CBA
A new secure protocol for Web-Cellular Phone SMS communication

Developed By:

Zeeshan Shafi Khan

(19-FAS/PhDCS/S05)

Qutubuddin

(286-FAS/MSCS/M06)

Supervised By

Prof. Dr. Khalid Rashid

Department of Computer Science
Faculty of Basic and Applied Science
International Islamic University Islamabad
2008

A thesis submitted to the

Department of Computer Science

International Islamic University Islamabad
as a partial fulfillment of requirements for the award of
the degree of

MS in Computer Science

International Islamic University, Islamabad

Dated: 08/05/08

Final Approval

It is certified that we have examined the thesis titled "CBA: A new Secure Protocol for Web-Cellular Phone SMS Communication" submitted by Zeeshan Shafi Khan, Registration No. 19-FAS/PhDCS/S05, and Qutubuddin, Registration No. 286-FAS/MSCS/M06, and found as per standard. In our judgment, this research project is sufficient to warrant it is acceptance by the International Islamic University, Islamabad for the award of MS Degree in Computer Science.

Committee

External Examiner

Dr. Ismail Shah
Head, Department of computer Science
Iqra University, Islamabad.

Internal Examiner

Mr. Qaisar Javed
Assistant Professor / Incharge
Cisco Networking Academy,
Department of Computer Science
Faculty of Basic and Applied Sciences
International Islamic University, Islamabad.

19/5/8

Supervisor

Prof. Dr. Khalid Rashid
Head, Department of Computer Science
Comsats Institute of Information Technology
Islamabad.

In the Name of ONE

Who has all the names, and who does not need any name

Declaration

We hereby declare that this work, neither as a whole nor as a part has been copied out from any source. It is further declared that we have developed the protocol and the accompanied report entirely on the basis of our personal efforts and under the sincere guidance of our supervisor Prof. Dr. Khalid Rashid. If any part of this project is proved to be copied out from any source or found to be reproduction of some other project, we shall stand by the consequences. No portion of the work presented in this dissertation has been submitted in support of any application for any other degree or qualification of this or any other university or institute of learning.

Qutubuddin

(286-FAS/MSCS/M06)

Zeeshan Shafi Khan
(19-FAS/PhDCS/S05)

Acknowledgment

All praise to Almighty Allah who has guided us in undertaking work on CBA (A secure protocol for web-cellular phone SMS communication) and has helped us through each step when there was no hope of pulling through.

We remember with thanks the pioneers who established educational infrastructure in the country enabling us to reach the stair of this effort. This step was not possible without earlier struggle of the great researchers leading the world to more civilized position. We are thankful to our supervisor Prof. Dr. Khalid Rashid for his kind help and supervision. We shall remember his support, building our capacity for research methodology and always guiding us to next bold step.

We would also like to thank our colleagues and faculty members of the university for their help and support, with special thanks to Mr. Rehan for helping us in the coding and Mr. Bashir for facilitating us in implementation of the CBA.

Finally we owe a lot to our beloved parents and our families for their love, guidance, moral and financial support.

Zeeshan Shafi Khan

Qutbuddin

Project in Brief

Project Title:	**CBA: A new secure protocol for Web-Cellular Phone SMS communication**
Undertaken By:	**Zeeshan Shafi Khan** **Qutubuddin**
Supervised By:	**Prof. Dr. Khalid Rashid**
Start Date:	**September 2007**
Completion Date:	**April 2008**
Tools and Technologies:	**Microsoft Visual C#.Net**
Documentation Tools:	**MS word, MS Excel**
Operating System:	**MS Windows XP Professional**
System Used:	**Pentium 4, 1.73 GHz**

Abbreviations Used

AUC	Authentication Center
AMPS	Advance Mobile Phone System
ANSI	American National Standard Institute
BSC	Base Station Controller
BSS	Base Station System
BTS	Base Transceiver Station
CAMEL	Customized Applications for Mobile networks Enhanced Logic
CBA	Call Back Authentication
CCH	Control Channel
CDMA	Code Division Multiple Access
DNS	Domain Name System
DoS	Denial of Service
EFR	Enhanced Full Rat
EIR	Equipment Identity Register
GIWU	GSM Internetworking unit
GMSC	Gateway Mobile Service Switching Center
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communication
HLR	Home Location Register
ISP	Internet Service provider
MHz	Mega Hertz
MMS	Multimedia Messaging Service
MSC	Mobile Service Switching Center
MSN	Mobile Service Node
MXE	Message Center
OMC	Operation and Maintenance Center
OSS	Operation and Support System
PCS	Personal communication system
PLMN	Public Land Mobile Network
PSTN	Public Switched Telephone Network
RF	Radio Frequency
SMS	Short Messaging Service
SS7	Signaling System 7
TCH	Traffic Channel
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
UDP	User Datagram Protocol
UMTS	Universal Mobile Telephone Service
VASP	Value added Service provider
VLR	Visitor Location Register
WFQ	Weighted Fair Queue

Abstract

In order to provide essential data services, the cellular networks have opened interface with Internet enabling communication with the users beyond the coverage of their wireless signals across the globe. Besides economical expansion of the GSM services, this interface has given birth to new security challenges. For example recent research has revealed that an adversary equipped with little more than a cable modem can block all voice communication in a metropolitan area or even in entire country of the size USA by sending enormous SMS traffic via Internet. To overcome above threat in particular and other client-server communication problems in general, a new secure protocol (CBA) for web to phone SMS communication is proposed that operates on proposed **Two-Tier Server Architecture** using a proposed **Pull Data traffic control mechanism**. The protocol is aimed to throat-knot malicious traffic at GSM-Internet interface gateway. Two Tier Server Architecture has two operating modules i.e. Tier-I called Token Server responsible for authenticating the legitimate clients and Tier-II called CBA Server responsible for providing service to the authenticated clients. Token server listens client requests via UDP, performs authenticity checks and sends the list of valid users to CBA server. After authentication process, the client is switched to listening mode for accepting connection initiated by the CBA Server.

Pull Data traffic control mechanism authorizes the CBA Server to initiate connection with the clients and pull the SMS data from queues maintained by the clients, on turn-by-turn basis. The CBA server maintains a list of clients; however the SMS traffic queues are maintained by the client machines. Since the flow of SMS traffic is controlled by the CBA Server, clients never can push traffic to server as prevailing in traditional client server communication. Clients in service always remain transparent of the trouble in case of DoS attacks. The attack possibility is confined to only one service i.e. Web-Cellular Phone SMS communication that ensures 100 % smooth operation of other GSM Services. The Servers are made resilient enough to defend the Internet originated SMS flood attack on Web-Cellular Phone SMS communication. The Protocol works with existing GSM architecture without any major change in existing system or putting exorbitant cost of deployment.

Table of Contents

List of Figures

1. Introduction

1. Introduction

GSM networks have global importance because of their major role in economies and social infrastructure of the nations. GSM companies have expanded their service from traditional voice service to TV broadcasts, Internet connectivity, email, short messaging and other value aided services. SMS is considered as future driver of the E-commerce, education, management and services due to its inherited characteristics of personal identification, low cost, location awareness and convenience of "any where any time". Interface of GSM networks with Internet has enabled delivery of many services by the operators at economical rate and communication beyond the range of GSM infrastructure on one hand, while inheriting the vulnerabilities and problems of Internet on the other hand. "Web to Cellular Phone SMS Chat", today's popular service provided by almost all the cellular companies is more utilitarian for SMS communication by the users across the globe especially the users beyond the coverage of particular GSM operator.

However, researchers have identified vulnerability of GSM networks to Internet originated SMS bursts that can result in denial of voice service to the cities of size Washington DC. Jamming of the GSM networks is considered server problem in view of resultant impact on economies and social infrastructure of the countries throughout the world. This critical situation is attracting the attention of the researchers of the field i.e. Computer Science to find a reasonable, viable and acceptable solution of the problem.

Provision of security against the attack without any compromise on availability of services, change in GSM architecture, exorbitant cost and without involvement of the users are the major challenges. Solutions that suggest decrease in priority level of SMS against the voice calls are not acceptable because of its use in real-time commercial applications like stock bidding and auctions etc. The solutions that suggest reserved bandwidth for voice and SMS are not viable because of resultant under utilization of the available bandwidth and implementation cost. Any solution that suggests participation of the GSM user is not acceptable because even low probability of customer discouragement is not tolerable in this competitive commercial age.

1.1 Global System for Mobile Communication (GSM)

Having 750 mobile networks and 2.83 billion subscribers across the globe, wireless networks are considered leaders in coverage. Amongst the wireless networks, GSM is the largest with 80 % share in total number of wireless subscribers. With 65 million subscribers by July 2007, Pakistan is ranked third in the list of top growth countries after China and India [2]. For its exceptional performance in establishment of mobile communication sector, the country has also won GSM Association Prestigious Government Award 2006 [3].

Because of their widest use, mobile networks are going to play vital role in productivity, economies and social infrastructure of the countries. Telecommunication companies have expanded their services from voice to TV broadcasts, Internet connectivity, email, short messaging and other value added services etc.[1]

Mobile phones are helpful in disaster management. During event of an emergency, disaster response crew can locate trapped people using the signals from their mobile phones or the small detonator of flare in the battery of every cell phone. The rescue aircrafts can detect the geographical position of trapped people carrying mobile even beyond the reach of coverage by radio signals attempting to connect the base station.[1]

The International Engineering Consortium defines GSM as "Global System for Mobile communication (GSM) is globally accepted standard for digital cellular communication GSM is the name of standardization group established in 1982 to create a common European Mobile telephone standard intended to formulate specification for a pan-European mobile cellular radio system operating at 900 MHz." Table 1 shows the major milestones achieved by the GSM technology since 1982.

Year	**Milestone**
1982	GSM formed
1986	Field test
1987	TDMA chosen as access method
1988	Memorandum of understanding signed
1989	Validation of GSM system
1990	Preoperation system
1991	Commercial system startup
1992	Coverage of large cities/airports
1993	Coverage of main roads
1995	Coverage of rural areas

Table 1.1: Milestones achieved by GSM [4]

GSM is a cellular network, which means that mobile phones connect to it by searching for cells in the immediate vicinity. GSM networks operate in four different frequency ranges. Most GSM networks operate in the 900 MHz or 1800 MHz bands. Some countries in the Americas (including Canada and the United States) use the 850 MHz and 1900 MHz bands because the 900 and 1800 MHz frequency bands were already allocated. [1]

In the 900 MHz band the uplink frequency band is 890–915 MHz, and the downlink frequency band is 935–960 MHz. This 25 MHz bandwidth is subdivided into 124 carrier frequency channels, each spaced 200 KHz apart. Time division multiplexing is used to allow eight full-rate or sixteen half-rate speech channels per radio frequency channel.

There are eight radio timeslots (giving eight burst periods) grouped into what is called a TDMA frame. Half rate channels use alternate frames in the same timeslot. The channel data rate is 270.833 Kbit/s, and the frame duration is 4.615 ms. [1]

The transmission power in the handset is limited to a maximum of 2 watts in GSM850/900 and 1 watt in GSM1800/1900. [1]

GSM has used a variety of voice codecs to squeeze 3.1 kHz audio into between 5.6 and 13 kbit/s. Originally, two codecs, named after the types of data channel they were allocated, were used, called Half Rate (5.6 kbit/s) and Full Rate (13 kbit/s). These used a system based upon linear predictive coding (LPC). In addition to being efficient with bit rates, these codecs also made it easier to identify more important parts of the audio, allowing the air interface layer to prioritize and better protect these parts of the signal. [1]

GSM was further enhanced in 1997 with the Enhanced Full Rate (EFR) codec, a 12.2 kbit/s codec that uses a full rate channel. Finally, with the development of UMTS, EFR was refactored into a variable-rate codec called AMR-Narrowband, which is high quality and robust against interference when used on full rate channels, and less robust but still relatively high quality when used in good radio conditions on half-rate channels. [1]

There are four different cell sizes in a GSM network—macro, micro, pico and umbrella cells. The coverage area of each cell varies according to the implementation environment. Macro cells can be regarded as cells where the base station antenna is installed on a mast or a building above average roof top level. Micro cells are cells whose antenna height is under average roof top level; they are typically used in urban areas. Picocells are small cells whose coverage diameter is a few dozen meters; they are mainly used indoors. Umbrella cells are used to cover shadowed regions of smaller cells and fill in gaps in coverage between those cells. [1]

Cell horizontal radius varies depending on antenna height, antenna gain and propagation conditions from a couple of hundred meters to several tens of kilometers. The longest distance the GSM specification supports in practical use is 35 kilometres (22 mi). There are also several implementations of the concept of an extended cell, where the cell radius could be double or even more, depending on the antenna system, the type of terrain and the timing advance. [1]

Indoor coverage is also supported by GSM and may be achieved by using an indoor picocell base station, or an indoor repeater with distributed indoor antennas fed through power splitters, to deliver the radio signals from an antenna outdoors to the separate indoor distributed antenna system. These are typically deployed when a lot of call capacity is needed indoors, for example in shopping centers or airports. However, this is not a prerequisite, since indoor coverage is also provided by in-building penetration of the radio signals from nearby cells. [1]

Followings are the main components of GSM architecture:

- **Home Location Register (HLR)**: "HLR is a database used for storage and management of subscriptions. Since it stores permanent data about the subscribers, it is considered as the most important database. It includes subscriber's service profile, location information and activity status. On buying the subscription a registry is made in the HLR of that operator". [4]
- **Mobile Service Switching Center (MSC):** "The MSC performs the telephony switching functions of the system. It controls calls to and from other telephone system.[4]
- **Visitor Location Register (VLR):** "VLR is a database that contains temporary information about subscribers that is needed by the MSC in order to service visiting subscribers. The VLR is always integrated with MSC. When a mobile set roams into a new MSC area the VLR connected to that MSC will request data about the mobile station from the HLR. Later, if the mobile station makes a call the VLR will have the information needed for call setup without having to interrogate the HLR each time".[4]
- **Authentication Center (AUC):** "It provides authentication and encryption parameters that verify the user's identity and ensure the confidentiality of each call. The AUC protects network operators from different types of frauds found in Today's cellular world".[4]
- **Equipment Identity Register (EIR):** "It is a database that contains information about the identity of mobile equipment that prevents calls from stolen, unauthorized or defective mobile stations. The AUC and EIR are implemented as stand alone nodes or as a combined AUC/EIR node".[4]
- **Mobile Service Node (MSN):** "The MSN is the node that handles the mobile intelligent network (IN) services".[4]
- **Gateway Mobile Service Switching Center (GMSC):** "A gateway is a node used to interconnect two networks. The gateway is often implemented in an MSC. The MSC is then referred to as the GMSC".[4]
- **GSM Internetworking Unit (GIWU):** "The GIWU consists of both hardware and software that provides an interface to various networks for data communication. Through GIWU user can alternate between speech and data during the same call".[4]
- **BSC:** "BSC provides all the control functions and physical links between the MSC and BTS. It is high capacity switch that provides functions such as handover, cell configuration data, and control of Radio Frequency (RF) power level in the Base transceiver stations. A number of BSCs are served by an MSC".[4]

- **BTS:** "BTS handles the radio interface to the mobile station. The BTS is the radio equipment needed to service each cell in the network" [4]. A group of BTS are controlled by a BSC. The Air Interface is divided into two general components – Control Channel (CCH) and Traffic Channel (TCH) [1]. It helps to think of control channels as a very small portion of radio frequency that allow cellular towers to send information pertaining to call setup, SMS delivery and network conditions (such as the availability of traffic channels) to mobile phones. Traffic channels are instead used to carry actual voice conversations after they have been established via the control channels. Figure 2 below, gives an intuitive representation of this setup. [5]

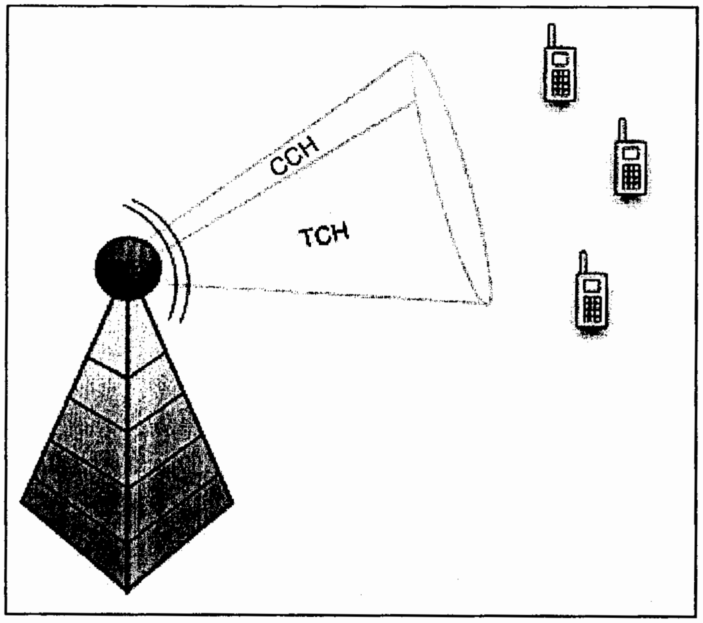


Fig.11 Air Interface of GSM [5]

GSM Services

Presently mobile phone companies are providing following services to their customers:

A. Voice calls

B. Short Messaging (SMS) and related value added services

   a. Corporate SMS for sending messages in bulk

   b. Common SMS intended for individual users

      ✓ Cellular Phone to Cellular Phone SMS

      ✓ Internet to cellular phone SMS

         i. Web to SMS Chat service

ii. Email to SMS Service

c. SMS-based on-demand value added services

- ✓ Entertainment information services
- ✓ Mobile banking
- ✓ News headlines & sports update

d. Email service

C. Multimedia Messaging Service (MMS)

a. Voice Mail

b. Fax Mail

c. Picture Messaging

D. Internet connectivity through General Packet Radio Service (GPRS)

E. Television broadcasts

Short Messaging Service (SMS)

Commercially SMS is a massive industry in 2006 worth over 80 Billion dollars globally. SMS has an average global price of 11 cents and maintains a near 90% profit margin. [1]

Short Messaging Service (SMS) has become major application to drive the mobile business because of its popularity in interpersonal, management, home, education and commercial communication. It is viewed as fast, economical and reliable method of communication when other means appear unavailable.

During the incident of 11th September 2001 the nature of text messaging proved to be far more utilitarian. Mobile companies reported that number of calls was far greater than capacity for voice communications in the affected areas. Due to TCH saturation voice services were almost unavailable; however SMS messages were still successfully received in even the most congested regions because the control channel responsible for their delivery remained available. [5]

SMS messaging has got wide scope in commerce applications due to inherited characteristics like Low cost, convenience of "any time and any where", personal identification and location awareness. A mobile device is able to receive or send a message at "any time" regardless of whether a voice or data call is in progress. In case the device is off, the messages are stored and delivered when it is on again. Since the SMS is received on mobile devices, it enjoys the availability "any where" making it most suitable for information alerting services, marketing campaigns and real-time auctions etc.

Weather reports, flight, train & bus information, stock business information, news headlines, real-time surveys, and auctions are examples of applications [6]. SMS can be used for identification purposes in information inquiries, reservation services and to provide remote point of sale. For example a consumer can purchase an item through the widely deployed automatic vending machine enabled with "Dial-an-Item" feature.

Presently SMS based services including Web-to-SMS Chat service is being managed by some cellular operators through third party service providers. Clients are offered to download an application program from the web page of cellular operator hosted by the third party service provider. The application program enables client's communication with third party server which acts as mediator between the client and SMSC. Client application program establishes TCP connection and sends SMS to third party server which forwards the same to SMSC for delivery to destination.

SMS is one of the most popular services of GSM. There is a dedicated message center that stores and forwards the SMS communication. Gateway MSCs are used by the Service Centre to communicate with the Public Land Mobile Network (PLMN) or PSTN. [4]

Subscriber originated messages are transported from a handset to a Service Centre either destined for mobile users, subscribers on a fixed network, or Value-Added Service Providers (VASPs) also known as application-terminated. [4]

Most digital mobile phones and some personal digital assistants support the Short Message Service. Most of the handsets are text enabled so they receive messages in text format, however messages can be delivered to non-enabled phones using text-to-speech conversion.

SMS derives its benefit from two absolute advantages compared to any other form of communication. SMS is the fastest form of communication if measured by actual communication throughput including instances such as the counterpart not being able to take a call, being out of radio coverage, listening to voicemail, put on hold etc. [4]

Web to Phone SMS chat service

SMS is one of the most popular services among the telecommunication industry. In order to facilitate users almost every cellular service provider has its web interface to enable clients for sending/receiving SMS to/from cellular phones. A user needs to download an application program from the provider's website that executes on client machine and establishes connection with the server machine. Users are provided with user friendly graphic interface for sending/receiving there text messages to desired cellular phone numbers. This service enables the users to have communication with the desired particular cellular telephone numbers irrespective of the geographical position of the user throughout the globe and without having cellular phone. The service overcomes the limit of coverage distance of particular cellular operator that depends upon physical infrastructure developed. Since the service is free of cost and easy to use, it is used by substantial number of users having Internet connectivity. Connecting cellular network with the Internet has enabled these networks to utilize global communication infrastructure of the Internet. However this interface has given birth to some security threats simultaneously as discussed in 3.1.

Phone Networks were designed to operate in homogenous isolated systems for providing only voice services to customers. Because of no interaction with heterogeneous networks, these networks were not exposed to variety of security threats except a few i.e. signal jamming, eavesdropping, physical damage by cutting wires or explosion etc. But now due to interaction with non- homogenous networks like Internet etc, the attacks to set the cellular networks non-responsive in the entire USA or even cyber warfare attacks capable of denying voice and SMS based services to an entire continent are also feasible.

1.2 Problem Domain

Identification of the vulnerability of the GSM networks to Internet originated SMS attack resulting in jamming the entire GSM network leading to denial of voice service is point of concern for the world. Because of its severity that it can give birth to cyber warfare among the countries, the problem has created uncertainty among the Stakeholders of Mobile Commerce. In the presence of this threat, it is almost impossible to build confidence amongst risk wary business community for potential investment in Mobile Commerce and dependent service sectors. Absence of the remedial measures to the problem may lead to status quo against revolutionary development in SMS commerce, dependent services and research in the field. Because of its different architecture and nature of services, Internet based defending mechanisms are insufficient to protect mobile networks. This research focuses over the acceptable and viable solution to the problem.

Denial of Service

A denial-of-service attack (DoS attack) is an attempt to make a resource unavailable to its intended users. It generally comprises the concerted, malevolent efforts of a person or persons to prevent an Internet site or service from functioning efficiently or at all, temporarily or indefinitely [1]. Although this type of attack was considered problem for Internet only, but is inherited by the cellular phone networks due to interface with the Internet.

One technique of attack is to saturate the target machine with dummy requests such that it can not respond to legitimate traffic or it becomes so slow that the resources become almost unavailable. DoS attacks are implemented by:

- Forcing the targeted computers to reset, or consume its resources such that it can no longer provide its intended service
- Obstructing the communication media between the intended users and the victim so that they can no longer communicate adequately

DoS attacks can also lead to problems in the network 'branches' around the actual computer being attacked. For example, the bandwidth of a router between the Internet and a LAN may be consumed by DoS, compromising not only the intended computer, but also the entire network [1].

A "denial-of-service" attack is characterized by an explicit attempt by attackers to prevent legitimate users of a service from using that service. Examples include: [1]

- Flooding a network, thereby preventing legitimate network traffic
- Disrupting a server by sending more requests than it can possibly handle, thereby preventing access to a service
- Preventing a particular individual from accessing a service
- Disrupting service to a specific system or person

Attacks can be directed at any network device, including attacks on routing devices and Web, electronic mail, or Domain Name System servers. A DoS attack can be perpetrated in a number of ways. Basic types of attack are: [1]

- Consumption of computational resources, such as bandwidth, disk space, or CPU time
- Disruption of configuration information, such as routing information
- Disruption of state information, such as unsolicited resetting of TCP sessions
- Disruption of physical network components
- Obstructing the communication media between the intended users and the victim so that they can no longer communicate adequately

Different types of DoS Attacks

There are many types by which a DoS attack can be launched. Few of them are discussed as under:

1. Smurf attack

Smurf Attack generates a lot of traffic at victim's end. The attacker sends a lot of ICMP echo messages (ping) to the IP broadcast address by spoofing the IP adress of the vicitim

One technique of attack is to saturate the target machine with dummy requests such that it can not respond to legitimate traffic or it becomes so slow that the resources become almost unavailable. DoS attacks are implemented by:

- Forcing the targeted computers to reset, or consume its resources such that it can no longer provide its intended service
- Obstructing the communication media between the intended users and the victim so that they can no longer communicate adequately

DoS attacks can also lead to problems in the network 'branches' around the actual computer being attacked. For example, the bandwidth of a router between the Internet and a LAN may be consumed by DoS, compromising not only the intended computer, but also the entire network [1].

