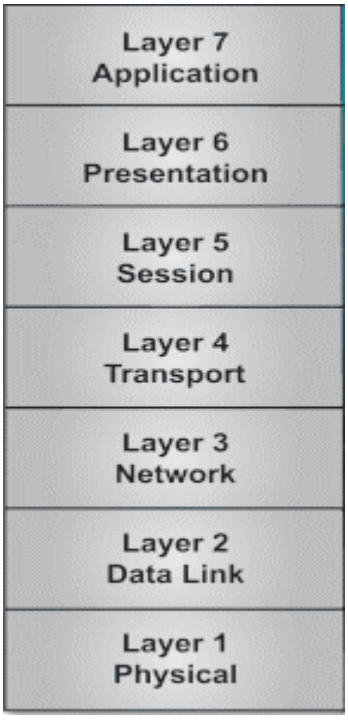


Figure 2.2: The OSI model [11].

Layer 1, the Physical layer defines the electrical and network components such as cables, network adapters and hubs. Primarily this layer specifies the data encoding format to be compatible with the transmission media.

Layer 2, the data link layer performs flow and error control. The layer provides a data communication link between two nodes and employs mechanisms to detect and correct errors in the exchanged data. The data link layer is marked by Media Access Control (MAC) which enables a host to gain access to network equipment or a communication medium and a Logical Link Control (LLC) to perform error checks and frame sequencing [9, 10].

The network layer, layer 3 is primarily concerned with conveying the data packets from a source node to a destination node residing on the network by a process known as routing. The layer provides an end-to-end delivery of messages using network addresses contained in the message.

The transport layer, layer 4 enables sequential delivery of data packets from one node to another on a network. Other functions performed by this layer include re-transmission of unsent packets, error control, segmentation of packets and reliable transmission.

The next layer, the session layer manages data communication between two nodes. The layer establishes a tunnel for applications on a local and remote host to communicate. It opens, coordinates and closes dialogue between applications running on different or the same host machines.

The presentation layer on layer 6 is primarily concerned with the format of data exchanged between end systems. The layer translates the data received into a more compatible format with the upper layer residing on the host.

Layer 7, the application layer is the nearest layer to users, accessed through software applications. Provision of network services to user's software is the main role of the application layer.

2.3.2 Overview of the SS7 model

The SS7 model is a direct reference to the OSI model but with certain layers omitted. Both models according to [3] were designed simultaneously. Layers on the SS7 network are usually referred to as *levels*. Four of these levels form the SS7 model as shown in the Figure 2.3 below. The physical layer on the SS7 network is marked by the message transfer part level one (MTP1). The data link and network layer are referred to as MTP level 2 and MTP level 3 respectively. The first three layers which are together referred to as Message Transfer Parts facilitate routing and transporting of signaling messages on the SS7 network. MTP level 3 together with the signaling connection control part (SCCP) makes the network service part (NSP) to provide complete network layer services to the SS7 network. Similarly to layer 7 of the OSI model, the upper level 4 of the SS7 network provides application services such telephone user part (TUP) and ISDN user part (ISUP), both for setting up and tearing of telephone calls [3].

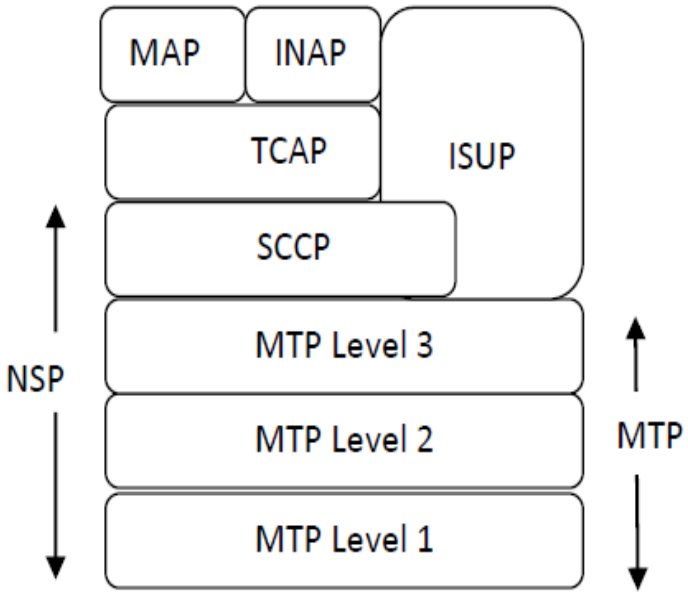


Figure 2.3: The SS7 model [12].

Message Transfer Part (MTP Level 1-3)

The MTP serves as a reliable means of transporting signaling messages between signaling points within the same PLMN. As a composite of the first three levels of the SS7 protocol stack, the MTP provides physical layer, data link layer and network layer functionalities to the SS7 network [3]. MTP ensures proper delivery of signaling messages through sequencing, error control checks and addressing schemes.

MTP level one, which is also referred to as the physical network of the SS7 model provides services similar to the physical layer of the OSI model. MTP1 chooses the network components for connecting the SPs to the transmission medium of the SS7 network.

Being the data link layer of the SS7 network, the MTP level 2 enables the SS7 SP to gain access to signaling links. MTP2 guarantees end to end transmission of signaling data between SP through error control checks, flow control, retransmission and frame segmentation and sequencing.

MTP level three is mainly concerned with routing signaling messages to the specified destination as well as delivering signals to their intended upper layer application (such as MAP). MTP3 uses point code addresses for delivering and routing signaling messages. MTP3 also has the task of providing signaling nodes with network management information such as link status. This Provides ample information to SPs when choosing the most appropriate link for transferring the signals.

Signaling Connection Control Part (SCCP)

The SCCP is used in conjunction with the MTP to provide routing and network management services on the SS7 network. The difference between the two is, whiles MTP provides information pertaining to the intra network of the SS7 network, SCCP gives information for routing and delivering signals between SS7 networks on different PLMNs. SCCP was developed as a result of the expansion of the SS7 network to reduce complexity in addressing network nodes on the SS7 network. SCCP reveals the SS7 network to other SS7 networks by means of Global Title addresses. GT addresses replace point code to enable SS7 connectivity to other PLMNS. GT addresses are translated to point codes by STPs by a process known as Global Title Translation before they are used in the intra SS7 network.

Transaction Capabilities Application Part (TCAP)

The Transaction Capabilities Application Part (TCAP) enables two applications services running on SS7 network to be bound together [3, 13]. With regards to the OSI model, TCAP is viewed as part of the application layer. TCAP establishes a dialogue between the upper layer applications on the SS7 network. To bind many of such applications to exchange signaling messages, TCAP employs the following components and terminologies: *Transaction* which represents the dialogue established between two TCAP protocols, *Dialogue and Dialogue ID* which is connection established between two TCAP entities and a value to identify different TCAP connections respectively.

Subsystem and Call Applications

Subsystem applications include applications which provide short message service, mobility management services, advance call handling services and intelligent services. Most of these subsystem applications are provided by the mobile application part (MAP) and the Customizable Applications for Mobile Enhanced Logic (CAMEL). Both MAP and CAMEL application protocols are discussed in the subsequent sections.

Initially the SS7 network was designed to provide call related services to establish and terminate telephone calls. SS7 employs two protocols namely TUP and ISUP on top of the MTP to provide such call services. ISUP allows ISDN connection to the SS7 network [3, 14].

Chapter 3

Vulnerabilities on SS7 Network

Very few telecommunication network operators existed at the time the SS7 network was developed and deployed. This made telephone companies to trust each other and believed that no fraudulent activities could be conducted on such a closed network. Being a network in a castle, the SS7 network which benefited from an inherent security from the trusted telecoms gave signaling messages a value of integrity and authenticity. The SS7 network was deployed with no cryptographic security mechanisms to check for messages originality (confidentiality service), no mechanism to verify the source of an incoming signaling message (authentication and integrity service), and neither a mechanism to prevent an unauthorized party from resending an approved signaling message at a later time (replay protection).

Currently SS7's security walls are devastated and exposed to a multitude of vulnerabilities and treat which degrades the security on the telecommunication network. There has been alleviation in the laws and regulations which governed the market of the SS7 network [3, 14]. Such liberalization has made the SS7 network less easy to access and acquire. In addition to that, attempts to merging the SS7 network to other networks for the appropriate interoperability has created many entry points on the network. The creation of entry points on the SS7 network has become a major source of vulnerability on the network [15].

As mentioned in the introductory chapter, mobility in telephone has introduces new services on the telecommunication network. A number of signaling messages has been created to facilitate mobility services on the mobile network. Unfortunately most of these legitimate signaling messages are wrongfully exploited on the SS7 network, due to the lack of cryptographic security on the network. Additionally, newer applications such as SMS and intelligent network services generate additional signaling messages and increase the chances of bringing the unsecured SS7 network under attack.

This chapter discusses the various entry points and their associated vulnerabilities to the SS7 network. Each of the three main signaling nodes on the SS7 is exposed to treats from the discussed entry points. The chapter also elaborates on the various signaling messages which can be abused to conduct attacks on the SS7 network. Together, these potentially abused signaling messages and the

type of attacks associated with them are described. Most of these abused signaling messages are created by the MAP layer (see section 3.2.1).

The two main messaging and call applications which are used on the mobile network are the SMS and CAMEL applications respectively. Section 3.2.3 discusses the routing procedures for sending SMS on the SS7 network and how it breeds vulnerabilities on the mobile network. Besides that, the architecture for the SS7 layer responsible for intelligent applications (CAMEL) is described and a further discussion shows an intrinsic vulnerability posed by CAMEL's architecture to the mobile network [16].

3.1 SS7 Entry Points

Alleviations in regulations and policies of the market for the SS7 network and attempts to provide a proper interoperability for the SS7 network with other networks (such as IP) have paved a way for many actors to access the SS7 network. These actors are the various channels (entry points) through which signaling messages are passed unto the SS7 network. Figure 3.1 below shows potential entry points to introducing malicious signaling messages on the SS7.

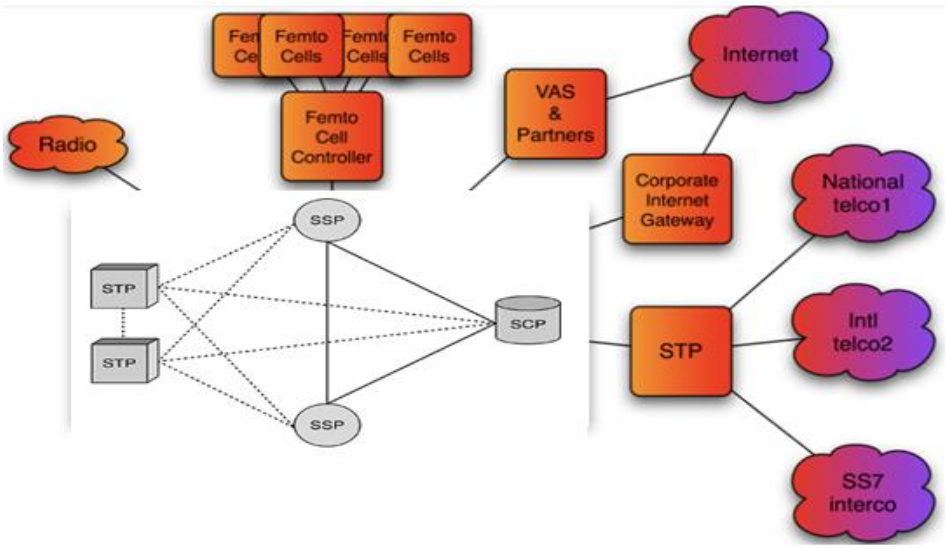


Figure 3.1: SS7 entry points [17].

Deregulations have made the SS7 network become ease to access. The three main signaling points are the major target points of attacks to the SS7 network, since all three signaling nodes have no security mechanisms to authenticate each other. The closest signaling point (SSP) to the

subscriber's mobile device is accessible through IP network by a device called Femtocell as shown in figure 3.2 [16]. SSPs can also be accessed through a voice trunk by ISDN users and analog lines, also see figure 3.2 [14]. The nearest signaling node on the SS7 network connected to the SSP is the STP. STP routes signaling messages on behalf of the SSP. SSPs residing in the same PLMN can also exchange signaling messages between each other.

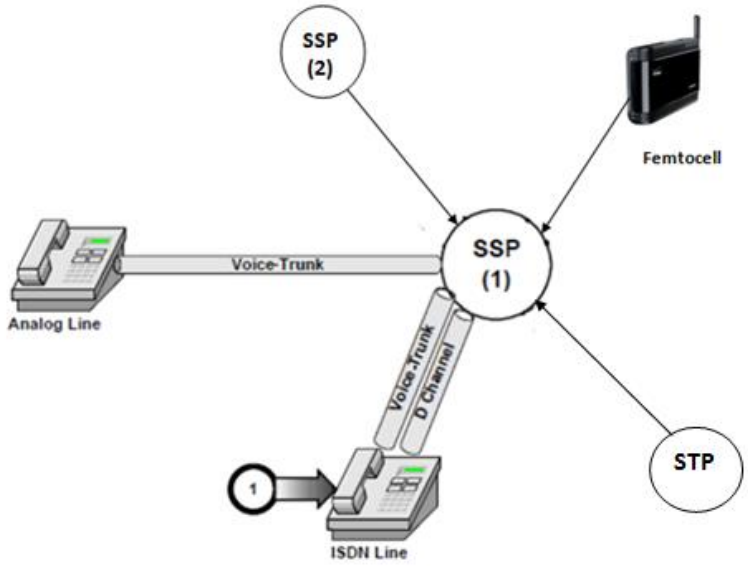


Figure 3.2: SS7 entry points at SSP [14].

The IP network is the most converged network to the SS7. Merging the IP network and SS7 produces a new protocol called SIGTRAN. Converging SS7 with the IP network is comparably gainful than other technologies such as time division multiple access (TDMA) [18]. IP network gains access to the SS7 network through the STP node. Besides that, both IP and SS7 can be connected to exchange signaling data through the session initiation signaling protocol (SIP), see figure 3.3. SIP communicates with the SS7 network via the network's gateway (STP). Within a PLMN, STPs communicate signaling messages with each other, whiles on the global SS7 network, STPs serves as a point of connection to other PLMNs.

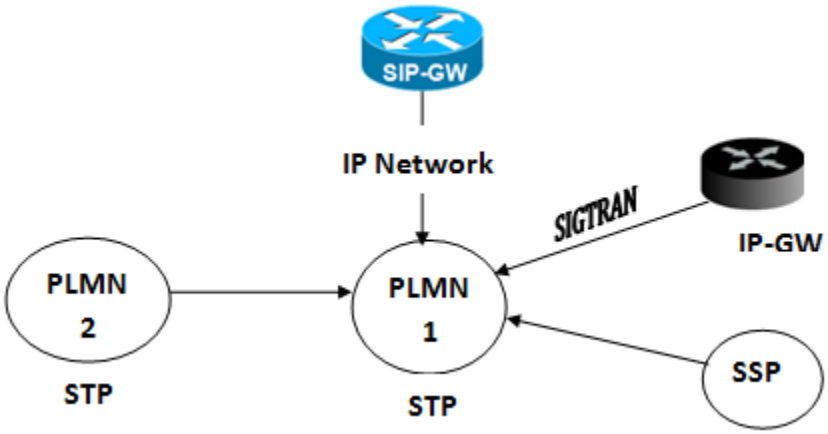


Figure 3.3: SS7 entry points at STP.

SCP is the entry point to database information on the SS7 network. SCP is usually hidden from other signaling nodes except the STP. Database information are routed to and fro the SCP through the STP as shown in figure 3.4. Thus any signaling message to request for database information is firstly passed to the STP.

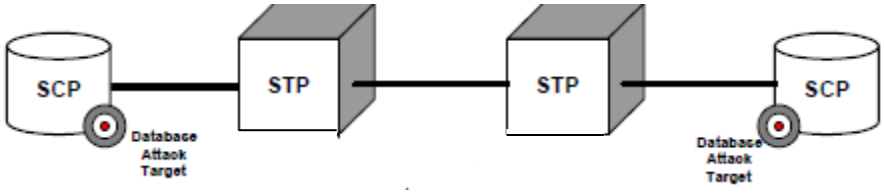


Figure 3.4: SS7 entry points at SCP [14].

