CS8792

CRYPTOGRAPHY AND NETWORK SECURITY

UNIT 5 NOTES

UNIT V SECURITY PRACTICE AND SYSTEM SECURITY

Electronic Mail security – PGP, S/MIME – IP security – Web Security – SYSTEM SECURITY: Intruders – Malicious software – viruses – Firewalls.

5.1 ELECTRONIC MAIL SECURITY

PRETTY GOOD PRIVACY (PGP)

PGP provides the confidentiality and authentication service that can be used for electronic mail and file storage applications.

The steps involved in PGP are

 Select the best available cryptographic algorithms as building blocks.

 Integrate these algorithms into a general purpose application that is independent of operating system and processor and that is based on a small set of easy-touse commands.

 Make the package and its documentation, including the source code, freely available via the internet, bulletin boards and commercial networks.

 Enter into an agreement with a company to provide a fully compatible, low cost commercial version of PGP.

PGP has grown explosively and is now widely used.

A number of reasons can be cited for this growth.

 It is available free worldwide in versions that run on a variety of platform.

 It is based on algorithms that have survived extensive public review and are considered extremely secure.

e.g., RSA, DSS and Diffie Hellman for public key encryption

 It has a wide range of applicability.

 It was not developed by, nor it is controlled by, any governmental or standards organization.

Operational description

The actual operation of PGP consists of five services:

1. Authentication
2. Confidentiality
3. Compression
4. E-mail compatibility
5. Segmentation.

1. Authentication

The sequence for authentication is as follows:

 The sender creates the message

 SHA-1 is used to generate a 160-bit hash code of the message

 The hash code is encrypted with RSA using the sender‟s private key and the result is pretended to the message

The receiver uses RSA with the sender‟s public key to decrypt and recover the hash code.

 The receiver generates a new hash code for the message and compares it with the decrypted hash code. If the two match, the message is accepted as authentic.

2. Confidentiality

Confidentiality is provided by encrypting messages to be transmitted or to be stored locally as files. In both cases, the conventional encryption algorithm CAST-128 may be used. The 64-bit cipher feedback (CFB) mode is used. In PGP, each conventional key is used only once. That is, a new key is generated as a random 128-bit number for each message. Thus although this is referred to as a session key, it is in reality a onetime key. To protect the key, it is encrypted with the receiver‟s public key.

The sequence for confidentiality is as follows:
 The sender generates a message and a random 128-bit number to be used as a session key for this message only.
 The message is encrypted using CAST-128 with the session key.
 The session key is encrypted with RSA, using the receiver‟s public key and is prepended to the message.
 The receiver uses RSA with its private key to decrypt and recover the session key.
 The session key is used to decrypt the message.

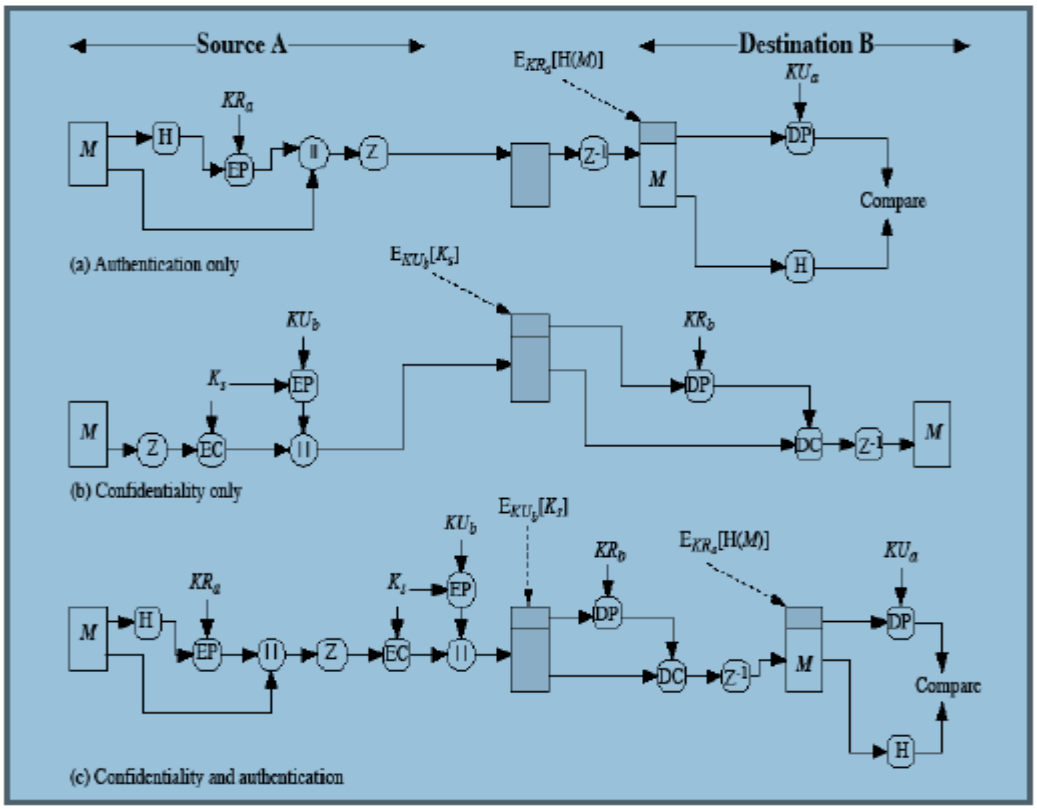
Confidentiality and authentication

Here both services may be used for the same message. First, a signature is generated for the plaintext message and prepended to the message. Then the plaintext plus the signature is encrypted using CAST-128 and the session key is encrypted using RSA.