A "denial-of-service" attack is characterized by an explicit attempt by attackers to prevent legitimate users of a service from using that service. Examples include: [1]

- Flooding a network, thereby preventing legitimate network traffic
- Disrupting a server by sending more requests than it can possibly handle, thereby preventing access to a service
- Preventing a particular individual from accessing a service
- Disrupting service to a specific system or person

Attacks can be directed at any network device, including attacks on routing devices and Web, electronic mail, or Domain Name System servers. A DoS attack can be perpetrated in a number of ways. Basic types of attack are: [1]

- Consumption of computational resources, such as bandwidth, disk space, or CPU time
- Disruption of configuration information, such as routing information
- Disruption of state information, such as unsolicited resetting of TCP sessions
- Disruption of physical network components
- Obstructing the communication media between the intended users and the victim so that they can no longer communicate adequately

Different types of DoS Attacks

There are many types by which a DoS attack can be launched. Few of them are discussed as under:

1. Smurf attack

Smurf Attack generates a lot of traffic at victim's end. The attacker sends a lot of ICMP echo messages (ping) to the IP broadcast address by spoofing the IP adress of the vicitim as a source address. Now when all the machines those are the part of that broadcast domaion recieve the ICMP echo request they sends an ICMP echo reply to the sender. Since attacker use the IP address of the victim as a source address so all the replies arrived at the victim results in Denial of service. There are two methods to prevent this type of attack. First is to configure all the hosts and router in such a way that they should not reply to the broadcasted ICMP echo requests. Second technique is to configure routers in such a way that they should not forward the ICMP echo request sent on the broadcast address. A smurf amplifier is a computer network that can be used in a smurf attack. [1]

2. Ping flood

Ping flood is to send enomerous numbers of ping request to the victim. Reply to these requests by victim results in DoS. This attack is very easy to launch, what the attacker is required to have is more bandwidth than the victim. This attack can be prevented by using Firewalls. Common solution to prevent this problem is to no reply to ping requests beyond threshold limits. [1]

3. SYN flood

When a client and server want to eastablish a TCP connection, a three way handshake takes place between them. Theses three steps are

- Client sends a Syn request
- Server acknowledge his Syn and sends it its on Syn packet
- Client acknowledge the server's Syn packet

To launch a Syn flood attack an attacker sends a Syn packet, when server recieves the Syn packet it reserves resources for that connection and acknowledge the client plus sends its own Syn, now when client gets the Syn it does not send acknowledgement. So the resources remain occupied until the server timeout. Now attacker sends a bulk of Syn packet but does not complete the three way handshake. Server has a backlog that tells the limit that how many half open TCP connection it can serve at atime. Now when attacker fills this backlog with the half open connection so now a legitimate user is not been able to eastablish a TCP connection with that server. Few mechanism to prevent or reduce the effect of this attack are:

- Increase the Backlog
- Reduce the value of Timeout
- Launch a firewall between client and server
- Do not reserve resources until connection is fully eastablished

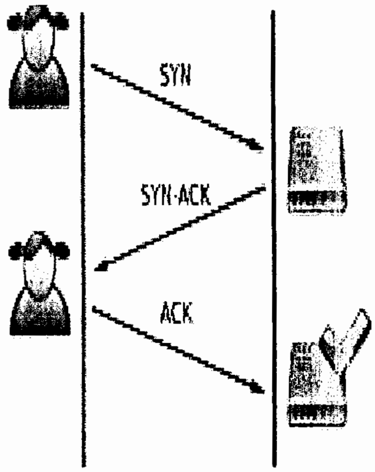


Fig.1.2: Normal Connection Establishment

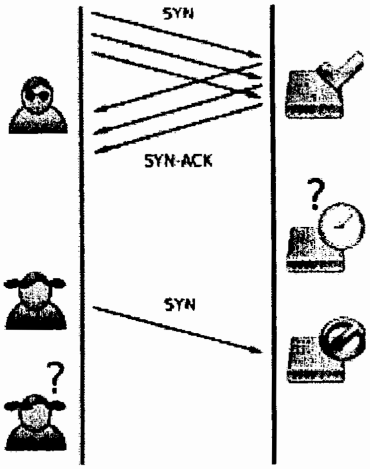


Fig.1.3: Attack through Syn Flood

4. Teardrop attack

Teardrop attack sends over-sized IP packets to the destination. The destination recieves these packet and perform exhaustive reassembly resulting in operating system crash. Almost all the Operating sytem of Microsoft and few versions of Linux are also valunerable to this type of attack. [1]

5. Nuke

It is an old denial-of-service attack that consistss of sending invalid ICMP packets to the destination, thus slowing down the destination computer until it comes to stop. [1]

6. Distributed denial of service attack (DDoS)

DDoS occurs when multiple compromised systems flood the bandwidth or resources of a targeted system, usually one or more web servers. These systems are compromised by attackers using a variety of methods [1].

Malware can carry DDoS attack mechanisms; one of the more well known examples of this was MyDoom. Its attack mechanism was triggered on a specific date and time. This type of DDoS involved hardcoding the target IP address prior to release of the malware and no further interaction was necessary to launch the attack.

It is important to note the difference between a DDoS and DoS attack. If an attacker mounts a smurf attack from a single host it would be classified as a DoS attack. In fact, any attack against availability would be classified as a Denial of Service attack (e.g. using High-energy radio-frequency weapons to render computer equipment inoperable, would be a DoS attack, albeit an exotic one.). On the other hand, if an attacker uses a thousand zombie systems to simultaneously launch smurf attacks against a remote host, this would be classified as a DDoS attack. [1]

The major advantages to an attacker of using a distributed denial-of-service attack are that multiple machines can generate more attack traffic than one machine, multiple attack machines are harder to turn off than one attack machine, and that the behavior of each attack machine can be stealthier, making it harder to track down and shut down. These attacker advantages cause challenges for defense mechanisms. For example, merely purchasing more incoming bandwidth than the current volume of the attack might not help, because the attacker might be able to simply add more attack machines. [1]

7. **Distributed reflected denial of service attack**

This is a way to send forged requests to a large number of computers by using IP address of the victim as source IP address. When those computers reply to that request a huge traffic takes place at victim site resulting in Denial of Service. [1]

DOS attack on GSM network

Long ago the phone networks were isolated haivng interface with homogenous networks limited to voice service. The networks were vulnerable to traditional threats like signal jamming, evesdropping, physical damamge and other similar attacks having limited impact. However the interface of these networks with Internet like hetrogenous networks has resulted in variety of new services as well as threats like UDP bursts on GPRS Ips, Internet originated SMS bursts and targeted SMS attacks discussed in chapter 3. Although various techniques are proposed by the researchers to defend against the DOS attack on the Internet, however these techniques are not sufficient to defend the GSM networks against the Internet originated SMS floods because of architural difference between the Internet and the GSM. Here major problem that we address is Internet originated SMS bursts injected by an adversary in GSM networks to saturate the bandwidth of the control channel leading to denial of voice service.

1.3 Proposed Approach

Various approaches including change in GSM mechanisms and channels etc are possible to tackle the problem. However we propose remedial/preventive security measures at GMS-Internet Interface Gateway to filter the malicious SMS traffic. The approach suggests entry of legitimate Internet originated SMS traffic into GSM network that puts responsibility of defending the attack over the Internet Interface Gateway.

The first step of the defense is identifying the attack using traffic threshold parameter for the purpose as proposed by different anti DOS techniques. Second step is activation of the security and authentication of the legitimate users. The third step is identification of the legitimate traffic to be serviced and malicious traffic to be filtered. The authenticated user has to be kept controlled because the Internet originated SMS attack on the GSM is even possible with limited resources.

Example of this type of security in normal life is security operating at a city/area gate that only allows entry of persons into the city/area having valid authority, eliminating the cost of deploying security at all vital points within the city/area. Here someone can face problem of delay during checking at the city/area gate, but as soon as entry is allowed the person can avail all the services available within the city/area irrespective of the intensity of the attack at city gate.

The approach further suggests keeping Internet users as well as GSM network transparent of the operational complexities of the security mechanisms. Once authenticated, the user remains in service irrespective of the attack traffic threshold. Trade strategies suggest that the supply should not push the commodities to market, but it is discretion of the market to pull the supply from the stocks in accordance with the demand. Similarly here server should pull the data from the client's queue where the client should maintain the queue rather than allowing client to push data traffic to server. The server should not be held responsible for maintaining the data traffic queues.

1.4 Thesis Outline

The remainder of the thesis is organized as follow. In chapter 2 we review the previous work in this domain. In chapter 3 we highlight the requirement analysis of our proposed protocol, while the chapter 4 elaborates the design requirements. In chapter 5 implementation issues are discussed and steps of the CBA protocol are described through sequence diagrams. In chapter 6 we analyze our protocol under different scenarios and parameters and results are presented in the form of graphs. Finally in Chapter 7, we present our conclusions with some open questions and future work.

2. Literature Survey

2. Literature Survey

Research in the field of the business management and commerce is considering the SMS as the driver of the future E-commerce because of its inherent characteristics like personal identification, location awareness any time access to the customers. But simultaneously the sensitivity of commercial operations against the security threats is realized. Where on the one hand, the risk wary business community and investors are not in position to tolerate security risk, the interface of the Internet with GSM networks has widened the scope of DOS attacks from Internet to GSM networks on the other hand. We studied thesis, research publications, articles, books etc in order to determine what has been done in the area in which our problem domain resides.

2.1 Related Research/Technologies

We divide our literature survey into following broad categories based on the particular research area:

- Denial of Service Attack
- Role of SMS in Business and commerce
- TCP and UDP based Client-Server communication
- SMS based DoS attack over cellular networks

2.1.1 Denial of Service Attack

Jun Li et.al. (2002) described a new protocol, named Source Address Validity Enforcement (SAVE) that can provide routers with the information needed for source address validation [7]. SAVE messages propagate valid source address information from the source location to all destinations allowing each router along the way to build an incoming table of valid source addresses. Through simulation they show the results of the protocol that routers successfully generate the list of authentic IP addresses and filter the malicious traffic at attack time on the basis of these lists. This mechanism needs to be implemented over all the routers of the world those are part of the Internet which is practically difficult.

Wang et.al. (2004) presented congestion puzzle as a countermeasure to defend against bandwidth-exhaustion attacks. In their mechanism the routers impose check puzzles [14]. Since puzzle based attacks require vast amount of resources at attacker's side so authors argue that this mechanism can mitigate the bandwidth-exhaustion attack launched through Zombie computers. The mechanism presented is applicable to packet switching networks and has no application scope in GSM.

Sirivianos et.al. (2006) proposed a protocol, named "A Source Authenticating Network Architecture Limiting DoS Attack (SANDLA)" that enforces user authentication on the IP level [15]. According to the authors, periodically stamping each IP packet with a unique Identity based signature of the sender. When a spoofed packet is transmitted over the network they are detected as soon as they arrive at the source authenticating router by matching signatures. Authors discuss the use of capabilities to enable the receiver to decide the senders those are allowed to send. Accordingly SANDLA is intended to limits the impact of DoS Attack. The application of this mechanism is feasible subject to Inter-cooperation between ISPs for accepting the overhead of adding/ stripping off the extra header attached by SANDLA.

Goodrich [22], Snoeren et.al. [23], sung et.al [24], wang et.al. [25,26], Basheer et.al. [27], Li et.al. [28], Abraham et.al. [29] all proposed traceback mechanism to detect and prevent the DoS attack

Wang [30] proposed pushback mechanism in order to detect and prevent DoS attack

2.1.2 Role of SMS in Business and Commerce

Stone et.al. (2002) have analyzed some field results to assess the comparative popularity level of World Wide Web, Email and SMS messaging [8]. The researchers have concluded that SMS is more popular and is an effective future tool for education. Their findings highlight the importance of SMS, which leads to significance of its security arrangements.

Ching et.al. (2002), has proposed SMS based transport information system intended to provide requisite information by transport companies to commuters about bus schedule, route and point to point travel solutions [10]. The system is able to automatically determine the sender's location using GSM base station. On a Query-SMS to system about route, schedule and availability of transport to desired destination, the commuters are replied through SMS about various options, routes, schedule of transport available and recommend solution. This research highlights potential use of SMS in travel services that again necessitate for security measures to enable its practical implementation.

HengXu et.al. (2002) have derived hypothesized success factors for emerging SMS commerce [11]. They are of the view that success of SMS commerce is positively related to infrastructure, interoperate-ability, low cost, high penetration of mobile phones, cooperation, government support and secure platform etc. According to them guaranteed security would be the major factor of success. The research can be considered as an indirect appeal from business sector to IT sector for security arrangements to enable revolutionary development in SMS Commerce.

Ke Wan (2003) has presented application of SMS in franchise retail business to enhance its competitive advantages [12]. He has comparatively analyzed SMS with other means of communication in terms of capital cost, operational cost, training cost, security, accessibility, and information volumes. The results show that SMS is best choice for franchise business if security needs are met.

Sang et. al. (2003), have proposed SMS application in remote controlling and monitoring of different equipment [13]. They are of the view that information can be retrieved from a database via query-SMS. The research shows significance of SMS in terms of its wide range application scope.

Kalinga et. al. (2006), have proposed Remote Controlling and Monitoring System to control electric circuitry through SMS using Microcontroller [16]. Proposed System enables authorized users to remotely ON/OFF different appliances to monitor their status through exchange of SMS messages. The research concludes that with slight modifications, the system can be applied to many applications such as remote sensing of meteorological information and vehicle security system with location finder. By inventing useful application of SMS, the research has given wide scope to SMS that indirectly necessitate for security features to enable its smooth operations.

Zerfos et. al. (2006) have presented a measurement study of SMS traffic i.e. short message size distribution, their arrival process, latency, number of messages sent/received by mobile phones and External Short Message Entities [17]. The results of the study show that 94.9 % messages are successfully delivered and 73.2 % messages reach the destination in less than 10 seconds. The study is based on the one week log entries of Charging Data Records. The research reveals that SMS failure ratio of 5.1% may further increase substantially in case of the problem discussed in [5].

Meng et. al. (2006) have examined reliability by analyzing the Charging Data Records of a nationwide cellular network. Research shows that bulk messaging can significantly affect the reliability of cellular system because bulk messages are injected into cellular network via wired links having high bandwidth as compared to wireless interface; responsible for delivery of those messages to cellular hosts [18]. They conclude that with 5.1 % failure ratio under normal conditions, SMS service is not more reliable as compared to other conventional communication systems. The research reveals that the reliability of SMS communication is at high risk in case of problem discussed in [5].

2.1.3 TCP and UDP based Client-Server Communication

Rabinovich et.al. presented a new protocol named as DHTTP [9]. In this protocol he mentioned that there are certain web requests those are too small. So it is not feasible to open TCP connection for those requests. So he proposed that in order to get web service from the server DHTTP sends a UDP request to the server. Server analyzes what client wants from him and on the basis of this information server decides whether to send data through UDP or through TCP. The main purpose of the author is to reduce the load from the server.

2.1.4 SMS based Denial of Service Attack over Cellular Networks

William Enck et. al. (2005) identified the vulnerabilities and risk originated from the interface of cellular network with Internet [5]. They have investigated the feasibility of exploiting the vulnerabilities by an adversary equipped with limited resources. The authors have demonstrated that sending enormous malicious SMS traffic originated from Internet deprives the bandwidth capacity of the cellular network resulting in denial of voice as well as legitimate SMS traffic. The attack that they identified can be launched from a single computer equipped with a dialup modem. Through the generation and use of large hit lists it is possible to deny the voice services of the cities like Manhattan and Washington D.C. Moreover voice traffic of entire United States can be blocked with the resources available to medium sized zombie networks.

Authors also discussed few methods of creating hit lists. Some of the methods are NPA/NXX, Web scraping, web interface attraction. According to their calculations it requires only 165 messages per second to deny the voice traffic of Manhattan city and 240 messages per second for Washington D.C. Furthermore the author calculates that it requires 3, 25,525 messages per second to attack on entire USA's cellular traffic. They also mentioned about targeted attacks in which the purpose of the attacker is to attack on a particular user rather than attacking on entire network.

Against these attacks the authors suggested few solutions to overcome the attack. These solutions include Separation of voice and data, Rate limitation, resource provisioning. Overall the main purpose of the author to write this paper is to describe the problem. At the end he presented few temporary solution but they emphasize on finding a better and complete solution of the problem.

Serror et. al. (2006) have identified vulnerability in cellular networks that offer Voice & Data services like General Packet Radio Service [19]. The research reveals that Paging Channel of air interface between Base Station and the Mobile Host can be saturated by injecting User Datagram Packets (UDP) from Internet destined for the mobile hosts. The researchers have successfully attempted to increase the traffic load by 10 % and identified that creating more traffic loads are also possible. The research reveals another vulnerability of GSM caused by GPRS service that could be exploited by attackers to halt the GSM network. Since the GPRS service is not simultaneously used by all users, exploitation of the vulnerability is not severe as compared to problem discussed in [5].

William Enck et. al. (2006), proposed a solution to the problem discussed in [5]. They proposed a mechanism to separate the voice from SMS traffic by implementing weighted fair queue in order to reserve bandwidth for voice and SMS so that attack through Internet originated malicious SMS traffic may have minimum effect on the voice traffic [20]. Their mechanism mitigates the attack and lessens its effect on the voice traffic. However the protection of the legitimate SMS and other traffic needs to be addressed. The application of the proposed mechanism results in under utilization of available bandwidth.

2.2 Limitations

The limitations of research findings are mentioned along with summary of each research finding described above. Perusal of the above research findings reveals the importance of SMS because of its wide scope in commercial and scientific applications. Secure SMS communication is highlighted as primary requirement for its application in commercial sector. Simultaneously the research in the field of communication networks has discovered inherited vulnerabilities in GSM networks. Feasibility of exploiting the open functionality by an adversary equipped with limited resources to the level of jamming the GSM network is considered a major problem. One can saturate the bandwidth capacity of a cellular network by sending enormous malicious SMS traffic from Internet resulting in denial of voice as well as legitimate SMS traffic.

Solution to the problem stated above, suggested by William Enck et. al. (2006) imposes limitations of underutilization of available bandwidth and implementation cost fro existing GSM networks. Since the solution needs to be implemented on GSM devices, which are part of closed proprietary systems provided by the specific vendors, the implementation of solution by the third party competitors is difficult and expensive.

2.3 Summary

Possibility of successful denial of service attack on GSM networks has given birth to uncertainty and various questions about the future of SMS in commercial applications. Such security risk involved in use of SMS in business applications may result in adverse effects over present and future e-commerce activities. Due to its limitations i.e. underutilization of the available bandwidth and implementation cost, the proposed solution is not sufficient to meet the requirement. The problem requires efficient and economical solution to the problem without putting any limitation on the availability of services or on convenience of the customers.

3. Requirement Analysis

3. Requirement Analysis

In cellular networks, the air interface is a critical component because of bandwidth constraints. For a cellular operator, achieving bandwidth in air interface is more costly. Air interface is divided into two parts, the Control Channel (CCH) and the Traffic Channel (TCH). CCH is responsible for call setup functions and delivery of SMS traffic, whereas the TCH is used for delivery of voice data. Since voice needs more bandwidth as compared to text data, always more bandwidth is allocated for it as compared to CCH.

In this new competitive age, provision of variety of services by the cellular phone networks has increased the dependency of other sectors on these services. Simultaneously interface of cellular networks with Internet has become inevitable because majority of the services are not viable without it. Wide range of vulnerabilities in cellular networks and remedial measures are being analyzed by the researchers, some of which are as under:

1. UDP bursts on GPRS IPs
2. Internet originated SMS bursts
3. Targeted SMS attack
4. Signal jamming
5. Eavesdropping
6. Other

One of the major threats to cellular network is the attack launched through Internet originated SMS bursts. Because text messages and mobile-phone call setups rely on the same limited resource, namely control channels, it is easy to attack this system. If enough text messages are sent so that no more control channels are available, calls will begin blocking (i.e. will not be connected) as depicted in figure 3.1. Saturation of the control channel with malicious traffic blocks the call setup data and legitimate SMS traffic traveling through this channel.

The risk of jamming the essential cellular services is point of concern for network operators and also results in uncertainty among the Stakeholders of Mobile Commerce. In the presence of this threat, it is almost impossible to build confidence amongst risk wary business community for potential investment in Mobile Commerce and dependent service sectors. Absence of the remedial measures to the problem may lead to status quo against revolutionary development in SMS commerce, dependent services and research in the field.

Because of its different architecture and nature of services, Internet based defending mechanisms are insufficient to protect mobile networks. The technique of dividing the bandwidth between Voice and other Traffic by implementing weighted Fair Queue

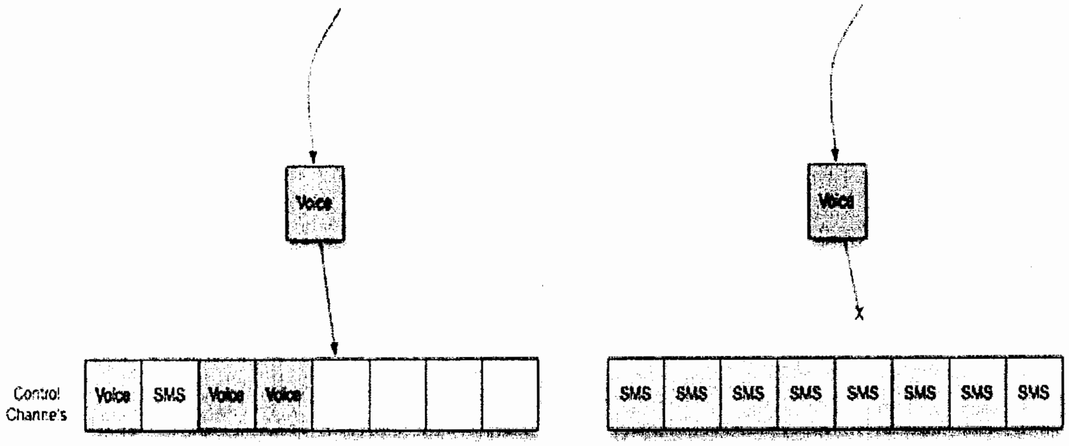


Figure 3.1: (a) (b)

In fig 3.1 (a), a request to set up a voice call is sent to the control channels. Because a number of unused control channels are available, the call will be connected. In fig 3.1 (b), the control channels have been filled by SMS messages. If the attacker sends enough SMS messages to this particular tower, they can ensure that voice calls will always be blocked with a very high probability [5].

(WFQ) etc., proposed by the researchers, mitigates the effect of attack on the voice services leaving other services vulnerable [20].

3.1 Problem Scenarios

An attack to jam a cellular network by generating Enormous Internet originated SMS traffic can take place in different scenarios. In order to accomplish the task, an adversary/group of adversaries can use different ways and means to generate the required threshold of SMS traffic. Different ways/techniques used by the adversary/adversaries create different problem scenarios as discussed below:

3.1.1 Attack using actual IP address - By Single Adversary

Single adversary equipped with resources can create the problem by sending enormous SMS traffic to a cellular network using its actual IP address. This scenario makes the defense difficult because a user needs to be kept controlled even after pre login authentication. In case of user is proved legitimate and is allowed to login after completion of the authentication process, still there is feasibility of sending SMS flood by this logged in user.

3.1.2 Attack using Spoofed IP addresses - By Single Adversary

Single adversary equipped with resources can create the problem by sending enormous SMS traffic to a cellular network using spoofed IP addresses. This scenario makes the defense difficult because filtering the attacker IP addresses and restricting the login on IP address basis is not feasible.

3.1.3 Attack using Zombie Network

Adversary can create the problem by sending enormous SMS traffic to a cellular network using a zombie network. This scenario makes the defense difficult because the zombie nodes use their actual IP addresses and take part in the communication steps impersonating as the legitimate user.

3.1.4 Attack using actual IP address - By group of adversaries

A situation can arise when numbers of adversaries unite by using their social contact and simultaneously launch an attack to help one another for the common purpose. Such group of adversaries can also send SMS bursts to a cellular network using actual IP addresses. Problem occurred in this scenario is also difficult to handle because defense system needs each user to be kept restricted of sending burst even after pre login authentication.

3.1.5 SYN flood attack

Presently majority of the "Web to Cellular Phone SMS Chat" services are implemented using TCP for communication between the web client and the server. Message sent by the client is received by the GSM-Internet interface gateway, which forwards the message to SMSC for delivery to the destination cellular phone etc. Although SYN flood attack on GSM-Internet interface gateway can not result in denial of GSM voice service, but the denial of one GSM service i.e. "Web to Cellular Phone SMS Chat" is possible.

3.2 Focus of Research

Keeping in view of the critical scenarios discussed in section 3.2, following goals and targets were set:

- Enable reliable and smooth Web-Cellular Phone SMS communication.
- Bring resilience in cellular network by defending the Internet Originated SMS burst attacks while simultaneously making cellular services available.
- To decrease the degree of risk involved in the use cellular services by different businesses, industrial, scientific and service sectors.
- Confine the scope of Internet originated SMS attack to only one cellular service (i.e. Web to SMS chat) that ensures smooth operation of the other cellular services at attack time.
- To motivate the business community and IT researchers for advancement in cellular technology and its application scope.

