

Library copy

Lib  
17/1/98

# SECURE COMMUNICATION AND FIREWALL DESIGN

Dissertation submitted in partial fulfilment  
of requirements for the award of the  
degree of

**Master of Technology  
in  
Computer Science**

by  
**Vinod kumar**



**SCHOOL OF COMPUTER & SYSTEM SCIENCES  
JAWAHARLAL NEHRU UNIVERSITY  
NEW DELHI - 110 067**

January, 1998.

005.8

TH

K9606

Se




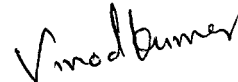
TH6844

## CERTIFICATE

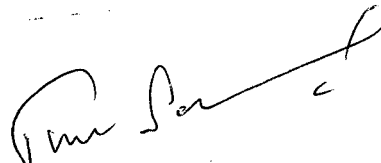
This is to certify that the dissertation entitled "SECURE COMMUNICATION AND FIREWALL DESIGN" being submitted by Mr. Vinod Kumar to the School of computer and system sciences, Jawaharlal Nehru University, New Delhi, in partial fulfillment of the requirement for the award of the degree of Master of Technology in Computer Science, is a bonafide work carried by him under the guidance and supervision of Dr.(Mrs.) S. Minz.

The matter embodied in the dissertation has not been submitted for the award of any degree or diploma.

  
5/11/98  
**Dr.(Mrs.) S. Minz**  
Ass. Prof., SC & SS  
Jawaharlal Nehru University  
New Delhi - 110 067.

  
**Vinod Kumar**



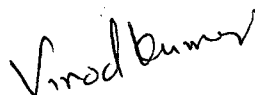
  
**Prof. P.C. Saxena**  
Dean, SC&SS  
Jawaharlal Nehru University  
New Delhi - 110 067.

## ACKNOWLEDGMENT

I would like to express my profound gratitude and personal regards to my guide **Dr.(Mrs.) S. Minz** who provided the primary incentives for development of this project. I am very thankful for her time to time questions, suggestion and constructive criticism which shaped the project design and development.

It is my pleasure to acknowledge the assistance of many friends and colleagues who encouraged me while I was doing my project which contributed measurable to quality of the thinking.

Last, but not least, my friends Gangesh, Mrinal, Rajib and F. minz gave me lot of emotional support without which I could never have succeed. Whenever I needed some encouragement, they some how sended that and provided it at just the right time.

  
**Vinod kumar**

# CONTENTS

<b>Preface</b>		<b>VI</b>
<b>Chap.1</b>	<b>INTRODUCTION</b>	<b>1</b>
	1.1. Networks	1
	1.1.1 Internetwork	1
	1.2 Need for Secure Communication	3
	1.3 Firewall	4
	1.4 Problem Definition	5
	1.5 Organizaiton of the dissertation	6
<b>Chap. 2</b>	<b>BACKGROUND AND MOTIVATION</b>	<b>7</b>
	2.1 Cryptography	7
	2.1.1. Secret Key	7
	2.1.2 Public key Cryptography	7
	2.2. Key Distribution Problem	9
	2.2.1 The RSA Alogrighm	9
	2.2.2 Authentication Protocal	11
	2.2.3 Digital Signature	12
	2.3 Firewalls	13

<b>Chap. 3</b>	<b>DESIGN ISSUES</b>	16
	3.1 Firewalls	16
	3.2 Router	18
	3.3 IP Datagram Format	20
	3.4 Design of Specific IP Datagram Packet Filter	24
<b>Chap. 4</b>	<b>ROUTER ALGORITHM AND PERFORMANCE</b>	29
	4.1 Algorithms	29
	4.1.1. Algorithm for send to Receiver functions	29
	4.1.2 Algorithm for check route function	30
	4.1.3 Algorithm for creat Table function	31
	4.2 Testing	33
	4.2.1 Test 1	33
	4.2. 2 Test 2	33
	4.2.3 Test 3	34
	4.2.4 Test 4	34
	4.3 Algorithm Evaluation	35
<b>Chap. 5</b>	<b>CONCLUSIONS</b>	36
	<b>BIBLOGRAPHY</b>	37

## LIST OF FIGURES AND TABLES

Figure 3.1	20
Figure 3.2	20
Figure 3.3	21
Figure 3.4	23
Table 3.1	24
Table 3.2	26
Table 4.1	33
Table 4.2	33
Table 4.3	34
Table 4.4	34

## ABSTRACT

Data transformation or encryption/decryption techniques are often utilized among senders and receivers to achieve secure communication. Since the data encryption/decryption requires sharing of a secret key, finding an efficient way to distribute the public key in a large-scale, truly distributed network has been a nontrivial task. We assume the target network consists of many locally trusted centers, and each center has many user attached to it. The scheme incorporates the public-key distribution concept and the RSA encryption scheme as the basic mathematical tools.

Firewalls are just a modern adaptation of the old security standby. When each packet entering or leaving a network is checked for its validity. However a high security LAN/Organization for a specific time duration and limiting the permission or access of facilities. Here we deal with the problem of time bound limited facility for an improved high security organization using firewall.

# CHAPTER 1

## INTRODUCTION

### 1.1 Networks

Communications between two or more computers are called computer communication and physical medium used for the communication is known as network.

The old model of a single computer serving all of the organisational needs, is rapidly being replaced by one in which a large number of separate but interconnected computers do the job. These systems are called computer networks.

Computer network is an interconnected collection of autonomous computers. Two computers are said to be interconnected if they are able to exchange information, knowledge and data. The connection may be via a copper wire, lasers, microwaves, and satellites communication.

#### 1.1.1 Internetwork

To exchange information from one computer to other distant computer, it requires network or collection of networks like Local Area Network (LAN), Wide Area Network (WAN), and Metropolitan Area Network (MAN) etc. The networks in the world, are often with different hardware and software. A person connected to one network often wants to communicate with people attached to a different network. This desire requires connection with different, and frequently incompatible networks. Sometimes machines called gateways are used to make the connection and provide the necessary translation, both in terms of Hardware



and Software. A collection of interconnected networks is called an internetwork or just Internet.