3.1.1 Femtocell-to-SSP Vulnerability

Femtocell device are replacements for macro base transceiver stations (BTS) on the PLMN. They serve as access points (AP) for the mobile device to connect to the telecommunication network through a multiple access mode (usually code division multiple access (CDMA)) on a radio network [19]. Femtocells are comparably smaller in size than the traditional BTS used on the mobile network and are developed to be used in remote areas or business offices. A major advantage of using Femtocell is that users enjoy a better signal. As shown in figure 3.5, Femtocell devices are connected to the mobile network via an IP network through a secure IPSec tunnel.

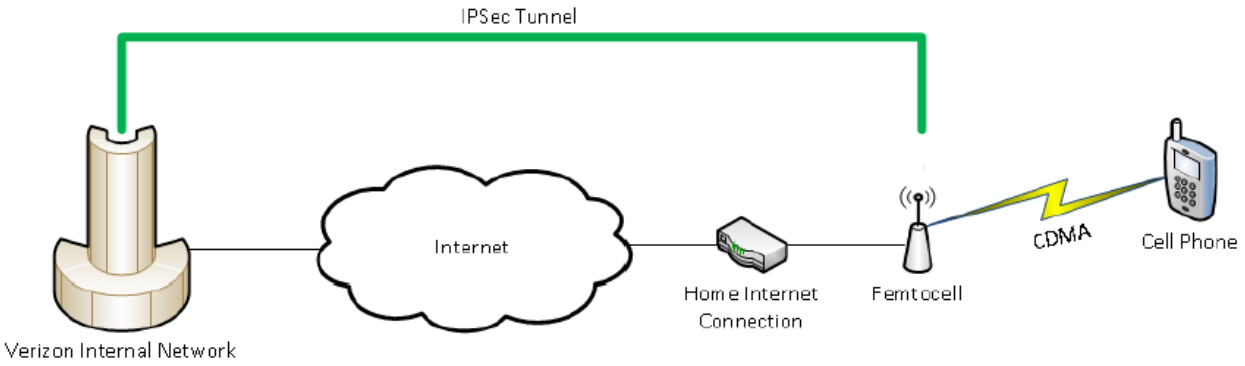


Figure 3.5: Architecture of Femtocell on CDMA link [20].

According to [19], the architecture for Femtocell devices which are designed to be used on air interfaces such as WiMAX, WCDMA and CDMA should adhere to the following features:

- The Femtocell device should provide a network configuration environment to enable users have control over the device. The device should possess enough configuration grounds as compared to that found in the traditional BTS. Such network configuration options include the ability to specify the mobile phone numbers permitted on the Femtocell.

- Consist of a security gateway which resides between the Femtocell device and the MSC (core network) responsible for performing the necessary conversions to expose the Femtocell to the core network as a device situated on a radio link as shown in figure 3.6 below.

- An operation, administration and management (OAM) system for performing the necessary update checks on the Femtocell. The standard employs the technical report 069 (TR-069) as the remote management protocol.

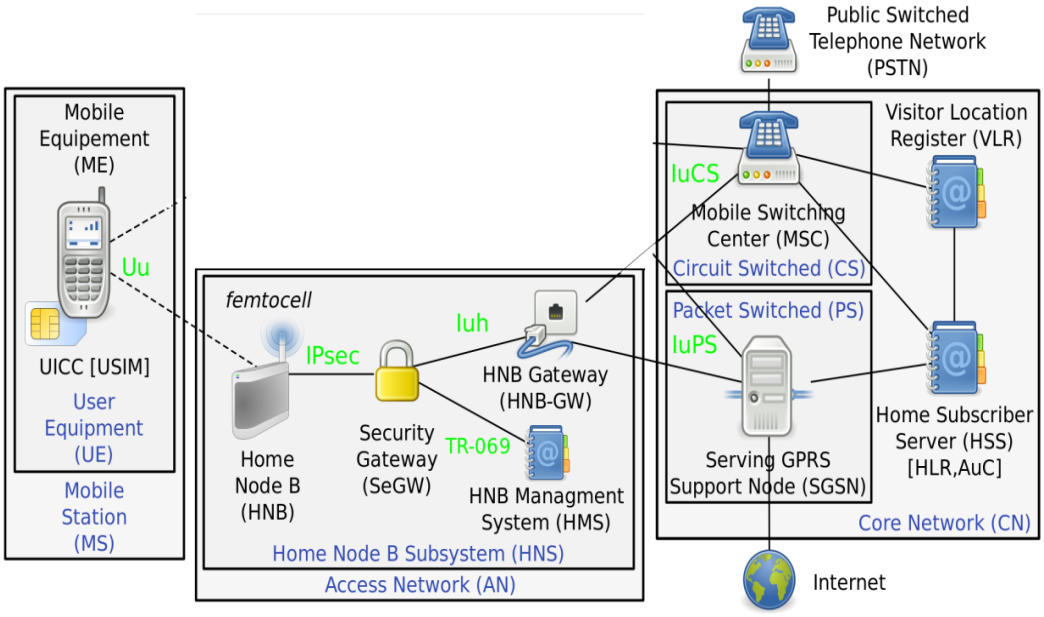


Figure 3.6: Femtocell subsystem architecture [19].

Vulnerabilities

A possible vulnerability which the Femtocell specification standards counteract is the exposure to the IP network. A standard requirement for every Femtocell device is that data should be sent in an encrypted manner and Femtocell devices should be authenticated by the mobile network. This requirement spares the use of Femtocell from IP vulnerabilities. However, the device is possessed by mobile subscribers and gives them physical control over it. Users can gain access to the terminal console and files in the Femtocell system. According to [20], not very much exploitation can be done from the devices terminal console, however, the system files could be retrieved and exploited on other sophisticated tools to gain SMS and voice data which is an indirect attack on the SS7 network.

On the CDMA system, a mobile subscriber is uniquely identified by electronic serial number (ESN) and mobile identification number (MIN). The former identifies the mobile device by the manufacturer whereas the latter is assigned to the mobile by the network operator. As shown in figure 3.7, the mobile network approves the mobile by the ESN and MIN, which is similar to authentication by international mobile subscriber identity (IMSI) on the GSM network.

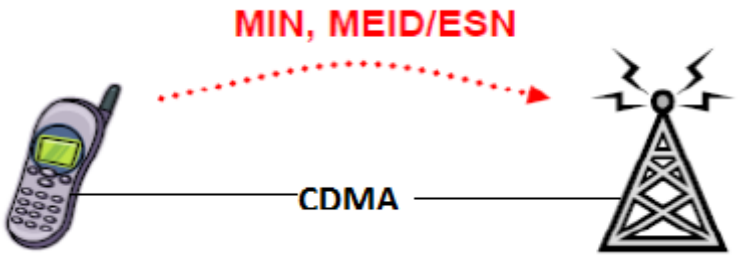


Figure 3.7: Subscriber authentication on CDMA system [20].

Masquerading with the MIN and ESN is a possible attack on the CDMA air interface. This attack is described in [21] as mobile cloning. To prevent cloning, the CDMA system employs a cryptographic mechanism known as cellular authentication and voice encryption (CAVE) to authenticate the mobile user. However, some Femtocell devices fail to implement the CAVE mechanism [20]. This gives attackers the chance to clone mobile devices and introduce unauthorized data unto the core network (SS7) of the mobile network through the Femtocell.

3.1.2 ISDN-to-SSP Vulnerability

Integrated service for digital network (ISDN) is a network standard which provides telecommunication services including data and voice transfer. The prefix “Integrated Service” refers to the networks ability to enable voice, data and other network services to be sent simultaneously on a single copper telephone line, yielding higher data rates [22, 23]. ISDN integrates with the SS7 network using the SS7/ISUP upper layer application protocol through a common link (channel D) [3, 23]. To implement an end to end communication between two ISDN telephone devices via the SS7 network requires mapping of signaling messages between both networks. Figure 3.8 shows a mapping of signaling messages exchanged between ISDN and SS7 nodes in an attempt to set up call between two ISDN users. ISDN initiates the call setup procedure with the “setup” message to the SS7 network to alert the network of a call request. SS7 nodes further process the call request setup using the initial address message (IAM) signaling message.

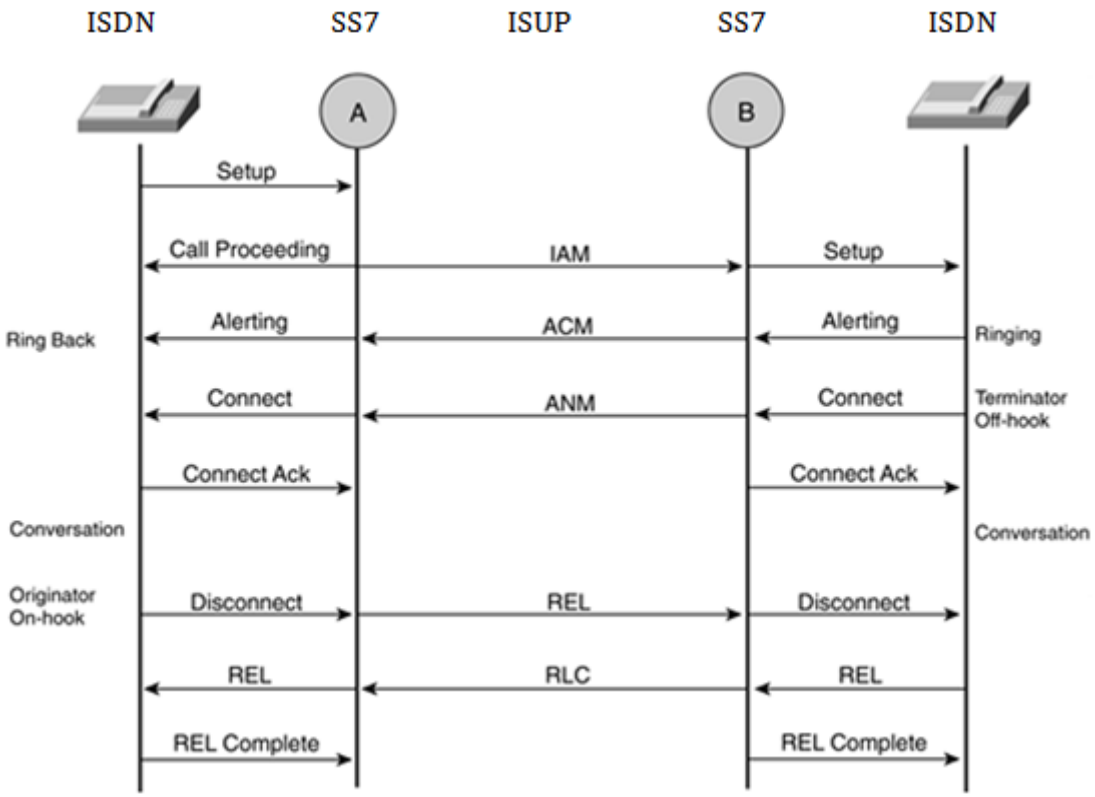


Figure 3.8: ISDN call setup over SS7/ISUP [3].

Vulnerabilities

A major vulnerability for the interoperation between ISDN and SS7 is the inadequate of authentication mechanisms to identify the sender and the originality of the received signaling message. A fake ISDN can masquerade as an authorized ISDN device and infuse the SS7 network with malicious ISUP messages. Likewise, this vulnerability can breed a denial of service attack (DOS) [14, 24]: An ISDN impersonator can initiate several call set up which will trigger a release of huge volumes of ISUP signaling messages (IAM) on the SS7 network. An attempt to process the requested calls will keep the network busy and denial services to other legitimate ISDN users seeking for a similar call service.

3.1.3 STP-to-SSP Vulnerability

STP is the closest signaling node connected to the SSP on the SS7 network. SSP is the connection point for entry points residing in the same PLMN and hence a request for telecommunication service from subscribers within the same PLMN gets to the SSP before they are passed to other signaling points. To process such services requires that the SSP retrieves database information stored on the network. To access database information, SSPs formulate database queries to the STP.

STP routes queries and return query results to the SSP. As demonstrated in section 3.1.2, an SSP which receives a "setup" signal from an ISDN node will be required to send query to the STP requesting for the destination route of the called party. The information contained in the returned query will be used to prepare the IAM signal.

Vulnerabilities

Inadequate authentication mechanism to proof the originality of signaling messages is again the source of vulnerability for the STP-to-SSP connection. Without cryptographic security mechanisms to authenticate the message originating node, an STP can easily deceive an SSP with a fabricated signaling message, where the fabricated message could be a query response.

3.1.4 IP-to-STP (SIGTRAN) Vulnerability

Transporting telecommunication signaling messages by the IP network relives the SS7 network from being overloaded with signals arising from newer telephone services. Currently the SS7 network is stuffed with signaling messages pertaining to SMS service [25]. Offloading such signaling messages unto the IP network will reduce congestion on the SS7 network. Additionally, in cost wise, such a transition is perceived as a gain as compared to transporting SS7 signals on TDMA lines. The IETF defines a signaling standard known as SIGTRAN for the IP and SS7 interoperation [26].

IP replaces MTP on the SS7-over-IP protocol stack and performs MTP functions on the SIGTRAN network. As shown in figure 3.9, IP lies at the bottom of the SIGTRAN protocol stack and takes full responsibility of the SS7 layer one protocol. IP is primarily concerned with addressing and routing signaling messages to their specified destinations. Similarly to the IP network which uses TCP/UDP as transport protocols, the SIGTRAN stack employs a new transport protocol called stream control transport protocol (SCTP) [27]. SCTP exhibits both TCP and UDP network transport functionalities: when acting as UDP, SCTP transport SS7 signals over the IP network without any assurance that the message will be delivered, but SCTP as TCP guarantees message delivery through retransmission mechanisms. An end to end message delivery on the SS7-over-IP network requires binding an IP address to an SCTP port number to form a *socket*. The SIGTRAN stack defines additional layer protocols called user adaption (UA) to support services related to the SS7 lower levels. These adaptation layers reside on top of the SCTP layer as shown in the figure 3.9 below.

MTP level two user adaptation **(M2UA)** carries MTP3 messages over the IP/SCTP socket. MTP level three user adaptation **(M3UA**) substitute MTP3 of the SS7 network to enable message application part messages (such as MAP and ISUP) to be carried on the IP network. Likewise SCCP user adaptation **(SUA)** functions like MTP and SCCP of the SS7 to transport user application messages on the SIGTRAN.

The SIGTRAN standard defines three new signaling nodes which are Signaling Gateway (SG), Media Gateway (MG) and Media Gateway Controller (MGC) on top of the SS7 network to facilitate the SS7-over-IP transition, see figure 3.10. All the three nodes sits on the edges of the SS7 network to expose the SS7 to IP and perform the necessary translations of signaling messages from SS7 to IP

compatible format and vice versa. Mostly the STP connects and routes SS7 messages to the SG. SG haven received the SS7 messages convert the message to IP compatible packets by a process called *encapsulation*. The SS7 encapsulated IP packets are further sent to MGC or MG for transportation over the IP network.

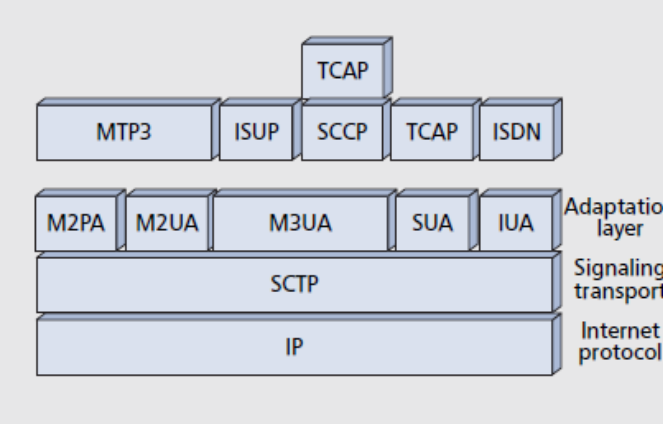


Figure 3.9: SS7-Over-IP encapsulation [29].