- To discourage targeted attack (End user based attack).

The requirement is a comprehensive solution that may resolve the problem by defending the attack in all scenarios and achieve the desired goals and targets. Different methodologies for different scenarios are discussed as follows:

3.2.1 Dynamic time-sharing and data transmission Control

In case of scenario discussed in 3.2.1, the solution system should not permit the monopoly on system time and transmission of data beyond the specific limit. Thus preemptive scheduling for time-sharing can be used implemented. Since the single user is capable of creating enormous malicious traffic, the server should discard the data beyond the specific size sent by client even during its time quantum. In case when there is no user competing the system time except the adversary, the system should allow utilization of the time but never beyond the handling capacity of GSM.

3.2.2 Resilience against spoofed IP requests

Keeping in view of the scenario stated at 3.2.2, authenticity of client IP address is required to be checked before its entry into service so that spoofed IPs could be filtered. Solution system should be able to identify the spoofed IP addresses and discard all the service requests having spoofed source IP addresses.

3.2.3 Resilience against Zombie requests

As discussed in 3.2.3, adversary can use zombie network to send enormous SMS traffic to the server. The solution system should be capable of identifying the requests created by the zombie network and discard them.

3.2.4 Resilience against group of adversaries

The scenario stated at 3.2.4 necessitates for a mechanism to be incorporated in the solution system to defend against such type of attack. The attack becomes more complicated and difficult to defend when each member of the adversary group uses actual IP rather than spoofed IP address. Here the large number of members can put the system under attack without employing any zombie network for the purpose.

3.2.5 Resilience against SYN floods

Although the attack scenario discussed in 3.2.5 cannot result in adverse effect on the GSM network and its services. But in case of successful SYN flood attack on the GSM-Internet interface gateway, the Internet users will not be able to send SMS to the cellular phones of the GSM network under the attack. However this type of attack will have no any adverse effect over the SMS traffic sent and received between the cellular phones and other GSM services.

3.3 Summary

There are many different scenarios in which this specific vulnerability of GSM network can be exploited by the adversary or group of the adversaries. Possibility of all the attacks simultaneously or at different times has made the situation more critical as well as complicated. This position necessitates a comprehensive and dynamic solution system to avoid the occurrence of the problem. There are two major operation scenarios i.e. the normal operation of the solution system when only the legitimate users are served and the other is abnormal operation when adversary launches attack on the system with malafides intention of denial of service. Here denial of service can occur in two scenarios- one is jamming of GSM network because of saturation of the Control Channel that results in denial of all the GSM services and other is exhaustion of the GSM-Internet interface gateway that results in denial of web to phone SMS chat service.

As customer convenience is the primary condition of today's services industry, any mechanism that involves user's intervention is not considered viable under the apprehension of customer discouragement. Since GSM networks are presently operating in services sector, desired solution system is required to work under these constraints.

Security always puts limitations on the availability, but the availability of GSM services cannot be compromised because of its use in proprietary services industry open to all the customers. So it is primary requirement that the solution should ensure maximum availability of the GSM services.

4. Proposed Solution

4. Proposed Solution

Analysis of the requirement and constraints as discussed in Chapter 3 realizes the necessity of designing the solution system to cope the problem caused by the inherent vulnerabilities in existing GSM and computer networks. Design of a theoretical or any system that suggest major change in existing architecture of GSM/Internet or a solution that require global cooperation of stakeholder may provide a solution having implementation constraints.

Keeping in view of the above we focused to design a secure protocol that protects the services and network from jamming, avoiding uncertainty amongst the network operators and the stakeholders of the mobile commerce.

Accordingly, we have developed CBA, a new secure protocol for Web-Cellular Phone SMS communication having defense features against the Internet originated SMS attack. It substantially decreases the probability of halting the entire network. It also brings resilience in the mobile network by assuring smooth operation of the essential services to legitimate users at attack time. The proposed protocol is intended to confine the effects of these Internet originated SMS bursts to only one sub-service i.e. Web-to-Phone SMS that translates to smooth operation of all other cellular services at attack time. The protocol is intended to keep the Web-to-Phone SMS service available, however 100% smooth operation and availability of this service at attack time might not be feasible.

The Protocol is intended to convey a confidence-building message to stakeholders and relevant business community in order to boost the development in SMS commerce. By mitigating the attacks, it may reduce the risk factor, bring certainty and act as catalyst for further research & development advancement in the sector.

4.1 Design requirements

Many solutions proposed for different problems of the Internet are still not being practiced because their implementation requires global cooperation of different stakeholders that is not feasible. The proposed Protocol should work with existing GSM architecture without any major change in system or putting exorbitant cost of deployment. In order to meet the major constraints of avoiding customer involvement in implementation, the security features of the CBA will be invisible to end-users

As the system designed for GSM requires implementation on GSM devices may not be viable because the prevailing GSM systems are closed and of proprietary nature. These systems are supplied by specific vendors who do not provide any range of flexibility for adopting any change in infrastructure. These implementation constraints necessities that a solution to this problem of GSM may be incorporated in the future devices to be supplied by the specific vendors as well as taking the existing installations on board. This interprets that existing installations may also have maintenance support from third party service providers rather than complete dependability on the specific vendors. In CBA protocol our main focus is to develop such a solution that not only solve the security threats but should also provide ease of implementation and range of flexibility irrespective of the vendor/brand of the GSM devices.

GSM networks are presently operated by the service industry directly. Here one thing that cannot be compromised is customer convenience and support without which service providers cannot survive. Any activity that involves risk of connivance is considered threat in business sector. So any solution to cop this vulnerability of GSM should strictly avoid involvement of the client, who most probably would be customer of GSM service providers. Incase if a solution suggested to this problem involves any activity that may lead to customer inconvenience, will occupy space in academia but never viable for the service providers. Accordingly another design requirement that we keep in mind while developing CBA protocol is to make it transparent from the end user as much as possible. This transparency makes the service easy to use and attract more customers.

4.2 CBA-Reference Architecture

The architecture of the CBA is designed in a modular way. The first module of the system is responsible for authentication of the legitimate clients. This is module that is further responsible for filtration of all the malicious requests for service. The client should pass the authentication check prior to login into service. The system is dynamic as it activates different security features against different types of attack. However at normal operation time the security features do not take part in the communication. The authentication process as a first step identifies the attack, determines its type and activates the counter feature to keep the system secure and smooth service to legitimate clients.

4.2.1 A New Client Server Architecture for security

In existing client server architecture a machine runs a program that is called client and the machine itself is called as client machine, and the machine that runs server program is called Server. The communication starts when client requests for a certain service and server responds according to the client's request. So the client initiates the communication and Server always remains in listening mode. [1]

When a client requests for service a virtual connection is established between client and server. In most of the cases this is the TCP connection. Establishment of TCP connection consumes many resources and it can give birth to different type of attacks as well. So server always has some security threats due to the existing client server architecture.

We purpose new client server architecture for security in which server initiates the TCP connection and client listens to it. So the role of the client and server has been changed. The machine that needs some service remains in listening mode and server contact it to provide service.

In our protocol client sends a UDP request to token server and goes to listen mode, token server authenticate it, and forward its information to the CBA. Now CBA call back the client to provide service.

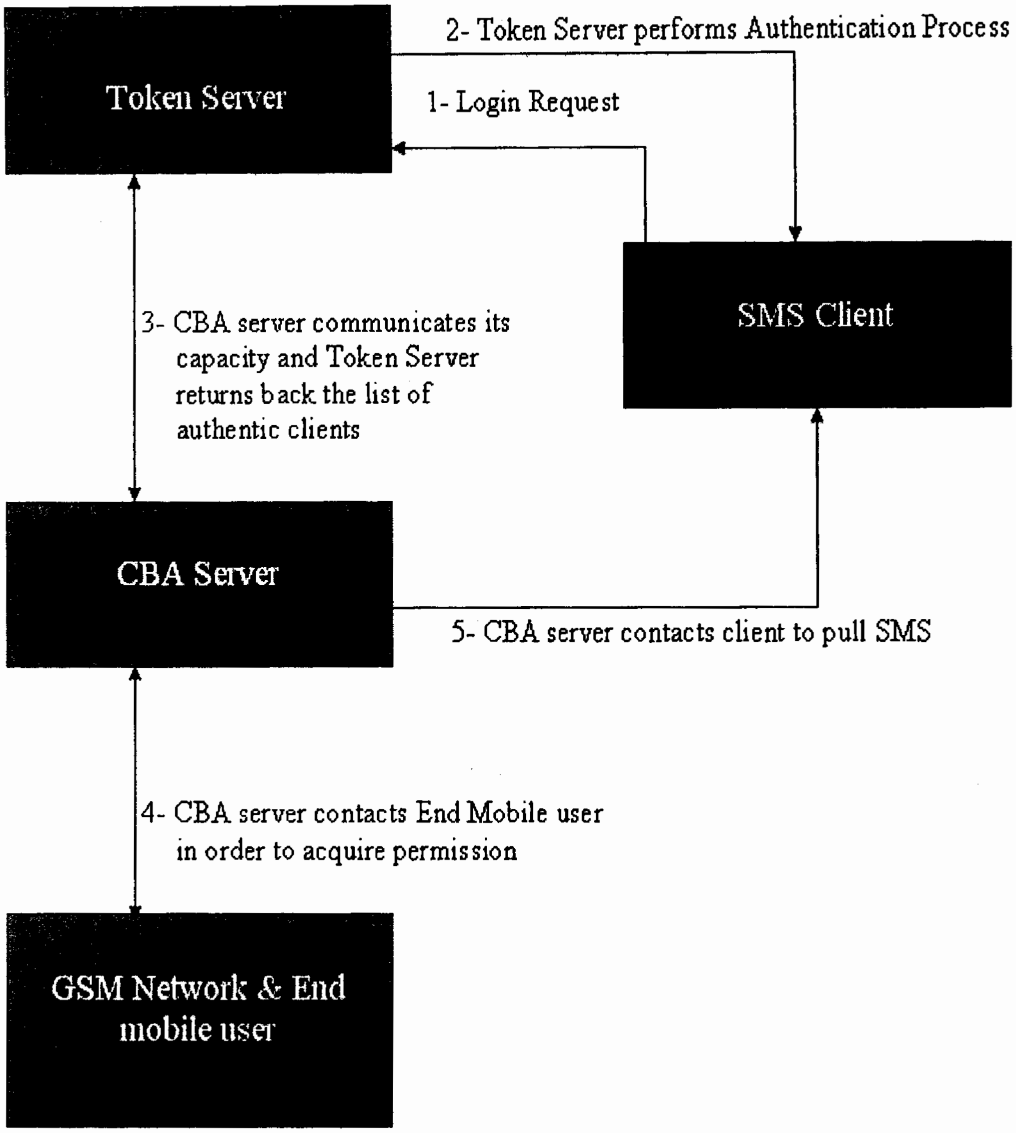

Token Server
2- Token Server performs Authentication Process
1- Login Request
SMS Client
3- CBA server communicates its capacity and Token Server returns back the list of authentic clients
CBA Server
5- CBA server contacts client to pull SMS
4- CBA server contacts End Mobile user in order to acquire permission
GSM Network & End mobile user

Fig 4.1: Client Server architecture for secure communication

This new architecture for security secures the server from different types of security threat, and it also provides some sort of authentication.

4.2.2 Push versus Pull mechanism

The existing client server architecture works on push mechanism (Client pushes the traffic towards server). Since client always initiates traffic so the flow of the traffic belongs to client.

In push mechanism waiting queues are established at the server results in consuming server's resources. Moreover in push mechanism it is very easy for the attacker to launch an attack (Specially DoS attack) by generating a huge amount of traffic at server.

We purpose a Pull mechanism for client server communication. Client goes to listening mode and whenever server sees that now it has enough resources to serve it contacts the client (who is in listening mode).

In pull mechanism waiting queues are established at client site that results in saving the server's resources.

Our proposed pull security mechanism is very much safe as compared to push mechanism because it is up to the server that how much traffic it wants to accept. So the control is in the hand of server that makes it more secure and reliable.

4.2.3 Throat knotting the attack at Gateway

Another security feature of the CBA is to throat knot the attack at the gateway. All the security is implemented on the token server that acts as a gateway to cellular network. So if attacker wants to attack on the cellular network token server stops it at the gateway. It is possible for the attacker with huge resources to become successful to down the token server but even then it is not possible for the attacker to enter into the cellular network illegally.

4.2.4 Authentication

When a client requests sends a login request to chat with a Cellular Phone, a server sends a token for the purpose of authentication. Upon receiving the token client needs to send back the same token to the server.

The issuance of the simple token to the client almost eliminates the threat of IP spoofing attacks. If a client used a spoofed IP address it will never be able to get the token and to access the web to SMS chat service.

When the login requests exceed a certain number during a particular time, server considers itself under attack, so it starts issuing image token. Image token consists of an image showing combination of 5 randomly selected letters (A-Z) or digits (1-9). The client needs to read the image and send back the letters or digits in the textual format by typing them from his keyboard.

The issuance of the image token to the client almost eliminates the Zombie attack (an attack launched through compromised nodes) because a zombie machine is not been able to read the image and interpret the text written on image.

4.2.5 Multi tier Architecture

We divide our server into two main components, Token Server and CBA server. The objective of the token serve is authentication. It performs authentication by issuing simple or image token. While the objective of the CBA server is to contact the client and forwards the messages.

In multi tier architecture if client becomes successful to launch any type of attack on token server, it will not affect the CBA server. So the clients who are already in service will remain secure from the attack.

4.2.6 Token server at Third Party Location

With the two-tier server (Token server and CBA server), it is very easy for the service provider to place the token server under the control of third party. Since the token server consumes most resources and it still has some sort of security threats so allocating the token server to third party will reduce the load from the service provider.

4.3 Methodology

In our protocol there are four main components:

1. Client (C)
2. Token Server (TKS)
3. CBA Server (CBA)
4. SMSC

We modularized our protocol on the basis of the communication pairs out of these four components. So our protocol consists of four modules:

1. Client-Token Server
2. CBA Server-Token Server
3. CBA Server-SMSC
4. CBA-Server-Client

4.3.1 Client-Token Server communication with simple Token

Our protocol starts working as soon as client starts the application program of the client. After starting the application program client types his nickname and targeted mobile number and sends a login request to the token server. When the Token server receives the login request it generates an 8 bytes simple token and sends it back to the client. Token server also stores that Simple token against the client's IP address. Before issuing the token to the client, token server checks the token limit against that IP. The administrator can configure the system to limit the number of attempts for certain IP address. When this limit reaches no more token is issued against that IP address.

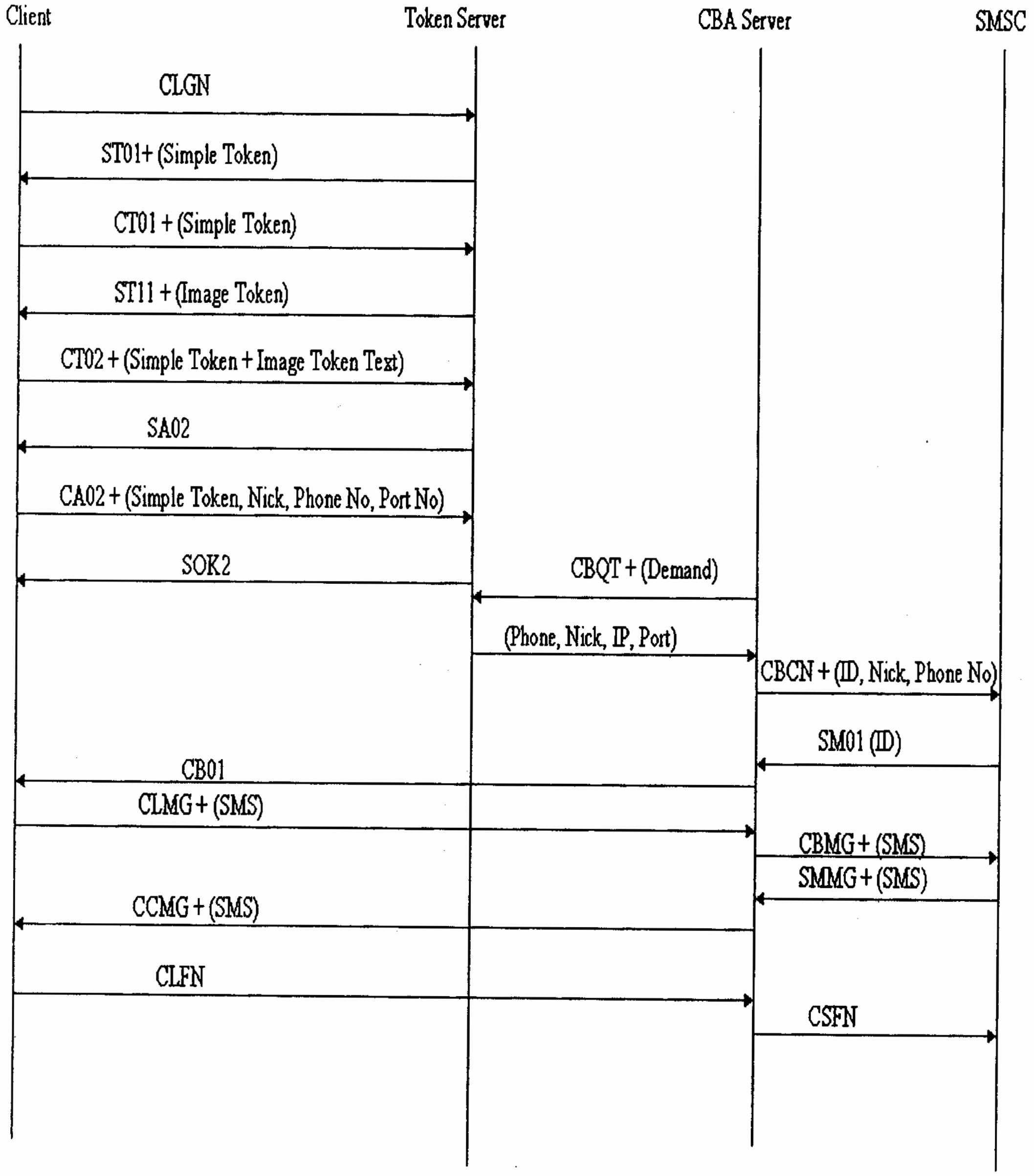


Fig 4.2: SMS communication session using CBA

When the client receives the simple token, it sends back the same token to the token server. This process almost eliminates the use of spoofed addresses to attack because when the token server receives a request from an IP address it sends token back to same IP address. So if that IP address is a spoofed IP it will not be able to get the token back. So attack through spoofed IPs is controlled through the use of these simple tokens. When client sends back the token to the token server, it matches it with the issued token. If the match found client is authenticated else client's request is aborted.

The process of sending and receiving back the simple token is totally transparent from the client. Client has no knowledge about it.

4.3.2 Client-Token Server communication at Threshold Time

There can be a situation in which token server may receive a bulk of login requests at a same time. In this situation token server considers itself under attack and starts issuing image tokens along with simple tokens. Image tokens are issued when the number of login requests exceeds a certain threshold value. The administrator that may change time to time configures this threshold value. The purpose of issuing the image token is to block the compromised nodes.

Compromised nodes are those machines that even do not have the knowledge that they are requesting the token server. Attacker writes a program and by using any mechanism installed that program on the compromised machine. Compromised machine after installing that program starts sending login requests to the token server. The user of the compromised machine is totally unaware of this process.

When a token server sends a simple token to the compromised node, the installed program sends the same token back to token server and gets authenticated. But when it receives an image token it is not possible for the installed program to read the image and sends back the text whatever written on the image. So in this way compromised nodes are blocked.

Image token consists of an image that has five random letters (A to Z except O) or digits (1 to 9) written over it. When client receives the image token it reads the image and types the letters or digits written on the image and sends back the text to the token server. When token server receives the image token back in textual format it matches it with the issued token. If it matches client is authenticated. If it does not match another image token is issued to that client. If again clients sends back wrong token a third chance is provided to the client. After the third chance clients request is aborted.

When a client sends back token to the token server it also sends few other things to the token server and those are the nickname of the client and target mobile number with whom client wants to chat.

There is maximum limit that bound a single client to get authenticated again and again. When this maximum limit reaches no more tokens are issued to the client and the client is added to the black list for certain period of time. When a request from an IP address arrives at token server, it first checks the black list, if that IP address lays in black list no token is issued to that IP address. An IP address remains in black list for a specific period of time, configured by the administrator, and after that it is released from the black list.

4.3.3 CBA Server-Token Server

CBA server is the main component of our protocol. It has TCP connection with the Token server. CBA calculates the maximum number of clients that it can serve during a certain period of time so that the other services should not be disturbed. In our protocol administrator tells the CBA about the maximum number of clients those it can server during a specific time period.

Once the administrator configured this limit, it is communicated to the token server. As soon as token server receives the request from the CBA, it starts a counter from that limit. Then token server starts sending information of the authenticated clients to the CBA. With every send it decreases the counter by one. As soon as the counter reaches to zero token server stops sending client's information to the CBA.

In case token server does not have as many authenticated tokens as demanded by the CBA, then on their arrival the authenticated clients are forwarded to CBA immediately in continuity till the counter reaches to zero.

When the counter reaches to zero, CBA sends another request for the certain number of tokens. Since CBA demands token according to its capacity by keeping in view the current situation of the network, so it is almost near to impossible to jam the voice and other data traffic of the cellular network through web to cellular phone SMS chat.

4.3.4 CBA Server-SMSC

After receiving authenticated client's requests from token server, CBA server stores this information. It takes the nick and targeted phone number and sends this information to the SMSC. SMSC upon receiving the nick and phone number sends a message to end mobile user stating that the particular nick wants to chat with you, would you like to chat?

If the destination mobile user wants to chat he sends "Yes" message to SMSC. SMSC will forward this message to the CBA. If destination user does not want to chat he will not send any reply or he can send No message.

4.3.5 CBA server-Client

Upon receiving the reply from SMSC, CBA starts communicating with client.

If the targeted user does not reply or say no I do not want to chat CBA simply sends a UDP message to inform the client that the destination user does not want to chat with you.

If SMSC sends "Yes" to CBA, CBA initiates a TCP connection with the client. Since connection is initiated by CBA so it has no threat like Syn flooding.

When the connection establishment is complete CBA tells the client to start sending SMS. Again to provide security and reliability different types of checks are applied on sending the SMS. First check is on the number of messages that can be sent over one TCP connection or in other words against one authenticated token. Second check is the time check. Id client does not send any message for a certain period of time it is connection is teardown.

Now when a client sends a message, CBA receives it and forward it to the SMSC. SMSC deliver it to the destination. When the destination reply it first came to the SMSC that forward it to the CBA and CBA deliver it to the client.

When the client logged off or maximum message limit reaches or client does not send the message for a long period of time (configured by administrator) the connection with the client is terminated.

4.4 Operational Details

In this section we will discuss the operational details of each of the components of CBA.

4.4.1 SMS Client

When a client wants to chat with a particular mobile phone it starts the client application program.

Now first client enters a Nick and destination phone number and press Login button. On pressing the Login button client opens a UDP socket and sends a login request to the Token server. After that client blocks at receive method. Now here two situations are possible, either it will get a simple token or it will receive an image token.

In case of simple token clients sends the same token and after that it sends Nick and Phone number to the token server through UDP socket.

In case of image token, client receives an image consists of five letters (A-Z) or digits (1-9) at the place between the two buttons. Client reads the image and types the letters or digits printed on it in the field parallel to OK button. On pressing the OK button client program sends the typed letters as well as Nick and Phone number to the token server through UDP socket.

Now if the destination mobile phone user does not want to chat with the client CBA sends a UDP message to the client and terminate the whole process.

But if destination mobile phone user wants to chat with the client, CBA establishes a TCP connection with the client and communication between Mobile phone user and client get start.

Algorithm 1 describes the main events that take palace at client side.

```
Begin:
  If (Button clicked = Login){
          Verify nick and Destination phone number fields
          Sends a UDP Login request to Token Server
          Waits to receive response from the Token Server
          If (response = Simple Token){
               send the same token and as well as Nick and Phone number to the
               token server through UDP socket
               Wait on receive method()
          }
          else if (response = Image Token){
               Type the letters printed on the Image
               Send these letters to the token server through UDP socket
               Wait on receive method ()
               Send Nick, Phone number and port to token server
               Wait on receive method ()
          }
    }
    If (receive = no or Timeout){
          Logged off automatically
    }
```

```
        else{
                Accept a TCP connection from the CBA
                Type the message in Text area
                Sends the message to CBA
                Receive message from CBA
        }
End
```

Algorithm 1

4.4.2 Token Server

Token server has connection with both CBA and Client. When token server receives a Login request from client it generates a Token and sends through a UDP message to the client. When it receives a simple or image token from the client it verifies the token and records the client data in the log.