The word Internet is used in a generic sense. The Internet means a specified world wide Internet that is widely used to connect, government offices, universities, companies and of late, private individuals.

An internetwork is formed when distinct networks are connected together. Connecting a LAN and a WAN or connecting two LANs forms an internetworks. With exponential growth, the old informal way of running the Internet no longer works. In January 1992, the Internet Society was set up, to promote the use of the Internet and perhaps eventually take over managing it.

Traditionally, the Internet had four main applications, as follows:

1. **Email:** The ability to compose, send and receive electronic mail was found suitable to do this ARPANET and is enormously popular. Many people get dozens of messages a day and consider this to be their primary way of interacting with the outside world, far outdistancing the telephone and snail mail. Email programs are available on virtually every kind of computer these days.
2. **News:** Newsgroups are specialised forums in which users with a common interest can exchange messages. Thousands of newsgroups exist, on technical and nontechnical topics, including Computer Science, recreation and politics. Each newsgroup has its own etiquette, style, and customs, and woe to be anyone violating them.

3. **Remote Login:** Using Telnet, Remote login, or other programs, users anywhere on the Internet can log into any other machine they want to access or on which they have an account.
4. **File Transfer:** Using the FTP program, it is possible to copy files from one machine on the Internet to other. Vast numbers of articles, databases, and other information are available this way.

## 1.2 Need for Secure Communication

Computer networks are increasingly being used to exchange knowledge, access information, and process data in a distributed environment of modern technology. For sending e-mail, FTP etc. communication security did not require lot of attention. But these days Internet is being used by millions of users at thousands of sites around the world depending upon the global Internet as part of their daily working environment, it might seem that the Internet is completely a table production facility. Internet is also used for banking, shopping and business establishments etc. Thus, security has become an issue. Most companies have great amount of secret data on-line such as products development plans, marketing strategies, and financial analysis etc. Besides leaking in and leaking-out of information may get uncontrollable and damaging. In particular, viruses, worms, and other digital pests can breach security, destroy valuable data, and waste large amounts of administrators time to clean up the mess they leave.

Encryption is the standard means of rendering a communication private. The sender enciphers each message before transmitting it to the receiver. The authorised receiver knows the appropriate deciphering function to apply to the received message in order to obtain the original message. An unauthorised

receiver who receive the transmitted message receives only garbage which makes no sense to him since he does not know how to decrypt it. The large volume of personal and sensitise information currently held in computerised data banks of personal and transmitted over telephone lines make encryption increasingly.

### **1.3 Firewall**

For secure communication in a distributed environment on computer network to exchange data, information etc. The encryption/decryption hide the data or information between source and destination. To stop viruses and digital pests on computer networks, we required firewall method. In firewall design every incoming or outgoing data is checked at two levels. Firewall design has better mechanism to stop viruses and digital pests etc. It gives a better security in a distributed environment than RSA algorithm.

Firewall design forces the entering or leaving information to pass over a single drawbridge, where they could be inspected by the I/O packet filters. The Firewall includes two components.

1. Two routers that do packet filtering and
2. An application gateway

Simpler configurations also exist, but the advantage of firewall design is that every packet must transit two filters and an application gateway to go in or out. Except this route, there is no other route.

Each packet filter is a standard router equipped with some extra functionality. The extra functionality allows every incoming or outgoing packet to be checked. Packets meeting some criterion are forwarded normally. Those packet that fail the test are dropped.

Packet filters are typically driven by tables configured by the system administrator. These tables list sources and destinations that are acceptable, sources and destinations that are blocked, and default rules about what to do with packets coming from or going to other machines.

The second half of the firewall mechanism is the application gateway. The gateway operates at the application level. A mail gateway, for example, can be set up to examine each message going in or coming out. For each one it makes a decision to transmit or discard it based on header fields, message size, or even the content.

Firewalls are widely used in the distributed network like intelligence, military and banking service. The design of the inside and outside packet filter are different in design. The packet filter on the inside LAN checks outgoing packet and the packet filters on the outside LAN check incoming packets. Packets crossing the first hurdle go to the application gateway for further examination. The point of putting the two packet filters on different LANs is to ensure that no packet gets in or out without having to pass through application gateway: there is no other path around it.

#### **1.4 Problem Definition**

For secure communication in a distributed environment on computer networks to exchange information, knowledge, and data. The encryption/decryption method hide the data or information between source and destination. In RSA algorithm viruses and other digital pests can breach the security, destroy valuable data, and waste large amounts of administrator time to clean up the mess they leave. To stop viruses and digital pests on computer networks, we required Firewall method.

In firewall design every incoming or outgoing data are checked at two levels. Thus, firewall design can stop unauthorised data, viruses and digital pests etc. Firewall design gives an improved security in a distributed environment than RSA algorithm. However a high security LAN/organization may like to interact with the outside distributed environment for a given time duration and limiting the permission or access of facilities. Here we deal with the problem of time bound limited facility for high security organization using Firewall.

### **1.5 Organization of the dissertation**

This dissertation discusses security of computer network, and is organised as follows.

Chapter 1 deals with introduction, motivation, problem defination and concept of firewall

Chapter 2 concerns the background of secure communication, cryptography, RSA algorithm and Firewall.

Chapter 3 deals with the design issues of firewall, a study of format of an IP datagram., and design of a specific router.

Chapter 4 presents with the router alogrithm, its testing and performance.

Finally conclusion and discussion for the scope of future work. are presented in Chapter 5.

## **CHAPTER 2**

### **BACK GROUND AND MOTIVATION**

#### **2.1 Cryptography**

With the rapid growth of the Internet and electronic commerce, more and more people rely on computer networks to exchange data, knowledge and volumes of personal and sensitive information are electronically transmitted and stored every day. Various kinds of data transformation, or through open medium became meaningless to unauthorised person. But the original message can be recovered when data reach the destination. The encryption/decryption techniques, have been developed such that the data through open medium became meaningless to unauthorised person. But the encryption/decryption performed between the sender and the receiver requires to share a secret key between these two parties. Traditionally, private key were utilised to distribute the secret key.