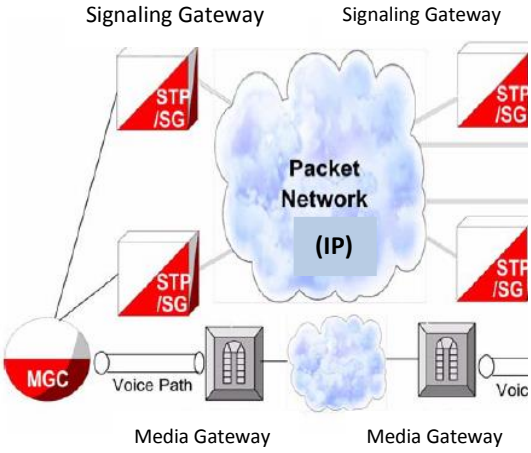


Figure 3.10: SS7-Over-IP architecture [28].

Vulnerabilities

Transporting SS7 signaling messages on the IP network exposes the signals to IP network vulnerabilities. The commonest of all is packet sniffing. Unless the telecom operator deploys cryptographic mechanisms (such as IPSEC) to authenticate and encrypt the SS7 encapsulated IP signaling messages, attackers can easily sniff and modify the encapsulated signaling packets. Deploying a SIGTRAN network without cryptographic security measures makes the network vulnerable to IP address spoofing. Any unauthorized node can easily impersonate an IP node by spoofing a legitimate IP address in the encapsulated packet. The impersonator can further flood the SIGTRAN network with encapsulated signaling packets resulting in a DOS attack.

SS7-over-IP network is vulnerable to SCTP port scanning. Performing a port scanning on a network is not the real attack, instead the aim is to detect loopholes in the network. Currently there are many SCTP scanning tools including that (SCTPscan) described in [30]. SCTP scan tools are purposely designed to conduct pentest on the telecom SS7-over-IP network, however, attackers can exploit these tools for their own gains.

Another vulnerability arising from the IP network allows an attacker to redirect SIGTRAN signaling messages to the attacker's desired destination. The vulnerability is such that an adversary is able to introduce malicious data on the domain name server (DNS) of the IP network [31]. DNS is an IP network server component responsible for receiving queries on address location of domain names (eg. mysite.com) and returning the corresponding IP address value. Usually the DNS forwards the query to other servers in case the server does not contain the requested query information. To

achieve higher DNS efficiency, the server stores the returned query information temporary in a cache memory. In figure 3.11, a server which fails to implement security mechanisms (such as DNSSEC) can be hoaxed by a fake DNS to redirect the encapsulated signaling messages to the attacker (step 1). The DNS can return a wrong IP address of the cached information received from the fake DNS to a client (SG) (steps 2 & 3). SG further sends the encapsulated signaling message over the IP network to the attacker's IP address (step 4).

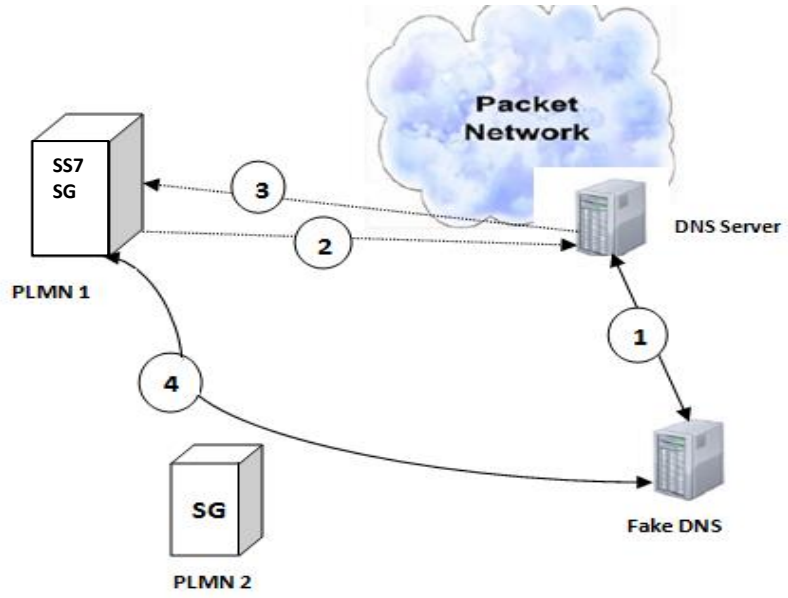


Figure 3.11: DNS spoofing on SIGTRAN.

3.1.5 SIP-to-STP Vulnerability

The driving force for internetworking between the telephone network and the IP network for video, audio and voice services is the session initiation signaling protocol (SIP). To connect two telephones over the IP network to exchange voice messages (VoIP) will require that a link is first of all established. SIP is used to setup such path over the IP for the call exchange. Once SIP has set the channel over the IP network, the real voice traffic can be exchanged between the two telephone terminals using the real-time transport protocol (RTP) [32]. To facilitate signaling for VoIP, the SIP protocol architecture deploys three fundamental nodes:

- **User Agent**: These are participants that either request or accept SIP services. The agent who request for service from SIP to reach the other party agent is the *client*, whereas the agent who responds to request from SIP is the server. User agents are simply software components installed on the SIP phone or computer.

- **SIP Proxy Servers**: The SIP proxy server acts as a mediator between the user agent client and user agent server to route service request and response between them. Apart from routing, the proxy performs other functions including, checking user agent availability and security checks (eg. authentication).

- **SIP Registrar Server**: The registrar is a database server to keep track of network information of user agent once they login. The registrar server monitors user agent location, hence it is required of user agents to submit their IP address to the registrar upon logging onto the network.

Internetworking between the SIP and the SS7 signaling protocols becomes necessary when a SIP device request to reach a device residing on a PSTN network, or vice versa. As shown in figure 3.12, a SIP phone requests for a voice service to a device on a PSTN. In this scenario, signaling messages which originates from the SIP protocol are carried over the IP network to the SG of the PLMN. SG based on a signaling mapping, converts the SIP messages to SS7 messages and transports to the STP on the SS7 network. Since SIP is an application layer protocol, it is mapped to a counterpart application layer protocol on the SS7 network, which is the ISUP.

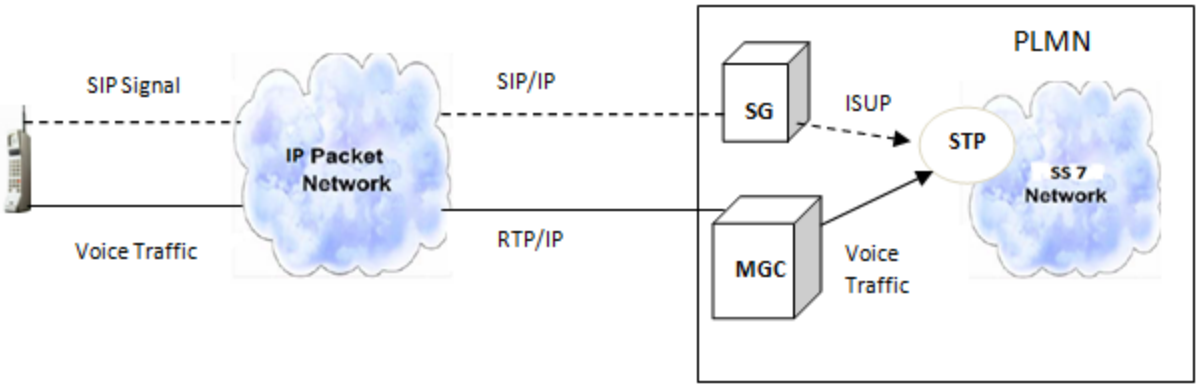


Figure 3.12: SIP/IP and ISUP/SS7 internetworking.

The message sequence diagram in figure 3.13 shows signaling message mapping for interworking on a call setup from a SIP/IP network to ISUP/SS7. An “INVITE” signal from the SIP protocol releases an IAM/ISUP message on SS7. ISUP replies to SS7 SG with an address complete message (ACM) which is mapped as message 100 to SIP. To terminate the call session, SIP signals “BYE” and receives release (REL) and release complete (RLC) from ISUP to confirm termination.

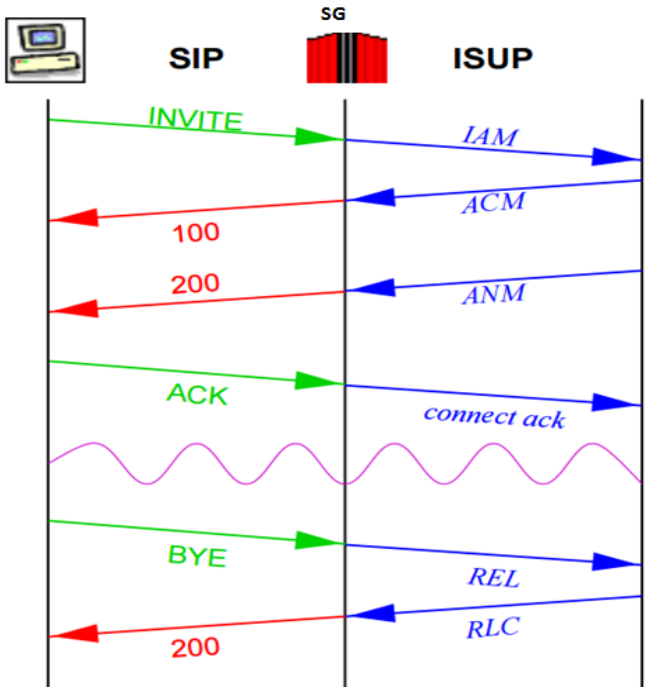


Figure 3.13: Call setup message mapping between SIP and ISUP [33].

Vulnerabilities

SIP and ISUP internetworking is liable to most IP vulnerabilities discussed in section 3.1.4. As shown in figure 3.13, signaling messages which originates from the SIP device are transported over IP to the PLMN's SG. An adversary on the IP network can easily sniff the SIP signals and eavesdrop on them. To counteract eavesdropping, SIP should deploy transport layer security (TLS) to authenticate and encrypt SIP users and signaling messages respectively [34]. The standard specification for the SIP architecture does not require TLS to be strictly deployed on every SIP network, however, it will be expedient to secure user privacy information such as voice communication.

Another vulnerability due the SIP and PSTN interoperation is threat to DoS attack. Usually UDP is used as the transport protocol for VoIP network, since a slight packet loss has less effect on performance. A network which receives a UDP packet performs the below steps [34]:

- Identifies the application opened for the port and checks whether any application awaits the port.

- Respond with an internet control message protocol (ICMP) to the specified IP address (address may be spoofed).

When a huge number of UDP packets are sent on any arbitrary port on the SIP network, the receiving host repeats the steps described above. When this happens, the receiving host waist

ample time responding to UDP senders with ICMP messages to indicate that the port is not reachable on the host. Other SIP users who request for UDP services will be denied due to the system's busy schedule.

Additionally, the SIP to ISUP architecture is vulnerable to flooding attacks arising from the SIP network (from user agent to SG, see figure 3.12). As described in [34, 35, 36], packet flooding is the commonest and easiest attack to conduct on SIP. Any of the three network components, being the registrar, proxy or user agent (SG of PLMN) could be a target for the flooding attack. In the case where the attacker targets the registrar server, the attacker simply formulates user registration messages and request to register the specified SIP accounts. Upon receiving heavy volumes of such request messages will compel the registrar to a busy state and denying any other SIP registration request [35, 36]. Another option for conducting flooding attack simply exploits SIP "INVITE" signaling message on the proxy server and the PLMN's signaling gateway. As already mentioned, the INVITE message signal initiates a SIP communication between SIP user agents. An attacker can masquerade to flood the proxy with INVITE messages, which will automatically results in flooding the other party (target SG) as shown in figure 3.14.

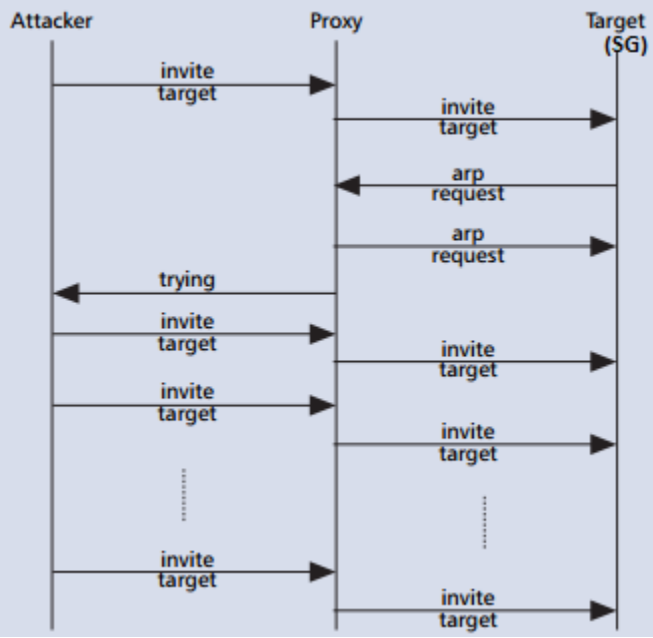


Figure 3.14: SIP flooding with INVITE signal [36].

3.1.6 STP-to-STP Vulnerability

STP serves as signaling gateway for routing signaling messages on both inter and intra network of the SS7. On the internal SS7 network, STPs intermediate between the SS7 switches (SSP) and the database (SCP) by routing query request from SSPs and returning results retrieved from the SCPs. STP is the access point for other PLMN's to access the SS7 network [15]. On the traditional SS7

network, STPs connect to other STPs from different PLMNs to share signaling messages to keep the telecommunication network live and running. Whiles on the SIGTRAN, STP is connected to the SG and the MGC to interconnect other SS7 networks to share signaling and payloads from other PLMNs. Message routing on the intra SS7 network is performed with point code addresses, whereas global title addresses are utilized to route signaling messages between PLMNs.

Vulnerabilities

STP is affected by the usual vulnerability on the SS7 network, inadequate authentication mechanisms to identify the origin of signaling messages. With the knowledge of OPC and DPC, an adversary can impersonate an STP and gain access to the SS7's database to retrieve, alter or delete subscriber information. Likewise an attacker with the knowledge of a GT address of a telecom operator can masquerade as a legitimate STP. Attacker simply formulates SS7 messages with spoofed GTs. since signaling messages are received without authentication, the attacker succeeds to gain illegitimate access to the global SS7 network.

3.2 Vulnerabilities arising from Mobility and advanced mobile services.

The SS7 network was initially designed to facilitate call services on the telecommunication network. Two SS7 protocols which were created to facilitate setting and tearing down calls are TUP and ISUP. Currently the telecommunication network is saturated by mobile users who transition from one PLMN to another. Even though it might sound simple, mobility on the telecommunication network has added additional signaling workload to the SS7 network. In facilitating mobility, the telecommunication network maintains databases (HLR and VLR) on every PLMN (or MSC) to keep record on subscriber location. Network subsystems (MSCs) are required to share the necessary information to update the respective databases to reflect subscriber mobility.

Development of newer services is another contributing factor for SS7's current workload. The short message service SMS, which is one of these new services is perceived as the most exploited mobile application on the telecommunication network [37]. The SMS architecture explained in section 3.2.3 shows that at least three signaling messages are exchanged between network subsystem nodes before the SMS text data is sent. Besides SMS, the telecom operators are allowed to define their own services. These services are referred to as *unstructured supplementary service* (USS) [3, 38]. USS services used by telecom operators include a service to demand for prepaid balance, a service to recharge prepaid card and a service to deliver a session password (one-time password). Additionally, the SS7 network supports message signaling to provide intelligent service. GSM intelligent services are usually provided to roaming customers, one of such service is *no prefix dialing*.

To enforce signaling between network subsystems on the PLMN to provide mobility and the above mentioned newer services, SS7 defines two new application protocols: Whiles the MAP layer defines signaling messages for mobility and GSM supplementary services, the CAP layer provides messages for intelligent applications. The telecommunication network has grown immensely for

providing support on mobility and other intelligent services. Unfortunately, most of the signaling messages generated by the SS7 application layer protocols to support mobility and the supplementary services are wrongly exploited. Anybody who gains access to the SS7 network can impersonate, masquerade, eavesdrop, intercept and perform fraud with such signaling messages [16]. The mentioned vulnerabilities are possible because the SS7 network fails to provide adequate authentication and encryption mechanisms. In addition to that, the architecture for signaling for SMS and the intelligent services intrinsically breeds vulnerabilities as discussed in sections 3.2.3 and 3.2.2 respectively.