3. Compression

As a default, PGP compresses the message after applying the signature but before encryption. This has the benefit of saving space for both e-mail transmission and for file storage. The signature is generated before compression for two reasons

- It is preferable to sign an uncompressed message so that one can store only the uncompressed message together with the signature for future verification. If one signed a compressed document, then it would be necessary either to store a compressed version of the message for later verification or to recompress the message when verification is required.
- Even if one were willing to generate dynamically a recompressed message from verification, PGP‟s compression algorithm presents a difficulty. The algorithm is not deterministic; various implementations of the algorithm achieve different tradeoffs in running speed versus compression ratio and as a result, produce different compression forms.
- Message encryption is applied after compression to strengthen cryptographic security. Because the compressed message has less redundancy than the original plaintext, cryptanalysis is more difficult. The compression algorithm used is ZIP



PGP Cryptographic functions

4. E-mail compatibility

Many electronic mail systems only permit the use of blocks consisting of ASCII texts. To accommodate this restriction, PGP provides the service of converting the raw 8-bit binary stream to a stream of printable ASCII characters. The scheme used for this purpose is radix-64 conversion. Each group of three octets of binary data is mapped into four ASCII characters.

5. Segmentation and reassembly

E-mail facilities often are restricted to a maximum length. E.g., many of the facilities accessible through the internet impose a maximum length of 50,000 octets. Any message longer than that must be broken up into smaller segments, each of which is mailed separately. To accommodate this restriction, PGP automatically subdivides a message that is too large into segments that are small enough to send via e-mail. The segmentation is done after all the other processing, including the radix-64 conversion. At the receiving end, PGP must strip off all e-mail headers and reassemble the entire original block before performing the other steps.

Cryptographic keys and key rings

Three separate requirements can be identified with respect to these keys:

- A means of generating unpredictable session keys is needed.
- It must allow a user to have multiple public key/private key pairs.
- Each PGP entity must maintain a file of its own public/private key pairs as well as a file of public keys of correspondents.

a. Session key generation

Each session key is associated with a single message and is used only for the purpose of encryption and decryption of that message. Random 128-bit numbers are generated using CAST-128 itself.

The input to the random number generator consists of a 128-bit key and two 64-bit blocks that are treated as plaintext to be encrypted. Using cipher feedback mode, the CAST-128 produces two 64-bit cipher text blocks, which are concatenated to form the 128-bit session key. The plaintext input to CAST-128 is itself derived from a stream of 128-bit randomized numbers.These numbers are based on the keystroke input from the user.

b. Key identifiers

If multiple public/private key pair are used, then how does the recipient know which of the public keys was used to encrypt the session key? One simple solution would be to transmit the public key with the message but, it is unnecessary wasteful of space. Another solution would be to associate an identifier with each public key that is unique at least within each user. The solution adopted by PGP is to assign a key ID to each public key that is, with very high probability, unique within a user ID. The key ID associated with each public key consists of its least significant 64 bits. i.e., the key ID of public key KUa is (KUa mod 264).

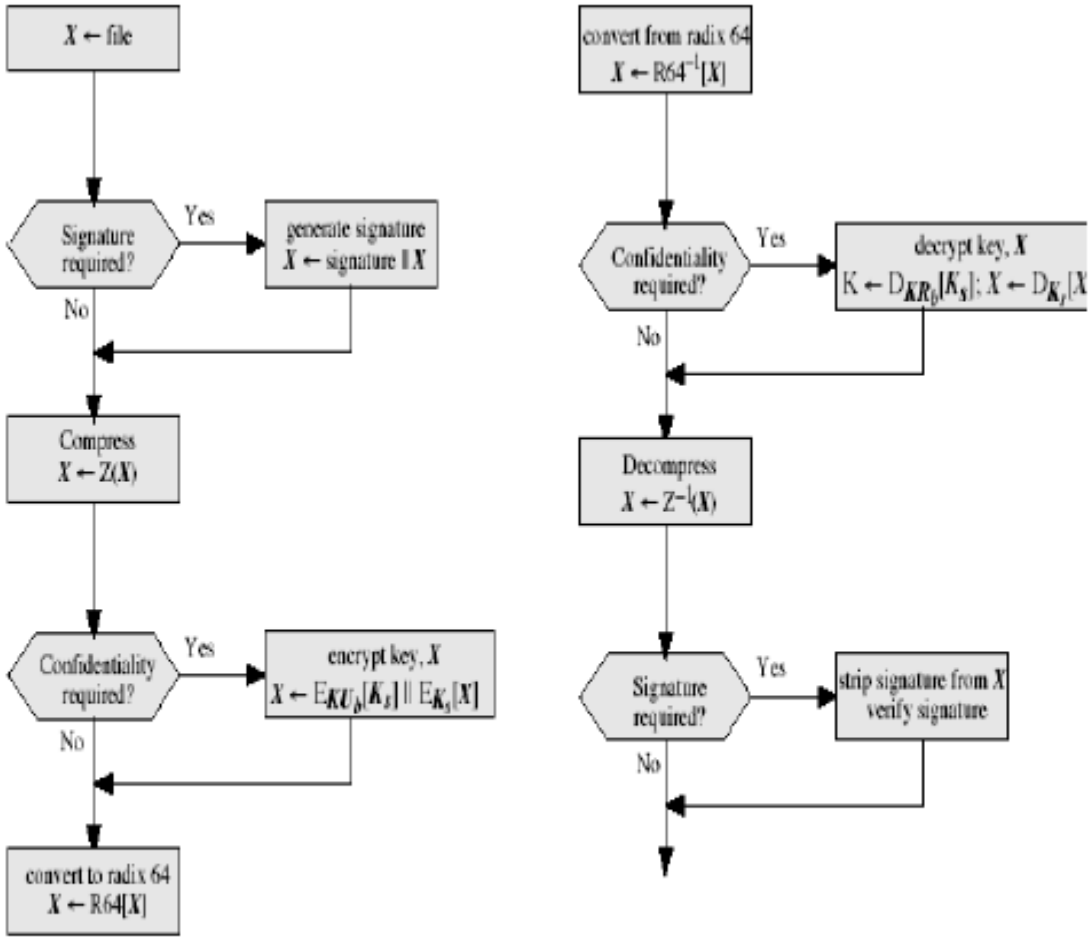
A message consists of three components.