##### **2.1.1 Secret Key**

Secret key encryption methods, such as the Data Encryption Standard (DSE), use the same key to both encrypt and decrypt data. The key must be known only to the parties who are authorised to encrypt and decrypt a particular message. On the other hand, public key, use the different keys to encrypt and decrypt data. The secret-key is kept confidential.

##### **2.1.2 Public-key cryptography**

Cryptography are typically classified as either public-key cryptography or secret key. A user wishing to exchange their public key encryption procedure, E, in a

public file. The user's corresponding decryption procedure,  $D$ , is kept confidential. Until Diffie and Hellman's public-key cryptography, all cryptographers simply took for granted that both the encryption and decryption keys had to be kept secret. If one think in terms of ciphers such as mono alphabetic substitution, it was obvious that encryption key, e.g.,  $xyz$  becomes  $ABC$ ; and the corresponding decryption key,  $ABC$  becomes  $xyz$ , each could be trivially derived from the other one.

Diffie and Hellman proposed an encryption algorithm,  $E$ , and a decryption algorithm,  $D$ , where  $E$  and  $D$  are chosen so that deriving  $D$  even given a complete decryption  $E$  would be effectively impossible.

These requirements differ strikingly from those of conventional cryptographic systems. There are three requirements.

(1)  $D(E(P))=P$

(2) It is very difficult to deduced  $D$  from  $E$

(3)  $E$  cannot be broken by a chosen plaintext attack.

The  $D(E(P))=P$ , says that if we apply  $D$  to an encrypted message,  $E(P)$ , we get the original plaintext message,  $P$ , back. The second requirement speaks for itself, that, it is very difficult to deduce  $D$  from  $E$ . The third requirement is needed, as, the unauthorised person may experiment with the algorithm to any extent. Under these three conditions, there is no way that  $E$  cannot be made public..

The public-key or encryption algorithm is then made public, hence the name public-key cryptography. Public-key might be done by putting it in a file that anyone want to read.

These requirements can be satisfied while establishing a secure communication between two parties,  $A$  and  $B$ , who have never had any previous contact. Both  $A$ 's

encryption key,  $E_A$  and B's encryption key,  $E_B$  are expected to publish their public encryption keys as soon as they became network users. A takes his first message,  $P$ , computes  $E_B(P)$ , and sends it to B. B decrypts it by applying his secret key  $D_B$  i.e., he computes  $D_B(E_B(P))=P$ . No one else can read the encrypted message,  $E_B(P)$ , because the encryption system is assumed strong and because it is too difficult to derive  $D_B$  from the public key  $E_B$

## **2.2 Key Distribution Problem**

Secret-key systems suffer from the key distribution problem. In order for a secure communication to occur, the key must first be securely sent to the other party. An unsecured channel such as a data network can not be used. Couriers or other secure means are typically used. Public-key systems do not suffer from this problem because of their use of two different keys. Messages are encrypted with a public-key and decrypted with a secret (private) key. No keys need to be distributed for a secure communication to occur.

There is no easy way for two total strangers who belong to different organisations to communicate in secure manner, except by physically getting together and agreeing upon a key on which they will work. It is as though you were not allowed to call someone on the telephone until the person had physically handed you this business card. So, A better method is needed.

### **2.2.1 The RSA Algorithm**

The Rivest-Shamir-Adleman (RSA) Algorithm is one of the most popular and secure public-key encryption methods. The algorithm capitalised on the fact that there is no efficient way to factor very large (100-200 digit) numbers.



Using an encryption key  $(e,n)$  :-

1. Represent the message as an integer between 0 and  $(n-1)$ . Large messages can be broken up into a number of blocks, each block than be represented by an integer in the same range.
2. Encrypt the message by raising it to be  $e^{\text{th}}$  power modulo  $n$ . The result is a ciphertext message  $C$ .
- 3 . To decrypt ciphertext message  $C$ , raise it to another power  $d$  modulo  $n$ .

The encryption key  $(e,n)$  is made public. The decryption key  $(d,n)$  is kept secret by the user.

To determine appropriate values for  $e,d$ , and  $n$ :-

1. Choose two very large (100+digit) prime numbers. Denotes these numbers as  $p$  and  $q$ .
2. Complete  $n=p*q$  and  $z=(p-1)*(q-1)$
3. Choose any large integer,  $d$ , such that  $\text{GCD}(d,z)=1$
4. Find  $e$  such that  $e*d=1(\text{mod } z)$

With all these parameters computed in advance now we can encrypt the message by dividing the plaintext into blocks, so that each plaintext message,  $p$ , falls in the interval  $0 < p < n$ . This can be done by grouping the plaintext into blocks of  $K$  bits, where  $K$  is the largest integer for which  $2K < n$  is true.

For encrypting a message,  $P$ , compute a ciphertext,  $C=P^e(\text{mod } n)$  and to decrypt a ciphertext,  $C$ , compute  $P=C^d(\text{mod } n)$ . It indicates that for all  $P$  in a specified range, the encryption and decryption functions are inverse. To perform the decryption you need  $d$  and  $n$ , and to perform the encryption, you need  $e$  and  $n$ . So, the secret-key consists of the pair  $(d,n)$  and the public key consists of  $(e,n)$ .

Cryptographic methods cannot be proven secure. Instead, the only test is to see if someone can figure out how to decrypt a message without having direct knowledge of the decryption key. The RSA method's security rests on the fact that it is extremely difficult to factor very large numbers. If 100 digit numbers are used for  $p$  and  $q$ , the resulting  $n$  will be approximately 200 digits. The fastest known factoring algorithm would take too long time for an unauthorized person to ever break the code. Other methods for determining  $d$  without factoring  $n$  and  $e$  are equally difficult.

As an example of RSA Algorithm, let us choose,  $p=3$ , and  $q=11$ , given  $n=33$  and  $z=20$ . A suitable value for  $d$  is 7, since 7 and 20 have no common factors. With these choices,  $e$  can be found by solving the equation  $7 \cdot e \equiv 1 \pmod{20}$ , which yields  $e=3$ . The ciphertext,  $C$ , for a plaintext message,  $P$ , is given by rule  $C = P^3 \pmod{33}$ . The ciphertext is decrypted by the receiver according to the rule  $P = C^7 \pmod{33}$ .