3.2.1 MAP Signaling Messages Abuse

On the level four of the SS7 protocol stack, GSM defines the mobile application part (MAP) [39] to facilitate mobility services on the telecommunication network as well as providing support to realize telecommunication newly developed services (as already mentioned). MAP messages are carried in TCAP messages and are routed by the SCCP protocol. Currently the MAP layer provides a pool of signaling messages to enhance **mobility management**. MAP mobility management signaling messages include:

- Authentication signaling message (*send_authentication_info*) which is used to request and respond to subscriber authentication information when it roams to a new PLMN (MSC). The visited MSC uses the IMSI of the mobile device to fetch subscriber's authentication information from the home network.

- Signaling messages to facilitate subscriber location management. Such messages include *cancelLocation* message which is used to erase user's data from a preceding MSC/VLR, an *updateLocation* message to notify the home network upon completing a location update process and a *purgeMS* which is a message sent from the visiting switching center to the home network notifying the network how dormant the subscriber has been.

Another collection of signaling messages defined by the MAP layer are used for **administrative and management** purposes to decide whether or not to trace a particular subscriber as well as request for sensitive information about a subscriber. Such messages include:

- An *activateTraceMode* message used to turn on trace mode. As shown in figure 3.15, the message is usually ordered by a management center and sent from the home network (HLR) to the visited network (VLR) to enable tracking of the mobile user.

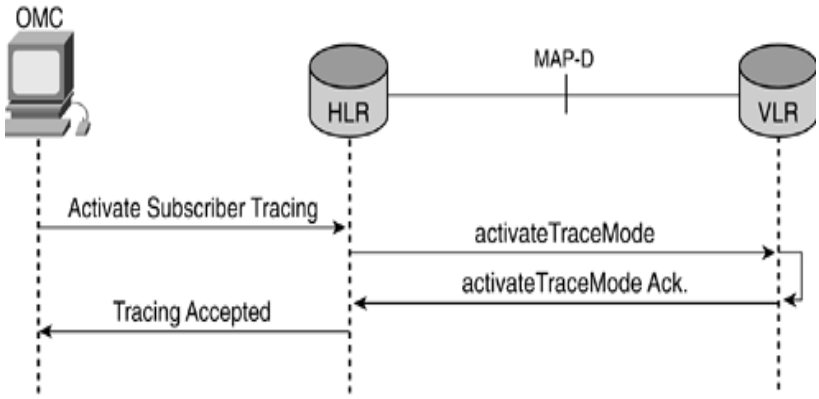


Figure 3.15: MAP_activateTraceMode request [3].

- A *deactivateTracemode* message used to disable subscriber tracking on the mobile network.
- Likewise the management centre can order for the IMSI number of a particular subscriber using the *sendIMSI* message as illustrated in figure 3.16.

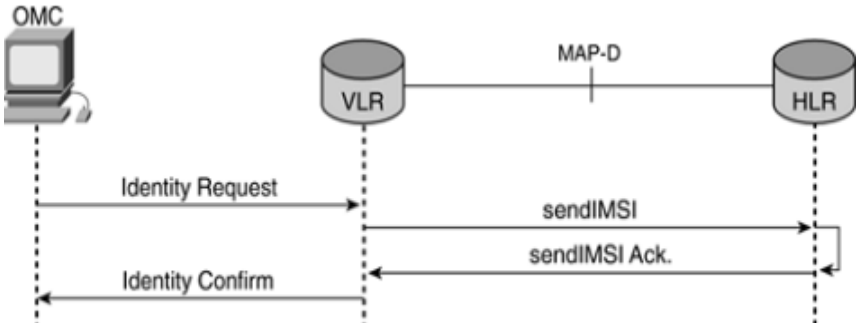


Figure 3.16: MAP_sendIMSI request [3].

The MAP protocol also provides signaling messages for call related services to supplement the ISUP as shown in figure 3.17. These messages are used to fetch routing information from the visited network haven received a call request from an IAM/ISUP signal. An IAM request for a call service from the mobile subscriber triggers the MSC gateway (GMSC) to send a request for information necessary to route the call to the location area currently serving the callee. MAP uses the *sendRoutingInfo* (SRI) message for that effect. The home network further request for the actual unique number required to route the call from the GMSC directly to the visited network. This is accomplished with the *provideRoamingNumber* message as depicted in the figure 3.17 below..

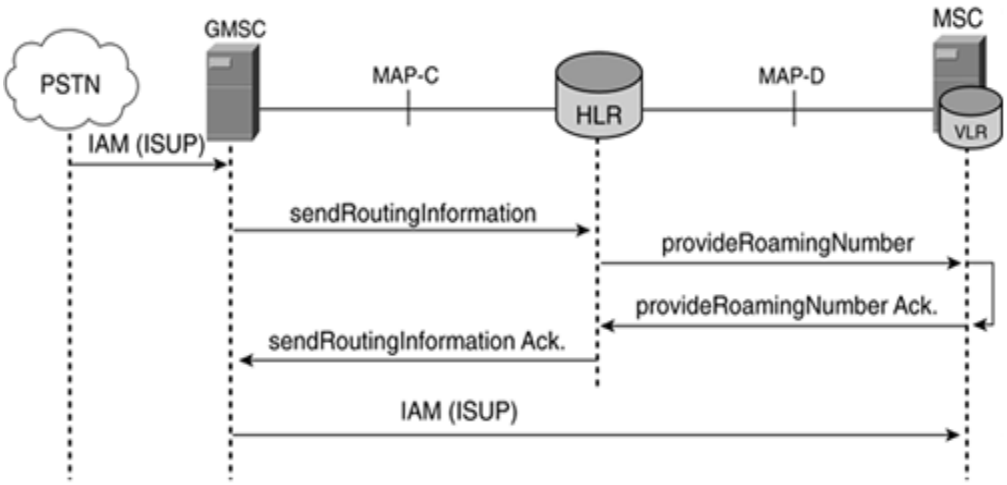


Figure 3.17: MAP_sendIMSI request [3].

Whiles the MAP layer defines messages to provide sophisticated services on the telecommunication network, the chances of brining the SS7 network under attack increases. Most of the above mentioned MAP messages are abused on the SS7 network, since the SS7 network fails to provide adequate authentication mechanisms to identify the origin of signaling messages. Signaling nodes are easily impersonated and exploited for the attackers own gain. The following demonstrations show how MAP messages are exploited to identify the location of a mobile subscriber (location tracking) and the secondly how MAP messages pertaining to intelligent applications are misused to intercept calls.

Subscriber Location Tracking Scenario

One of the many ways to perform location tracking on the SS7 network is to exploit the *anyTimeInterrogation* (ATI) MAP signaling message. The attacker is required to posses the mobile station international subscriber directory number (MSISDN) which is the phone number of the subscriber. The MAP ATI message instructs the home network to request for subscriber information from the visited network using the *provideSubscriberInfo* message. The visited network further retrieves the ID of the cell (base station) serving the victim (target subscriber) through a paging request [40] and return to the attacker as shown in the figure 3.18. According to [40], most telecom operators block the ATI request on their network.

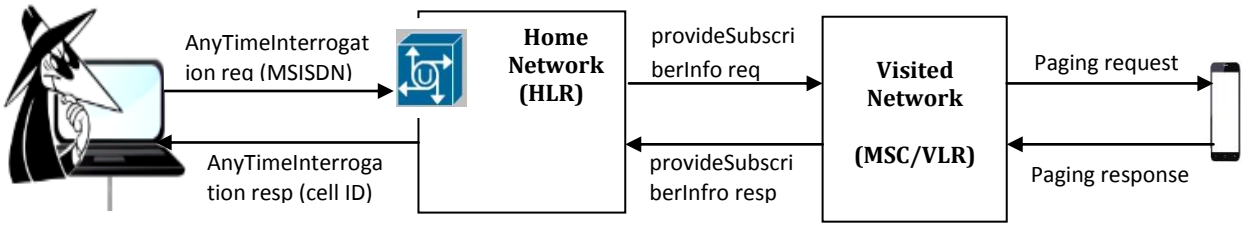


Figure 3.18: Subscriber tracking with MAP_anyTimeInterrogation.

In an alternative approach to perform location tracking, the attacker impersonates the home network (eg. spoofing GT of GMSC) and request for the location of the victim from the visited network (VLR). Such impersonation will require that the attacker knows the IMSI number of the victim as well as the GT address of the switching centre currently serving the victim. One way of retrieving the victims IMSI and GT of the victim's network is to conduct a brute force search with a *sendIMSI* MAP signal on every MSC on the network. An MSC which replies to the sendIMSI request implies that the victim resides on its network. Another approach exploits the *sendRoutingInfoForSM*. The sendRoutingInfoForSM signaling message can be used by anybody who request for SMS service from the network. An attacker takes the advantage to abuse the *sendRoutingInfoForSM.* Haven gotten the victims IMSI and GT of the visited network, the attacker further impersonate the home network with signaling messages such as *provideSubscriberInfo* and request for the victims location as shown in figure 3.19.

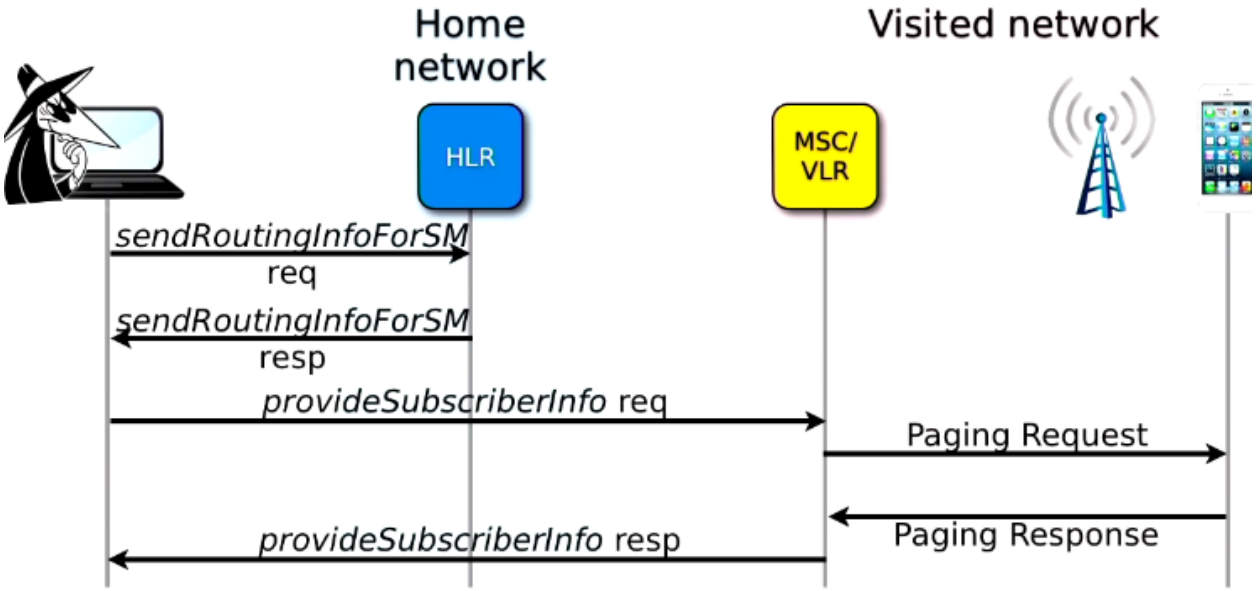


Figure 3.19: Subscriber tracking through HLR impersonation [40].

3.2.2 Intercepting Calls with CAMEL Application Part Protocol

To provide intelligent network (IN) services on the telecommunication network, the SS7 network integrates a new protocol to its application layer, this is the customized application for mobile networks enhanced logic (CAMEL) application part (CAP) [41]. CAP messages are carried as signaling packets encapsulated in TCAP messages on the SS7 network. CAP enables the mobile network to provide sophisticated services which surpasses the standard telecom services. Most of CAMEL's services gears towards adding value to call and SMS services [42]. Such intelligent application services include *no prefix dialing* service, which permits a mobile network to accept and process a call even when the subscriber omits the country code (prefix) of the callee's number.

Another CAMEL IN service allows a call in progress to be redirected to another mobile number by a process called *call transfer*.

The CAMEL application network is identified by the functionalities it provide as described in [42, 43] as "CAMEL Phases". Currently there are four distinct phases of the CAMEL network, each phase providing different telecom functionality services. However, each phase incorporates and improves the preceded phases.

To implement IN services on the SS7 network, CAMEL defines two fundamental nodes on the telecom network:

- A network node which resides in the home network, which contains the procedures and rules governing the implementation of an IN service. This is the **GSM Service Control Function (gsmSCF)**.

- A network node deployed on the visited network to intermediate between the MSCs and the gsmSCF. When a mobile subscriber who is registered on a CAMEL service request for the service, the visited network request for directives for handling the requested IN service from the gsmSCF. The node used to retrieve the CAMEL service directives from the gsmSCF is the **GSM service switching function gsmSSF**.

The CAMEL architecture provides a way to identify the various states at which an IN service is invoked; these are the point in call (PIC). Anytime a subscriber visits a new switching centre, the home network submits the subscribed IN services to the visited network. This is marked as triggering points in the MSC as illustrated as step (1) in figure 3.19. Unfortunately an attacker is able to masquerade as a legitimate gsmSCF due to inadequate authentication. In this scenario, the attacker takes the position of the home network and sends a CAMEL request on behalf of the gsmSCF of the mobile subscriber, asking the visited network for verification and correction of the callees phone number (steps 3 & 4). The attacker orders the visited network to route the call to the attacker's phone number (step 5). An attacker can further assume the role of a man-in –the-middle (MITM) to relay calls to the callee (step 6) as shown in the figure 3.20 below.

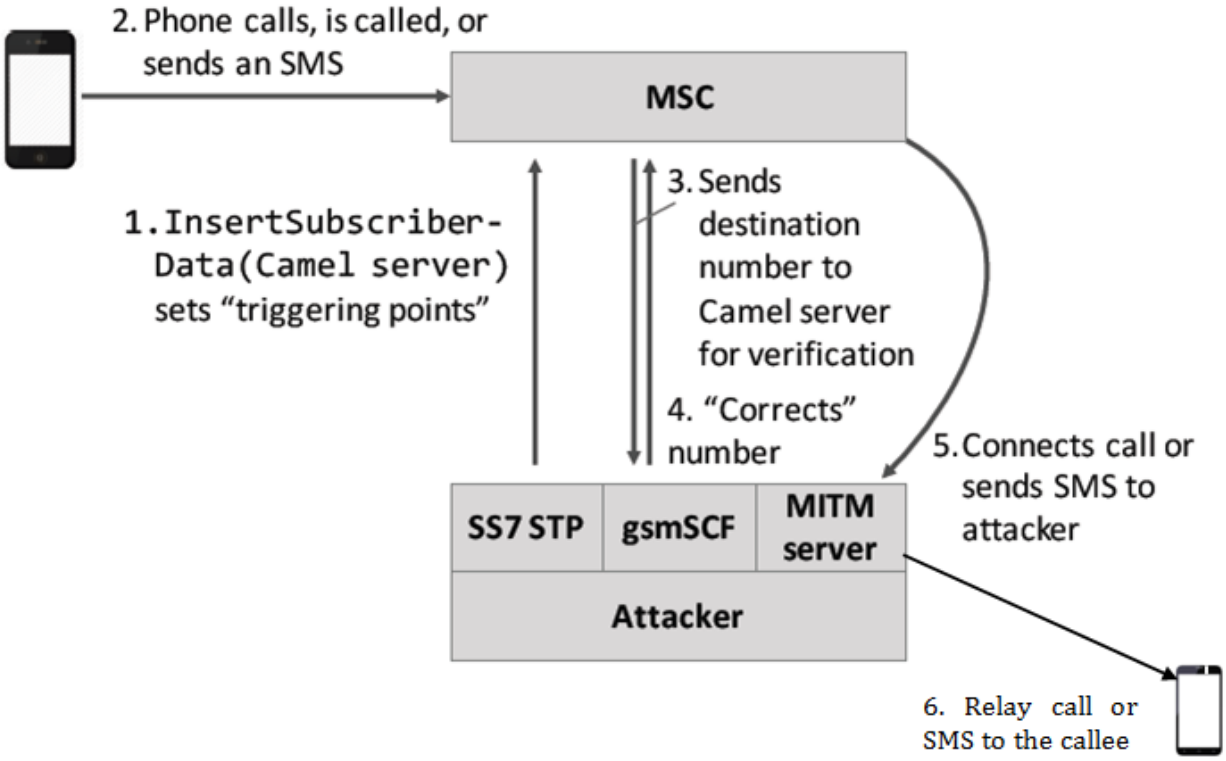


Figure 3.20: Call interception with CAMEL [16].