Upon receiving Authenticated connections request from CBA, it provides the required number of tokens to CBA.

Algorithm II describes the working of Token Server.

```
Begin:
        Start the Token Server by clicking on Start button
        Create 2 threads. First will communicate with client and 2nd with the CBA.
        Start thread 1{
                Accept a login request from the client
                Generates an 8 byte simple token
                If (threshold limit reaches){
                        Generates an image token
                }
                Sends the token to the client and waits to receive it back
                Upon receiving back the token verify it
                If (verified){
                        Take Phone number, Nick and port from client
                Authenticate that client
                Stores the information of the client
            }
            else{
                terminate the process
            }
        }// End of Thread 1

        Start thrad 2{
            Establish a TCP connection with the CBA
            Receive Connection demand from the CBA
            If(Connection demand <= the available authenticated Connection){
                Sends required number of connections with all the details to the CBA
            }
            else{
                Sends all the available connections to CBA
                As soon as a client authenticated forward its information to the
                CBA until the demand expires
            }
End
```

Algorithm II

4.4.3 CBA Server

CBA is the main component of our protocol. CBA communicates with all the other three components, client, Token server and SMSC. CBA when gets start it receives a TCP connection from Token server. It takes the number of connection that it can serve from the administrator and sends request to the Token server. Upon receiving a connection from token serve, CBA sends the Nick and Phone number to the SMSC. When the response of the Mobile phone user arrived through SMSC, CBA reads that response. If the response is negative CBA sends a UDP message to client and terminate the process. If the response is positive CBA creates a new thread and that thread establishes a TCP connection with the client to send and receive messages. When client wants to Log off or maximum limit to send the messages reaches TCP connection is terminated and Thread is destroyed.

Algorithm III describes the processes that occur at CBA Server.

```
Begin:

    Accept a TCP connection from Token server
    Take the value of maximum number of clients that the network can serve from
    administrator

    Sends the above value to the Token server
    Receive the desired number of authenticated connections from token server
    Upon receiving a connection for each connection{
        Generate a Thread with a new ID
        Sends the Nick and Phone number to SMSC
        Wait to receive response from SMSC
        If (response = negative or No response until Timeout){
            Sends a UDP message to client
            Terminate the Process
        }
        else{
            Creates a new Thread
            Establish a TCP connection with the client
            Forward the messages of the client to SMSC and vice versa
            If (client Logged off or maximum limit= true){
                Drop the connection
                Destroy the Thread
            }
        }
    }
End
```

Algorithm III

4.4.4 SMSC

SMSC's main purpose is to communicate with the Mobile phone. It takes the Nick and Mobile phone from the CBA and forwards the messages to that mobile phone. On the other side it takes the messages from mobile phone and forwards to CBA so that it should be delivered to client. We develop a simulator for SMSC who receives the request from CBA and returns it back to CBA by adding client's response (Yes, No, or Timeout).

Algorithm IV specifies the steps that SMSC performs:

```
Begin:
    Case 1:

        Receive Nick and phone number from the CBA
        Send initial message to the received mobile phone number
        Receives the response from the mobile phone
        Forwards the response to CBA

    Case 2:

        Receive message from CBA
        Forward the message to the Mobile phone
        Receive reply from mobile phone
        Forward the reply to CBA

End
```

Algorithm IV

4.5 Summary

The main issues that are considered in the design of CBA protocol include security against application level SMS burst attacks, integration with the existing GSM architecture and transparency from the end user. Two-tier server architecture makes the system more reliable and capable to throat-knot the attacker at the gateway. Our proposed PULL traffic mechanism keeps the server safe from application layer flood attacks as well as from few network layer DoS attacks.

Our Design is modular and we divide it in to the four modules. First module is responsible for Authentication of users, second module provides interaction between the two tiers of servers, third module is to connect the CBA server with the GSM network and in the last module service is provided to the end user.

5. Implementation

5. Implementation

C and C++ are the most widely used languages in the field of software development. These two languages provide the user millions of options and controls. Very few languages are as rich in functionality as C and C++. Compared to other languages such as Microsoft Visual Basic, C and C++ takes more time and effort to produce the same output as Visual Basic. Due to complexity and uneasiness programmers want a language which has features of C++ and controls of some other languages like Microsoft Visual Basic. There is tradeoff between functionality and productivity. When we want to enhance one of them the other automatically reduces.

To solve the problem Microsoft developed a new language named as C# (C-Sharp). C# is an object oriented language that provides almost all the functionality provided by the C++ as well as more and more controls to the user. It is very easy to develop an application in C#. C# takes lesser time and provides maximum controls as compared to C++.

Because of its elegant object-oriented design, C# is a great choice for architecting a wide range of components-from high-level business objects to system-level applications. Using simple C# language constructs, these components can be converted into XML Web services, allowing them to be invoked across the Internet, from any language running on any operating system. [21]

5.1 Deployment Environment

In order to acquire the results of the CBA protocol, we test it over a Local area network of 30 computers. Each computer is installed with Windows XP professional and Microsoft .NET framework 2.0. We take two computers as server one for Token server and the other for CBA server. One machine is used to run SMSC simulator.

Different attack programs are written in order to take accurate results. These attack programs are divided into spoofed and zombie programs. These programs send login requests of different sizes to token server.

5.2 Configurable Features

In our protocol there are few features those vary from time to time. These features are configured and controlled by the administrator. These features include:

I. Login Threshold

When client sends a UDP request, token server respond to it by issuing an 8 byte token. But when number of requests per second increases to a certain limit, token server assumes that now it is under attacked. So it starts to issue image token in order to block zombie attack. We call this limit as Login Threshold. The administrator can increase or decrease the value of this limit by observing the current situation of the network.

II. IP Filtering

When there are two many requests at the token server and token server assumes itself under attack then Administrator has the option to configure the IP filtering. IP filtering is help to block the attacker.

III. Tokens against one IP Address

When IP filtering is on a certain number of tokens are issued against one IP address. After that the IP address is added to the Block list and is not able to request any more tokens. The value of the tokens issued to one IP address depends upon the administrator.

IV. Time Period of Blocking an IP Address

When we issue the maximum number of Tokens allowed by the administrator against one IP address, then that IP address is added to the Black list. Now the amount of time for which the IP address will remain blocked again depends on Administrator.

V. Maximum number of Clients to be Served at a time

During a particular time period CBA can serve a limited number of clients depends upon the current situation of the network. So administrator by keeping the current situation of the network in mind configures the value of this parameter.

5.3 Flow Diagrams

We divide our flow charts into following categories:

1. Client-Token Server communication with simple Token
2. Client-Token Server communication with Image Token
3. SMSC-CBA-Client communication

5.3.1 Simple Token Based Authentication

Followings are the protocol commands used in the simple token based authentication:

Step 1: CLGN

A client who needs service sends a fresh request to Token server with the protocol code CLGN.

Step 2: ST01

When a server receives a fresh request from a client it generates an 8 byte simple token. After generating token, it records the token in the RAM as well as writes a log against IP address. After that authentication level is incremented. Authentication level is used to determine at which level of authentication client exists. If client wants to jump forward, authentication level stops the client to do that. As well as if he repeats the already performed steps authentication level plays its role and stops the client to do so. After performing these steps server sends the token to client with the protocol code ST01.

Step 3: CT01

After receiving the simple token from token server, client returns the same token to token server with the protocol code CT01.

Step 4 (a): ER01

When server receives back the token it compares that whether it is the same token what was issued to client. If this token does not match the one issued to the client an error is reported with the protocol code ER01 and whole process is terminated.

Step 4 (b): ER02

After the successful token matching, authentication level is verified. If it mismatches again an error is generated with the protocol code ER02.

Step 4 (c): SA01

If ER01 and ER02 both do not exist client is authenticated and authentication level is incremented. Client is informed about authentication with the protocol code SA01.

Step 5: CA01

When client receives SA01 authentication code, it sends additional data required for further communication with the protocol code CA01. Additional data includes nick, destination phone number and port.

Step 6 (a): ER03

When server receives CA01 with additional data, it again matches the simple token. If at this stage simple token mismatches an error is reported with the protocol code ER03.

Step 6 (b): ER04

If ER03 does not exist, authentication level is checked. If authentication level does not match error with ER04 protocol code is generated.

Step 6 (c): SOK1

If ER03 and ER04 both do not exist, an OK message is send to the client with the protocol code SOK1. After receiving SOK1 client goes to listen mode.

Case 1: Client returns an invalid simple token

C →TKSS	:	CLGN
TKSS → C	:	ST01 + (Simple Token)
C →TKSS	:	CT01 + (Simple Token)
TKSS → C	:	ER01 //Simple Token Mismatches

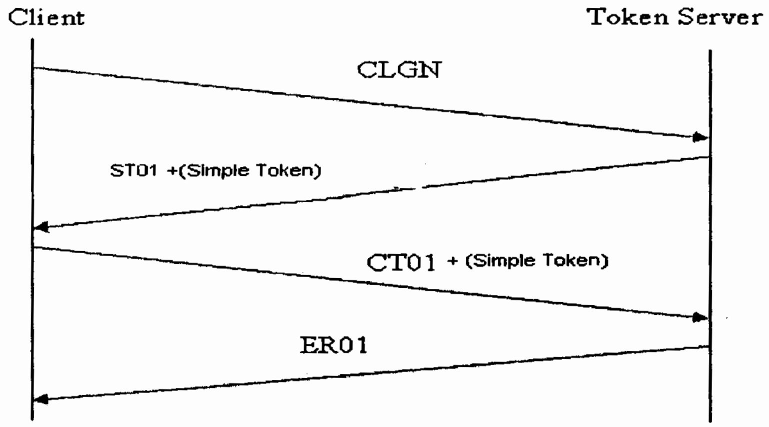


Fig 5.1: Client returns an invalid simple token

Case 2: Authentication level mismatch at the time of submitting simple token

C →TKSS	:	CT01 + (Simple Token)
TKSS → C	:	ER02 //Authentication Level mismatch

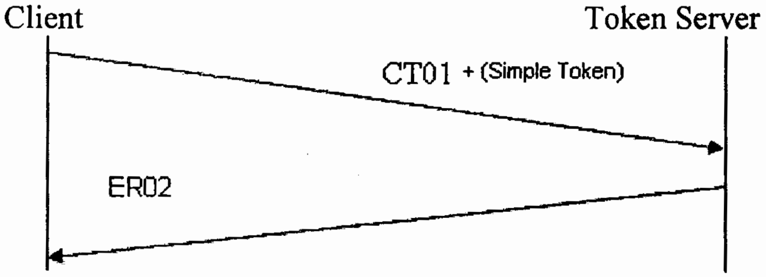


Fig 5.2: Authentication level mismatch upon of submitting simple token by the client

Case 3: Client returns a valid simple token

C →TKSS : CLGN
TKSS → C : ST01 + (Simple Token)
C →TKSS : CT01 + (Simple Token)
TKSS → C : SA01 //Authenticated

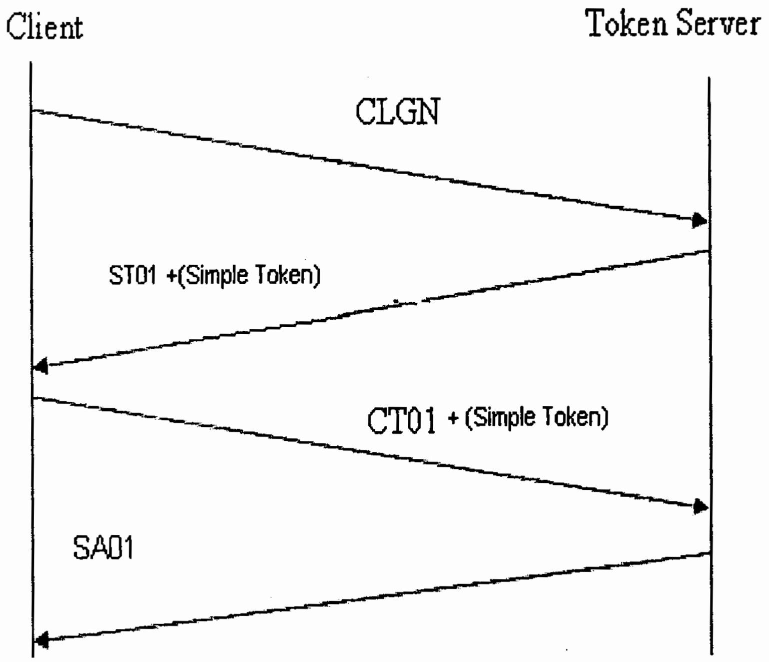


Fig 5.3: Client returns a valid simple token

Case 4: Simple token mismatches at the time of submitting requisite information

C →TKSS : CLGN
TKSS → C : ST01 + (Simple Token)
C →TKSS : CT01 + (Simple Token)
TKSS → C : SA01 //Authenticated
C →TKSS : CA01 + (Simple Token, Nick, Phone No., Port No.)
TKSS → C : ER03 // Simple Token Mismatched

Case 5: Authentication level mismatches at the submission of requisite information

C →TKSS : CA01 + (Simple Token, Nick, Phone No., Port No.)
TKSS → C : ER04 // Authentication Level Mismatched

Case 6: Client returns a valid simple token along with requisite information

C →TKSS : CLGN
TKSS → C : ST01 + (Simple Token)

C →TKSS : CT01 + (Simple Token)
TKSS → C : SA01 //Authenticated
C →TKSS : CA01 + (Simple Token, Nick, Phone No., Port No.)
TKSS → C : SOK1 // Accepted

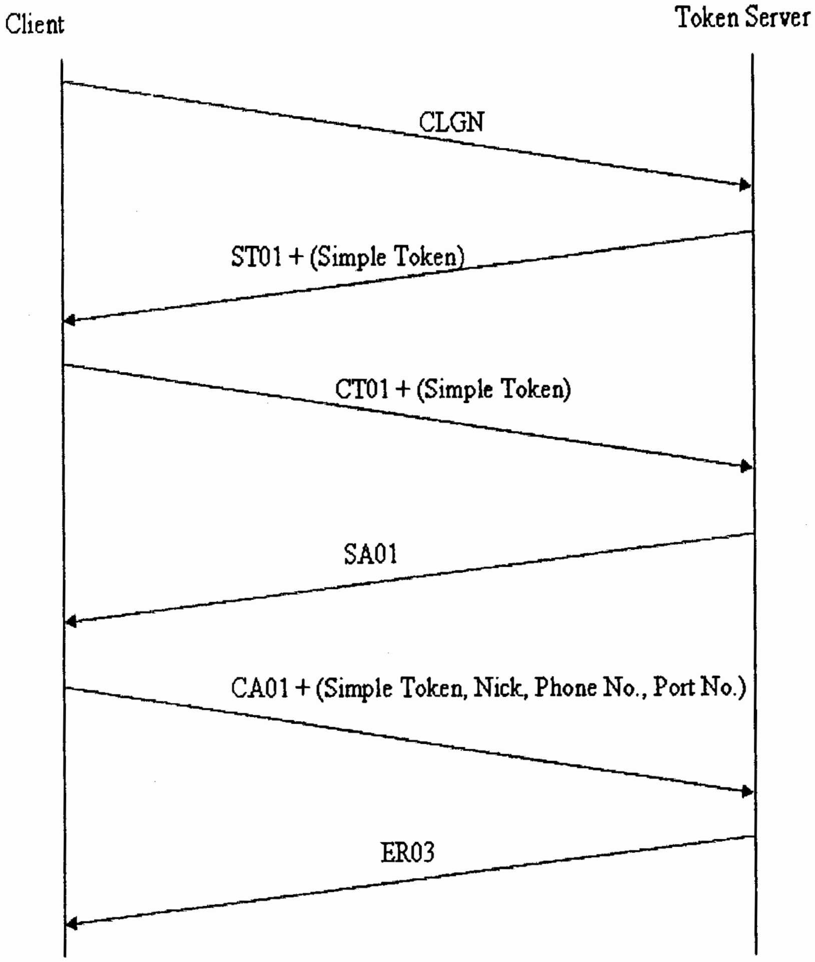


Fig 5.4: Simple token mismatches at the time of submitting requisite information

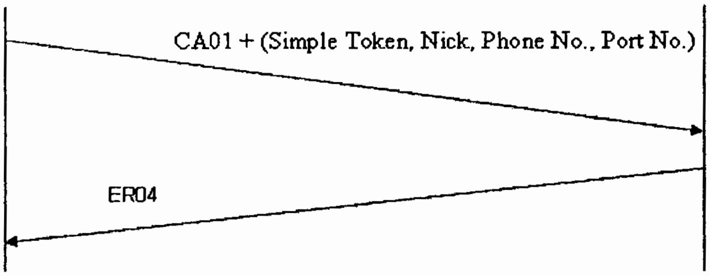


Fig 5.5: Authentication level mismatches at the submission of requisite

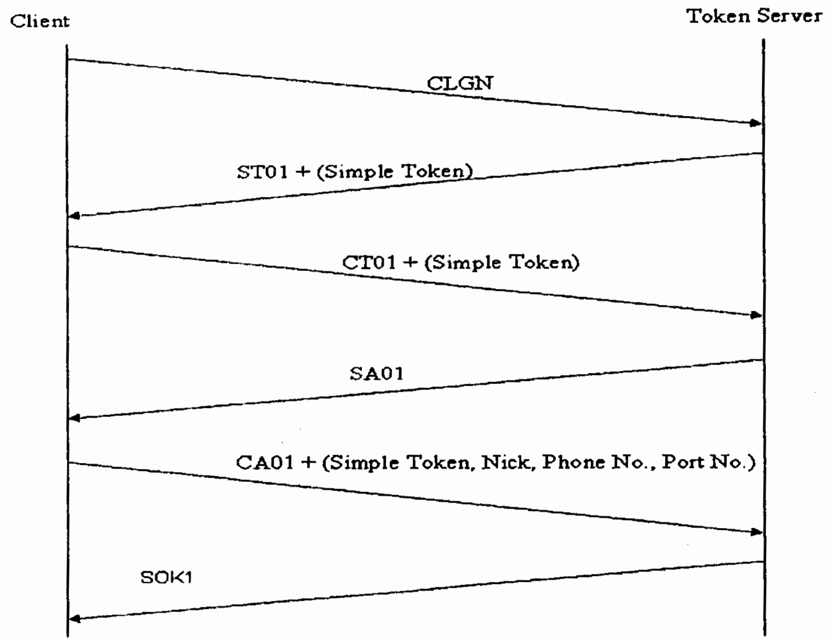


Fig 5.6: Client returns a valid simple token along with requisite information

5.3.2 Image Token Based Authentication

Step 1: CLGN

A client who needs service sends a fresh request to Token server with the protocol code CLGN.

Step 2: ST01

When a server receives a fresh request from a client it generates an 8 byte simple token. After generating token, it records the token in the RAM as well as writes a log against IP address. After that authentication level is incremented. Authentication level is used to determine at which level of authentication client exists. After performing these steps server sends the token to client with the protocol code ST01.

Step 3: CT01

After receiving the simple token from token server, client returns the same token to token server with the protocol code CT01.

Step 4 (a): ER05

When server receives back the token it compares that whether it is the same token what was issued to client. If this token does not match the one issued to the client an error is reported with the protocol code ER05 and whole process is terminated.

Step 4 (b): ER06

After the successful token matching, authentication level is verified. If it mismatches again an error is generated with the protocol code ER06 and the process terminated.

Step 4 (c): ER07

An attacker can launch an attack by demanding more and more image tokens. So the administrator set an upper limit in order to block this type of attack. An IP address can get a limited number of Image tokens and after that no more tokens will be issued to that IP address. Moreover that IP address will be added to block list for some period of time. When an IP address is added to the block list it means no new image token will issued to that IP address but the image tokens which are already issued to that IP will have no effect of this blocking. The will continue smoothly. So when a client request for image token his maximum limit is checked. If it exceeds an error with the protocol code ER07 issued to client.

Step 4 (d): ST11

If all the three errors of the step 4 do not exist, an Image token is issued to the client. Image token consists of an Image on which a random combination of 5 letters from A to Z (excluding O) and 9 digits from 1 to 9 printed. Image token is issued with the protocol code ST11. Authentication level is incremented.

Step 5: CT02

After receiving image token client reads the image and typed the printed letters or digits in a textbox and sends back to token server with the protocol code CT02.

Step 6 (a) ER08

After receiving image token first simple token is verified. If this token does not match the one issued to the client an error is reported with the protocol code ER08 and whole process is terminated.

Step 6 (b) ER09

After the successful matching of the simple token, authentication level is verified. If it mismatches again an error is generated with the protocol code ER09 and the process terminated.

Step 6 (c) ER10

When client sends back an image token in textual format, it may be possible it has some typographical mistakes. In this case another chance is provided to the client by issuing a new image token. But there is an upper limit defined by the administrator that how many times a new token will be issued to the client against a mistake. When that upper limit exceeds an error with the protocol code ER10 is reported to client.

Step 6 (d) ER11

If the upper three errors do not exist, the textual image token returned by the client is compared to the issued image token. If it does not match an error is generated with the code ER11 and a new image token is issued to the client if its maximum limit does not exceed. If its limit exceeds ER10 is generated and the whole process terminated.

Step 6 (e): SA02

If all the above 4 errors do not exist, client is authenticated and notified with the protocol code SA02. Authentication level is incremented.

Step 7: CA02

When client receives SA02 authentication code, it sends additional data required for further communication with the protocol code CA02. Additional data includes nick, destination phone number and port.

Step 8 (a): ER12

When server receives CA02 with additional data, it again matches the simple token. If at this stage simple token mismatches an error is reported with the protocol code ER12.

Step 8 (b): ER13

If ER12 does not exist, authentication level is checked. If authentication level does not match error with ER13 protocol code is generated.

Step 8 (c): SOK2

If ER12 and ER13 both do not exist, an OK message is send to the client with the protocol code SOK2. After receiving SOK2 client goes to listen mode.