### **2.2.2 Authentication Protocol**

In a connection oriented systems, authentication can be done when a communication between two or more networks are established. In a traditional approach a user prove his identity typing a password. Besides of authentication expose the user to passive wiretapping, it is also require the authentication computer to maintain the list of password internally, which is itself a potential security problem. Using public-key cryptography, it is possible to perform the authentication in a secure way, without storing any password.

Authentication is the technique by which a process verifies that its communication partner is who it is supposed to be and not an imposter. Verifying the identity of a

remote process in the face of a malicious, active intruder is surprising difficult and required complex protocols based on cryptography. Some of the many authentication protocols that are used an insecure computer network.

There is some confusion between authorisation with authentication. Authentication deals with the question of whether or not you are actually communicating with a specific process. Authorisation is concerned with what that process is permitted to do.

### **2.2.3 Digital signature**

Digital signature is one of the major key in modern cryptography and computer security. In a digital data processing, for the receiver to keep as an evidence that an electronic document was indeed sent originally for a specific signer, the technique of digital signature is employed. So many digital signature schemes have been developed so far.

When you write a cheque and send it to another party, it is your signature that authenticates the document. With commerce moving slowly but surely, to the Internet, the time is not far away when cybershopping and its inevitable fallout, cybercash becomes the rule rather than the exception. An electronic or digital signature is an abbreviated and encrypted message, ensure that the accompanying document originated from the one signing it and that it was not tempered with route.

The sender compresses his text message in the form of a digest uses what is called a private key, know only to him, to encrypt the digest, thus turning it into a digital signature. He encrypts if second time using a public-key, know also to the recipient. Then he transmits it.

The recipient uses his own private key, known only to him, to decrypt the message. Then he uses the public-key, known to both of sender and receiver, to reconstitute the digest. His computer then decompresses the message from the digest.

The property,  $E(D(M))=M$ , of public-key cryptosystems allows a user to digitally "sign" a message they send. The digital signature provides proof that the message originated from the designated sender. In order to be effective, digital signature need to be both message-dependent as well as sender-dependent. This would prevent electronic "cutting and pasting" as well as modification of the original message by the recipient.

Suppose user A wanted to send a "digitally-signed" message, M, to user B:

1. User A applies their decryption procedure to M. This results in ciphertext C.
2. User A applies the encryption procedure of user B to C. This results in message S.
3. Ciphertext message S is sent over some communication channel.
4. Upon receipt, user B applies their decryption procedure to S. This results in ciphertext message C.
5. User B applies user A's encryption procedure to message C. This results in original message, M.

User B cannot alter the original message or use the signature with any other message. To do so would require user B to know how to decrypt a message using A's decryption procedure

### **2.3 Firewalls**

Firewalls control the flow of services between a site and the Internet outside, and the direction in which these services flow. A firewall's protection weakens as

the threat becomes more deeply embedded within the data it carries. By limiting outside access to secured servers sitting in a demilitarised zone outside the enclave, most firewall can also reduced the risk of outside clients attacking vulnerable servers with in the enclave. By integrating additional features, firewall can provide additional forms of protection. For instance, a firewall that supports strong user authentication can enforce additional restrictions on Internet access, controlling which users have access and what type of activities users may perform. Firewall builds a blockade between an internal network that is assumed to be secure and trusted, and another network, usually an external network, such as the internet, that is not be assumed to be secure and trusted. The general reasoning behind firewall usage is that without a firewall, a network's systems are more exposed to inherently insecure internet protocols and corresponding services, as well as attacks from hosts everywhere as the internet. In a firewall less environment, network security is solely a function of each host on the network. A firewall is to prevent unwanted and unauthorised communication into or out of the network, and to allow an organisation to enforce a network security policy on traffic flowing between its network and the internet.

A firewalls system usually consists of packet filtering routers and application gateway. A packet filtering router is a multiported IP router that applies a set of rules to each incoming IP packet, and decides whether it is to be forwarded or dropped. The packet filtering router filters IP packet, based an information that is available in packet headers, such as protocol numbers, source and destination IP address and port number, and some other IP options.

An application gateway is a server process running on a firewall system to perform a specific TCP/IP functions as a gateway on behalf of the network users.

An application gateway is, in essence, an application layer gateway; a gateway from one network to another for a specific network application. The user contacts a gateway using TCP/IP application, such as Telnet or FTP, and it asks the user for the name of the remote host to be accessed. When the user provides and responds a valid user identification and authentication information, the gateway contacts the remote host, and relays IP packets between the two communication points.

## CHAPTER 3

### DESIGN ISSUES

#### 3.1 Firewalls

Firewalls are computers or communication devices that restrict the flow of information between two or more networks. Firewalls are configured to protect one or more machines, inside a protected domain, erected between an organisational network and the Internet or any other outside network. A firewall is intended to prevent a malicious attacker who has control of computers outside the organisation's firewalls from gaining an access foothold inside. Used within an organisation, a firewall can limit the amount of damage from an intruder who does penetrate the organisation's internal network. An intruder may break into one set of machines but the firewall can protect others, perhaps containing more sensitive data or computations.

A firewall implements a security policy, the set of rules, that define what kind of interactions are allowed between the protected domain and the unprotected outside. Firewalls screen traffic flowing both into and out of the protected domain, rejecting any data that does not conform to the policy the firewalls are configured to implement. They can protect many kinds of attacks, including the following.

Firewalls can maintain audit logs that can indicate after the effect what kind of damage occurred in case the firewall did not block an attack. Firewalls implement either a simple blocking technology, proxy, or real service. Firewall inspects incoming traffic, decides which data items to admits perhaps admitting e-mail but blocking remote logins. We limited functionally emulator for a service that

protects an internal host providing the actual service. For example, a firewall could provide a proxy FTP service, reflecting certain commands to the inside for actual implementation by the protected hosts, and blocking others (which might do things such as try to being access to a protected directory). A firewall may instead provide a full service itself, such as reporting the status of only these known users that the protected organisation wants to have reported, perhaps shielding (because the firewall is not configured to know) the names of addresses of some internal user.