- **Message component –** includes actual data to be transmitted, as well as the filename and a timestamp that specifies the time of creation
- **Session key component** – includes session key and the identifier of the recipient public key.

- **Signature component** – includes the following
- T**imestamp** – time at which the signature was made.
- **Message digest** – hash code.
- **Two octets of message digest** – to enable the recipient to determine if the correct public key was used to decrypt the message.
- **Key ID of sender’ s public key** – identifies the public key

Notation:

- EkUb= encryption with user B‟s Public key
- EKRa= encryption with user A‟s private key
- EKs = encryption with session key
- ZIP = Zip compression function
- R64 = Radix- 64 conversion function



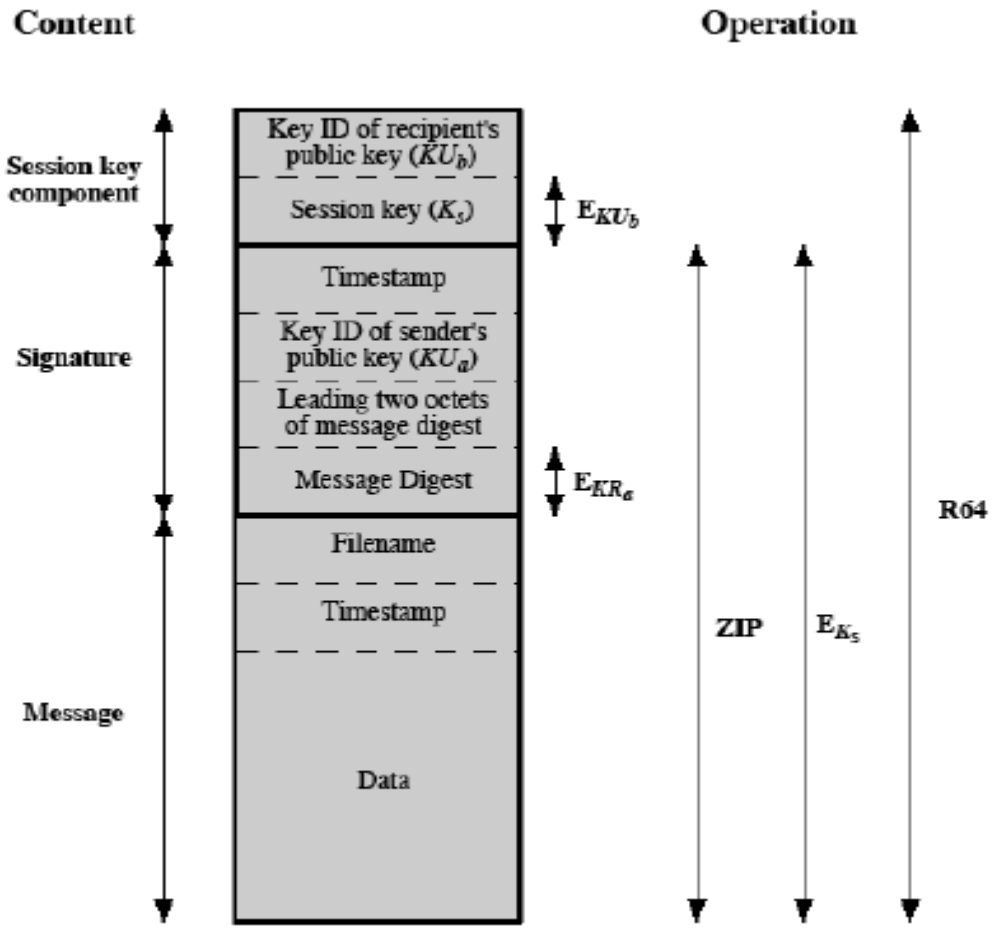
(a) Generic Transmission Diagram (from A)

(b) Generic Reception Diagram (to B)

Transmission and Reception of PGP message

PGP provides a pair of data structures at each node, one to store the public/private key pair owned by that node and one to store the public keys of the other users known at that node. These data structures are referred to as private key ring and public key ring. The general structures of the private and public key rings are shown below:

Timestamp - the date/time when this entry was made.
Key ID - the least significant bits of the public key.
Public key - public key portion of the pair.
Private Key - private key portion of the pair.
User ID - the owner of the key
Key legitimacy field – indicates the extent to which PGP will trust that this is a valid public key for this user.