3.2.3 An intrinsic Vulnerability from SMS Routing

The procedure for processing call service request on the telecommunication network authorizes the callee's home network to have full control over the call routing; telephone calls are routed from the callee's home network (HGMSC) to the location where the callee is currently being served (VMSC). This procedure is different for sending SMS text message. Instead, the sender's network takes full responsibility in routing the SMS text to the recipient's network. The routing procedure for sending an SMS message on the SS7 network is illustrated in figure 3.21. The sender's network (originating SMSC) simply asks the receiver's home network (destination HLR) of the current location of the receiver using the *sendRoutingInfoForSM* message signal. The HRL responds with the receiver's IMSI and the GT address of the VMSC. Based on the provided information, the originating SMSC forwards the SMS text data to the VMSC in an *mtForwardSM* MAP signaling message as illustrated below.

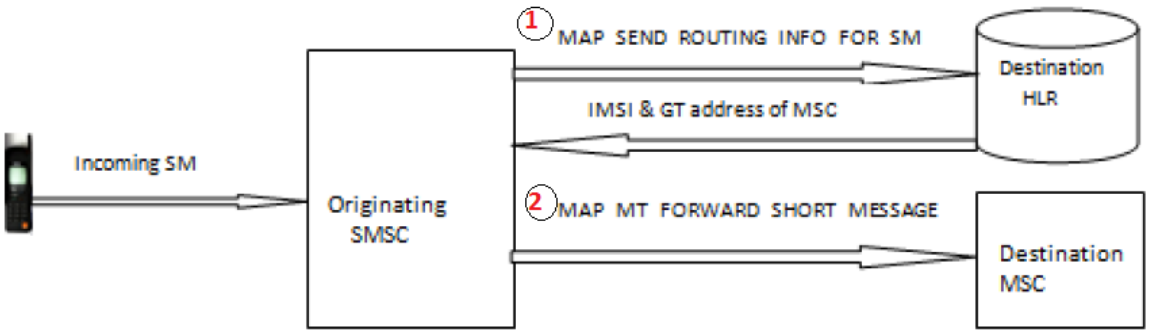


Figure 3.21: Routing procedure for SMS on SS7 [12].

Heading towards SMS Spamming

In a critical analysis of the SMS routing, it can be deduced that, first of all as shown in the figure 3.21 above, there is no coordination between the HLR and the MSC, thus the step to request for the receiving subscriber's location (step 1) and the actual sending of the SMS (step 2) are not bound to each other. Secondly, the SMS message is sent directly from the originating SMSC to the destination MSC without the knowledge of the HLR. This autonomy in the SMS routing procedure can be termed as *SMS routing flaw*. As a result of this flaw, an attacker is able to create a database comprising of valid information (MSISDN, subscriber IMSI and GT of VMSC) retrieved from the HLR of the SMS receivers as shown in figure 3.22. Note that, the attacker only have to know the phone numbers (MSISDN) of the recipients. An attacker can guess to compose a list of MSISDNs and retrieve their location information from the HLR. After obtaining recipients location information, the attacker needs one step further to send unsolicited messages to the victims (the guessed phone numbers). This is achieved by sending a huge volume of mtForwardSM MAP signaling messages to the VMSC of the recipients, which further forwards the spam messages to the mobile stations. In an attempt to deceive the VMSC, the spammer sets the following in the mtForwardSM message:

- A spoofed GT address as originating SMSC.
- Fake originating phone number, identified as transmission path originating address (TP-OA).
- Spam text message

The above SMS flaw permits an attacker to masquerade with other telecom operator's SMSC by simply spoofing their GT address. As illustrated below, spammer X spoofs the GT address of operator Y's SMSC, as a result of that, the receiving VMSC ends up charging the wrong operator (Y in this case) for rendering the SMS service.

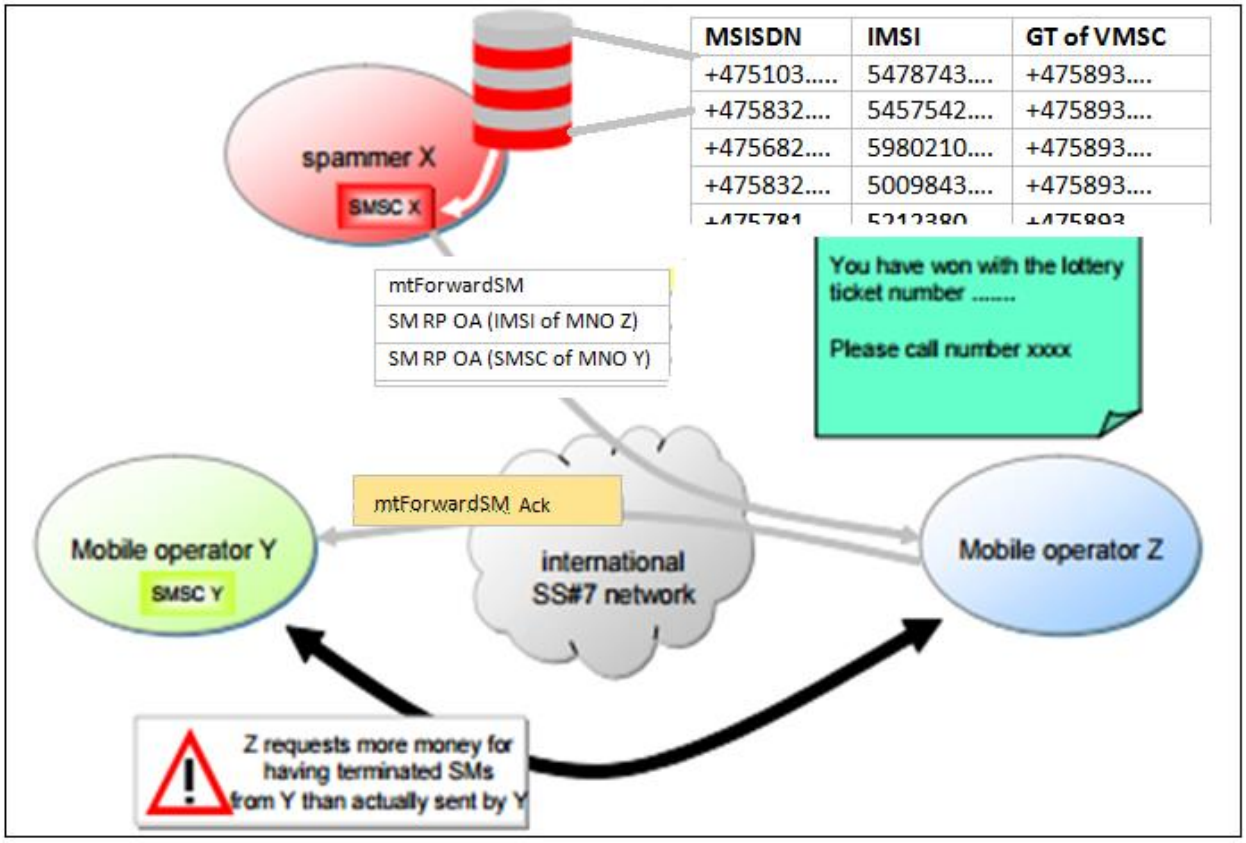


Figure 3.22: SMS spam illustration [44].

Chapter 4

An IDS for Potential Spam Signaling Messages on SS7

This chapter is mainly concerned with proposing and implementing an intrusion detection method to mitigate signaling for spam SMS on the SS7 network. The primary function of the IDS is that, the system should be able to detect similar volumes of signaling messages (mtForwardSM) sent simultaneously on a telecom's SS7 network over a predefined time for SMS service. Such signaling messages should be reported as potential spam messages. As already described in section 3.2.3 on figure 3.21, a spammer after gathering the necessary information required for routing SMS MAP signaling messages can further masquerade as a legitimate SMSC and send out spam text encapsulated in a mtForwardSM MAP signaling message. To prevent subscribers from being victims to such unsolicited messages, the proposed IDS can be deployed on the network of the SS7 to detect and report such simultaneous emission of similar volumes of mtForwardSM messages as potential spam signaling messages.

The impression is that, a high volume of similar mtForwardSM is received simultaneously on the SS7 network over a short period of time. It could be thousands or millions of mtForwardSM received on the SS7 network for a spam camping. In an attempt to propose a method to detect such similar bulky mtForwardSM MAP messages, the following are addressed:

- A definition for similarity and the type of similarity function to base our model on.
- Features to be selected from mtForwardSM.
- Storage facility to store mtForwardSM features before further processes.

As mentioned, the proposed IDS detects similar MAP signaling messages which are simultaneously received on the SS7 requesting to forward SMS texts to mobile subscribers. Since the mtForwardSM signaling message originates from the same spammer who wishes to achieve a particular goal with the spam messages, it is highly probable that the text message conveys the same information or the

spammer in an attempt to hoax the telecom network makes slight changes in the spam text. Hence the proposed scheme provides a near duplicate detection based on some selected features in the mtForwardSM message. The system receives mtForwardSM message and checks whether they are similar or same as the preceded mtForwardSM messages received within a time frame.

As illustrated in figure 3.21, spammers usually masquerade with the SMSC in the mtForwardSM message, however is this proposal we assume such vulnerability will be catered for. Meaning that, the received mtForwardSM signaling messages will be originated from the spammers SMSC. An elaboration on how to prevent against masquerading with SMSC is given in the subsequent sections. Subsequently, it can be perceived that, the SMS text and the GT address of the originating SMSC are the two consistent features in the mtForwardSM signaling messages. Another feature in the mtForwardSM message which could have been employed as an input to the similarity check is phone numbers (also known as TP-OA) of the victim subscribers. Studies have shown that similar SMS spam text messages originate from similar spam phone numbers [45]. However, in this dissertation, phone numbers are not used as mtForwardSM feature for the similarity check because we assume the spammers are experienced enough to generate random phone numbers for the spam messages.

With regards to similarity check, the basic intuition is that, two mtForwardSM messages are similar if some portions of message content they share surpass a given threshold. Applying this logic to two different mtForwardSM signaling message will imply that, when the selected features (SMS text + GT of SMSC) are broken into blocks, then similar mtForwardSM will have most of their blocks alike. In view of that, the *Jaccard similarity function* is employed to model the similarity check for the IDS. One additional requirement for the intrusion detection system is a function or an algorithm to test whether or not blocks of mtForwardSM feature message received are the same as previously received blocks. It should be noted that, thousands or millions of mtForwardSM messages might have already been received. To store such bulky blocks of mtForwardSM feature message for a later membership query, a sophisticated and a space efficient data structure algorithm will be required. A proper solution used in this thesis work is the *Bloom filters*.

4.1 Related Work

Several approaches to mitigate SMS spamming on the telecommunication network employ machine learning algorithms. Such machine learning approaches trains and learn from SMS spam reports collected from mobile subscribers to detect spam SMS messages [46, 47]. To depend on SMS spam reports from mobile users before training an algorithm to detect spamming can be described as a reactive approach to SMS spam detection. The Reactive approach fails to detect spam messages whose reports are not yet realized by the telecommunication network.

In the works of [48], they present a proactive method to detect potential SMS spam messages. In their method, they design a scheme which tries to assess a received SMS message against other SMS messages previously received over some time by comparing to figure out any matches in their message contents. In their method, they deploy counting bloom filter which is a variant of the standard bloom filter to keep track of the occurrence of blocks from a text message. The authors used YouTube dataset to test their detection scheme. Their results shows that using a counting

bloom filter of size 50000 and above can provide a 100% detection rate for an unusual number of similar text messages received on a network in a brief period.

This thesis demonstrates a similar proactive approach to detecting unusual number of similar mtForwardSM messages received on the SS7 network over a short period. Unlike [48] where authors used YouTube dataset, real SMS data are used as the SMS text feature to test the proposed detection scheme. An additional feature from the mtForwardSM signaling message added to the SMS text data to test the detection method is the GT address of the originating SMSC.

4.2 A Model for Similarity Check

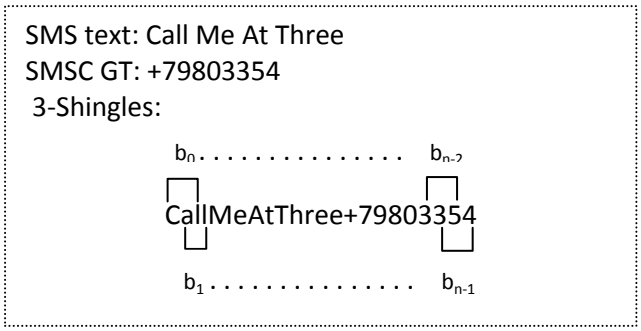
Smaller chunks of blocks from GT address of the originating SMSC concatenated with SMS text feature blocks represent the mtForwardSM signaling message to be used for testing the proposed detection method. Ideally, two mtForwardSM signaling messages are near duplicate if the blocks generated from their features are substantially the same. Let **F=** $\{b_0 . . . . , b_{n-1}\}$ represent a set of blocks extracted from mtForwardSM message features. One of the many ways to compute for similarity given two sets of mtForwardSM feature blocks $\mathbf{F_1}$ and $\mathbf{F_2}$ can be done through the well-known Jaccard similarity function [49], as shown in (1).

$$J_s(\mathbf{F_1} \text{ and } \mathbf{F_2}) = \frac{|\mathbf{F_1} \cap \mathbf{F_2}|}{|\mathbf{F_1} \cup \mathbf{F_2}|} \qquad (1)$$

To declare two mtForwardSM features $\mathbf{F_1}$ and $\mathbf{F_2}$ as near duplicate, the Jaccard similarity index needs to exceed some threshold, thus J_s ($\mathbf{F_1}$ and $\mathbf{F_2}$) ≥ J_{s0}, given that J_{s0} is a similarity threshold. The following properties about message similarity however should be noted [49, 50]:

- If the value for J_s ($\mathbf{F_1}$ and $\mathbf{F_2}$) is big, then both features are close.
- If the value is small, it implies $\mathbf{F_1}$ and $\mathbf{F_2}$ are not close enough.
- J_s ($\mathbf{F_1}$ and $\mathbf{F_2}$) with value 1 implies that both features are the same.
- Lastly, the value of J_s ($\mathbf{F_1}$ and $\mathbf{F_2}$) lies between [0, 1].

Blocks b_0 to b_{n-1} are substring of consecutive progression of all characters in mtForwardSM message features. Such consecutive blocks of characters can be formed using a method called *shingling* [50, 51]. To apply shingling on a message means to break the message into a consecutive substring of size *k* called k-shingles which is also known as k-grams. To generate all blocks of k-shingles for **F**, take the first substring of *k* characters which represent the first block, the next block can be derived by moving one position to the right and taking characters of size *k*. To get the next block, shift one position to right and take *k* size of character, repeat the steps to generate the rest of the blocks till the last character in the message is realized. An example of k-shingle on mtForwardSM message is illustrated below.