A more complex firewall structure uses two or more separated firewalls to implement a virtual private work. Suppose an organisation has two or more sites connected by an insecure medium such as the Internet. If the organisation uses encryption implemented by the two firewalls for all traffic between them, the encryption can effectively provide a network, protected against attacks that target confidentiality and integrity.

The standard firewall has two component: (i) Two routers that do packet filtering and (ii) an application gateway. Simpler configuration is also exist that the advantage of this design is that every packet must transit two filters and an application gateway to go in or out and cannot follow any other route.

Each packed filter is a standard router equipped with some extra functionally allows every incoming or outgoing packet to be inspected with packets that satisfy the already defined criteria are forwarded normally, others being rejected and dropped.

The packet filter on the insider LAN checks outgoing packets and the one on the outside LAN checks incoming packets. Packet crossing the first hurdle go to the



application gateway for further inspection. The justification of putting two packet filters on different LANs is to ensure that no packet get in or out without having to pass through the application gateway as there is no path around it.

Packet filters are typically driven by the tables configured by the authorised person (say system administrator). Those table list sources and destinations that are acceptable, sources and destination that are blocked and default, action about what to do with packets coming from or going to other machines.

Blocking outgoing packets is trickery because although most sites stick to the standard port naming conventions, they are not forced to do so. Also, for some important services, such as FTP, port numbers are assigned dramatically. In addition to this, blocking UDP packets are much more difficult than TCP packets as little information are available about what these UDP packets contains.

The second part of the firewall mechanism is the application gateway. Rather than just looking at raw packets, the gateway operates at the application level. A mail gateway, e.g., can be set up to examine each message going in or coming out. For each one, it make a decision to transmit or discard it based on header fields, message size or even the contain. More than one application gateways can be installed, each one for specific purposes, as the need arises. Combined with encryption and packet filtering, this arrangement offers a limited amount of security at the cost of same inconvenience.

### **3.2 Router**

A special purpose, dedicated computer that attached to two or more networks and forwards or discards the IP datagram from one machine to another machine packets from one to other. In particular, an IP router forwards IP datagrams among

the networks to which it connects. A router uses the destination address on a datagram to choose a next-hop to which it forwards the datagram.

When Internet connections becomes more complex, routers next to know about the topology of the Internet beyond the networks to which they connect. For a large Internet composed of many networks, the router task of making decisions about where to send packet becomes more complex.

The idea of a router seems simple, but it is important because it provides a way to interconnect networks, not just machine.

Routers must know how to route packets to their destination are large machines with enough primary or secondary memory to hold information about every primary or secondary memory to hold information about every machine in the Internet to which they attach. Routers used with TCP/IP internets are usually small computers. They often have little or no disk storage and limited main memories.

The trick to build-up a small Internet router lies in the following concept: "Routers use the destination network, not the destination host, when routing a packet," and "In a TCP/IP Internet, components called routers or gateways provide inter connections among physical networks".

If routing is based on networks, the amount of information that a router needs to keep is proportional to the number of the networks in the network, not the number of computers.

An Internet is composed of multiple physical networks interconnected by computers called routers. Each other has direct connections to two or more networks. Routers in TCP/IP Internet form a co-operative, interconnected

structure. Datagrams pass from router to router until they reach a router that can deliver the datagram directly.

In a highly secured network, such as intelligence, military there is a need for greater security than those already proposed by existing firewall design. Supposing that server is having to receive some data from another server in a network for a particular time period is over it does not want to remain accessible to that machine, then a time factor has to be introduced in the router so that no packet can get through after that specified time duration. It also works for multimedia application, where synchronised (timely) audio and video data have to be route and received over the communication line.

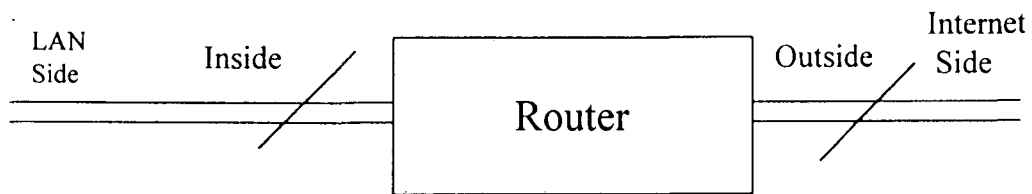


Figure 3.1 Firewall Router

### 3.3 IP Datagram Format

The different fields of the IP datagram are as shown in figure 3.2. Format of an Internet datagram, the basic unit of transfer in a TCP/IP internet.

0	4	8	16	19	24	31
VERS		HLEN	SERVICE TYPE		TOTAL LENGTH	
IDENTIFICATION			FLAGS	FRAGMENT OFFSET		
TIME TO LIVE		PROTOCOL		HEADER CHECKSUM		
SOURCE IP ADDRESS						
DESTINATION IP ADDRESS						
IP OPTIONS (IF ANY)					PADDING	
DATA						
...						

Figure 3.2 I.P. Datagram Format.

As the datagram processing occurs in software, the contents and format are not constrained by any hardware. The first 4 bits field in a datagram (VER) contains the versions of the IP protocol that was used to create the datagram. It is used to verify that the sender, receiver and any routers in between then agree on the format of the datagram. All IP software is required to check the version field before a datagram to ensure it matches the format the software expects.

The header length field (HLEN), also 4 bits, given the datagram header length measured in 32 bit words. The TOTAL LENGTH fields gives the length of the IP datagram measured in octets, including octets in the header and data. The size of the data area can be computed by subtracting the length of the header (HLEN) from the TOTAL LENGTH. As the TOTAL LENGTH field is 16 bits long, the maximum possible size of an IP datagram is  $2^{16}$  or 65535 octets.

The 8 bit SERVICE TYPE field specifies how the datagram should be handled and is broken down into five subfield as in the following figure 3.3. The five subfields that comprise the 8-bit SERVICE TYPE field.