General Format of PGP message (From A to B)

Signature trust field − indicates the degree to which this PGP user trusts the signer to certify public key.
Owner trust field - indicates the degree to which this public key is trusted to sign other public key certificates.

PGP message generation

First consider message transmission and assume that the
message is to be both signed and encrypted. The sending PGP entity performs the following steps

1. Signing the message

- PGP retrieves the sender‟s private key from the private key ring using user ID as an index. If user ID was not provided, the first private key from the ring is retrieved.
- PGP prompts the user for the passphrase (password) to recover the unencrypted private key.
- The signature component of the message is constructed.

2. Encrypting the message

- PGP generates a session key and encrypts the message.
- PGP retrieves the recipient‟s public key from the public key ring using user ID as index.

Private Key Ring

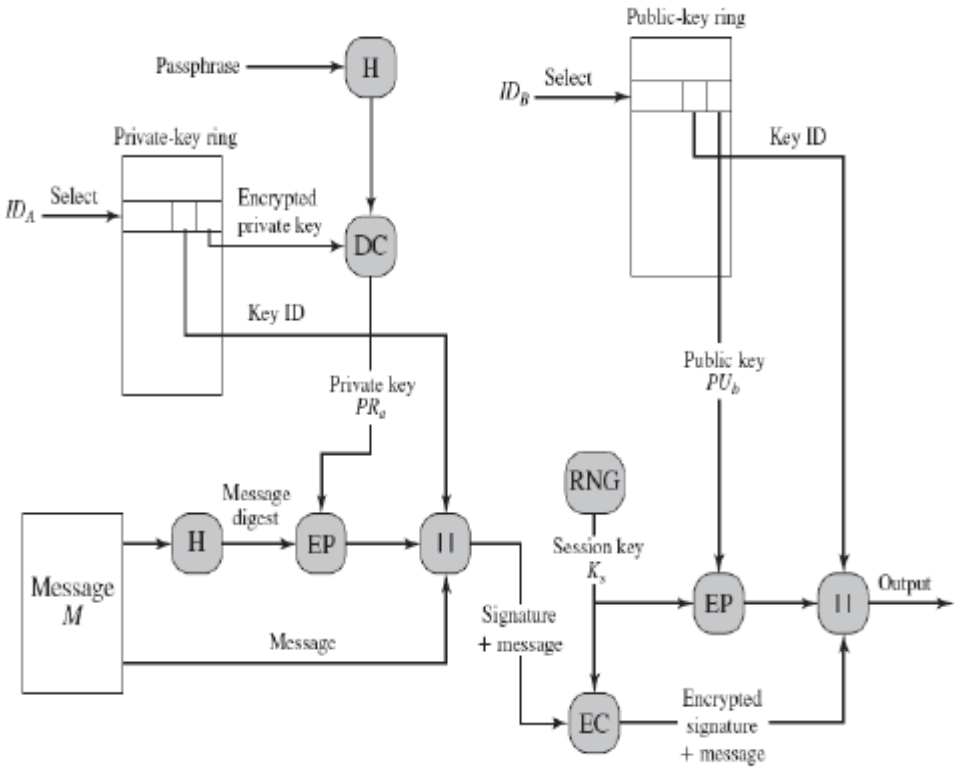
Timestamp	Key ID*	Public Key	Encrypted Private Key	User ID*
•	•	•	•	•
•	•	•	•	•
•	•	•	•	•
T_i	$PU_i \bmod 2^{64}$	PU_i	$E(H(P_i), PR_i)$	User i
•	•	•	•	•
•	•	•	•	•
•	•	•	•	•

Public Key Ring

Timestamp	Key ID*	Public Key	Owner Trust	User ID*	Key Legitimacy	Signature(s)	Signature Trust(s)
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
T_i	$PU_i \bmod 2^{64}$	PU_i	$trust\_flag_i$	User i	$trust\_flag_i$		
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•

* = field used to index table

General Structure public and private key



PGP Message Generation

The receiving PGP entity performs the following steps:

1. Decrypting the message

• PGP retrieves the receiver‟s private key from the private key ring, using the key ID field in the session key component of the message as an index.

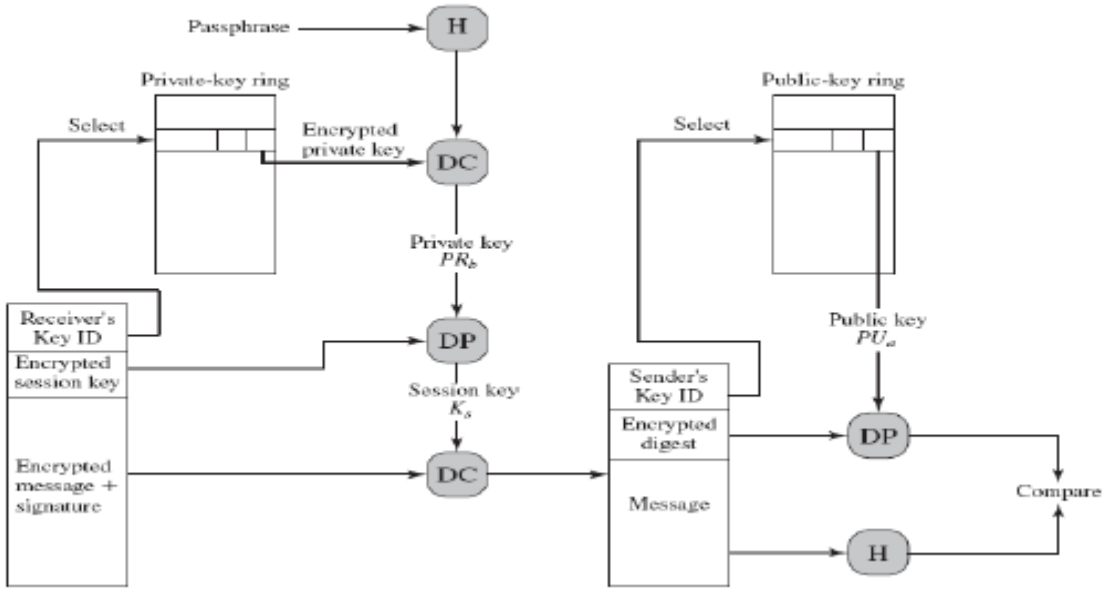
• PGP prompts the user for the passphrase (password) to recover the unencrypted private key.