After performing K-shingling on the mtForwardSM message to generate the blocks **F**, the aim is to determine similarity between two sets of such blocks based on the Jaccard similarity shown in the equation 1. As already mentioned in this chapter, we assume that the spammer in an attempt to circumvent an anti-spam security measure makes slight changes in the SMS spam text. The spammer either omits some characters (deletion), or input some additional text to the spam message (insertion), or possibly modifies some existing text in the spam message (substitution). Such alterations transform spam messages with a difference which can be described as *edit distance* [52, 53]. The edit distance e_d shows the dissimilarity between spam messages. Given an mtForwardSM message with SMS text feature of size t, a k-shingling on the SMS text feature of size t will produce a maximum of t-k+1 blocks. If a spammer wants to generate two similar spam messages with an e_d (where e_d is the number of changes), then for each alteration of a particular character z in the original spam message, there will be at most $e_d \cdot k$ dissimilar blocks from the original spam message. In this case, every k-shingle block which contains character z will be different from all the k-shingle blocks generated in the original spam message. Therefore the portion of k-shingle blocks which are dissimilar to the original spam message after performing e_d operations is:

$$\frac{e_d \cdot k}{t\text{-}k+1} \qquad (2)$$

One of the above listed properties of message similarities states that two messages are the same if their Jaccard similarity index value is 1. So haven known the fraction of dissimilar blocks, the approximate portion of similar blocks will be:

$$1 - \frac{e_d \cdot k}{t\text{-}k+1} \qquad (3)$$

Hence to declare two messages to be similar, then equation (3) needs to exceed the preset similarity threshold, thus:

$$J_{s0} < 1 - \frac{e_d \cdot k}{t\text{-}k+1} \quad (4)$$

Or

$$J_{s0} \cdot t\text{-}k+1 < t\text{-}k+1 - e_d \cdot k \quad (5)$$

The implemented detection scheme in this thesis is based on equation (5) to identify huge volumes of similar mtForwardSM messages sent simultaneously on the SS7 network over a short period of time. The left hand side (LHS) of equation (5) is simply the product of the preset similarity threshold and the total number of k-shingle blocks derived from the received mtForwardSM message features, whereas the right hand side (RHS) computes the number of similar blocks which are already received. To be able to find how many similar blocks of mtForwardSM message features are received over a short period, thus in attaining the RHS, it is required to store all k-shingle block of each mtForwardSM message features F_0 , F_2 , F_3 . . . . , F_{n-1} received over the time period and then later computes how many of them are similar to the currently received mtForwardSM message F_n. To store and process huge volumes of mtForwardSM message feature blocks, a memory and space efficient data structure called *Counting Bloom filters* is introduced.

4.3 Bloom filters

A bloom filter is a memory and space efficient data structure which provides an optimal computational complexity for membership query on a set of large data items [54, 55]. The two fundamental operations of the bloom filter is the insert and query operation. The bloom filter maintains an array of bits and allows an insert operation to add 1 bit to the array based on the hash values obtained from hash functions (usually more than one hash function) on a particular data item. The filter can later be queried whether it contains some specified item. A query results which says “No, such item is not in the set”, is an absolute answer, meaning that the item is definitely not in the set. On the other hand if the returned results gives “Yes, the item is in the set”, then there is the likelihood that the answer is false. Thus a query results from a bloom filter can be false positive [55]. Figure 4.1 demonstrate an insert and query operation on a bloom filter with *n* bit array which is initially set to all 0s and a *d* distinct hash functions each capable of producing *n* different hash values.

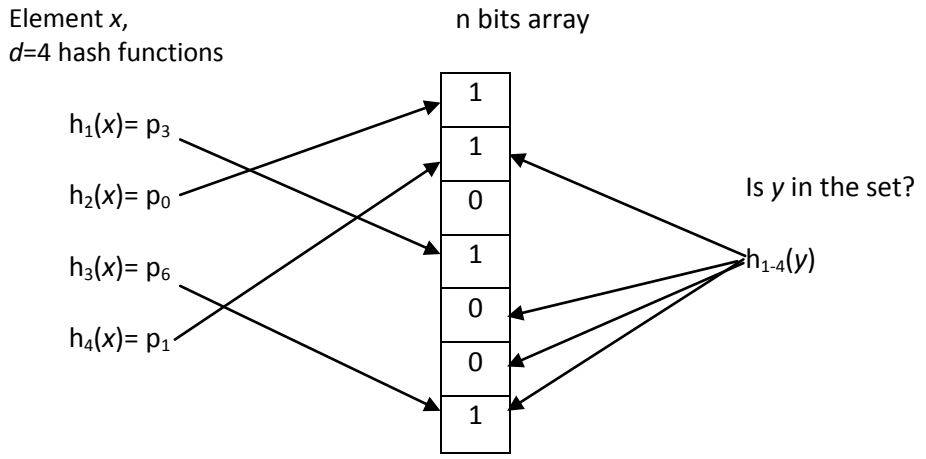


Figure 4.1: A Bloom filter illustration [56].

Upon receiving a membership query request for the element *y*, all four hash values produced from *y* are compared to the corresponding bloom filter array values. From figure 4.1 above, two of the array positions corresponding to the membership query on item *y* are 0s, meaning element *y* is not in the set. The general rule for membership test for bloom filters is that, if any of the returned query is 0, then the filter does not contain the element, but if all bit positions corresponding to the query are 1s, then the filter contains the queried item.

One peculiar feature about the insert operation on the bloom filter is that, when a bit position is already set to 1, other request to insert bit 1 to the same bit position are ignored, since the position is already used. Thus a bit position of value 1 does not guarantee that only one item is found in that position, instead two or more items may be represented by the same 1 bit. Unfortunately this does not permit a delete operation on the bloom filter, since the bit to be deleted could represent many items matched on that particular bit position. One more drawback on bloom filters is that the returned query could be positive whiles the item is actually not in the filter (false positive results). To reduce false positive rate, [54, 57] suggests that more than one hash functions are used, hash functions should be capable of producing good hash results, and lastly a bloom filter with a large space reduces false positives.

Unfortunately, the regular bloom filters are not able to achieve one of the requirements of the proposed detection scheme, which is that, the system keeps track of all similar mtForwardSM messages received within a short period of time. To maintain a count on the number of similar features from several mtForwardSM messages received within a time frame, a variation of the standard bloom filter called counting bloom filters (CBF) is implemented. CBF makes provision for delete operation on the filter [57], as well as keeps track of the number of similar mtForwardSM feature blocks received in a specified period of time. CBF replaces every array position in the standard bloom filter with a bucket counter instead of one bit; this enables the filter to keep count on the number of matches in every bucket of the bit positions. An insert and delete operations on the CBF increments and decrements the corresponding counters respectively.

4.3.1 Choice of Hash Function

The fundamental requirement for selecting a hash function for the counting bloom filter is that, the hash function should be fast as possible [58]. It is therefore not advisable to implement cryptographic hash functions (eg. SHA1) when using bloom filters. Most bloom filter implementations including Hadoop and python-bloomfilter use hashes different from cryptographic hash functions [58]. [58, 59] suggest that an easier way to generate hash values for the bloom filter can be done by simply combining two independent hash functions. However, the hash function should be able to produce *n* different CBF size hash values. A simple example of double hashing which is capable of producing *n* different CBF size hash values is:

$$hash = hash_1 + (hash_2 * i) \bmod CBFSize$$

In this regard, i takes values from 1 to the size of the CBF to produce *n* different hash values. The below is a java hash code for the CBF of the proposed detection scheme in this thesis:

```
public int hashCode()
             {

                    int h1, h2, size,i=0;
                    long hash;
                     Random r          = new Random();


                    size  = r.nextInt(cbfSize);
                    h1 = r.nextInt(cbfSize) % size;
                    h2 = r.nextInt(cbfSize) % size;

                    //Generate k different hash functions with a
simple loop

                    hash = h1;
                    while(i<cbfSize)
                    {
                        hash = h1 + (h2*i);
                        i++;
                    }
                    return new Random(hash).nextInt();
             }
```

4.4 PROPOSED METHOD OF DETECTION

This section shows a proposal of a detection method to identify an unusual high volume of similar mtForwardSM signaling messages received on the SS7 network over some brief period of time. The

approach is such that, a counting bloom filter, which is an array of buckets $CBF = [B[0], \ldots, B[n-1]]$ is used to keep a count of the occurrence of blocks **F** from selected features (SMS Test + GT of SMSC) of mtForwardSM messages. Upon the arrival of a new mtForwardSM message, the CBF which has membership query capabilities can be asked for the number of previously received blocks **F** which are the same as the newly received mtForwardSM message feature blocks **F**. Based on the CBF query return, we compute from the equation 5 (see section 4.2) to determine whether a volume of similar blocks **F** have been received within the time frame. The following three stages describe the implementation of the proposed detection method:

- **Stage 1:** To determine an unusual number of similar blocks, there is the need to define a threshold $T_{sh}[0], T_{sh}[1], \ldots, T_{sh}[n-1]$ corresponding to each CBF bucket $B[0], B[1], \ldots, B[n-1]$, which will enable the network to know the limited number for each block expected to be received within a time frame. Keeping a threshold for each CBF bucket block will help identify abrupt changes in the rate for each block received on the network.

  To compute the bucket thresholds, insert mtForwardSM messages into counting bloom filters $CBF_0, \ldots, CBF_{n-1}$, each for a specific time within the time frame. Afterwards compute the threshold for each bucket. In this regard, we compute the mean value for each bucket based on the number of filters N_cbf used within the time frame, thus a summation of all bucket counters in the same position of different filters divided by the total number of filters as shown in the equation 6.

  $$\overline{B}[i] = \frac{1}{N\_cbf} \sum_{j=1}^{N\_cbf} Bj[i] \qquad (6)$$

  The average value for each bucket threshold is set to the computed mean value from equation 6 or in case the computed mean is 0, then set the bucket value to an arbitrary number n of choice: $B[i] = max(\overline{B}[i], n)$, where n is some small number. The purpose of replacing every 0 mean value with n is that, at least every bucket threshold will be set to a minimum number n to prevent blocks which are previously unrecognized to be classified as potentially spam blocks. In this thesis, we set n to 1 to ensure that every CBF bucket has a minimum of 2 counts in order to be declared as a potential spam block.

- **Stage 2:** Upon the arrival of a new mtForwardSM message, break the message features into k-shingle blocks and increase their respective filter bucket counters based on their hash values. Having done that, check how many of the incremented filter buckets values are greater than the corresponding bucket thresholds (let counter represent the number of the exceeded bucket threshold). Now based on the equation 5 (see section 4.2) which was deduced from the Jaccard similarity function, if counter is greater than the product of the total number of the received blocks and the similarity threshold (thus $J_{s0} \cdot t\text{-}k+1 <$ counter), then report the mtForwardSM message as a potential spam message.

- **Stage 3:** When the time frame is due, update both the counting bloom filter and the bucket threshold. For the next allocated time frame, use an empty filter and also compute a new bucket threshold taking into consideration the values of the immediate past bloom filter.

4.5 mtForwardSM Message Features

The mtForwardSM message features selected for training and testing the proposed detection method are the SMS text attribute and the originating global title address of the SMS center (SMSC) attribute as shown in figure 4.2. The SMS text feature is the SMS message payload which is the message meant for the mobile end users. Usually spammers make slight changes in the SMS text messages they dispatch to mobile users. As already mentioned, spammers can also masquerade with the GT of other SMSCs in the mtForwardSM message for their SMS spam campaigns. These are the two consistent attributes in the mtForwardSM message which are deployed for training and testing the proposed detection method.

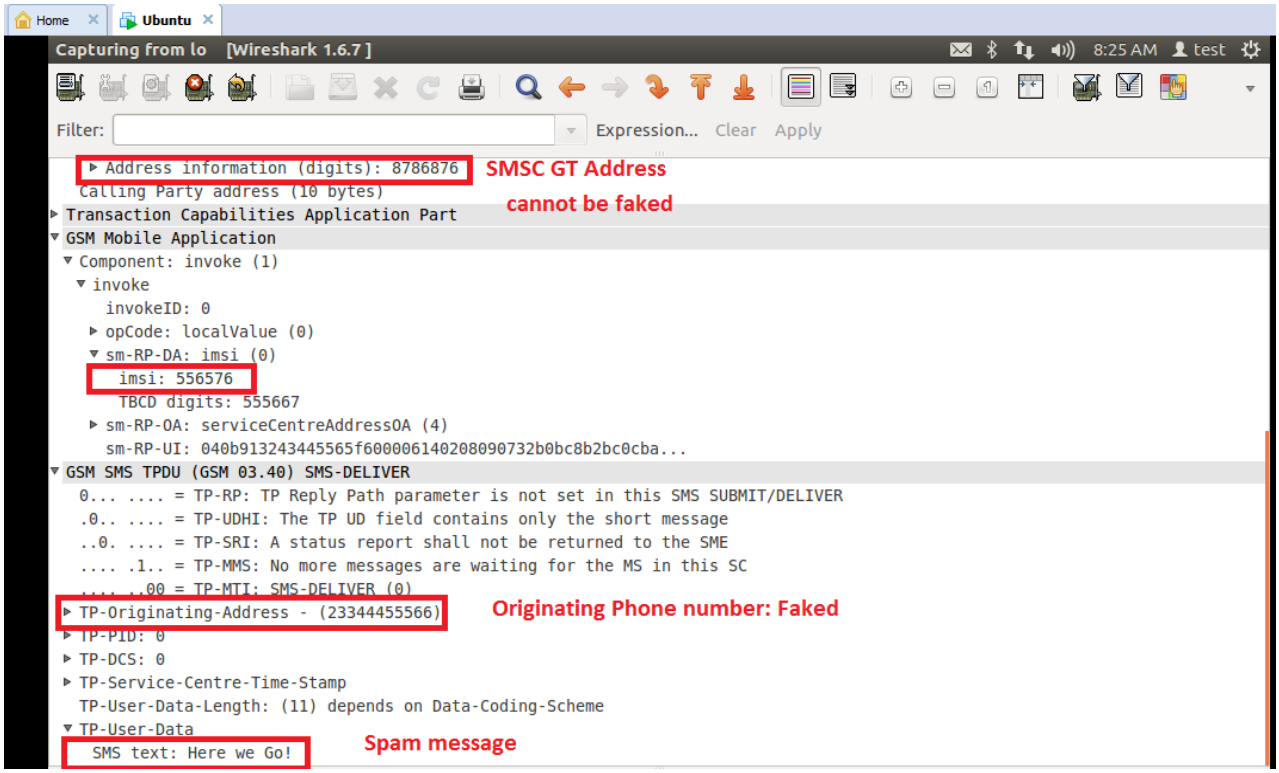


Figure 4.2: Wireshark capture of simulated Map_mtForwardSM message.

One of the possible means to prevent masquerading with SMSC GT can be achieved with the TCAP handshake protocol. TCAP handshake simply authenticates SMSCs before accepting mtForwardSM messages from them. The protocol enables negotiations to exchange mtForwardSM message between the sending and receiving SMSCs. The sender initiates the TCAP handshake dialogue with a TC_Begin message, specifying the originating SMSC addresses. The recipient replies with a unique transactional ID and expects the sender to authenticate itself with the ID. The TCAP four way handshake procedure as shown in figure 4.3 compels the mtForwardSM sender to use the right GT address.

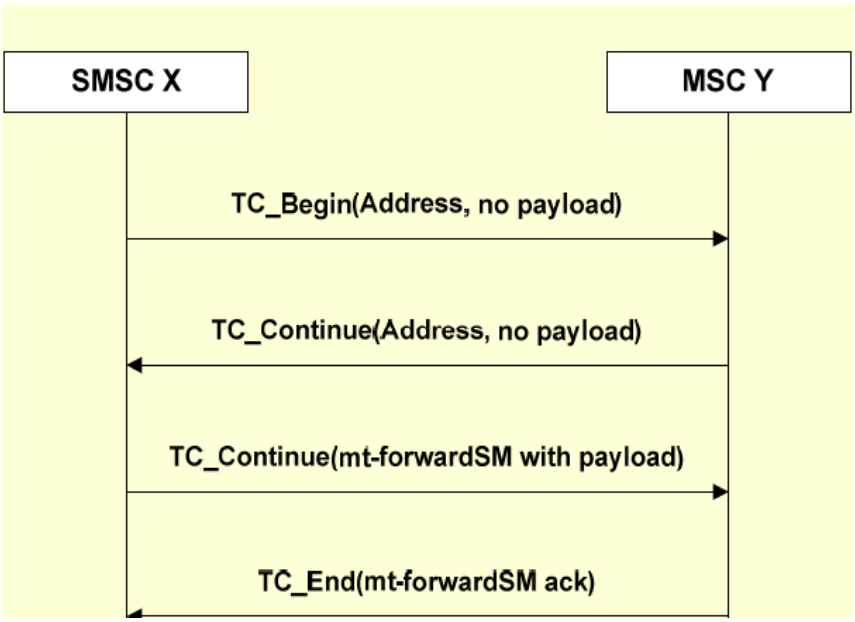


Figure 4.3: TCAP Handshake procedure [13].