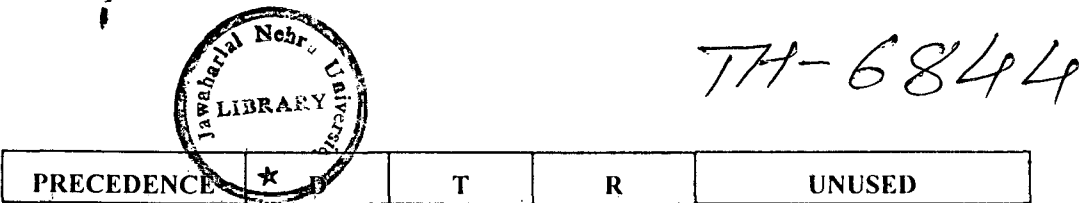


Figure 3.3 Service Type Field

Three PRECENCE bits specify datagram procedure with values ranging from 0 to 7, allowing sender to indicate the importance of each datagram. But P,T,R, specify the type of the transport the datagram derives. When set the bit D required low delay, the bit T requires high through put and the bit R requests high reliability.

Now each packet switching technology places a fixed upper bound on the amount of data that can be transferred in one physical frame and this upper bound is called Maximum Transfer Unit (MTU). The datagram is divided into MTUs and it is called fragmentation. The four fields in the datagram header, IDENTIFICATION, FLAGS, and FRAGMENT OFFSET, control fragmentation and reassembly of datagrams. Field IDENTIFICATION contains a unique integer that identifies the datagrams. Its primary purpose is to allow the destination to know which arriving fragments belong to which datagrams. For a fragment, field FRAGMENT OFFSET specifies the offset in the original datagram of data being carried in the fragment, measured in units of 8 octets, starting at offset zero. To reassemble the datagram, the destination must obtain all fragments, starting with the fragment that has offset zero through fragment with the highest offset.

The low order two bits of the 3 bit FLAGS field control fragmentation, the first of which specifies whether the datagram may be fragmented and the second one specifies whether the fragment contains data from the middle of the original data datagram or from the end.

Field Time To Live (TTL) specifies how long in seconds, the datagram is allowed to remain in the Internet system. Whenever a machine injects a datagram into the Internet, it sets a maximum time that the datagram should survive. Routers and hosts that process datagrams must decrement the TTL field as time passes and remove the datagram from the Internet when its time expires.

Field PROTOCOL is analogous to the type field in a network frame. The value in the PROTOCOL field specifies which high-level protocol was used to create the

message being carried in the DATA area of a datagram. In essence, the value of PROTOCOL specifies the format of the data area.

The HEADER CHECKSUM ensures integrity of header values. The IP Checksum is formed by treating the header as a sequence of 16 bit integers, adding them together using a one's complement arithmetic and then taking the one's complement of the result.

Fields SOURCE IP ADDRESS and DESTINATION IP ADDRESS contain the 32 bit IP address of the datagrams, senders and intended recipient. Although the datagram may be routed through many intermediate routers, the source and destination fields never change; they specify the IP address of the original source and ultimate destination.

The field labelled DATA shown the beginning of the data area of the datagram. It's length depends on what is being sent in the datagram.

The IP OPTIONS field following the destination address is not required in every datagram and they are included primarily for network testing and debugging.

The option code octet is divided into three fields as in figure 3.4.

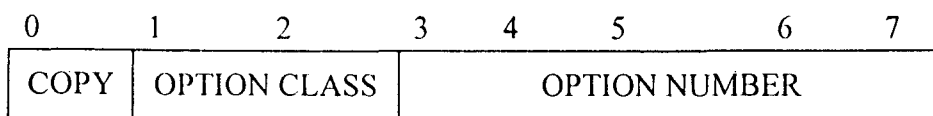


Figure 3.4. Option Code Octet

The fields consists of 1 bit copy flag, a 2-bit option class and the 5 bit option number. The copy flag controls how routers treat options during fragmentation. Where the COPY bit is set to 1, it specifies that the option should be copied into all fragments. The OPTION CLASS and ON bits specify the general class of the options and give specific option in that class, as shown in table 3.1.

Option Class	Option Number	Length	Description
0	0	-	End of option list. Used if options do not end at end of header (also see header padding field).8
0	1	-	No operation (used to align octets in a list of options).
0	2	11	Security and handling restrictions (for military applications).
0	3	var	Loose source routing. Used to route a datagram along a specified path.
0	7	var	Record route. Used to trace a route.
0	8	4	Stream identifies. Used to carry a SATNET stream identifier (obsolete).
0	9	var	Strict source routing. Used to route a datagram along a specified path.
2	4	var	Internet timestamp. Used to record timestamps along the route.

Table 3.1 IP Option table.

The field labelled PADDING, depends on the option selected. It pads zeros to make the length of datagram header to an exact multiplier of 32 bits.

### 3.4 Design of Specific IP Datagram Packet Filter

Firewalls, may be generally classified into two components: Packet filter and Application gateway. Packet filters block the transmission of packets based on the protocol, address and port identifier, while application gateway filter traffic using application specific requirement. More than one of these methods can sometimes be used in combination. In this work we concentrate on packet filtering. All routers have filtering capability and understanding packet filter required an understanding of the TCP/IP protocols that is also helpful in establishing other gateway protections, and also packet filtering in most easily accessible form of firewall protection in most environments.

At the time of designing a packet filter, one has to determine what filtering capabilities the router has and where to place the filtering, e.g., if the router has one or more LAN (“Inside”) ports and/or one or more WAN (“Outside”) ports, one may intend to put the filter outside to protect the router (fig). Most router, do in fact, allows to build packet filter and apply them port wise.

The proposed parameters:-

1. **Direction:** This parameter tells about whether the filter is being used for an in-bound or out-bound packet from the perspective of the router.
2. **IP Source:** An optional IP source address to identify the relevant hosts.
3. **IP Destination:** An optional IP destination address to which the packet is directed to.
4. **Protocol:** The protocol to which this specifications applies (like IP, TCP, and UDP, etc.)
5. **Source Port:** An optional source port address to identify the application to which this specification apply.
6. **Destination Port:** An optional destination port address, to which this specification applies e.g., let us assume that local networks FTP server and mail server have IP addresses 202.41.10.1 and 202.41.10.2 respectively. Assuming that these are class C address, the domain network identifier will be 202.41.10.0.