• PGP then recovers the session key and decrypts the message.

2. Authenticating the message

• PGP retrieves the sender‟s public key from the public key ring, using the key ID field in the signature key component of the message as an index.

• PGP recovers the transmitted message digest. • PGP computes the message digest for the received message and compares it to the transmitted message digest to authenticate.



PGP Message Reception

5.2 S/MIME

S/MIME (Secure/Multipurpose Internet Mail Extension) is a security enhancement to the MIME Internet e-mail format standard, based on technology from RSA Data Security.

5.1.2.1 Multipurpose Internet Mail Extensions

MIME is an extension to the RFC 822 framework that is intended to address some of the problems and limitations of the use of SMTP (Simple Mail Transfer Protocol) or some other mail transfer protocol and RFC 822 for electronic mail.

Following are the limitations of SMTP/822 scheme:

1. SMTP cannot transmit executable files or other binary objects.
2. SMTP cannot transmit text data that includes national language characters because these are represented by 8-bit codes with values of 128 decimal or higher, and SMTP is limited to 7-bit ASCII.
3. SMTP servers may reject mail message over a certain size.
4. SMTP gateways that translate between ASCII and the character code EBCDIC do not use a consistent set of mappings, resulting in translation problems.
5. SMTP gateways to X.400 electronic mail networks cannot handle non textual data included in X.400 messages.
6. Some SMTP implementations do not adhere completely to the SMTP standards defined in RFC 821. Common problems include:

• Deletion, addition, or reordering of carriage return and linefeed

• Truncating or wrapping lines longer than 76 characters

• Removal of trailing white space (tab and space characters)

- Padding of lines in a message to the same length
- Conversion of tab characters into multiple space characters

MIME is intended to resolve these problems in a manner that is compatible with existing RFC 822 implementations. The specification is provided in RFCs 2045 through 2049.

OVERVIEW

The MIME specification includes the following elements:

1. **Five new message header** fields are defined, which may be included in an RFC 822 header. These fields provide information about the body of the message.
2. **A number of content formats** are defined, thus standardizing representations that support multimedia electronic mail.
3. **Transfer encodings** are defined that enable the conversion of any content format into a form that is protected from alteration by the mail system.

In this subsection, we introduce the five message header fields. The next two subsections deal with content formats and transfer encodings.

The five header fields defined in MIME are as follows:

- **MIME-Version**: Must have the parameter value 1.0. This field indicates that the message conforms to RFCs 2045 and 2046.
- **Content-Type**: Describes the data contained in the body with sufficient detail.
- **Content-Transfer-Encoding**: Indicates the type of transformation that has been used to represent the body of the message in a way that is acceptable for mail transport.
- **Content-ID:** Used to identify MIME entities uniquely in multiple contexts.
- **Content-Description**: A text description of the object with the body; this is useful when the object is not readable (e.g., audio data).

MIME Content Types

There are seven different major types of content and a total of 15 subtypes

MIME Content Types		
Type	**Subtype**	**Description**
Text	Plain	Unformatted text; may be ASCII or ISO 8859.
	Enriched	Provides greater format flexibility.
Multipart	Mixed	The different parts are independent but are to be transmitted together. They should be presented to the receiver in the order that they appear in the mail message.
	Parallel	Differs from Mixed only in that no order is defined for delivering the parts to the receiver.
	Alternative	The different parts are alternative versions of the same information. They are ordered in increasing faithfulness to the original, and the recipient's mail system should display the "best" version to the user.
	Digest	Similar to Mixed, but the default type/subtype of each part is message/rfc822.
Message	rfc822	The body is itself an encapsulated message that conforms to RFC 822.

	Partial	Used to allow fragmentation of large mail items, in a way that is transparent to the recipient.
	External-body	Contains a pointer to an object that exists elsewhere.
Image	jpeg	The image is in JPEG format, JFIF encoding.
	Gif	The image is in GIF format.
Video	mpeg	MPEG format.
Audio	Basic	Single-channel 8-bit ISDN mu-law encoding at a sample rate of 8 kHz.
Application	PostScript	Adobe Postscript.
	octet-stream	General binary data consisting of 8-bit bytes.

Multipurpose Internet Mail Extensions

Multipurpose Internet Mail Extension (MIME) is an extension to the RFC 5322 framework that is intended to address some of the problems and limitations of the use of Simple Mail Transfer Protocol (SMTP), defined in RFC 821, or some other mail transfer protocol and RFC 5322 for electronic mail. [PARZ06] lists the following limitations of the SMTP/5322 scheme.