4.6 Data Set Description and Preprocessing

For testing the proposed detection method, a publicly available SMS corpus comprising of legitimate and illegitimate (or spam) text messages has been used. The SMS dataset consist of 4,827 legitimate and 747 spam messages, totaling 5,301 SMS messages. The SMS dataset which was used by a different author to test an SMS-Spam-Classifier is available at [60]. The dataset is a composition of other sources of SMS data websites including Grumbletext [61], SMS messages from a PhD thesis by Caroline Tag [62], NUS SMS corpus meant for the University of Singapore [63] and finally a corpus collected by Maria Gomez Hidalgo [64]. Whiles testing the proposed detection method, both the legitimate and spam messages within the dataset are regarded as legitimate messages. The spam messages in the collected dataset are not referred to the spam messages used in this project, instead we formulate spam messages based on the procedure described in chapter 5.

The dataset only shows the SMS text messages without showing neither the originating phone numbers nor the originating mobile network (SMSC). However the GT address of the SMSC of the originating network is a selected feature in the mtForwardSM message to be used for testing the proposed detection method. To make provisions of SMSC GT address for the SMS text messages in the collected dataset, we assume that out of every hundred text messages, fourteen of the messages originated from a specific SMSC, whiles the rest (100-14) came from a random SMSCs (see the Java program in the appendix). This assumption is based on the fact that some SMS text messages are expected to might originate from the same SMSC.

Before the insertion of the message features into the counting bloom filter, each SMS text is processed to remove trivial symbols (such as spaces, #, !, etc) which adds no meaning to the text message.

4.7 Suitable Block Size

The size of the block is a determining factor for detection accuracy of the proposed detection method. Ideally, choosing smaller block size yields higher detection rate, however, if most of the words used in different messages are the same, then the chances of getting uncorrelated messages to be similar is higher. As shown on figure 5.1, using smaller block sizes reduces the detection rate of the proposed method, because, most SMS text messages contain similar words and therefore end up having similar blocks. Bigger block sizes (say 7 to 9) give a better detection results since uncorrelated messages with similar words turns out having dissimilar blocks. However, too much bigger block sizes according to our experiment tends to reduce the detection accuracy, for that matter, we choose 8 for the block size parameter.

Chapter 5

Detection Accuracy Results

5.1 Results and Discussion

This chapter shows a step by step procedure to testing the proposed detection method given the described dataset. Based on different parameter settings, we show the detection accuracy of the proposed method. Detection results using different block sizes and different number of spam messages are shown.

In the first step, the entire dataset is used to generate threshold values for the buckets in the CBF. To establish the bucket thresholds, we select messages without replacement from the dataset and insert their features (SMS text + GT of SMSC) into the counting bloom filters (with N_cbf=2). For each message features, we apply two distinct hash functions to generate two hash values and increase the corresponding bucket counters. Afterwards, each bucket threshold is computed from equation 6; by simply adding counters in the same bucket positions of the two filters and dividing the sum by two. The average value for each bucket threshold as described in stage 1 of the proposed method (see section 4.4) is set to:

$$B[i] = \max\left[\frac{B_1[i] + B_2[i]}{N\_cbf=2}, 1\right] \quad (7)$$

The next step focuses on determining the detection accuracy of the proposed detection method. Remember the primary objective for the detection scheme is to identify similar volumes of mtForwardSM messages sent simultaneously on the SS7 network over some short time. And it is mentioned in the previous sections that the spammer been professional enough tries possible means to circumvent anti-spam mechanisms by making slight changes in the SMS text messages. Based on this intuition, we create spam messages by taking any of the SMS text from the dataset which represents the seed (or parent spam message) and generate a number of similar messages given an edit distance e_d, where edit distance shows the dissimilarity between spam messages. For

example, given an edit distance of 8 and the text message below, similar spam messages can be generated by replacing 8 letters (italicized) in the spam seed with arbitrary letters:

Spam Seed: "You have won a £100 prize GUARANTEED. Text *KQZDRCJR* to 09050001808"

Similar Spam Messages:

You have won a £100 prize GUARANTEED. Text *SIHKAVIM* to 09050001808
You have won a £100 prize GUARANTEED. Text *OCPEZETC* to 09050001808
You have won a £100 prize GUARANTEED. Text *EGJSUHHD* to 09050001808
You have won a £100 prize GUARANTEED. Text *RNCNCCHI* to 09050001808
You have won a £100 prize GUARANTEED. Text *WSHIYLCB* to 09050001808
You have won a £100 prize GUARANTEED. Text *IVEMKXCS* to 09050001808

Now, we insert the generated spam messages with the SMSC GT addresses into a counting bloom filter. It should be noted that all spam mtForwardSM messages are originated from the same SMSC (see TCAP handshake at section 4.5), hence each spam message is assigned the same GT address. We apply two distinct hash functions on each spam message features and generate two hash values and increase the corresponding bucket counters. Having filled the spam features in the filter, the spam filter was topped up with an arbitrary number of SMS messages from the dataset.

To test the performance of the described detection method, we generated one more spam message from the spam seed and together with the GT address (GT same as other spam messages), increased the bucket counters of the filter which is already filled with previous spam messages based on their hash values. Afterwards, checked the bucket positions whose counters are incremented against their corresponding thresholds. Based on equation 5 (see section 4.2), if the total number of exceeded bucket threshold surpasses the product the total number of the received blocks and the similarity threshold ($J_{s0} \cdot t\text{-}k+1$), then declare the message as potential spam message. Detection accuracy for different block sizes and different CBF sizes are shown with the following parameter settings: N_cbf = 2 filters for generating bucket thresholds, edit distance $e_d = 8$ for spam messages, preset similarity threshold $J_{s0} = 0.64$ and 3 distinct spam seed, where each detection accuracy is determined from the three different spam seeds, thus to compute for the detection accuracies, we repeat the steps to testing for the performance of the detection method thrice. Detection accuracy is determined by the ratio of the number of similar blocks detected to the total number of blocks obtained from the message features, given the similarity threshold J_{s0}.

Detection Accuracy by sizes of blocks (k-shingles)

According to figure 5.1, it can be observed that smaller block sizes yields poor detection rate on the proposed detection method. The reason is because, SMS messages usually contain similar words, and so if the messages are broken into smaller block sizes, uncorrelated messages end up having similar blocks. Therefore whiles establishing the bucket thresholds, many blocks are inserted into the same buckets and as a result generating higher bucket thresholds. A spam messages is therefore required to possess higher bucket values to exceed bucket thresholds before it can be classified as potential spam message. According to the figure, detection accuracy tends to increase as the number of block sizes increases (4 to 8-shingles). With 8-shingle blocks, the proposed detection scheme gives 100% detection accuracy for 10 to 90 spam messages. It can also be observed that detection rate decreased when the block size reached 10, hence, for such a detection method, the choice of too much higher block size parameters is not encouraged.

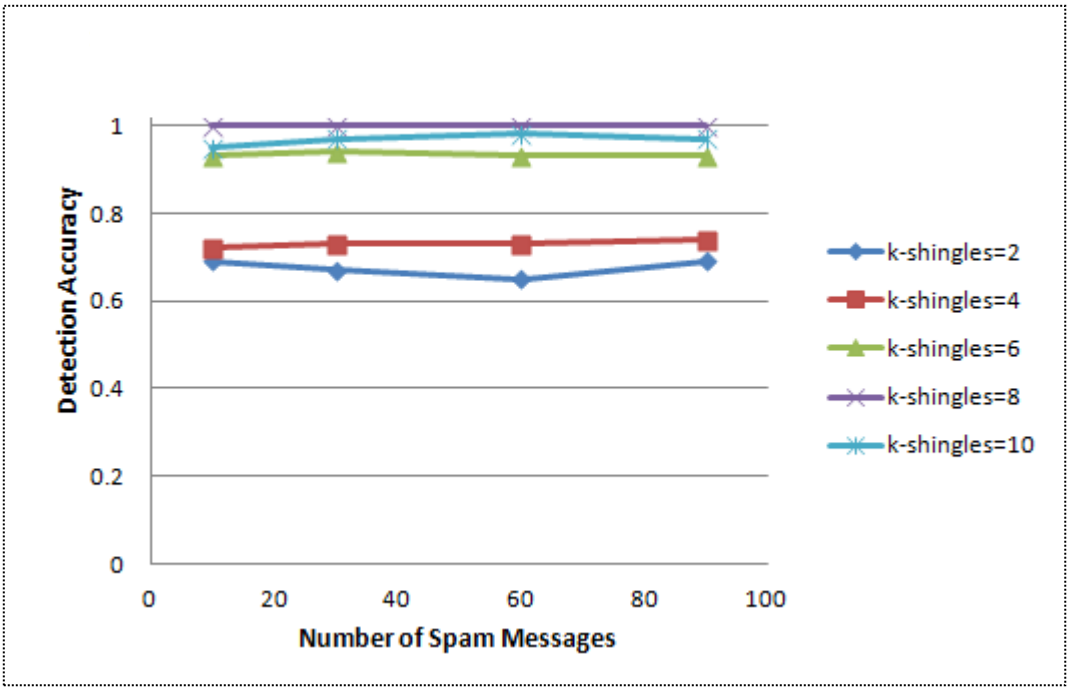


Figure 5.1: Detection Accuracy by sizes of blocks (k-shingles)

Detection Accuracy by different counting bloom filter sizes

Figure 5.2 below shows that larger bloom filters gives better detection results on the proposed detection method. Detection accuracy for spam messages of 40 to 160 tends to increase as the counting bloom filter size increases from 5×10^3 to 1×10^5. CBF of size 50000 and above gives approximately 100% detection accuracy. Higher CBF sizes yields better detection results because larger filter sizes provides enough room to prevent differing blocks from being inserted in the same buckets. Hence accurate values are maintained for the bucket thresholds and spam messages are correctly identified. Otherwise, smaller size filters will allow dissimilar blocks to be mapped in the same buckets whose values will exceed the spam bucket values and fail to identify spam campaigns. Another observation drawn from figure 5.2 is that, the proposed method gives better detection accuracy as the spam messages rises.

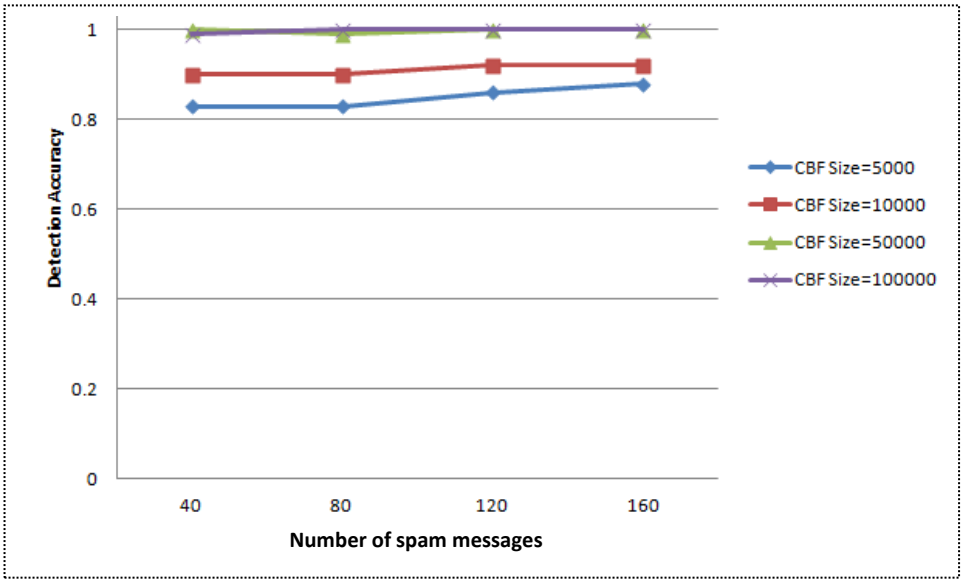


Figure 5.2: Detection Accuracy by sizes of CBF.

Chapter 6

Conclusion

The current wall-less signaling system no.7 network is exposed to a multitude of vulnerabilities which degrades telecommunication network security. The castle walls of the network have been taken captive by alleviations in laws and regulations which governed the market of the network. Attempts to merging the network to other networks for the appropriate interoperability has added numbers to SS7 castle wall attackers. Likewise telephone mobility and advanced telecom services has led to the creation of many signaling messages which are abused to perform attacks including subscriber tracking, call interception, SMS spamming and denial of services attacks.

A comprehensive discussion on the threats posed to the various SS7 network elements is shown. For the three fundamental SS7 network nodes, we list the various entry points which connect to them and elaborate on the kind of vulnerabilities they breed on the network. We described that the SS7 network converges with the internet protocol (IP) and the session initiation protocol (SIP) to reduce traffic load on the network. Converging SS7 with the IP network allows attacks such as packet sniffing, port scanning and DNS spoofing to be easily carried out on the SIGTRAN. Likewise SS7 and SIP internetworking makes room for eavesdropping and DOS attacks. Additionally, the SS7 network fails to provide adequate authentication mechanism to authenticate other PLMNs SS7 nodes (eg. STP), whereby permitting illegitimate access to sensitive data on the telecommunication network. Telecommunication applications responsible for text messaging and advanced call services are the SMS and CAMEL respectively. The later provides intelligent services beyond the standard telecom services. We have shown that an attacker can take advantage of the network's inability to authenticate CAMEL nodes to masquerade as a legitimate node and intercept calls. The former breeds an intrinsic vulnerability whiles in the process of routing signaling messages pertaining to the SMS texts. The flaw in routing SMS messages allows an attacker to spam the network with unsolicited SMS MAP signaling messages (mtForwardSM).

The latter part of this thesis shows a proposal and evaluation of a detection method to mitigate signaling for spam SMS on the SS7 network. The proposed method sorts to detect similar volumes of mtForwardSM messages sent simultaneously over a short period of time in request for SMS service. The proposed near duplicate detection method was modeled after the Jaccard similarity function and deploys counting bloom filters to record the appearance of mtForwardSM message features. Test results on the proposed detection method shows that, selecting a very small block size parameter yields low detection accuracy rate, however, with a block size parameter 8, the scheme provides 100% accuracy on detecting 10 to 90 spam messages with edit distance (dissimilar characters) of 8 in 5,301 SMS messages. Likewise test results proofs that with the same

number of SMS messages and a counting bloom filter of size 50000 and above, the proposed detection method gives approximately 100% detection accuracy for spam messages of 40 to 160 with edit distance of 8 characters.

It will be necessary to test the proposed detection method with a larger number of SMS text messages as further work, because ideally, more than 5,300 SMS MAP signaling messages can be received on a telecom's SS7 network within a short time.

Reference

[1] Wikipedia. https://en.wikipedia.org/wiki/Telecommunication.
Retrieved February 18, 2016.

[2] Wikepedia. https://sv.wikipedia.org/wiki/Antonio_Meucci.
Retrieved February 18, 2016

[3] Lee, D., & Jeff, H. (2005). Signaling System No. 7 (SS7/C7): Protocol, Architecture, and Services.

[4] Telektronikk. (2011). Mobile Virtual Network Operators.

[5] Wikipedia. https://en.wikipedia.org/wiki/Common-channel_signaling
Retrieved March 18, 2016.

[6] Wikipedia. https://en.wikipedia.org/wiki/Signalling_System_No._7#cite_ref-q700_1-0
Retrieved March 18, 2016

[7] Dodo, A, Z. (2001). The Essential Guide to Telecommunications.

[8] TOP Business Interactive. Web-Based Training: SS7 Basic.

[9] Wikipedia. https://en.wikipedia.org/wiki/OSI_model
Retrieved March 31, 2016.

[10] Global Knowledge. (2006). The OSI Model: Understanding the Seven Layers of Computer Networks.