7. **Time Duration:** Time domain for which the service is being blocked or allowed to pass through the router.

A sample set of packet filtering action which may be embedded in assigned to the router's WAN (outside) interface are given as in table 3.2.

#	Direction	IP Source	IP Destination	Protocol	Source Port	Destination Port	Time duration		
							T <sub>1</sub>	to	T <sub>2</sub>
1	IN	*	*	TCP	23	*		*	
2	OUT	*	*	TCP	*	23		*	
3	IN	*	*	TCP	21	*		*	
4	OUT	*	*	TCP	*	21		*	
5	IN	*	*	TCP	20	1023		*	
6	OUT	*	*	TCP	*	20		*	
7	IN	*	202.41.10.1	TCP	*	21	10:30	to	22:30
8	OUT	202.41.10.1	*	TCP	21	*	10:30	to	22:30
9	IN	*	202.41.10.1	TCP	*	20	10:30	to	22:30
10	OUT	202.10.10.1	*	TCP	20	*	10:30	to	22:30

Table 3.2 IP Option Table

The first action applied to Telnet, an application that runs over TCP. Standard Telnet server use the port #23. The action #1 and #2 together allowed outgoing packets to a Telnet Server and it also allows incoming Telnet packets from a Telnet server, if they are part of an existing connection. The communicative effect of these actions is that any of the LAN uses can establish a Telnet connection to an Internet

Internet host but no one on the Internet can establish a Telnet connections to one of the LANs hosts.

The next 8 actions deal with the File Transfer Protocol (FTP), another protocols that works that works over TCP. When a user (client) opens an FTP session with a remote host (the server), the client set up an FTP control connection to the server on TCP port #21. When the client sends FTP commands to obtain directory listing or to indicate file transfer, e.g., the FTP server establishes on FTP, data connections with the client using TCP port #20. The FTP client is assigned same port number greater than 1023. The server set up a new FTP data connections for each directory get or put command and that the client is assigned a new TCP port number for each, FTP data connections. Thus actions #3 & #4 allow outgoing FTP control packets, but limits incoming FTP. Control packets to existing connections, while actions #5 and #6 allow incoming and outgoing FTP data packets. These four action together allow to establish an FTP connections with an FTP server on the Internet.

Action #7 to #10 also deal with FTP. They limit new incoming FTP. Control connection and now outgoing FTP data connections only between the LAN's FTP server (IP address 202.41.10.1)and a remote host for only between the time duration of 10:30 to 22:30 after which it automatically stands cancelled. These four actions than allow any Telnet host to established on FTP connection on to the designed FTP server but to no other host on the LAN for the specified time period.

IP datagram format initially does not include the fields direction, source port, Destination port and Time Duration. So, we are proposing to introduce these fields by breaking the IP options field so that any incoming and outgoing packet can be checked in the router on the basis of these fields for its validity.

# CHAPTER 4

## ROUTER ALGORITHMS AND PERFORMANCE

In this chapter we present the proposed algorithm based on the design specification discussed in detail in section 3.4. The main components of the proposed router to accommodate time bound permission of a specified network. Various steps of the algorithm are hand tested as presented in section 4.2 followed by the observations based on the tests carried out.

### 4.1 ALGORITHMS

-Call Socket System Call to Create Socket

/\*Socket System call create one end point of the communication \*/

-if (Socket fd<0) then

error socket cannot be created

/\*socketfd is the descriptor returned by socket system call \*/

-write specified number of null bytes to destination

-call bind system call to bind the socket

/\* Address of the communications \*/

-Call function Sent\_to\_Receiver .

#### 4.1.1 Algorithm for Send to Receiver function.

while (1) do /\* infinite loop\*/

{ read destination address;

read messages; /\* message to be send \*/

convert destination address to 32 bit integer ;

### Call **Check\_rout** function

```
/* This function takes all the fields define in routing table and checks that route exists or not */
```

If route exist then

```
{  
  
    print "route exists"  
  
    call send to system call to send the message;  
  
    call receive from system call to receive the message;  
  
    /* Both way communication */  
  
    print the received message;  
  
}  
  
else  
  
{  
  
    print " route does not exists"  
  
    break the while loop  
  
}  
  
}
```

#### **4.1.2 Algorithm for check route function :-**

Call **Create\_Table** function to create link list

```
/* Using Create_Table function it gets the starting pointer to the link list representing routing Table */
```

while not null do

```

{
if (a_direction != Direction) /* a_direction is arrival sample direction */
    return false;
else if (d_add != D_IP) /* d_add is arrival sample IP destination address */
    return false;
else if (s_add != S_IP) /* s_add is arrival sample IP source address */
    return false;
else if (a_protocol != Protocol) /* a_protocol is arrival sample protocol */
    return false;
else if (as_port != S-port) /* as_port is arrival sample source port number */
    return false;
else if (ad_port != D-port) /* ad_port is arrival sample destination port number */
    return false;
else if (T3 > T2 or T3 < T1) /* T3 is sample arrival time at router */
    return false;
else
    return true;
}

```

#### 4.1.3 Algorithm for create Table function

Structure used for creating list is as follows

type def. unsigned short u-short.

struct **route\_table**

```
{
```

```

u-short    direction ;

char       S_IP [30] ;

char       D_IP [30] ;

char       Protocol [10] ;

u-short    S_Port [5] ;

u-short    D_Port [5] ;

char       T_Duration [20] ;

};

```

```

n ← 10

```

```

Start ← Starting pointer of link list

```

```

While (n ≤ 10) do

```

```

{

temp ← melloc ( ) ;

temp.direction ← IN or Out ;

temp.S_IP ← Source IP address ;

temp.D_IP ← Destination IP address ;

temp.Protocol ← "TCP" ;

temp.S_Port ← Source port number ;

temp.D_Port ← Destination port number ;

temp.T_Duration ← Time duration ;

temp ← start ;

n ← n + 1;

}

```

## 4.2. Testing

The program is tested by considering the local FTP server with IP addresses 202.41.10.1 & 202.41.10.2. The IP datagram samples pass through the routing table and checked by the router algorithm.