Case 1: Client returns an invalid simple token

C →TKSS : CLGN
TKSS → C : ST01 + (Simple Token)
C →TKSS : CT01 + (Simple Token)
TKSS → C : ER05 //Simple Token Mismatches

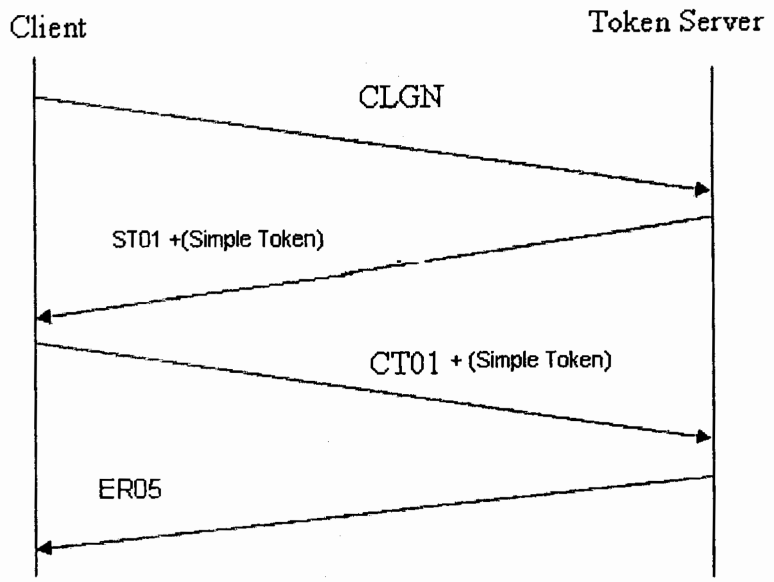


Fig 5.7: Client returns an invalid simple token

Case 2: Authentication level mismatches at the time of submitting simple token

C →TKSS	:	CT01 + (Simple Token)
TKSS → C	:	ER06 //Authentication Level mismatch

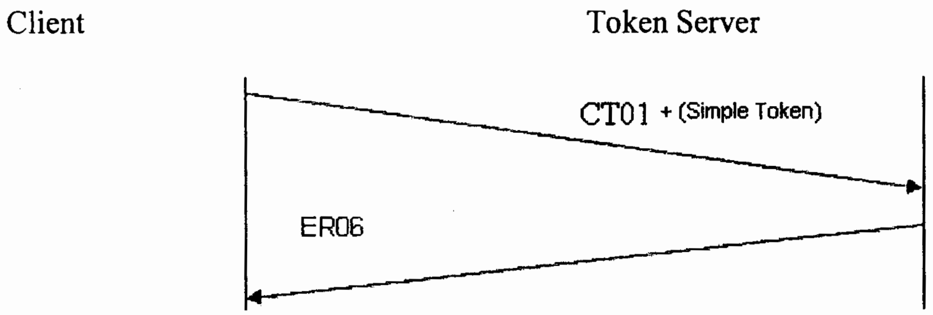


Fig 5.8: Authentication Level Mismatches at the Time of Submitting Simple Token

Case 3: Maximum Limit to Issue Image Token against One IP Address Exceeds

C →TKSS	:	CLGN
TKSS → C	:	ST01 + (Simple Token)
C →TKSS	:	CT01 + (Simple Token)
TKSS → C	:	ER07 // Maximum Limit to Issue Image Token Exceeds

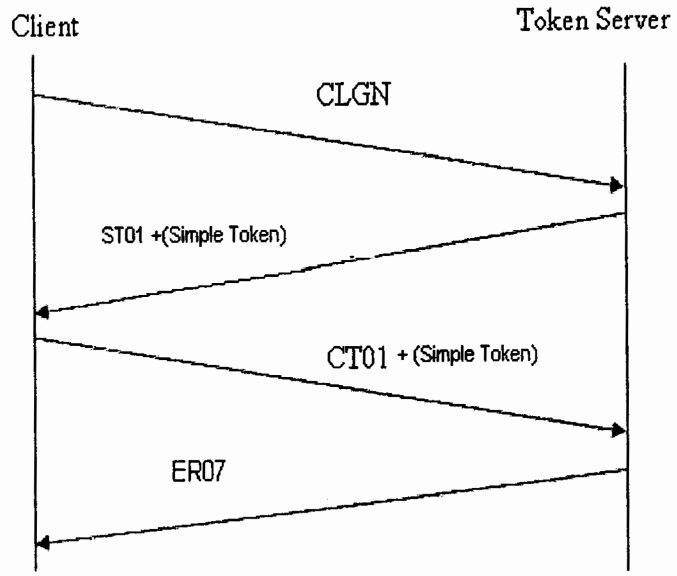


Fig 5.9: Maximum Limit to Issue Image Token against One IP Address Exceeds

Case 4: Issuance of Image Token

C →TKSS	:	CLGN
TKSS → C	:	ST01 + (Simple Token)
C →TKSS	:	CT01 + (Simple Token)
TKSS → C	:	ST11 // Image Token Issued

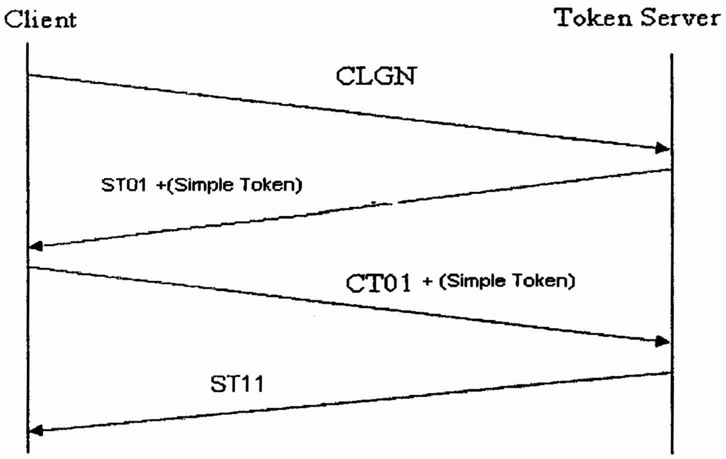


Fig 5.10: Issuance of Image Token

Case 5: Simple token mismatches at the time of submission of image token

C →TKSS	:	CLGN
TKSS → C	:	ST01 + (Simple Token)
C →TKSS	:	CT01 + (Simple Token)
TKSS → C	:	ST11 (Image Token)
C →TKSS	:	CT02 + (Simple Token + Image Token Text)

TKSS → C : ER08 // Simple Token Mismatch

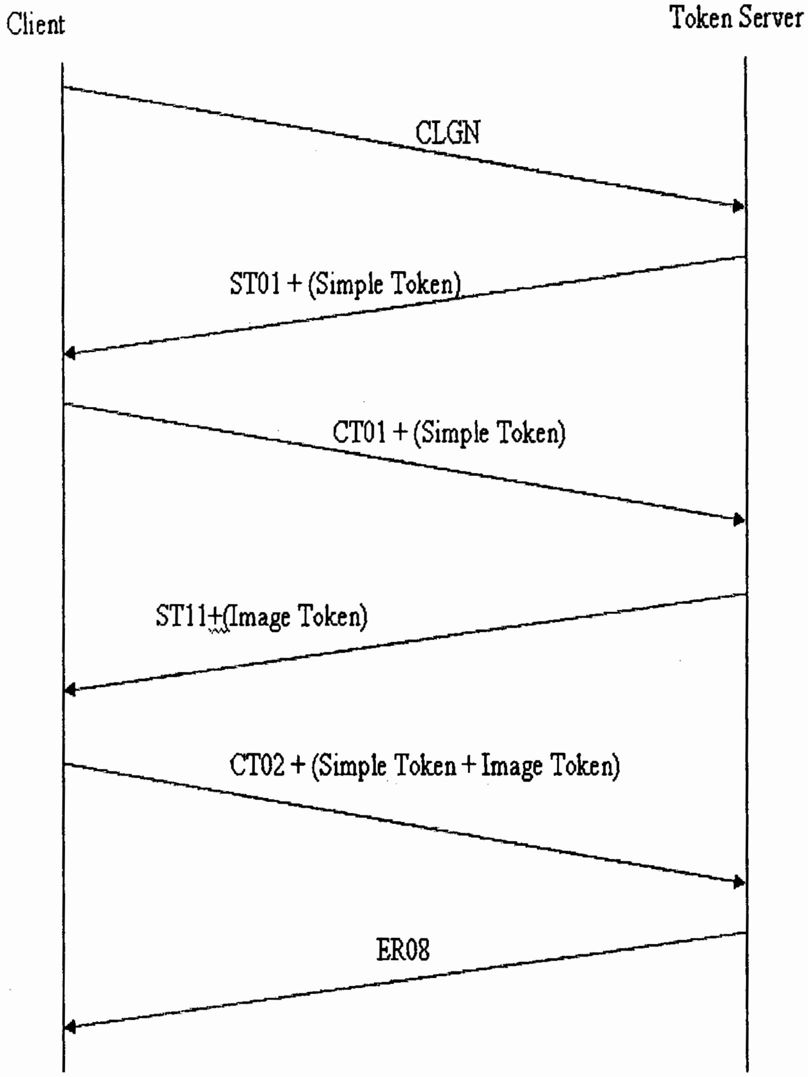


Fig 5.11: Simple token mismatches at the time of submission of image token

Case 6: Authentication level mismatches at the time of submission of image token

C →TKSS : CT02 + (Simple Token + Image Token)
TKSS → C : ER09 // Authentication Level Mismatch

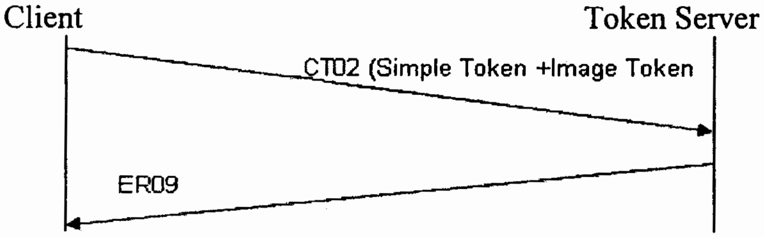


Fig 5.12: Authentication level mismatches at the time of submission of image token

Case 7: Maximum number of attempts to issue image token exceeds

C →TKSS	:	CLGN
TKSS → C	:	ST01 + (Simple Token)
C →TKSS	:	CT01 + (Simple Token)
TKSS → C	:	ST11 (Image Token)
C →TKSS	:	CT02 + (Simple Token + Image Token Text)
TKSS → C	:	ER10 // Maximum Limit to issue an Image Token is over

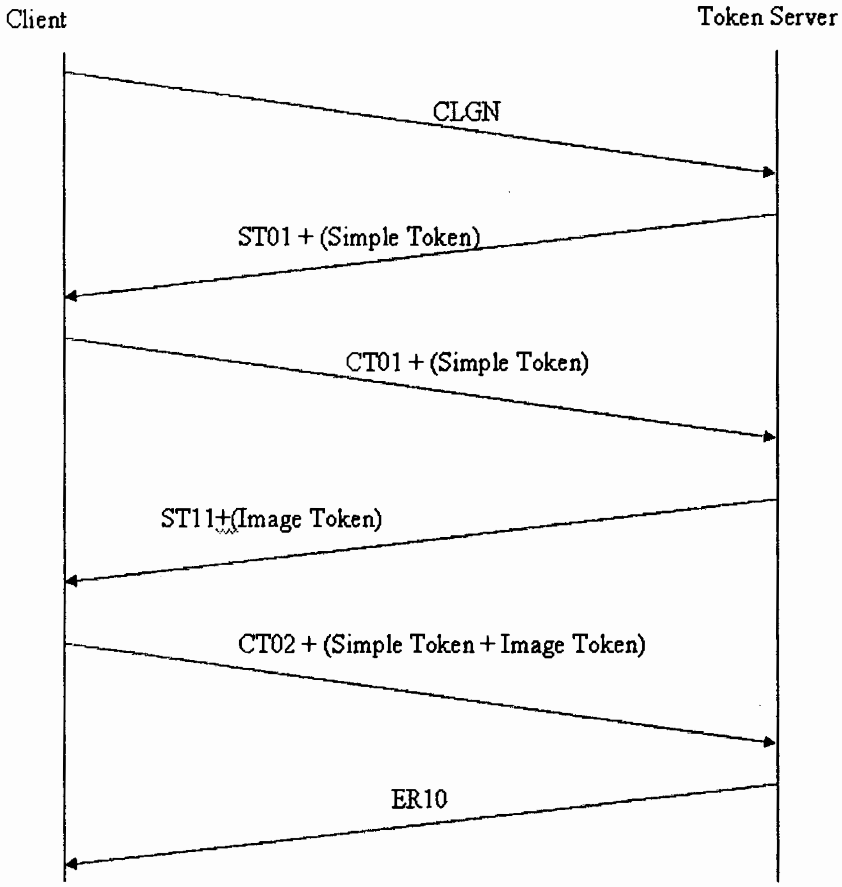


Fig 5.13: Maximum number of attempts to issue image token exceeds

Case 8: Image token mismatched

C →TKS	:	CLGN
TKSS → C	:	ST01 + (Simple Token)
C →TKS	:	CT01 + (Simple Token)
TKS → C	:	ST11 (Image Token)
C →TKS	:	CT02 + (Simple Token + Image Token)
TKS → C	:	ER11 // Image Token Mismatched

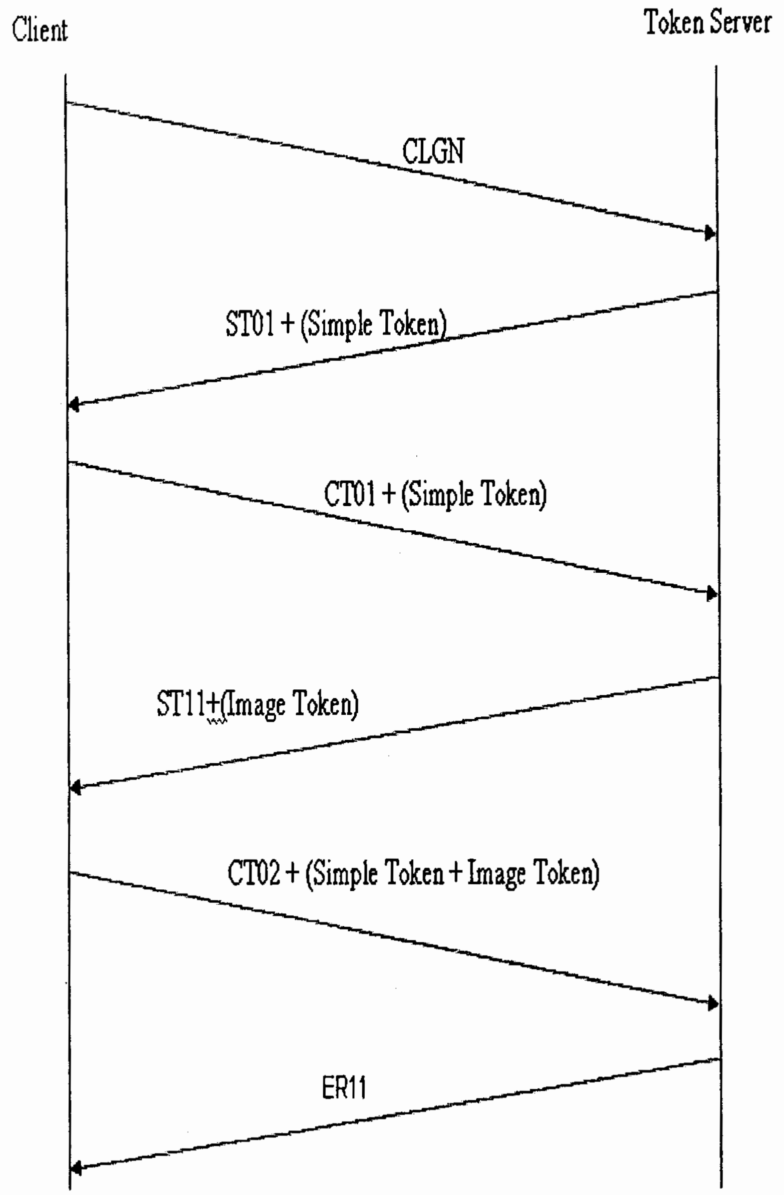


Fig 5.14: Image token mismatched

Case 9: Image token matched

C →TKS	:	CLGN
TKS → C	:	ST01 + (Simple Token)
C →TKS	:	CT01 + (Simple Token)
TKS → C	:	ST11 (Image Token)
C →TKS	:	CT02 + (Simple Token + Image Token)
TKS → C	:	SA02 // Image Token matched

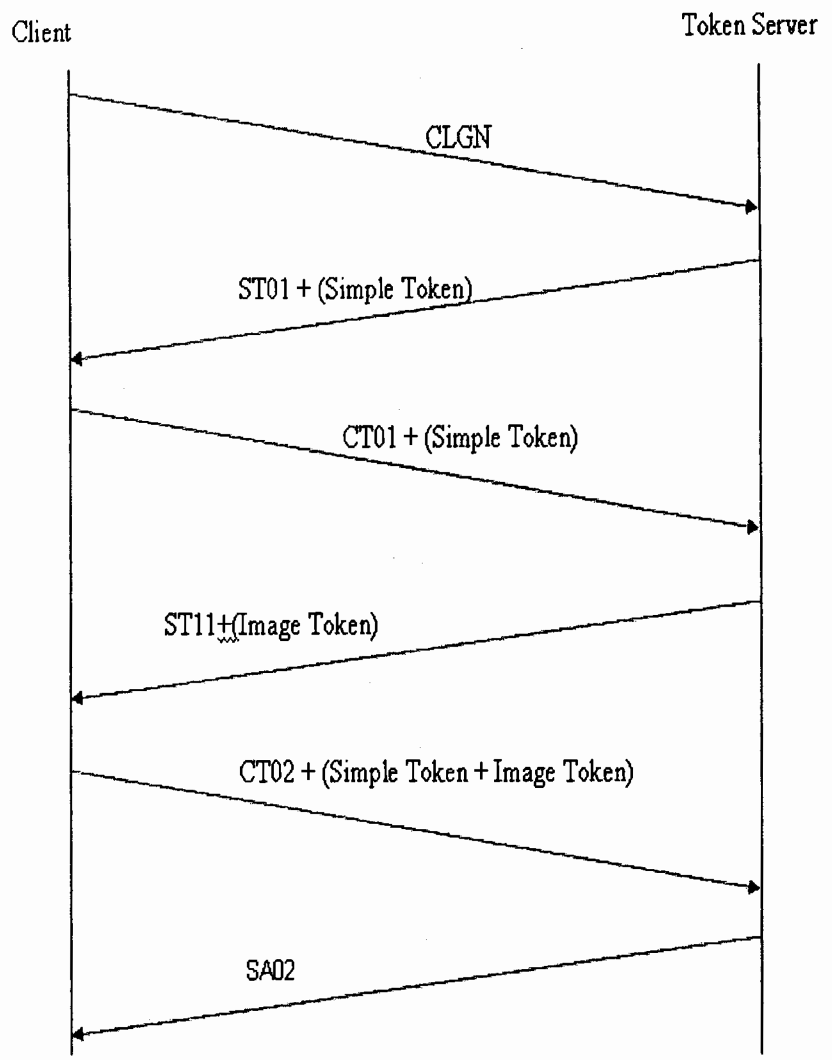


Fig 5.15: Image token matched

Case 10: Simple token mismatched at the submission of requisite information

C →TKS	:	CLGN
TKS → C	:	ST01 + (Simple Token)
C →TKS	:	CT01 + (Simple Token)
TKS → C	:	ST11 (Image Token)
C →TKS	:	CT02 + (Simple Token + Image Token)
TKS → C	:	SA02 // Image Token matched
C →TKS	:	CA02 + (Simple Token + Nick, Phone No, Port No.)
TKS → C	:	ER12 // Simple Token Mismatched

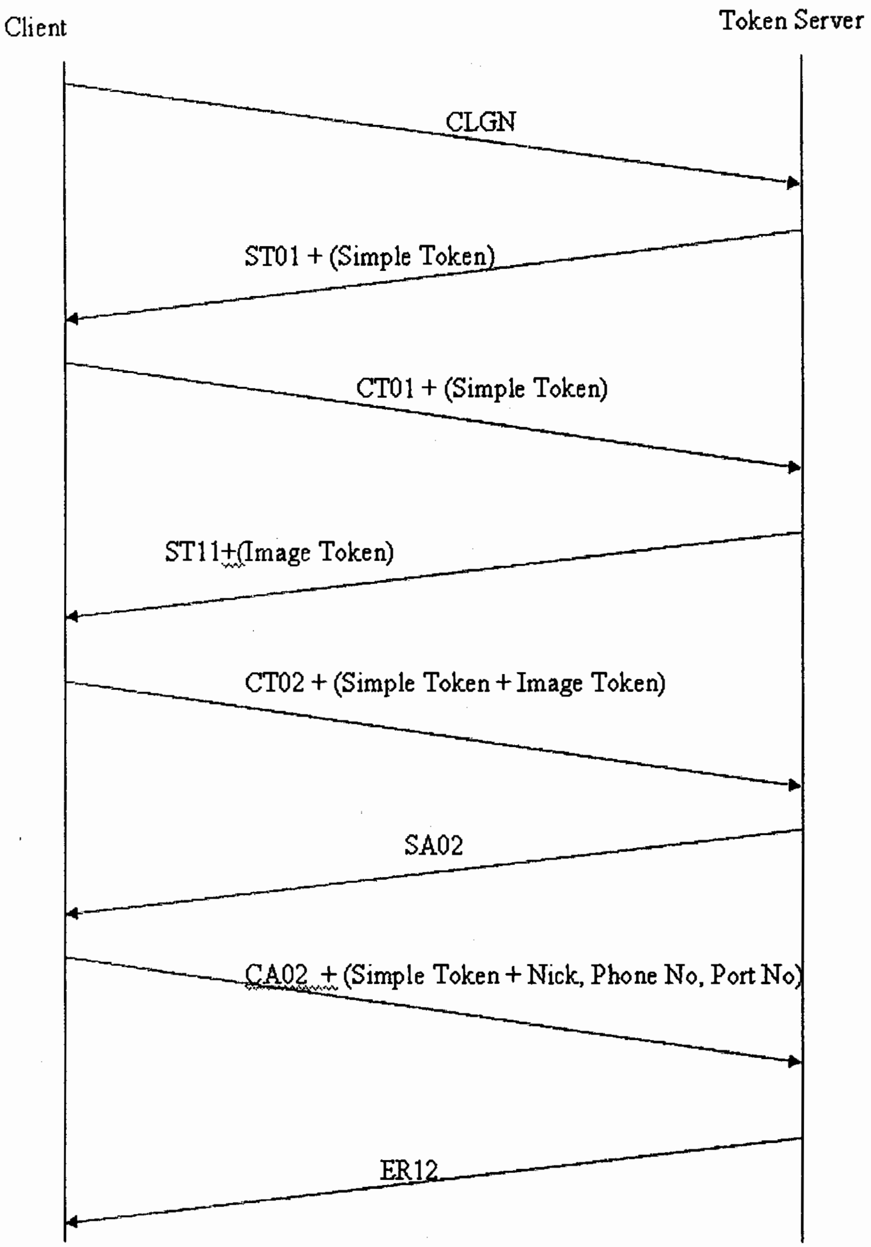


Fig 5.16: Simple token mismatched at the submission of requisite information

Case 11: Authentication Level Mismatches at the Submission of information

C →TKS	:	CA02 + (Simple Token + Nick, Phone No, Port No.)
TKS → C	:	ER13 // Authentication Level Mismatched

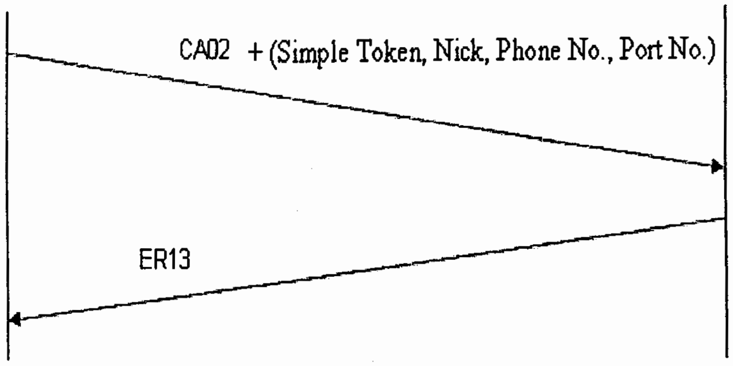


Fig 5.17: Authentication level mismatches at the submission of requisite information

Case 12: Successful submission of requisite information

C →TKS	:	CLGN
TKS → C	:	ST01 + (Simple Token)
C →TKS	:	CT01 + (Simple Token)
TKS → C	:	ST11 (Image Token)
C →TKS	:	CT02 + (Simple Token + Image Token)
TKS → C	:	SA02 // Image Token matched
C →TKS	:	CA02 + (Simple Token + Nick, Phone No, Port No.)
TKS → C	:	SOK2 // Accepted

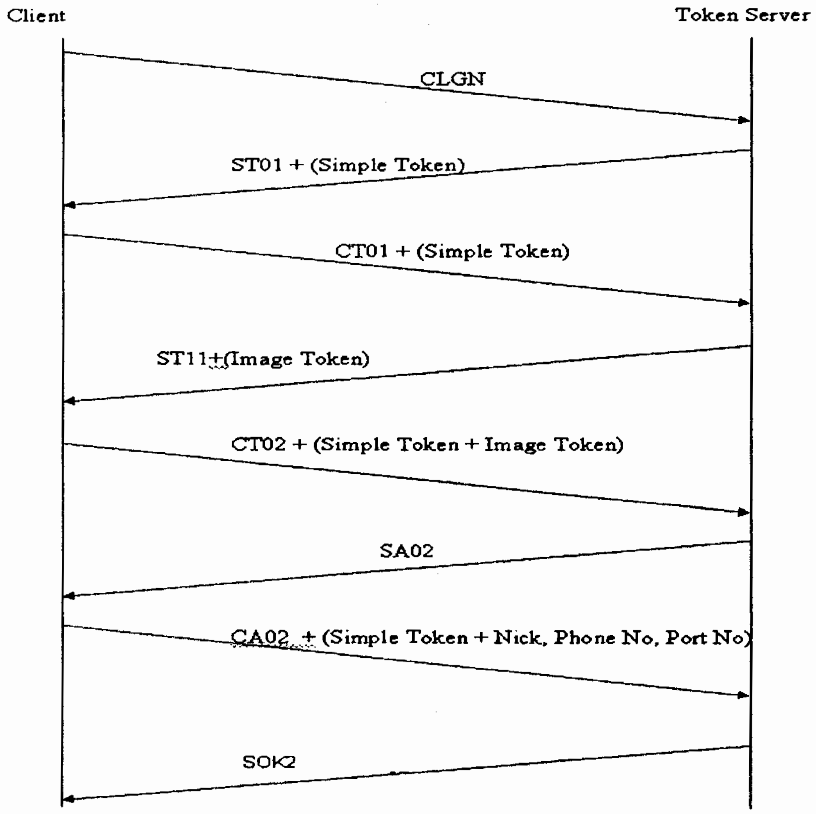


Fig 5.18: Successful submission of requisite information

Protocol Commands: CBA-Token Server communication

Step 1: CBQT

As soon as the token server gets a client's request, it initiates a TCP connection with the CBA. When the TCP connection completes CBA send the maximum number of clients that it can serve at a time along with the protocol code CBQT. After receiving demand token server sends back the requested number of clients. Client's data received by CBA consists of IP address of the client, nick, destination phone number and port.

CASE 1: CBA Demands List of Authenticated clients from Token Server

CBA → TKS :CBQT + (Token Demand)

5.3.3 SMSC-CBA-Client communication

Step 1: CBCN

CBA sends client's request to SMSC. It tells the SMSC that a user with an ID X (allocated by CBA), and Nick (provided by the client) wants to communicate with the mobile number provided by the client.