1. SMTP cannot transmit executable files or other binary objects. A number of schemes are in use for converting binary files into a text form that can be used by SMTP mail systems, including the popular UNIX UUencode/ UUdecode scheme. However, none of these is a standard or even a de facto standard.
2. SMTP cannot transmit text data that includes national language characters, because these are represented by 8-bit codes with values of 128 decimal or higher, and SMTP is limited to 7-bit ASCII.
3. SMTP servers may reject mail message over a certain size.
4. SMTP gateways that translate between ASCII and the character code EBCDIC do not use a consistent set of mappings, resulting in translation problems.
5. SMTP gateways to X.400 electronic mail networks cannot handle nontextual data included in X.400 messages.
6. Some SMTP implementations do not adhere completely to the SMTP standards defined in RFC 821. Common problems include:

• Deletion, addition, or reordering of carriage return and linefeed
• Truncating or wrapping lines longer than 76 characters
• Removal of trailing white space (tab and space characters)
• Padding of lines in a message to the same length
• Conversion of tab characters into multiple space characters

MIME is intended to resolve these problems in a manner that is compatible with existing RFC 5322 implementations. The specification is provided in RFCs 2045 through 2049.

Functions

• **Enveloped data:** This consists of encrypted content of any type and encryptedcontent encryption keys for one or more recipients.
• **Signed data:** A digital signature is formed by taking the message digest of the content to be signed and then encrypting that with the private key of the signer. The content plus signature are then encoded using base64 encoding. A signed data message can only be viewed by a recipient with S/MIME capability.
• **Clear-signed data:** As with signed data, a digital signature of the content is formed. However, in this case, only the digital signature is encoded using base64. As a result, recipients without S/MIME capability can view the message content, although they cannot verify the signature.
• **Signed and enveloped data:** Signed-only and encrypted-only entities may be nested, so that encrypted data may be signed and signed data or clear-signed data may be encrypted.

Cryptographic Algorithms

• **MUST:** The definition is an absolute requirement of the specification. An implementation must include this feature or function to be in conformance with the specification.

• **SHOULD:** There may exist valid reasons in particular circumstances to ignore this feature or function, but it is recommended that an implementation include the feature or function.

Enhanced Security Services

• **Signed receipts:** A signed receipt may be requested in a SignedData object. Returning a signed receipt provides proof of delivery to the originator of a message and allows the originator to demonstrate to a third party that the recipient received the message. In essence, the recipient signs the entire original message plus the original (sender's) signature and appends the new signature to form a new S/MIME message.

• **Security labels:** A security label may be included in the authenticated attributes of a SignedData object. A security label is a set of security information regarding the sensitivity of the content that is protected by S/MIME encapsulation. The labels may be used for access control, by indicating which users are permitted access to an object. Other uses include priority (secret, confidential, restricted, and so on) or role based, describing which kind of people can see the information (e.g., patient's health-care team, medical billing agents, etc.).

• **Secure mailing lists:** When a user sends a message to multiple recipients, a certain amount of per-recipient processing is required, including the use of each recipient's public key. The user can be relieved of this work by employing the services of an S/MIME Mail List Agent (MLA). An MLA can take a single incoming message, perform the recipient-specific encryption for each recipient, and forward the message. The originator of a message need only send the message to the MLA with encryption performed using the MLA's public key.

5.3 IP SECURITY

OVERVIEW OF IPSEC

Applications of IPSec

IPSec provides the capability to secure communications across a LAN, across private and public WANs, and across the Internet. Examples of its use include the following:

- Secure branch office connectivity over the Internet
- 2 Secure remote access over the Internet
- Establishing extranet and intranet connectivity with partners
- Enhancing electronic commerce security

Benefits of IPSec:

- When IPSec is implemented in a firewall or router, it provides strong security
- IPSec in a firewall is resistant to bypass if all traffic from the outside must use IP, and the firewall is the only means of entrance from the Internet into the organization.
- IPSec is below the transport layer (TCP, UDP) and so is transparent to applications. There is no need to change software on a user or server system when IPSec is implemented in the firewall or router.
- IPSec can be transparent to end users. There is no need to train users on security mechanisms
- IPSec can provide security for individual users if needed.

Routing Applications

IPSec can play a vital role in the routing architecture required for internet working.
The following are examples of the use of IPSec. IPSec can assure that

- A router advertisement (a new router advertises its presence) comes from an authorized router
- A neighbor advertisement (a router seeks to establish or maintain a neighbour relationship with a router in another routing domain) comes from an authorized router.
- A redirect message comes from the router to which the initial packet was sent.
- A routing update is not forged.

IP SECURITY ARCHITECTURE

Covers the general concepts, security requirements, definitions, and mechanisms defining IPSec technology.

Encapsulating Security Payload (ESP):

Covers the packet format and general issues related to the use of the ESP for packet encryption and, optionally, authentication.