[11] http://pubs.opengroup.org/architecture/togaf8-doc/arch/Figures/figa-9.gif .

[12] Yeboah, P, N. (2015). SS7 Vulnerabilities and Countermeasures.

[13] Chung, K., & Gustafsson. (2016). Prototyping and Evaluation of TCAPSec

[14] Lorenz, G., Moore, T., Manes, G., Hale, J., & Shenoi, S. (2011). Securing SS7 Telecommunication Networks.

[15] Vauboin, P, O., &Oliveira, O,D. (2014). Worldwide Attack on SS7 Networks.

[16] Karsten N. (2016). Recap: SS7 Attack Potential

[17] Philippe, L. (2002). Telecommunication Infrastructure Security.

[18] Agilent Technologies. (2002). SS7 over IP White Paper.

[19] Wikipedia. https://en.wikipedia.org/wiki/Femtocell

[20] DePerry, D., Ritter, T., & Rahimi, A. (2014). Traffic Interception & Remote Mobile Phone: Traffic Cloning with a Compromised CDMA Femtocell.

[21] Balani A. Authentication and Encryption in CDMA Systems.

[22] Beckman M. (2013). An Introduction to ISDN: http://www.jet.net/isdn/isdnintro.html.

[23] Wikipedia. https://en.wikipedia.org/wiki/Integrated_Services_Digital_Network Retrieved June 13, 2016

[24] P1 Security. (2012). Telecom Signaling attacks on 3G and LTE networks.

[25] Agilent Technologies. (2002). SS7 over IP White Paper.

[26] IETF RFC 2719. (1999). Framework Architecture for Signaling Transport.

[27] IETF RFC 2960. (2000). Stream Control Transmission Protocol.

[28] Tekelec Eagle. (2007). SS7-over-IP Networks Using SIGTRAN.

[29] Jajodia., Wijesekera, D., & Dantu, R. (2006). SS7 Over IP: Signaling Interworking Vulnerabilities

[30] P1 Security. Retrieved from http://labs.p1sec.com/2014/12/28/ss7map-country-risk-ratings/ Retrieved June 14, 2016

[31] Wikipedia. https://en.wikipedia.org/wiki/DNS_spoofing Retrieved June 15, 2016.

[32] Wikipedia. https://en.wikipedia.org/wiki/Session_Initiation_Protocol Retrieved June 16, 2016.

[33] IETF. (1998). Mapping between ISUP and SIP.

[34] 383_NTRL_VoIP_08.qxd. (2006). SIP Architecture.

[35] Geneiatakis, D., Dagiuklas, T., Kambourakis, G., Lambrinoudakis, C., & Gritzalis, S. (2006). Survey of Security Vulnerabilities in Session Initiation Protocol.

[36] Deng, X., & Shore M. Advanced Flooding Attack on a SIP Server.

[37] Wikipedia. https://en.wikipedia.org/wiki/Text_messaging . Retrieved June 20, 2016.

[38] Wikipedia. https://en.wikipedia.org/wiki/Unstructured_Supplementary_Service_Data. Retrieved June 20, 2016.

[39] Oracle Communication Network Charging and Control. (2010). Mobile Application part (MAP)

[40] Engel T. SS7: Locate Track Manipulate.

[41] Wikipedia. https://en.wikipedia.org/wiki/CAMEL_Application_Part
Retrieved June 24, 2016.

[42] Wikipedia. https://en.wikipedia.org/wiki/Value-added_service
Retrieved June 24, 2016.

[43] Meskauskas P. Customized Application for Mobile Networks Enhanced Logic (CAMEL)

[44] http://telecompk.net/wp-content/uploads/2008/09/smsf.png
Retrieved June 28, 2016.

[45] Jiang, N., Jin, Y., Skudlark, A,. & Zhang, Z. (2012). Understanding SMS Spam in Cellular Network: Characteristics, Strategies and Defense.

[46] Houshmand, S, M. SMS Spam Detection using Machine Learning Approach.

[47] Mujtaba G., & Yasin M. (2014). SMS Spam Detection Using Simple Message Content Features.

[48] Coskun, B., & Giura, P. (2012). Mitigating SMS Spam by Online Detection of Repetitive Near-Duplicate Messages.

[49] Wikipedia. https://en.wikipedia.org/wiki/Jaccard_index
Retrieved July 5, 2016.

[50] Philips, J, M. (2013). Jaccard Similarity and Shingling.

[51] Wikipedia. https://en.wikipedia.org/wiki/W-shingling
Retrieved July 5, 2016.

[52] Jurafsky, D. Stanford University. Minimum Edit Distance.

[53] Wikipedia. https://en.wikipedia.org/wiki/Edit_distance
Retrieved July 6, 2016.

[54] Wikipedia. https://en.wikipedia.org/wiki/Bloom_filter
Retrieved July 11, 2016.

[55] Tarkoma, S., Rothenberg, C, E., & Lagerspetz, E. Theory and Practice of Bloom Filters for Distributed Systems.

[56] Cao, P. (1998). Bloom Filters- the Math

[57] Tech-Effigy. (2016). Bloom Filters- Explained.

[58] Fork me on Github. Bloom filters by Example.

[59] Kirsch, A., & Mitzenmacher, M. (2007). Less Hashing, Same Performance: Building A better Bloom Filter.

[60] RevantKumar. (2014). SMS-Spam-Classifier: https://github.com/revantkumar/SMS-Spam-classifier.

[61] Grumbletext. http://www.grumbletext.co.uk/

[62] Tag Caroline. http://etheses.bham.ac.uk/253/1/Tagg09PhD.pdf

[63] University of Singapore. http://www.comp.nus.edu.sg/~rpnlpir/downloads/corpora/smsCorpus/

[64] Gomez, H., & Maria, J. http://www.esp.uem.es/jmgomez/smsspamcorpus/

Appendix

A Abbreviations

ACM	Address Complete Message
AUC	Authentication Centre
CAMEL	Customized Application for Mobile Networks Enhanced Logic
CAVE	Cellular Authentication and Voice Encryption
CCS	Common Channel Signaling
CDMA	Code Division Multiple Access
DNS	Domain Name System
DOS	Denial of Service
DPC	Destination Point Code
EIR	Equipment Identity Register
ESN	Electronic Serial Number
GSM	Global System for Mobile Communications
GSMSCF	GSM Service Control Function
GSMSSF	GSM Service Switching Function
GT	Global Titles
HLR	Home Location Register
IAM	Initial Address Message
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IMSI	International Mobile Subscriber Identity
IN	Intelligent Networks
IP	Internet Protocol
ISDN	Integrated Services Digital Network
ISUP	Integrated Services Digital Network User Part
ITU	International Telecommunication Union
M2UA	MTP Level Two User Adaptation
M3UA	MTP Level Three User Adaptation
MAC	Media Access Control
MAP	Message Application Part
MIN	Mobile Identification Number
MSC	Mobile Switching Centre
MSISDN	Mobile Station International Subscriber Directory Number
NFAS	Non-facility Associated Signaling
NSP	Network Service Part
OAM	Operation Administration Management
OPC	Originating Point Code
OSI	Open System Interconnection
PLMN	Public Land Mobile Network
PSTN	Public Switch Telephone Networks
UDP	User Datagram Protocol
USS	Unstructured Supplementary Service
RTP	Real-Time Transport Protocol
SCCP	Signaling Connection Control Point
SCP	Service Control Point

SCTP	Stream Control Transport Protocol
SG	Signaling Gateway
SIGTRAN	Signaling Transport
SIP	Session Initiation Protocol
SMS	Short Message Service
SMSC	Short Message Service Centre
SP	Signaling Points
SPC	Signaling Point Code
SS7	Signaling System No. 7
SSP	Service Switching Point
STP	Service Transfer Point
TCAP	Transaction Capabilities Application Part
TLS	Transport Layer Security
TUP	Telephone User Part
VLR	Visitor Location Register
VMSC	Visited Mobile Switching Centre
VoIP	Voice Over IP

B Java Implementation Code

B.1: Main function

```
import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;
import java.util.ArrayList;
import java.util.Random;

public class StringClass {

   public static void main(String[] args)
   {
         int data1[],data2[],data3[],tshold[],counter=0,i=0;
         double markValue;
         ArrayList usedIndex = new ArrayList();
         // inserting training data into counting bloom filter 1
         File file1 = new File("D:\\spam1.txt");
      ScannerClass trainingcbf1= new ScannerClass ( file1,100000,"training");
         data1=trainingcbf1.cbfData();
      while(i<100000)
         {
                  //System.out.println(data1[i]);
                  i++;
         }
    // inserting training data into counting bloom filter 2
      File file2 = new File("D:\\spam2.txt");
      ScannerClass trainingcbf2= new ScannerClass ( file2,100000,"training");
      i=0;
         data2=trainingcbf2.cbfData();
      while(i<100000)
         {
                  //System.out.println(data2[i]);
                  i++;
         }
    // inserting spam data into counting bloom filter 3
      File file3 = new File("D:\\spam.txt");
      ScannerClass trainingcbf3= new ScannerClass ( file3,100000,"spam");
      i=0;
         data3=trainingcbf3.cbfData();
      while(i<100000)
         {
                  //System.out.println(data3[i]);
                  i++;
```

```
}
// Call tshold method here
CBF <String> T=new CBF <String>(100000);
tshold=T.Tshold(data1, data2);
i=0;
while(i<100000)
{
//System.out.println(tshold[i]);
i++;
}
// insert data to determine the detection rate
File file4 = new File("D:\\spamTest.txt");
ScannerClass spamtest= new ScannerClass ( file4,100000,"detector");
spamtest.cbfData(); //new SMS message, break the msg into blocks and insert the hash values
//into the spam testing cbf.
usedIndex=spamtest.index();  // retrieve the indexes in the spam testing  cbf which are used
i=0;
while(i<usedIndex.size())
{
// add 1 to the affected indexes in the spam testing CBF
data3[(Integer)usedIndex.get(i)]+=1;
//System.out.println(data3[(Integer)usedIndex.get(i)]);
i++;
}
// check if the bin counters exceed the threshold for s0.z.Nb
markValue=0.64*usedIndex.size();
i=0;
while(i<usedIndex.size())
{
if(data3[(Integer)usedIndex.get(i)]>tshold[(Integer)usedIndex.get(i)])
{
counter++;
}
i++;
}
if(counter>markValue)
{
System.out.println("Spam Found");
}
else
{
System.out.println("Not Spam");
}
```

B.2: Counting bloom filter function

```
import java.io.File;
import java.io.FileNotFoundException;
import java.util.ArrayList;
import java.util.Random;
import java.util.Scanner;

public class ScannerClass {

	private File file;
	private stringProcess mystring;
	private CBF <String>cbf;//
	private String  Tdata;
	private int[] GT;

	public ScannerClass (File myFile, int cbfSize, String tdata)
	{
		file=myFile;
		cbf= new CBF <String>(cbfSize); //initializing instance of CBF  class
		mystring=new stringProcess();// initializing instance of StringProcess class
		Tdata=tdata;  // Choice of data, either training data or testing data
		//we initialize 14 different Global Titles, which are SMSCs serving in local areas
		GT = new int[]
{74543987,87654456,12099870,41100987,66557890,23123455,77889877,20022094,876544
53,12099878,41100984,66557897,23123453,77889878};
	}

	public int[] cbfData()
	{
		cbf.insertZeros();
		int k=0,c=0,n=0;
		Random r=new Random(10);  // this random generator is declared as a random
seed
				// to maintain consistencies in the random numbers used as global
titles
		String GTadr="";
try {
    Scanner scanner = new Scanner(file);
    while (scanner.hasNextLine())
    {
      String line = scanner.nextLine();
      int i=0,j=8,len;
```

```
GTadr="";
        String block="",compare="";
        boolean check;
        //processing the string
        line=mystring.processString(line);
        //.....// here we can add the MSC GT to the line, thus adding more
parameters
        // breaking the string into n-grams or n-shingles

       // append the GT address to the SMSs text. if
       // For testing purposes, we need to generate global titles for each SMS text.
       //for every 100 SMS, we assign a fixed global title to 14 SMS text and generate (100-14)
random global title for the other SMSC's.
        if(Tdata=="spam" && n<45)
        {
          line+="74543987";     //for this example we assume that the spammer's SMSC's GT
is 71405178
          n++;
        }
        else
        {
          if(k<14)
                                {

                //assign 14 global titles to every 100 SMS
                                        line+=GT[k];
                                }
          else
          {
                //This is a random generator method. it generates 8 digit numbers, which
represents the global title of the originating SMSC's
                while(c<8)
                                        {
                                                GTadr+=Integer.toString(r.nextInt(9));
                                                c++;
                                        }
                line+=GTadr;
          }
          if(k==100)
                                {
                                        k=0;
                                }
```

```
k++;
                                        c=0;
        }
        len=line.length();
        while (i<len)
        {
            if(len !=j-1)
          {
                    block=line.substring(i,j);
                            check=block.equals(compare);
                            if(check)
                            {
                                    //Skip empty n-grams
                            }
                            else
                            {
                            // inserting the binary of the hashed n-grams block into CBF
                     //if check is true, then we keep track of the index whose values are either
increased or decreased.
                            if(Tdata=="detector")
                            {
                                    cbf.check=true;
                            }
                            else
                            {
                                    cbf.check=false;
                            }
                            cbf.insertcbf(block);
                            }
          }
          else
          {
          i=0;
          j=0;
          }
            i++;
            j++;
        }
      }
      scanner.close();
   }
          catch (FileNotFoundException e)
          {
      e.printStackTrace();
          }
          return cbf.getData();
          }
           public ArrayList<Integer> index()
            {
                    return cbf.index();
            }
}
```

B.3: Hash code generator function

```
import java.util.ArrayList;
import java.util.Random;


public class CBF<V>
{
	private final int[] cbf;	//counting bloom filter data
		private final int cbfSize;	// size of CBF
		private int counter;		//keeps a count on the number of items in the cbf
		private final int noHash;	// number of hash bits
		public boolean check;
		private ArrayList <Integer> markIndx;

		//constructor for CBF class

		public CBF (int size)
		{
			cbf = new int [size];
			cbfSize=size;
			counter=0;
			noHash=2;
			markIndx=new <Integer> ArrayList();
			check=false;
		}

		//Initializing the cbf to 0s

		public void insertZeros()
		{
			int i=0;
			while (i<cbfSize)
			{
				cbf[i]+=cbf[i];
				i++;
			}
		}

		// Overriding the hashCode method

		public int hashCode()
			{

				int h1, h2, size,i=0;
				long hash;
				 Random r	= new Random();
```

```
size  = r.nextInt(cbfSize);
                        h1 = r.nextInt(cbfSize) % size;
                        h2 = r.nextInt(cbfSize) % size;

                        //Generate k different hash functions with a simple loop

                        hash = h1;
                        while(i<cbfSize)
                        {
                          hash = h1 + (h2*i);
                          i++;
                        }
                        return new Random(hash).nextInt();

                }

        //performing the hash operation

        private void performHash(V value, int action)
                {
                        Random r=new Random(value.hashCode());
                        int i=0;
                        while(i<noHash)
                        {
                                // hash position in the cbf array. and either insert or delete 1 bit
from that position

                                int cbfIndex= r.nextInt(cbfSize);
                                if(check==true)
                                {
                                markIndx.add(cbfIndex); //adding the used index to the arraylist
                                }
                                else
                                {
                                cbf[cbfIndex]+= action;
                                check=false;
                                }
                                i++;
                        }
                }

        public void insertcbf(V value)
        {
                performHash(value,1);
        }
        public void deletecbf(V value)
        {
                performHash(value,-1);
        }
```

```
public int[] getData()
	{
		return cbf;
	}
	// computing the threshold value for each bin
	public int[] Tshold(int[] cbf1,int[] cbf2)
		{
			int[] Tshd=new int [cbfSize];
			int i=0;
			while (i<cbfSize)
			{
				Tshd[i]=Math.max((cbf1[i]+cbf2[i])/2,1);
				i++;
			}
			return Tshd;
		}
	// returning the affected indexes in the cbf
	public ArrayList<Integer> index()
	{
		return markIndx;
	}
```