Step 2 (a): SM01

SMSC forwards the client request to the mobile number and waits for the response. No if that phone number replies that "Yes I want to communicate with this nick", SMSC informed the CBA about his decision with the protocol code SM01.

Step 2 (b): SM02

If in response destination phone number replies that "No I do not want to communicate with that nick", again SMSC informs the CBA with the protocol code SM02.

Step 2 (c): SM03

If the destination phone number does not send any reply (neither No nor Yes), SMSC waits for a certain period of time and after that informs the CBA that client does not send any reply with the protocol code SM03.

Step 3 (a): CB01

When CBA receives SM01 from the SMSC, it replaces the protocol code SM01 with CB01 and informs the client. After that CBA initiates the TCP connection with the client in order to send and receive SMS.

Step 3 (b): CB02

When CBA receives SM02 from the SMSC, it replaces the protocol code SM02 with CB02 and informs the client. After informing the client CBA terminate the whole process and client is logged off.

Step 3 (c): CB03

When CBA receives SM03 from the SMSC, it replaces the protocol code SM03 with CB03 and informs the client. After informing the client CBA terminate the whole process and client is logged off.

Step 4: PULL

Now after the destination phone number's acceptance a TCP connection is established with the client and CBA starts to pull SMS from the client by sending protocol code PULL. Remember that a client can send SMS up to a certain limit defined by the administrator, after that client connection will be teardown.

Step 5: CLMG

When a client wants to send an SMS, it sends it with the protocol code CLMG. An SMS is 160 characters long.

Step 6: CBMG

After getting an SMS from client, CBA forwards it to the SMSC with the protocol code CBMG. When CBA forwards an SMS to SMSC it also tells the SMSC about the client ID generated by CBA, nick that client used to communicate and the destination phone number where to send SMS.

Step 6: SMMG

When SMSC receives a response from the mobile phone, it forwards that reply to CBA with the protocol code SMMG.

Step 7: CCMG

CBA after receiving a message from SMSC, forwards it to the client with the protocol code CCMG.

Step 8: CLFN

Now when client want to finish the communication by clicking on Logout, it sends a message to the CBA with the protocol code CLFN.

Step 9: CSFN

When client sends CLFN, CBA forwards client FIN message to the SMSC with the protocol code CSFN.

Step 10: CLRT

When there is no communication on the TCP connection by the client, a timeout occurs after that CBA sends CLRT protocol code to the SMSC and teardown the connection with the client.

Step 11: CCML

There is an upper limit of the number of SMS that a client can send during a one connection. When this limit approaches, CBA sends a CCML protocol code to the SMSC and teardown the connection.

Step 12: SMFN

When the destination mobile phone wants to finish the communication it informs the SMSC and SMSC informs the CBA with the protocol code SMFN.

Step 13: CCFN

After receiving SMFN from SMSC, CBA forwards the FIN request of the mobile phone to the client with the protocol code CCFN.

Case 1: End Mobile User refuses to entertain chat request

CBA →SMSC	:	CBCN + (Nick, Phone No)
SMSC→CBA	:	SM02 // Negative reply from mobile user
CBA → C	:	CB02 // Negative reply forward to client

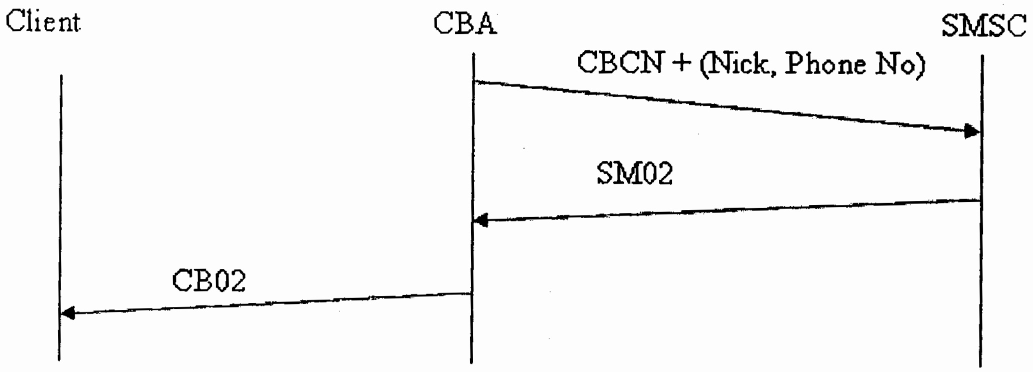


Fig 5.19: End Mobile User refuses to entertain chat request

Case 2: End mobile user does not reply to client's request

CBA →SMSC	:	CBCN + (Nick, Phone No)
SMSC→CBA	:	SM03 // Timeout
CBA → C	:	CB03 // Forward the response to client

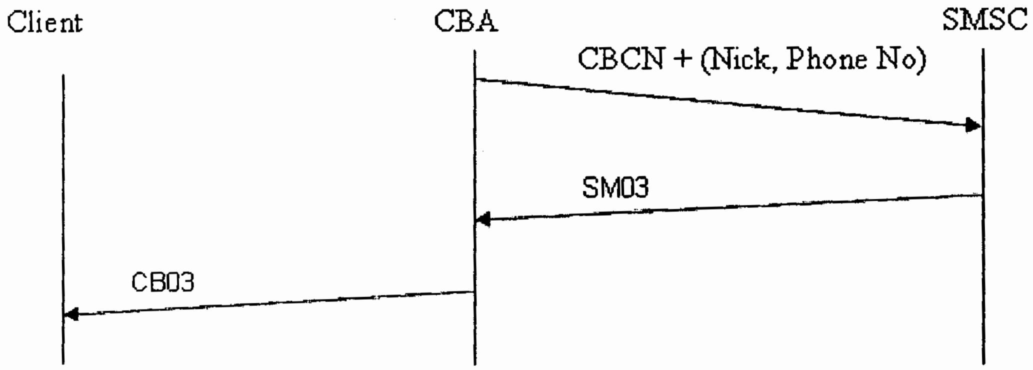


Fig 5.20: End mobile user does not reply to client's request

Case 3: End mobile user wants to chat and two way SMS communication

CBA →SMSC	:	CBCN + (Nick, Phone No)
SMSC→CBA	:	SM01 // Positive reply from mobile user
CBA → C	:	CB01 //Forward the positive reply to client
CBA → C	:	PULL // Pull SMS from client
C → CBA	:	CLMG + (SMS)
CBA →SMSC	:	CBMG + (SMS)
SMSC→CBA	:	SMMG + (SMS)
CBA → C	:	CCMG + (SMS)

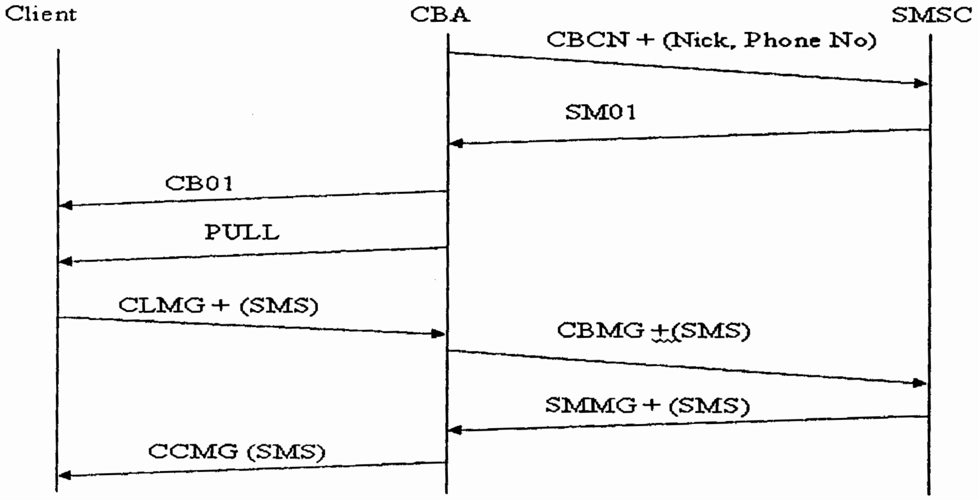


Fig 5.21: End Mobile User wants to Chat and Two way SMS communication

Case 4: When client sends FIN request in order to terminate the chat session

C → CBA : CLFN
CBA → SMSC: CSFN

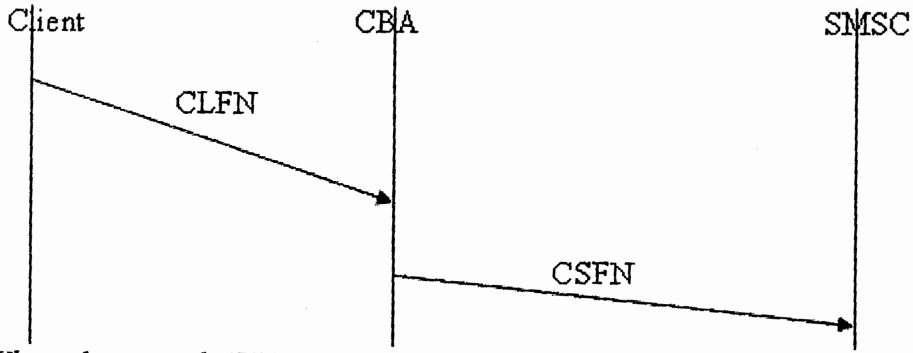


Fig 5.22: When client sends FIN request in order to terminate the chat session

Case 5: When end Mobile user sends FIN request in order to terminate the chat session

SMSC → CBA: SMFN
CBA → C : CCFN

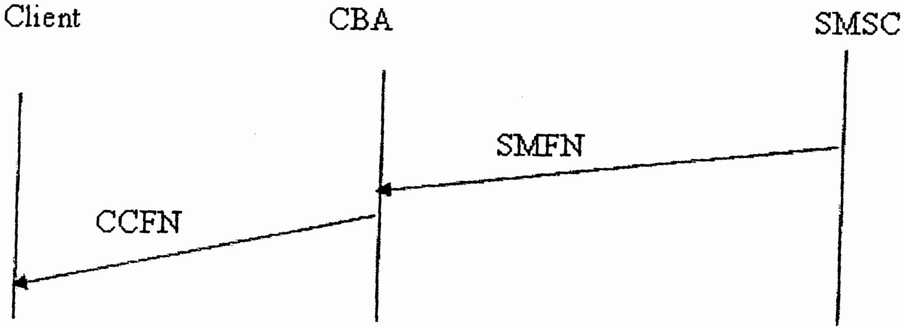


Fig 5.23: End Mobile user sends FIN

Case 6: Client does not send any SMS until timeout

CBA → SMSC: CLRT

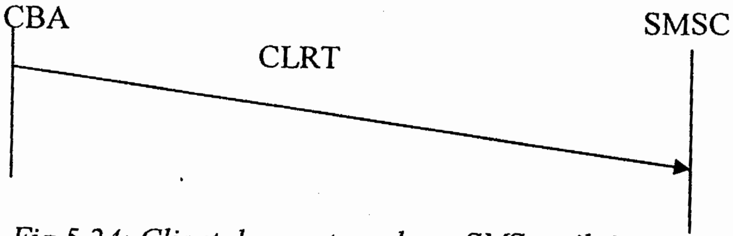


Fig 5.24: Client does not send any SMS until timeout

Case 7: The limit of maximum number of SMS allowed to be sent over one TCP connection reached

CBA → SMSC: CCML

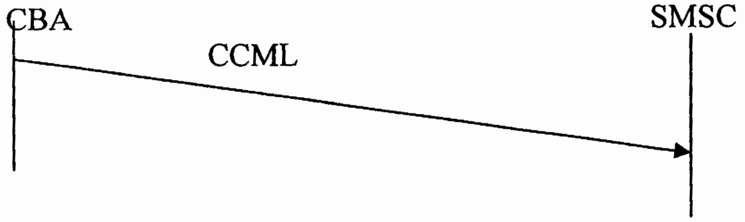


Fig 5.25: The limit of maximum number of SMS allowed for one TCP connection reached

5.4 Summary

Because of its elegant object oriented design, C# was taken on board to proceed implementation of the new dynamic and comprehensive solution as discussed in chapter 4. The implementation of the system is modular as well as complicated because of four separate modules that operate independently to achieve the common objectives. The SMS client is the module that supports the client directly; however it keeps the majority of security checks and features transparent from the user. SMS client keeps the traffic queue in its memory that is pulled by the CBA server. Token server accepts the client request, checks the authenticity and provides list of legitimate users to CBA server. Accordingly the CBA server interacts with the GSM network through SMSC in order to know willingness of the intended mobile user. As soon as destination is found willing to receive, the SMS communication is started between the nodes. The SMS traffic is pulled by the CBA server from the client's queue and delivered to the mobile user and vice versa.

6. Testing and Performance Evaluation

6. Testing and Performance Evaluation

For the purpose checking the practical performance and results, the CBA protocol was tested over a Local area network of 30 computers, each machine installed with Windows XP professional and Microsoft .NET framework 2.0. Two machines were installed with the Server software i.e. the Token Server and the other for CBA Server. Third one was used to run SMSC Simulator.

Different attack programs were written in order to take accurate results. These attack programs are divided into spoofed and zombie programs and send login requests of different sizes to Token Server.

6.1 Results of the CBA Protocol

Followings are the main scenarios we tested in order to evaluate the performance of CBA:

1. Authentication of clients when there was no spoofed or zombie request at Normal Traffic Time
2. Authentication of clients at the time of spoofed attack only at Normal Traffic Threshold
3. Authentication of clients at the time of zombie attack only at Normal Traffic Threshold
4. Authentication of clients when there was no spoofed or zombie request at Abnormal Traffic Threshold
5. Authentication of clients at the time of spoofed attack only at Abnormal Traffic Threshold
6. Authentication of clients at the time of zombie attack only at Abnormal Traffic Threshold
7. Number of authenticated clients when both zombie and spoofed attacks were launched simultaneous with Legitimate Requests
8. Authentication Rate of Legitimate and Zombie Requests
9. The Effect of Zombie and Spoofed Requests over the authentication rate of Legitimate Clients
10. Storage resources required by TCP and CBA to accommodate different numbers of connection requests
11. Comparison of storage resources consumed by simple SMS chat server and CBA protocol for Waiting clients

6.1.1 Authentication of clients when there was no spoofed or zombie request at Normal Traffic Threshold

To obtain the authentication rate of legitimate clients when there was no spoofed or zombie clients we performed various experiments with different number of legitimate login requests. When we perform these experiments at normal traffic time and the result shows that each and every request sends to Token Server gets authenticated. So the authentication rate was 100% for all the experiments performed with different values of legitimate requests.

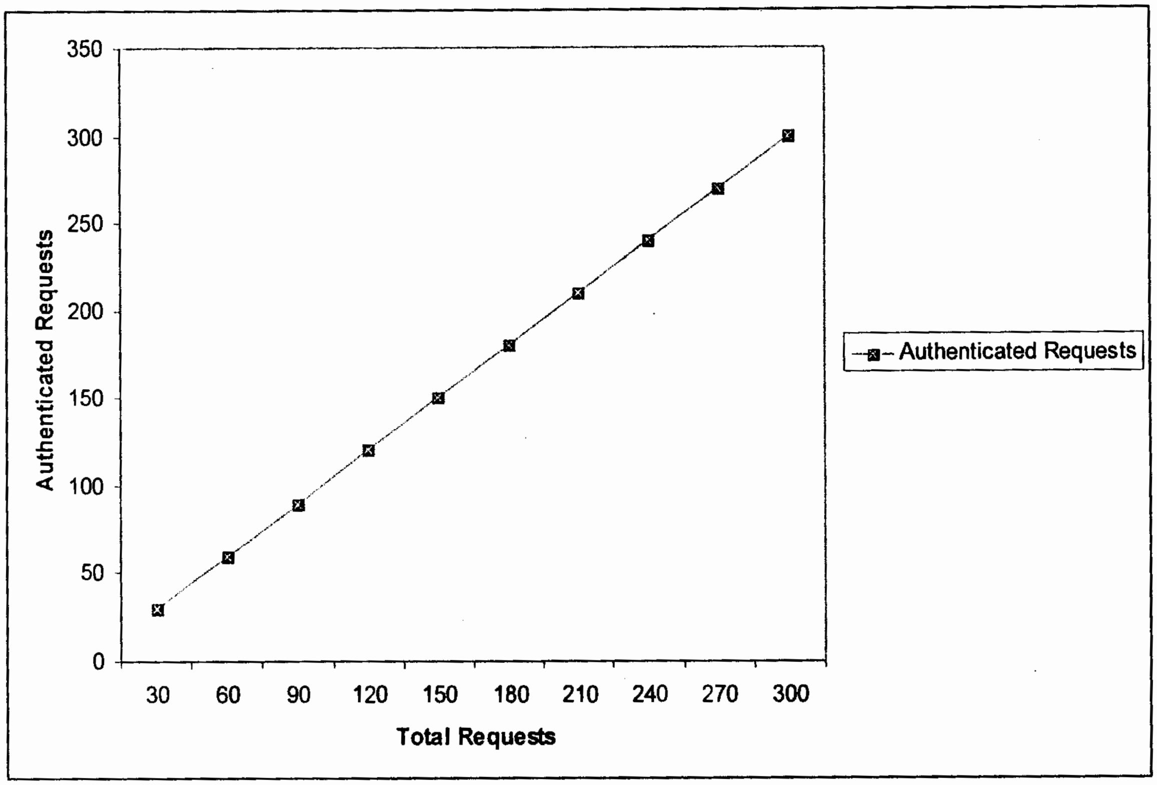


Fig 6.1: Authentication of Legitimate Clients when there is no Spoofed or Zombie Request at Normal Traffic Threshold

6.1.2 Authentication of clients at the time of spoofed attack only at Normal Traffic Threshold

The next set of experiment was performed to verify the authentication of clients when server was under attack by the spoofed requests at normal threshold value. The results show that only those clients get authenticated who were legitimate. Because client was required to submit the issued simple token back to Token Server that was not possible for the spoofed client so even not a single spoofed client got authenticated.

6.1.3 Authentication of clients at the time of zombie attack only at Normal Traffic Threshold

Next experiments were performed to verify the authentication of clients when server is under attack through zombie machines at normal traffic threshold. The results show that all the clients who were legitimate as well as zombie get authenticated. At the normal threshold time only simple token is used for the authentication and it is very easily possible for zombie machine to submit the same token back to the Token Server and get authenticated. So at the normal threshold all the zombie requests got authenticated.

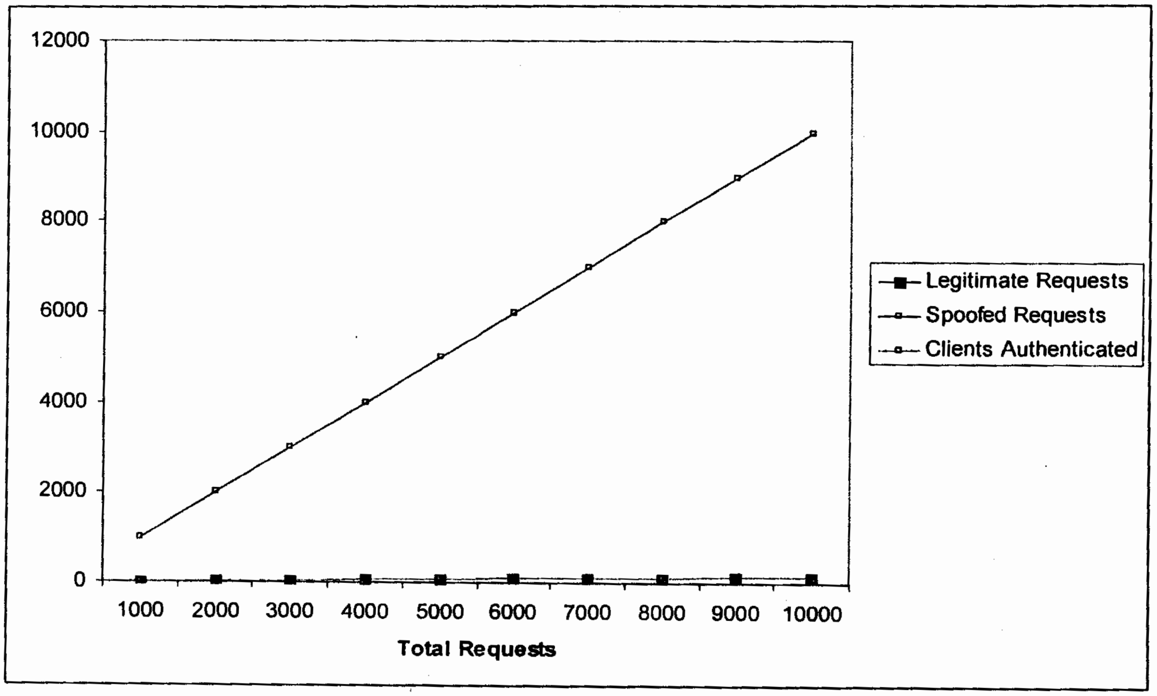


Fig 6.2: Authentication of Legitimate Clients at the time of spoofed attack only at Normal Traffic Threshold

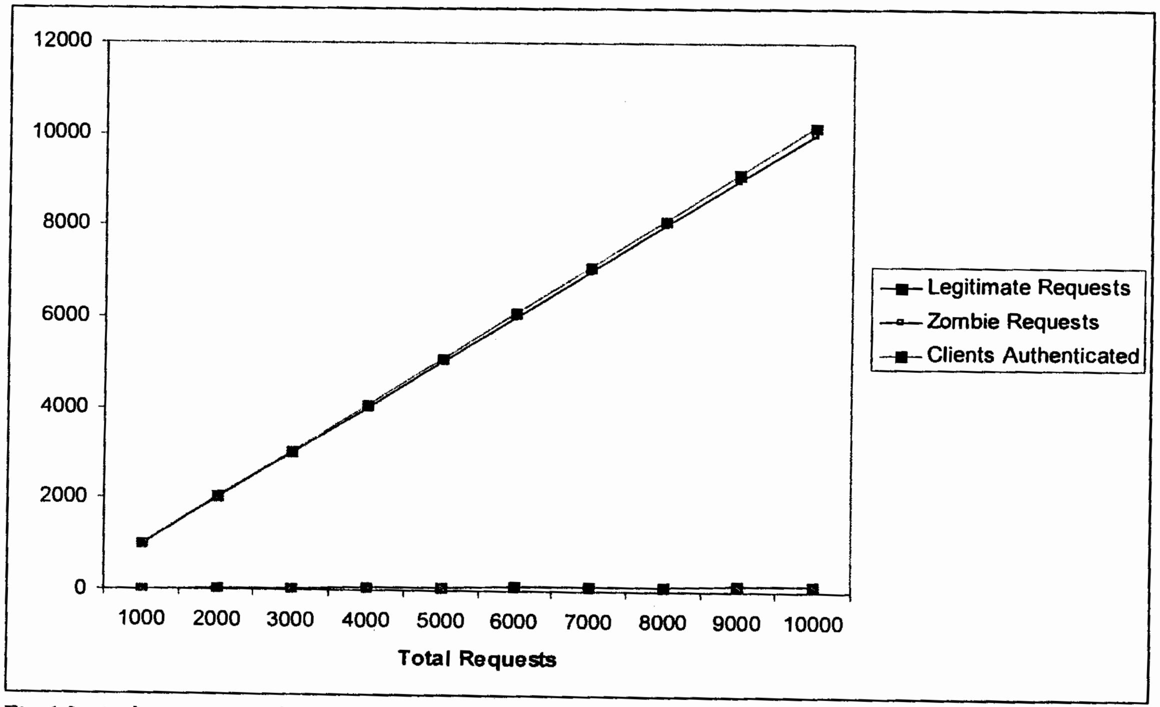


Fig 6.3: Authentication of Legitimate Clients at the time of Zombie attack only at Normal Traffic Threshold

6.1.4 Authentication of clients when there was no spoofed or zombie request at Abnormal Traffic Threshold

The results of theses experiments when performed at Abnormal Traffic Rate again shown 100% authentication rate but there were few user level errors. There user level errors occured due to typing wrong image token. While typing the letters printed on the image token few typing mistakes were made that resulted in user level errors. However all the clients who submitted valid image tokens got authenticated.