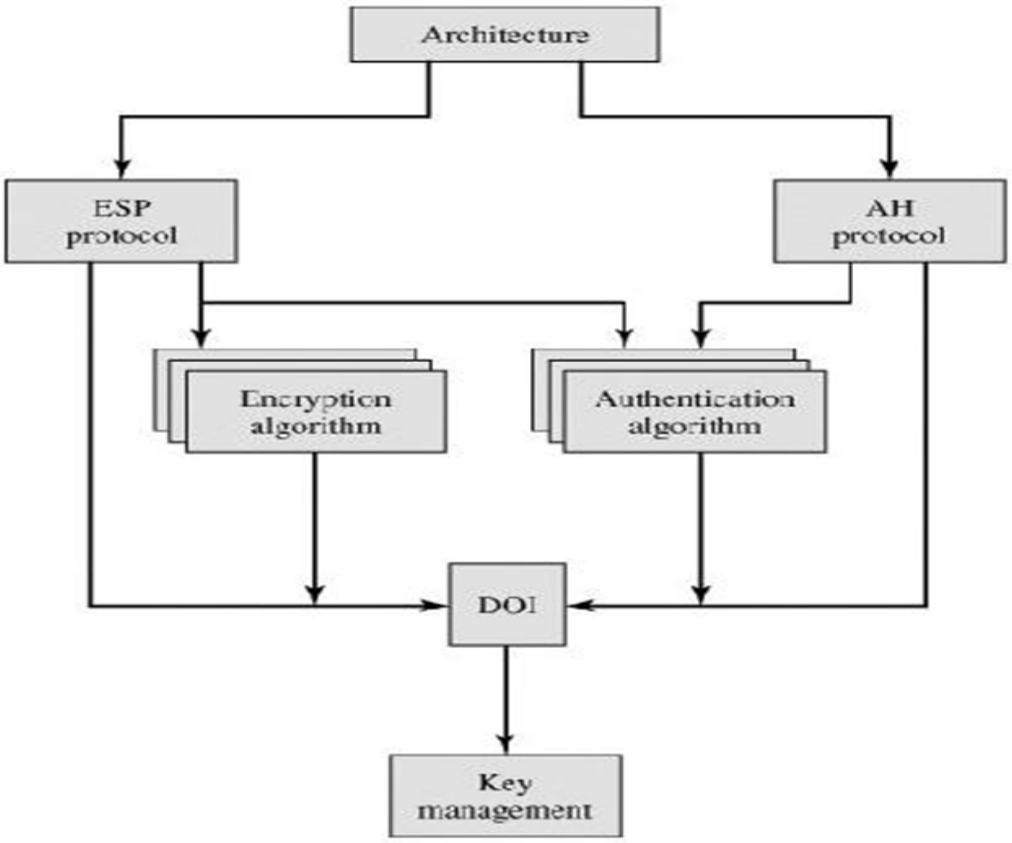
Authentication Header (AH):

Covers the packet format and general issues related to the use of AH for packet authentication.

Encryption Algorithm:

A set of documents that describe how various encryption algorithms are used for ESP.

Authentication Algorithm:
A set of documents that describe how various authentication algorithms are used for AH and for the authentication option of ESP.
Key Management:
Documents that describe key management schemes.
Domain of Interpretation (DOI):
Contains values needed for the other documents to relate to each other. These include identifiers for approved encryption and authentication algorithms, as well as operational parameters such as key lifetime



IP security Document overview

IPSec Services

IPSec provides security services at the IP layer by enabling a system to select required security protocols, determine the algorithm(s) to use for the service(s), and put in place any cryptographic keys required to provide the requested services.

Two protocols are used to provide security:
- An authentication protocol: Designated by the header of the protocol, Authentication Header (AH);
- Encryption/authentication protocol designated by the format of the packet for that protocol, Encapsulating Security Payload (ESP).

The services are

- Access control
- Connectionless integrity
- Data origin authentication
- Rejection of replayed packets (a form of partial sequence integrity)
- Confidentiality (encryption)
- Limited traffic flow confidentiality

Modes of Transfer

Both AH and ESP support two modes of use: transport and tunnel mode.

Transport Mode:

Transport mode provides protection primarily for upper-layer protocols. That is, transport mode protection extends to the payload of an IP packet.

Tunnel Mode:

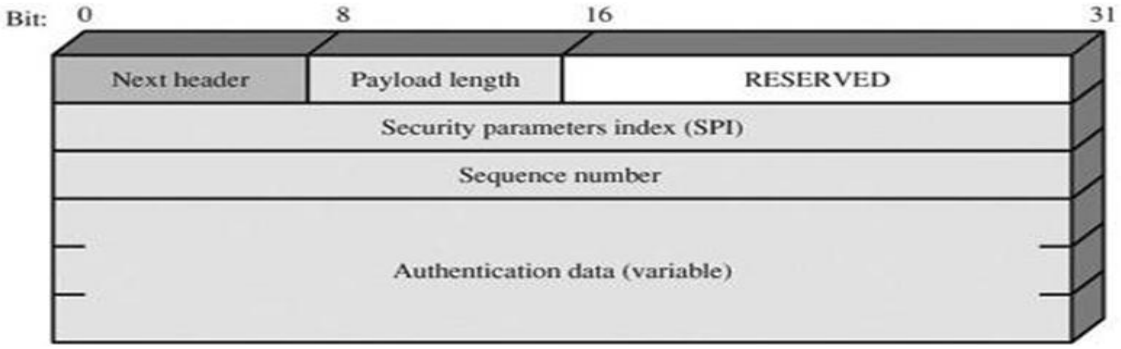
Tunnel mode provides protection to the entire IP packet. To achieve this, after the AH or ESP fields are added to the IP packet, the entire packet plus security fields is treated as the payload of new "outer" IP packet with a new outer IP header.

The entire original, or inner, packet travels through a "tunnel" from one point of an IP network to another; no routers along the way are able to examine the inner IP header. Because the original packet is encapsulated, the new, larger packet may have totally different source and destination addresses, adding to the security.