### 4.2.1 Test 1

#	Direction	IP Source	IP Destination	Protocol	Source Port	Destination Port	Arrival Time at router $T_3$
#	IN	*	202.41.10.1	TCP	*	21	10.30

Table 4.1 IP packet Sample 1

All the fields in this incoming sample packet will check in routing table according to the algorithm already defined in section 4.1.2 in which it satisfy all the fields direction, IP source address, IP destination address, protocol, source port number, destination port number and time duration ( $T_3 < T_2$  and  $T_3 > T_1$ ). So, it will return a true value according to the algorithm defined in section 4.1.1. So packet will be allowed to pass and “route exist” message is displayed along with the message.

### 4.2.2. Test 2

#	Direction	IP Source	IP Destination	Protocol	Source Port	Destination Port	Arrival Time at router $T_3$
#	OUT	202.41.10.1	*	TCP	21	*	10.30

Table 4.2 IP packet Sample 2



All the fields in this incoming sample packet will check in routing table according to the algorithm already defined in section 4.1.2 in which it satisfy all the fields direction, IP source address, IP destination address, protocol, source port number, destination port number and time duration ( $T_3 < T_2$  and  $T_3 > T_1$ ). So, it will return a true value according to the algorithm defined in section 4.1.1. So packet will be allowed to pass and “route exist” message is displayed along with the message.

#### 4.2.3 Test 3

#	Direction	IP Source	IP Destination	Protocol	Source Port	Destination Port	Arrival Time at router $T_3$
#	IN	*	202.41.10.1	TCP	*	80	10.30

Table 4.3 IP packet Sample 3

All the fields in this incoming sample packet check in routing table according to the algorithm already defined in section 4.1.2 in which it satisfy the fields, direction, IP source address. IP destination address, protocol & source port number but in routing table destination port number 21 is for TCP facilities and in the sample packet destination port no. 80 is for HTTP services, it will return a false value to the algorithm defined in section. 4.1.1. so packet will be discarded and “route does not exist” message will be displayed.

#### 4.2.4 Test 4

#	Direction	IP Source	IP Destination	Protocol	Source Port	Destination Port	Arrival Time at router $T_3$
#	IN	*	202.41.10.1	TCP	*	21	9.30

Table 4.4 IP packet Sample 4

All the fields in this incoming sample packet check is the routing table according to the algorithm already defined as section 4.1.2. In which it satisfy the fields, direction, IP source address, IP destination address, protocol, source port number & destination port number, but the time duration ( $T_3 > T_2$  or  $T_3 < T_1$ ) will return a false value to the algorithm defined in section 4.1.1. So, packet will be discarded and “route does not exist” message will be displayed.

### **4.3 Algorithm Evaluation :-**

Firewalls are designed to protect one or more machines inside or outside a protected domains. Firewall is erected between an organizational network and the Internet or any other outside network.

#### **Unauthorized Access:-**

An unauthorized user outside the protected domains seeks access to information, knowledge, data and services inside the domain, and internal user seeks access to unauthorized information, knowledge, data and services outside the domain. e.g., protecting companies, defence and intelligence confidential data.

#### **Session Stealing :-**

A session for one purpose is connected to another purpose, e.g., when an e-mail connected to another connection in progress is converted to one that opens a file transfer operation.

#### **Communication Security Services :-**

Firewall provides authentication, data confidentiality and integrity, as well as nonrepudiation services to communicate peers.

## CHAPTER 5

### CONCLUSIONS

In firewall method, the focus was on the various techniques that are being used to provide internet security. A firewall builds a blockade between an internal network that is assumed to be secure and trusted, and another network, usually an external network such as Internet.

One of the drawbacks of the proposed router design algorithm is that, if one wants to send a packet which violates the specification in atleast one of the fields of the already set router table then network administrator has to reset the router table again to let the packet pass through it. Also the proposed router algorithm is not implemented practically. So there is no actual results to verify the feasibility of such a router, due to shortage of time. All the results presented here are based on theoretical assumptions and computation.

Firewall can erected at the border gateways to the internet. As they are capable of selectively forwarding or dropping IP datagrams, firewall also restrict the connectivity of the Internet as whole.

In this firewall design we propose to block the IP datagram for a specific time domain in which datagram will be forwarded for a router. This method can be used in a multiple LAN system. In future it will help in multimedia to block the datagram or forward the datagram for a specific time domain.

## BIBLIOGRAPHY

- [1] Black Uyles,,: TCP/IP and Related protocols, McGraw-Hill, Inc.,  
Second Edition, International Editions, 1995.
- [2] Comer, D.E. and Stevens, D.L.,: Internetworking with TCP/IP, vol.  
1, PHI, New Delhi, 1997.
- [3] Comer, D.E. and Stevens, D.L.,: Internetworking with TCP/IP, vol.  
2, PHI, New Delhi, 1997.
- [4] Harn Lein and Huang David, "A protocol for Establishing Secure  
Communication chanals in a large Network," IEEE Transactions on  
knowledge and Data Engineering, vol. 6, no.1, Feb. 1994.
- [5] Oppliger R., "Internet Security: Firewalls and Bey," Communication  
of the ACM., vol.40, no. 5, May 1997.
- [6] Rivest, R.L., Shamir A., and Adleman L., "A method for obtaining  
digital signatures and public-key cryptosystems," Commun. ACM.,  
vol. 21, pp 120-126, Feb. 1978.
- [7] Stevens, W.R.: Unix Networking Programming, PHI, New Delhi,  
1994.
- [8] Tanenbaum, A.S.: Computer Networks, PHI, New Delhi, 1997.
- [9] Yen, S.M., Laih C.S., "Improved Digital Signature Algorithm," IEEE  
Transactions on Computers, vol. 44, no. 5, May 1995.
- [10] Yen, S.M., Laih C.S., "Improved Digital Signature Suitable for Batch  
Varification," IEEE Transactions on Computers, vol. 44, no. 7, May 1995.