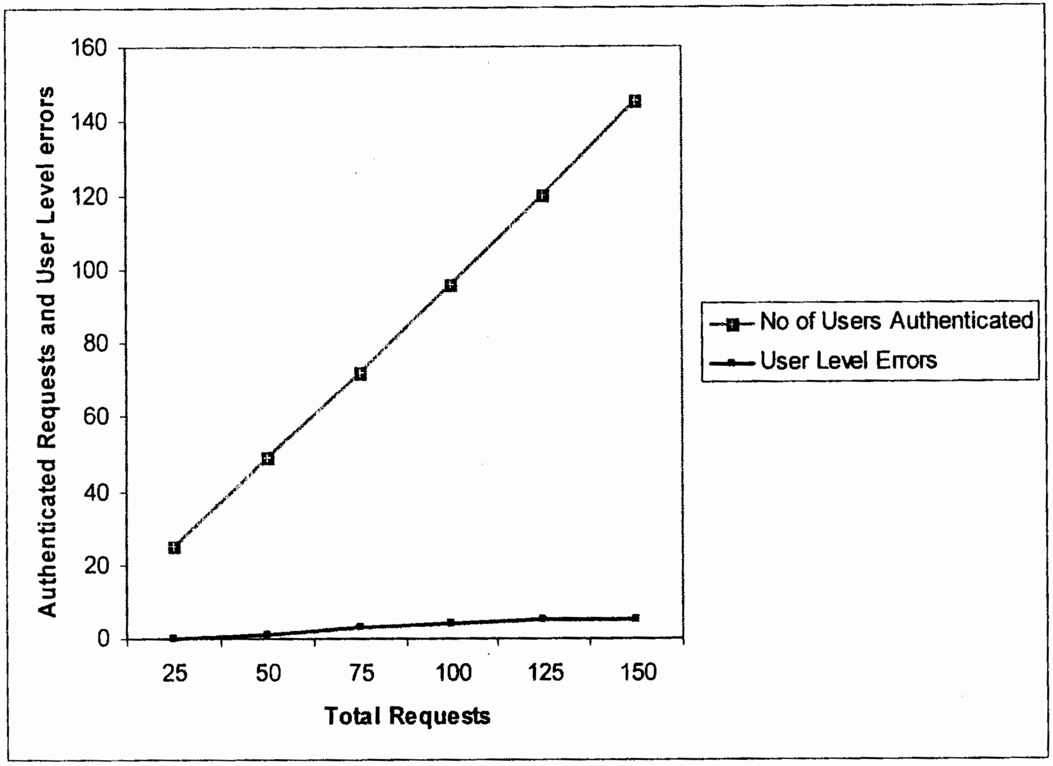


Fig 6.4: Authentication of Legitimate Clients when there is no Spoofed or Zombie Request at Abnormal Traffic Threshold

6.1.5 Authentication of clients at the time of spoofed attack only at Abnormal Traffic Threshold

These experiments were performed to verify the authentication of clients when the Server was under attack by the spoofed requests at abnormal traffic threshold. The results show that only those clients get authenticated who were legitimate. Even at abnormal threshold time spoofed requests were rejected with the help of simple token without issuing image token.

6.1.6 Authentication of clients at the time of zombie attack only at Abnormal Traffic Threshold

In these experiments we verify the authentication of clients when server was under attack through the zombie machines at abnormal traffic threshold. The results show that only those clients got authenticated who were legitimate. It is very easy for the zombie machine to submit the simple token back to the issuer but is near to impossible for the zombie to interpret the image and submit it in the textual format. So image token restricts the zombie requests from authentication.

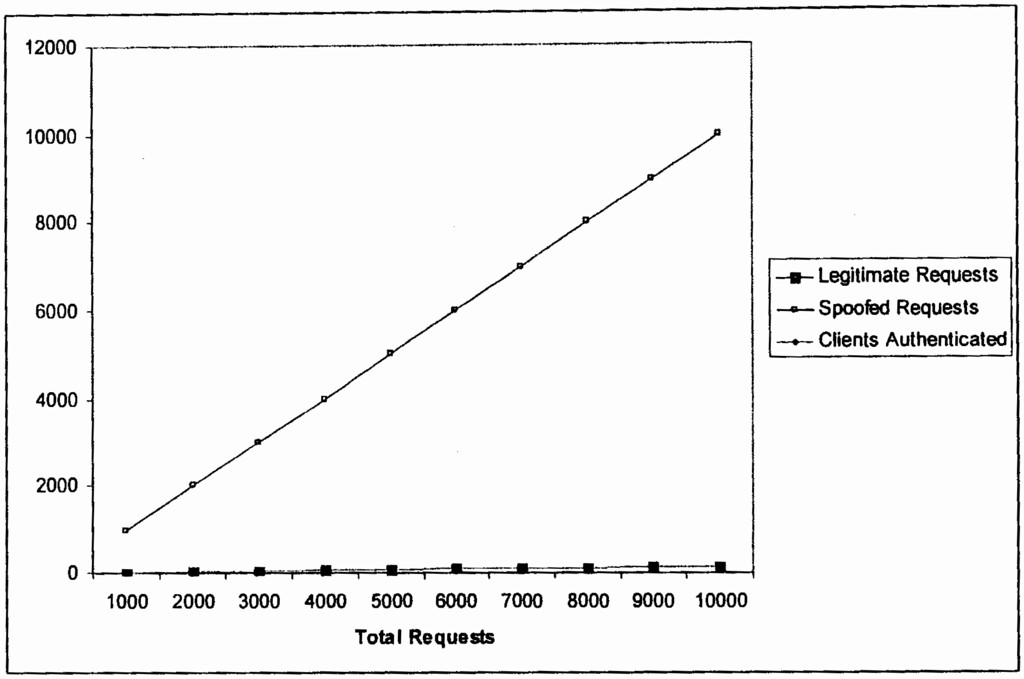


Fig 6.5: Authentication of Legitimate Clients at the time of spoofed attack only at Abnormal Traffic Threshold

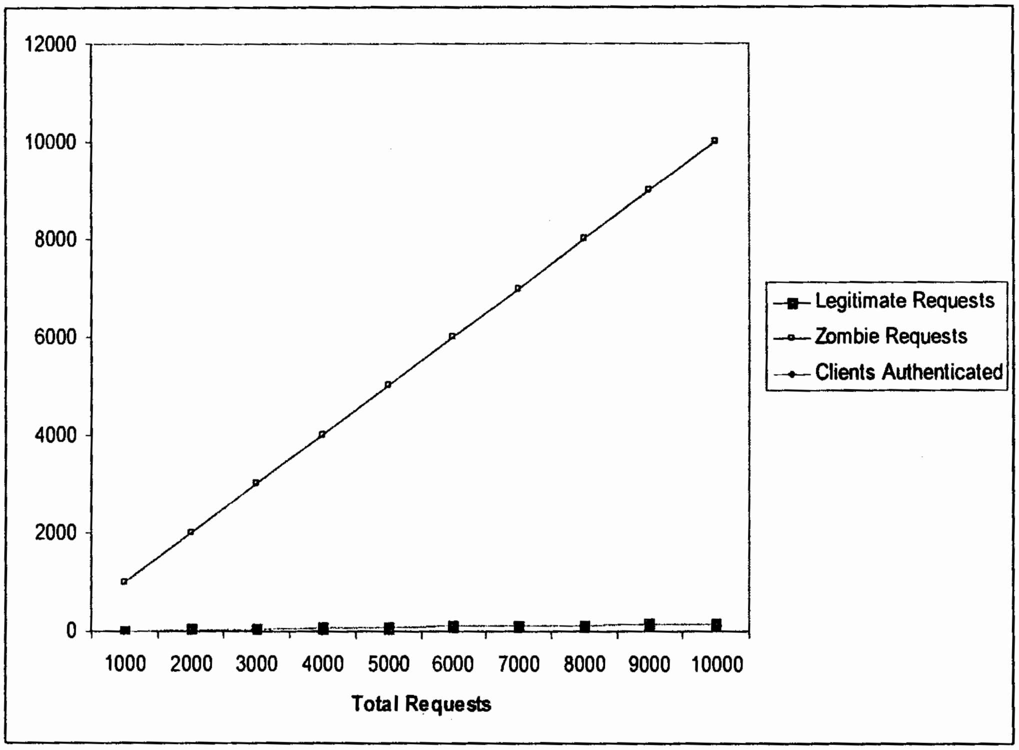


Fig 6.6: Authentication of Legitimate Clients at the time of Zombie attack only at Abnormal Traffic Threshold

6.1.7 Number of authenticated clients when both zombie and spoofed attacks launched at a time along with Legitimate Requests

In this experiment different numbėr of spoofed, zombie and legitimate requests were launched at different time.

First experiment was launched when there was normal traffic threshold. The results show that all the legitimate and zombie requests got authenticated while no spoofed request was entertained by the Token Server. So the total number of authenticated requests is the sum of legitimate and zombie requests.

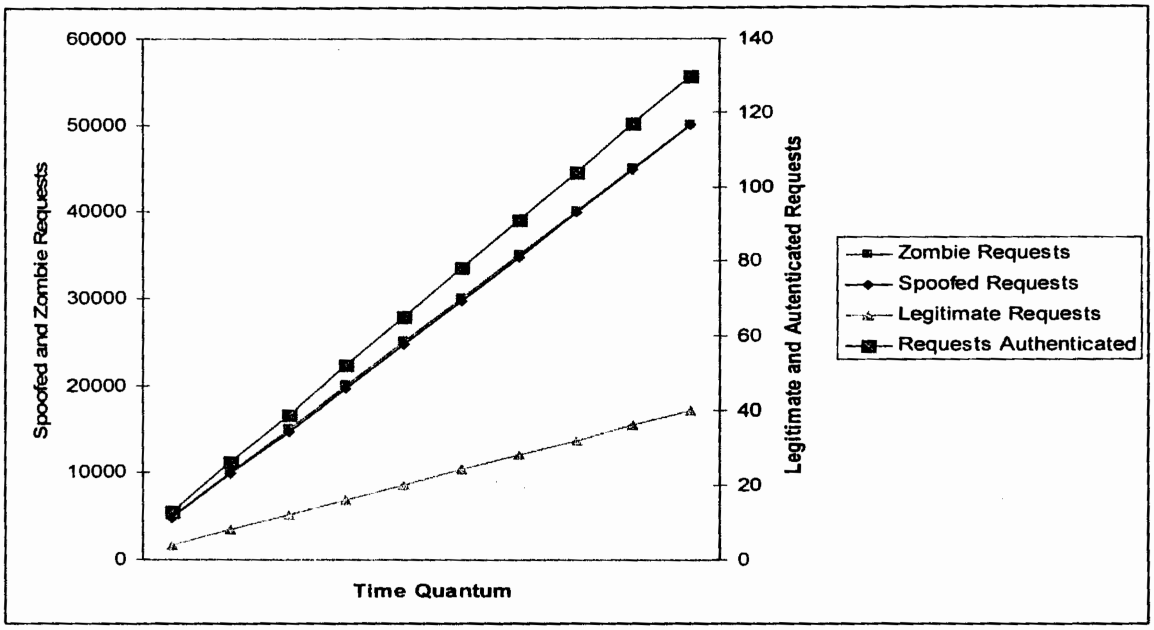


Fig 6.7: Number of authenticated clients when Threshold value set to 100

Above experiment was repeated at abnormal traffic threshold. That time results were very much different as compared to the normal traffic threshold experiment. First of all since abnormal traffic threshold involves submission of image token so few user level errors appeared in the results. Moreover no zombie or spoofed request was entertained by the Token Server. The total number of the authenticated clients consists of the difference of legitimate requests and user level errors.

So this experiment shows that at the abnormal traffic time Token Server does not entertain any of the zombie or spoofed requests. However at the normal traffic rate zombie requests were entertained by the Token Server.

6.2.8 Authentication Rate of Legitimate and Zombie Requests

In this section we perform various experiments with different number of legitimate and zombie requests at different threshold levels.

Results show that at Normal Traffic rate all the requests received from legitimate clients as well as from zombie machines got authenticated. So the authentication rate for both legitimate and zombie requests remain 100%. But as soon as traffic reached to threshold value and Token Server started issuing image token to the clients the authentication rate of the zombie clients drops to 0 while legitimate clients were getting 100% authentication.

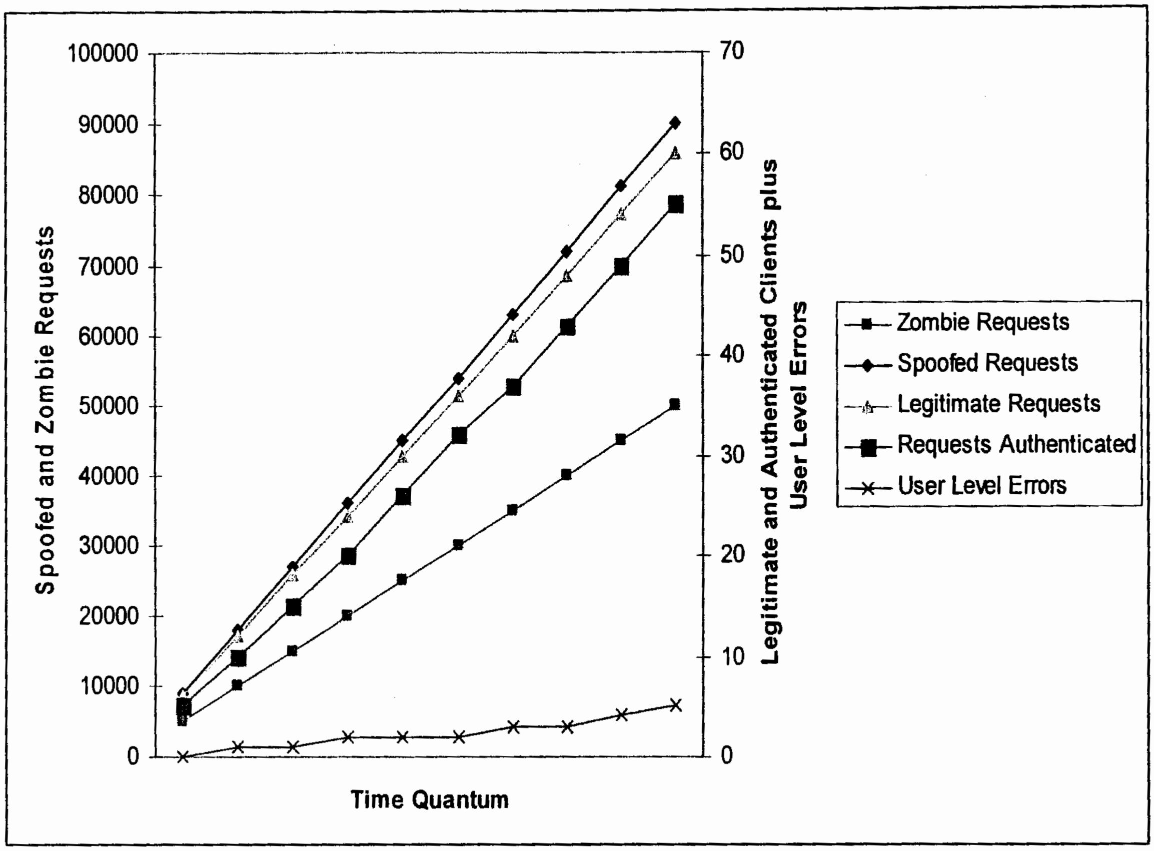


Fig 6.8: Number of authenticated clients when Threshold value set to 0

These results show that the zombie attacks can only be successful until there is normal traffic rate. As soon as the traffic rate gets abnormal no zombie request will be entertained. When there is normal traffic threshold the authentication of zombie client can not cause Denial of service.

6.2.9 The Effect of Zombie and Spoofed Requests over the authentication rate of Legitimate Clients

These results were obtained by performing various experiments with different sequences of legitimate, zombie and spoofed requests.

In the first experiment the legitimate requests were started in the first minute of the experiment. For the first three minutes there was no spoofed or zombie request, so the authentication rate of the legitimate requests did flow on 100%. At the start of fourth minute spoofed attack was launched that continued till the end of the experiment. But the launching of spoofed attack dis not affect the authentication rate of legitimate client at the rate of 100% success. At the start of

6th minute zombie attack also came into action and started sending zombie requests to Token Server. But again the mixture of spoofed and zombie requests did not affect the authentication rate of the legitimate clients that remained 100% till the end of experiment.

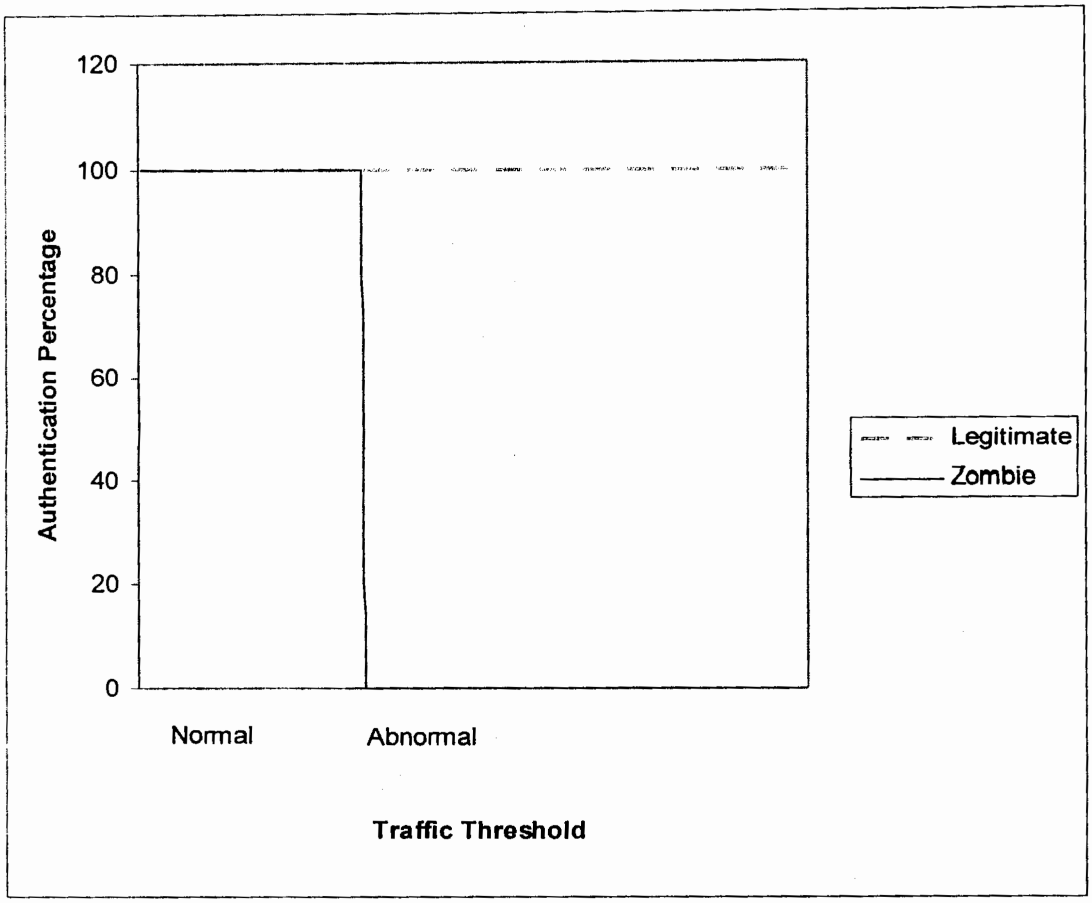


Fig 6. 9: Authentication Rate of Legitimate and Zombie Requests at Normal and Abnormal Traffic Threshold

Next experiment was started with zombie attack. First legitimate request was issued in the 4th minute of the experiment and spoofed attack was launched at the start of 9th minute. Again the results were same. Start with zombie attack also failed to reduce the authentication rate of the legitimate clients. Legitimate clients still got 100% authentication rate.

Last experiment was started with spoofed and zombie attacks simultaneously. The legitimate clients started sending requests in the 6th minute of the experiment. Results show that launching the spoofed and zombie attack simultaneously at the start of the experiment does not make any difference in the authentication rate of the legitimate clients. Legitimate clients even then get 100% authentication

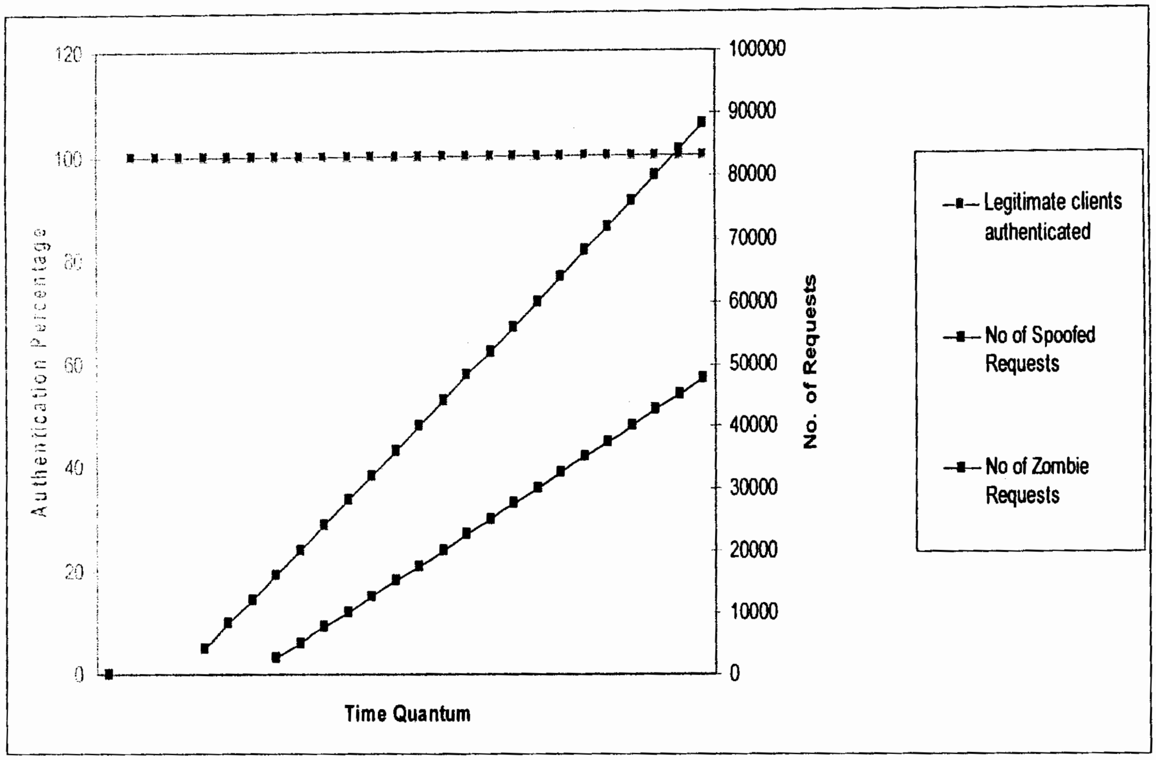


Fig 6.10 (a) The Effect of Zombie and Spoofed Requests over the authentication rate of Legitimate Clients when Legitimate clients were in action while spoofed and zombie attack launched in between

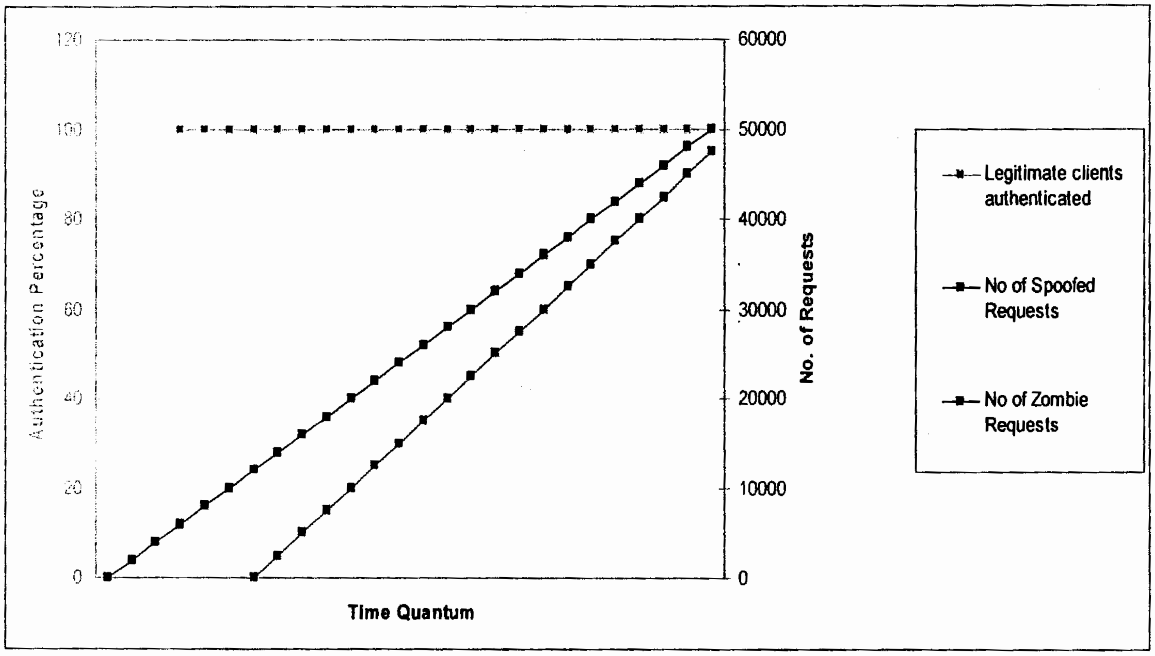


Fig 6.10: (b) The Effect of Zombie and Spoofed Requests over the authentication rate of Legitimate Clients when Zombie attack launched at the start and spoofed attack at the last

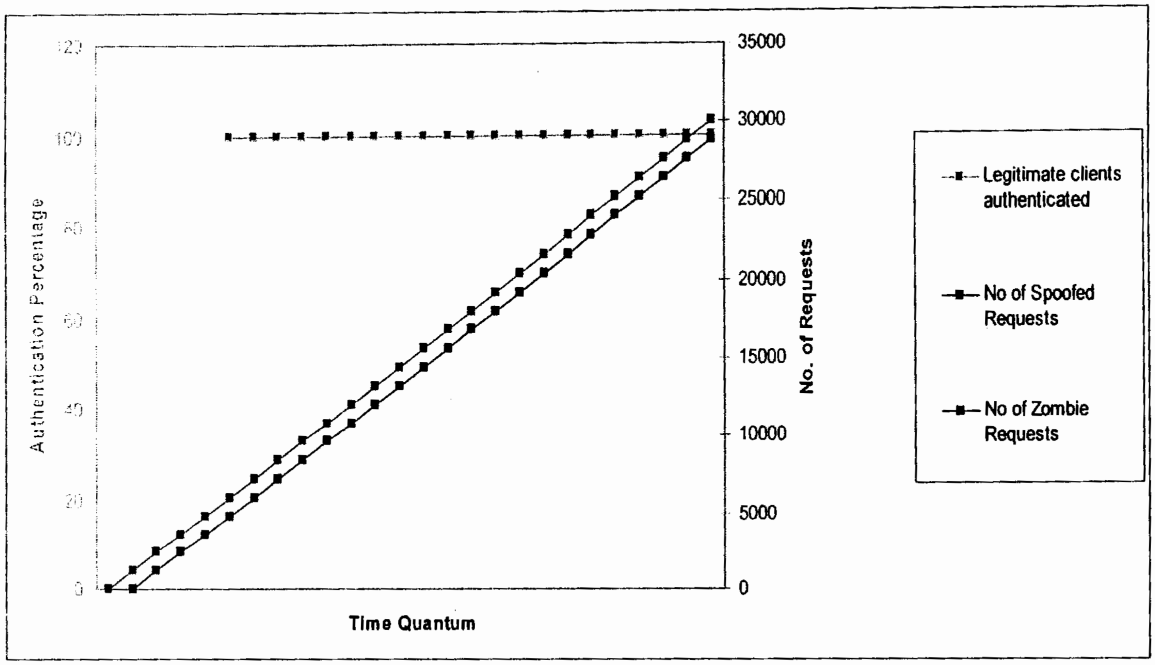


Fig 6.10: (c) The Effect of Zombie and Spoofed Requests over the authentication rate of Legitimate Clients when Zombie and spoofed attack launched at the start of the experiment

6.2.10 Storage resources required by TCP and CBA to accommodate different number of connection requests

TCP is the most widely used protocol of the Internet. According to [23] TCP reserves storage resources of 280 bytes for every incoming SYN request. Moreover it allocates these 280 bytes to spoof and zombie requests as well. So in other words normal working of the TCP can not differentiate among spoofed, legitimate, and zombie requests. Therefore it is relatively easy to launch DoS attack over TCP. Many solutions are proposed to prevent the TCP from DoS attack but neither of the solution is capable to fully eliminate this threat. Our CBA protocol performs the authentication process through UDP messages and transfer the SMS data through TCP.

In this regard we compare the storage resources required by the TCP and CBA to accommodate different number of connection requests.

CBA requires maximum 13 bytes per connection at the abnormal traffic threshold and 8 bytes at the normal traffic threshold. 13 bytes at the abnormal traffic threshold includes 8 bytes simple token and 5 bytes image token text while the 8 bytes of the normal traffic threshold consist of simple token only.

6.2.11 Comparison of storage resources consumed by simple SMS chat server and CBA for waiting clients

A normal SMS consists of 160 bytes. When simple chat server receives an SMS it also includes phone number (Normally of 11 bytes), Nick (Normally of 8 bytes) and IP address of the client that is of 4 bytes. So an SMS normally consists of 183 bytes. When client sends an SMS to the simple chat server it is stored at the server side. So the queue is established at the server that results in occupying storage space of the server.

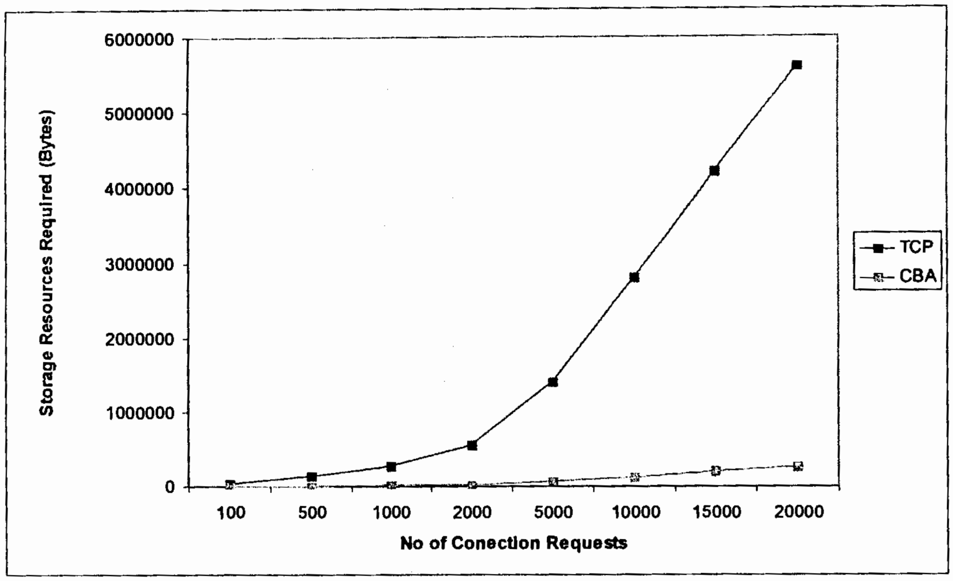


Fig 6.11: Storage resources required by TCP and CBA to accommodate different numbers of connection requests

CBA only keeps the necessary data in its queue while the SMS data is stored at the client side. CBA pulls the data from client. This feature of CBA protects it from many types of security threats including Denial of Service attack through data bursts or SYN flood.

The necessary data that CBA stores for each authenticated client includes IP address of the client (4 bytes), port of the client used for communication (2 bytes), Nick (8 bytes) and the Phone number of the other end user (11 bytes). So it totally stores 25 bytes per user. The SMS data of 160 bytes is stored at the client side.

In the graph below we present the comparison between simple chat server and CBA with respect to the storage space consumed by the clients in waiting.

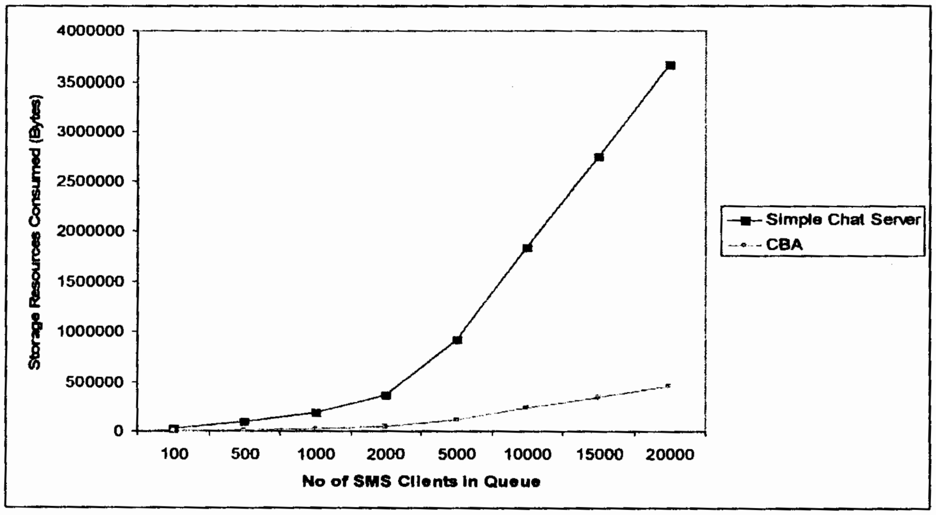


Fig 6.12: Comparison of storage resources consumed by simple SMS chat server and CBA for waiting clients

6.2 Summary

Spoofed and zombie requests are two major threats to the web to cellular phone SMS service. Most of the graphs we mentioned in this chapter reflect the effect of these two attacks over web to cellular phone SMS service. Results shows that CBA is very much capable to mitigate the effect of spoofed and zombie attack. Results shows that in the presence of spoofed and zombie requests the authentication rate of the legitimate clients remain 100%. We see that zombie requests get authenticated at normal threshold time but as the requests reaches to abnormal threshold value no zombie requests is entertained by the Token Server. Spoofed requests are denied by the simple token issuing mechanism both at normal and abnormal traffic threshold.

We also compare the storage resources of CBA protocol with that of TCP protocol and results show that CBA requires less storage resources as compared to TCP.

The traditional chat server storage resources required to queue an SMS are compared with the storage resources required by CBA server for an SMS. Again CBA requires very few resources as compared to chat server. It is because CBA does not stores the SMS at server rather it is stored at client machine and pulled by the CBA turn by turn.

7. Conclusion and Outlook

7. Conclusion and Outlook

7.1 Conclusion and Achievements

As we have seen that web to cellular phone SMS service is one of the popular services of GSM technology. This service attaches the cellular network with the Internet. So the security threats that Internet is facing now a days also become threats for the cellular network. Attack launched through this service not only effect the targeted service but can jam the whole cellular network including voice calls. So to secure the whole cellular network this service needs to be secure.

A protocol namely as "CBA" was designed that secures the web to cellular phone SMS chat in such a way that all the other services become fully secure and this particular service becomes partially secure.

In this research new Client-Server Architecture is presented for security in which Pull Data Traffic Control (PDTC) mechanism was used. PDTC mechanism not only secures the network but also reduces the traffic load and resources of the server.

In order to stop an attack launched through spoofed IP address a token issue schemes that requires the token to be return was used. Since spoofed IP addresses are not been able to return back the token so they are blocked. Zombie attacks are stopped by issuing an image token. The client needs to return the image token in textual format that is not possible for zombie.

Another important aspect of our research is Throat-Knot the attack at the gateway. Since attacker's maximum reach is up to the gateway, it is not possible for the adversary to get access inside the network.

The Two Tiers of the server not only secures the CBA Server but also provides an ease of implementation to the service provider to place their Token Server with the third party service providers.

The new client server architecture for security secures the server from SYN flooding. In the normal client server architecture TCP connection is initiated by the client and server listens to it. Since the client have the control to push traffic so it is up to the client when he pushes the traffic and how much. In our client server security architecture client goes to listening mode and server initiates the connection. So the control is with the server and it is up to the server to pull the traffic according to its capacity.

7.2 Improvements

Priority setting is a major productive parameter for efficient delivery of services and advancement in services sector. Equal priority to all the clients is neither a prevailing nor recommended practice in the service delivery. Clients are always discriminated and served according to priority settings. However the CBA server provides the services to all the clients on equal priority basis. Although web to cellular phone SMS communication service is being provided on equal priority basis by all the cellular operators free of cost, but future business dynamics may demand priorities on the basis of charges etc

Keeping in view of the different traffic scenarios that depend upon the number of customers served by particular GSM operator and the social events of a particular geographical area, the normal and abnormal values of traffic threshold is accepted by the CBA through manual configuration. However intelligent algorithms can be designed on the basis of historical data etc that can calculate the values of normal and abnormal traffic threshold.

7.3 Future Recommendations

There is high potential of Future enhancement in this work into different dimensions because of its dynamic and innovative nature. Few of the future goals are mentioned below:

First is to implement this new Client Server Security Suite on the other TCP based application layer protocols such as SMTP, FTP etc in order to check what level of security this suite can provide specially against SYN flooding attack.

GPRS is another service that connects the cellular phone with the web. So again security threats faced by the Internet become the security threat of the cellular network as well. CBA can be modified in order to secure the threats associated with the GPRS.

References

References

[1] Media Centre, GSM Association. http://www.gsmworld.com/news/statistics/index.shtml.

[2] Ministry of IT & Telecom, Government of Pakistan. http://www.moitt.gov.pk/

[3] International Engineering Consortium. http://ww.iec.org

[4] William Enck, Patrick Traynor, Patrick McDaniel, and Thomas La Porta "Exploiting Open Functionality in SMS Capable Cellular Networks" "ACM, CCS 05, Nov 7-11, 2005, Alexandria, Virginia, USA."

[5] Wikipedia: The biggest Encyclopedia. www.en.wikipedia.com

[6] Pakistan Telecom Authority. http://www.pta.gov.pk

[7] Jun Li, Jelena Mirkovic, Mengqiu Wang, Peter Reiher, and Lixia Zhang "SAVE: Source Address Validity Enforcement Protocol"

[8] Andy Stone, Jonathan Briggs, Craig Smith "SMS and interactivity – Some results from the field and its implications on effective uses of Mobile technologies in education" "Proceedings of the IEEE international workshop on wireless and mobile technologies in education, 2002"

[9] Lim Tai Ching, H K Garg "Designing SMS application for public transport service system in Singapore"

[10] Heng Xu, Hock Hai Teo, Hao Wang "Foundations of SMS Commerce Success: Lessons from SMS Messaging and Co-opetition" "Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS'03), 2003"

[11] Ke wan "An SMS-Based Sales Reporting System for a Fashion-Clothes Franchising Company"

[12] Beh Kok Sang , Abdul Rahman Bin Ramli, V Prakash, Syed Abdul Rahman Bin Syed Mohamed " sms gateway interface - remote monitoring and controlling via gsm sms"

[13] Heng Xu, Hock Hai Teo, Hao Wang "Foundations of SMS Commerce Success: Lessons from SMS Messaging and Co-opetition" "Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS'03), 2003"

[14] Michael Sirivianos Ersin Uzun Ines Viskic "SANALDA: A Source Authenticating Network Architecture Limiting DoS Attacks"

[15] S. R. D. Kalingamudali, J. C. Harambearachchi, L. S. R. Kumara, J. H. S. R. De Silva, R. M. C. R. K. Rathnayaka, G. Piyasiri3, W. A. N. Indika, M. M. A. S. Gunarathne, H. A. D. P. S. S. Kumara, M. R. D. B. Fernando "Remote Controlling and Monitoring System to Control Electric Circuitry through SMS using a Microcontroller" "First International Conference on Industrial and Information Systems, ICIIS 2006, 8 - 11 August 2006, Sri Lanka"

[16] Petros Zerfos, Xiaoqiao Meng, Starsky H.Y Wong, "A Study of the Short Message Service of a Nationwide Cellular Network" "IMC'06, October 25–27, 2006, Rio de Janeiro, Brazil. Copyright 2006 ACM 1595935614/06/0010"

[17] Xiaoqiao Meng, Petros Zerfos, Vidyut Samanta, Starsky H.Y. Wong, Songwu Lu "Analysis of the Reliability of a Nationwide Short Message Service"

[18] Jeremy Serror, Hui Zang, Jean C. Bolot, "Impact of Paging Channel Overloads or Attacks on a Cellular Network" "WiSe'06, September 29, 2006, Los Angeles, California, USA Copyright 2006 ACM 1XXXXXXXXX/ 06/0009 overloads."

[19] William Enck, Patrick Traynor, Patrick McDaniel, and Thomas La Porta "Mitigating attack on Open Functionality in SMS Capable Cellular Networks" "ACM, MobiCom'06, Sep. 23-26, Los Angeles, California, USA"

[20] Christoph L. Schuba, Ivan V. Krsul, Markus G. Kuhn, "Analysis of a Denial of Service Attack on TCP"

[21] http://www.gsmworld.com/news/statistics/index.shtml

[22] Michael T. Goodrich "Efficient Packet Marking for Large-Scale IP Traceback" CCS'02, November 18–22, 2002, Washington, DC, USA.Copyright 2002 ACM 1-58113-612-9/02/0011

[23] Snoeren, A.C. Partridge, C. Sanchez, L.A. Jones, C.E. Tchakountio, F. Schwartz, B. Kent, S.T. Strayer, W.T. "Single-packet IP traceback" Networking, IEEE/ACM Transactions on, page(s): 721- 734, Dec 2002

[24] Minh Sung ,Jun Xu "IP Traceback-Based Intelligent Packet Filtering: A Novel Technique for Defending against Internet DDoS Attacks" IEEE transactions on parallel and distributed systems, Sep 2003

[25] Bao-Tung Wang; Schulzrinne, H "An IP traceback mechanism for reflective DoS attacks" Canadian Conference on Electrical and Computer Engineering, 2004.

[26] Bao-Tung Wang; Schulzrinne, H "A denial-of-service-resistant IP traceback approach" Proceedings. ISCC 2004. Ninth International Symposium on Computers and Communications, 2004

[27] Basheer Al-Duwairi and G. Manimaran "A Novel Packet Marking Scheme for IP Traceback" 2004

[28] Jun Li, Minho Sung, Jun (Jim) Xu, Li (Erran) Li "Large-Scale IP Traceback in High-Speed Internet: Practical Techniques and Theoretical Foundation" 2003

[29] Abraham Yaar Adrian Perrig Dawn Song "Pi: A Path Identification Mechanism to Defend against DDoS Attacks" 2003

[30] B.-T. Wang (USA) "Tracing High Bandwidth Aggregates" Proceeding (440) communication, Network, and Information Security - 2003

Appendix I

Short Message Service: A Brief History

- In 1985 SMS was defined as a part of GSM series of standards
- Original message length was 160 characters
- The Mobile Application Part (MAP) of the SS7 protocol included support for the transport of Short Messages
- The first commercial SMS message was sent over the Vodafone GSM network in the United Kingdom on 3rd December 1992, from Neil Papworth of Airwide Solutions (using a personal computer) to Richard Jarvis of Vodafone (using an Orbitel 901 handset). [wikipedia]
- The first SMS typed on a GSM phone is claimed to have been sent by Riku Pihkonen, an engineer student at Nokia, in 1993. [wikipedia]
- The first consecutive commercial deployments were by Acision with Telenor in Norway and BT Cellnet (now O2 UK) in 1993. [wikipedia]
- By 2007 an average of 9 million texts were sent every hour during New Year's Day in the UK alone. []
- In 2001 e2sms, in conjunction with KMS Software (London), launched 2 way web SMS and email.

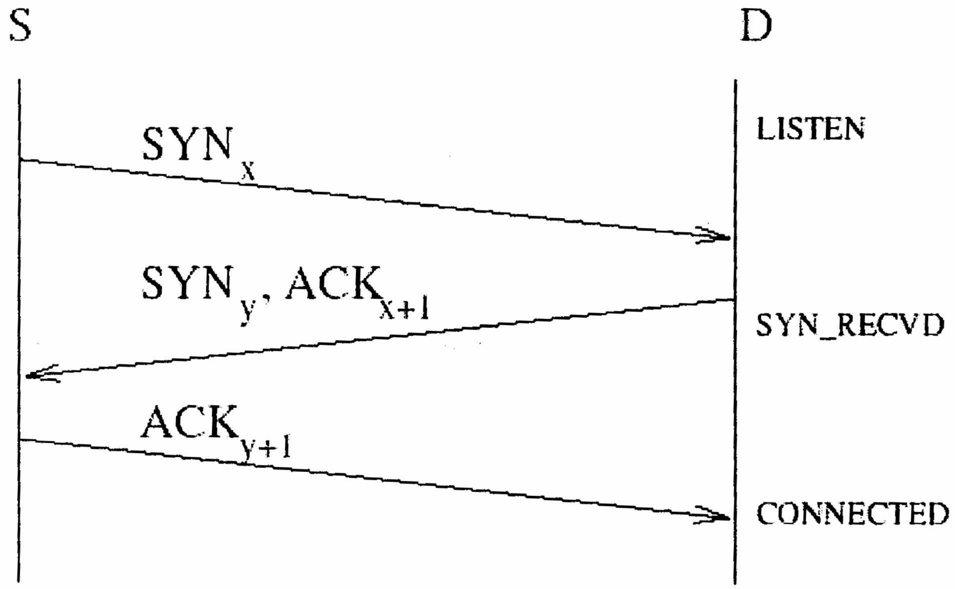
Appendix II

TCP Syn Attack and Counter Measures

From last 10 to 12 years Internet is under the threat of denial of service attack due to syn flooding. This attack exploits the weaknesses of TCP/IP protocol suit. The attackers can easily launch this type of attack and it is very difficult to identify the attacker.

To ensure the reliable communication between two users TCP is the best available protocol. It is connection oriented protocol. It guarantees the ordered delivery of data from sender to receiver. Before the transmission of the data a logical connection is established between the two endpoints. The connection establishment process consists of three way handshake.

First client sends a syn packet to the server, server in response acknowledge client's syn packet and as well as send its on syn packet. When client receives server's syn packet ie send acknowledgment to the server.



TCP: Three way handshake process

When a syn arrived at a server, server allocates many resources to the half open TCP connection. This behavior of TCP gives birth to denial of server attack. An attacker can send a bulk of TCP syn requests to a server in order to overload the half open queue of the server. Once the maximum number of half open connection per port reached, no more

- Another solution solve the problem is to change the connection establishment process. When a server receives a syn packet, it does not allocate resources to the connection, in fact it calculates Y , a hash function of the client's IP address and port plus a secret key use by the server. When server receives back the acknowledgment it can match the value of Y. if the value matched connection is established else it is denied.

  This solution seems better as compared to above mentioned solutions, but this solution again requires resources to calculate the hash function at the time of syn and matching process at the time of acknowledgment. So again attack can be launched but this time attack will be on the processing load of the server instead of the memory limit of the server.

- Few solutions are based on firewall. First solution is to induct a firewall between client and server. When a client requests firewall acknowledges it and sends its own syn. Upon receiving the acknowledgment from user it completes the connection by performing same steps with the server.

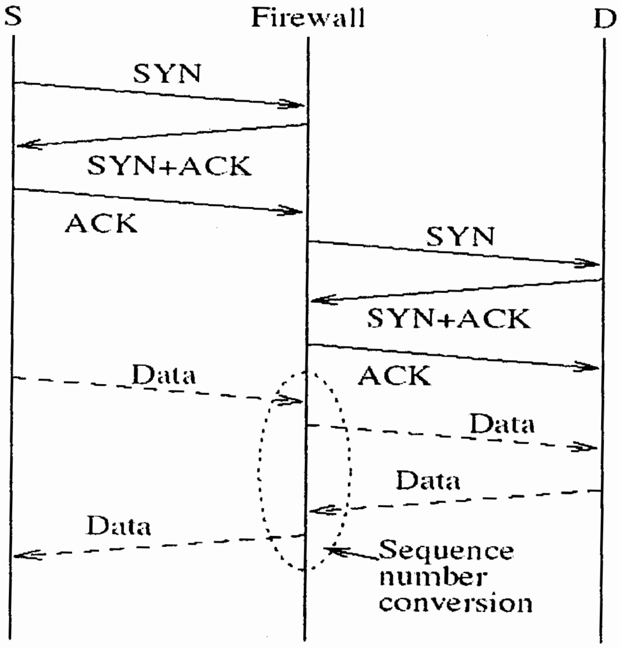


Fig B-1: Firewall based solution

This solution is a better option for the network security administrator. Timeout value should be selected very carefully in case of firewall based solution.

Appendix III

GSM Statistics

- GSM is the fastest growing communications technology of all time.
- Today, GSM accounts for 82% of the global mobile market.
- 29% of the global population use GSM technology.
- The GSM Association currently has operator members in 218 countries and territories.

Subscriber statistics at the end of March 2007 [21]

World	**2,831,345,390**	
• GSM	2,278,095,380	80.5%
• 3GSM (WCDMA)	114,664,827	4.0%
• CDMA	18,138,942	0.6%
• CDMA 1X	289,963,166	10.2%
• CDMA 1X EV-DO	57,376,347	2.0%
• TDMA	16,235,932	0.6%
• PDC	27,857,370	1.0%
• iDEN	26,494,743	0.9%
• Analog	2,518,683	0.1%

GSM Technologies [21]

World	**2,392,760,207**	
• Africa	208,498,137	9%
• Americas	234,821,455	10%
• Asia Pacific	924,047,562	39%
• Europe: Eastern	349,952,186	15%
• Europe: Western	444,426,302	19%
• Middle East	136,649,157	6%
• USA/Canada	94,365,408	4%

Top 10 growth countries [21]

GSM net additions in Q1 2007

• China	18,040,914
• India	13,728,036
• Pakistan	7,662,993
• Indonesia	5,394,269
• Iran	5,135,330
• Brazil	3,877,141
• Argentina	3,809,765
• Nigeria	3,321,118
• Thailand	3,255,817
• Russian Federation	3,215,204