Authentication Header

The Authentication Header provides support for data integrity and authentication of IP packets. The Authentication Header consists of the following fields:

- **Next Header (8 bits):** Identifies the type of header immediately following this header.
- **Payload Length (8 bits):** Length of Authentication Header in 32-bit words, minus 2.
- **Reserved (16 bits):** For future use.
- **Security Parameters Index (32 bits):** Identifies a security association.
- **Sequence Number (32 bits):** A monotonically increasing counter value.
- **Authentication Data (variable):** A variable-length field (must be an integral number of 32-bit words) that contains the Integrity Check Value (ICV), or MAC



KEY MANAGEMENT

The key management portion of IPSec involves the determination and distribution of secret keys. Two types of key management:

Manual: A system administrator manually configures each system with its own keys and with the keys of other communicating systems. This is practical for small, relatively static environments.

Automated: An automated system enables the on-demand creation of keys for SAs and facilitates the use of keys in a large distributed system with an evolving configuration.

The default automated key management protocol for IPSec is referred to as ISAKMP/Oakley and consists of the following elements:

Oakley Key Determination Protocol: Oakley is a key exchange protocol based on the Diffie-Hellman algorithm but providing added security. Oakley is generic in that it does not dictate specific formats.

Internet Security Association and Key Management Protocol (ISAKMP):ISAKMP provides a framework for Internet key management and provides the specific protocol support, including formats, for negotiation of security attributes

5.4 WEB SECURITY

WEB SECURITY CONSIDERATIONS

The World Wide Web is fundamentally a client/server application running over the Internet and TCP/IP intranets.

Web Security Threats

A Comparison of Threats on the Web

	Threats	Consequences	Countermeasures
Integrity	Modification of user data Trojan horse browser Modification of memory Modification of message traffic in transit	Loss of information Compromise of machine Vulnerability to all other threats	Cryptographic checksums
Confidentiality	Eavesdropping on the Net Theft of info from server Theft of data from client Info about network configuration Info about which client talks to server	Loss of information Loss of privacy	Encryption, web proxies
Denial of Service	Killing of user threads Flooding machine with Bogus requests Filling up disk or memory Isolating machine by DNS attacks	Disruptive Annoying Prevent user from getting work done	Difficult to prevent
Authentication	Impersonation of legitimate users Data forgery	Misrepresentation of user Belief that false information is valid	Cryptographic techniques

Two types of attacks are:

Passive attacks include eavesdropping on network traffic between browser and server and gaining access to information on a Web site that is supposed to be restricted.

Active attacks include impersonating another user, altering messages in transit between client and server, and altering information on a Web site.

SECURE SOCKET LAYER AND TRANSPORT LAYER SECURITY

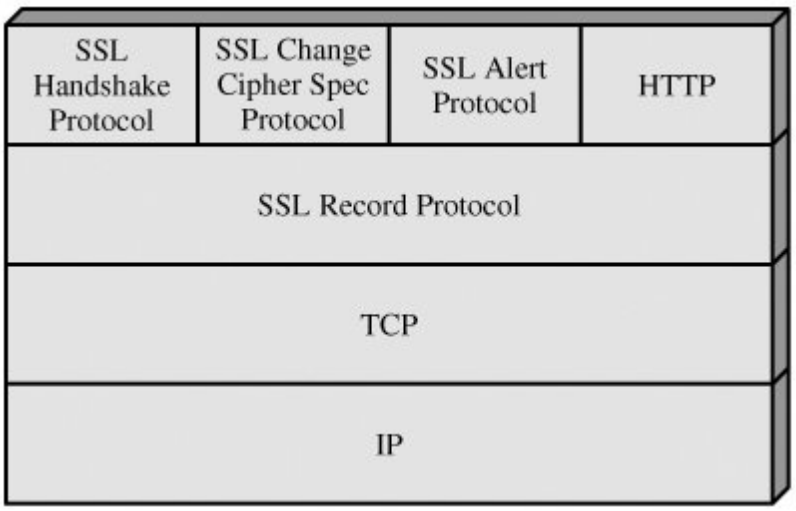
SSL Architecture

SSL is designed to make use of TCP to provide a reliable end-to-end secure service. The SSL Record Protocol provides basic security services to various higher-layer protocols. In particular, the Hypertext Transfer Protocol (HTTP), which provides the transfer service for Web client/server interaction, can operate on top of SSL. Three higher-layer protocols are defined as part of SSL: the Handshake Protocol, The Change Cipher Spec Protocol, and the Alert Protocol

Two important SSL concepts are the SSL session and the SSL connection, which are defined in the specification as follows:

Connection:

A connection is a transport (in the OSI layering model definition) that provides a suitable type of service. For SSL, such connections are peer-to-peer relationships. The connections are transient. Every connection is associated with one session.



SSL PROTOCOL STACK

Session:
An SSL session is an association between a client and a server. Sessions are created by the Handshake Protocol. Sessions define a set of cryptographic security parameters, which can be shared among multiple connections. Sessions are used to avoid the expensive negotiation of new security parameters for each connection

A session state is defined by the following parameters

- Session identifier
- Peer certificate
- Compression method
- Cipher spec
- Master secret
- Is resumable

A connection state is defined by the following parameters:

- Server and client random
- Server write MAC secret
- Client write MAC secret
- Server write key
- Client write key.
- Initialization vectors
- Sequence numbers

SSL Record Protocol

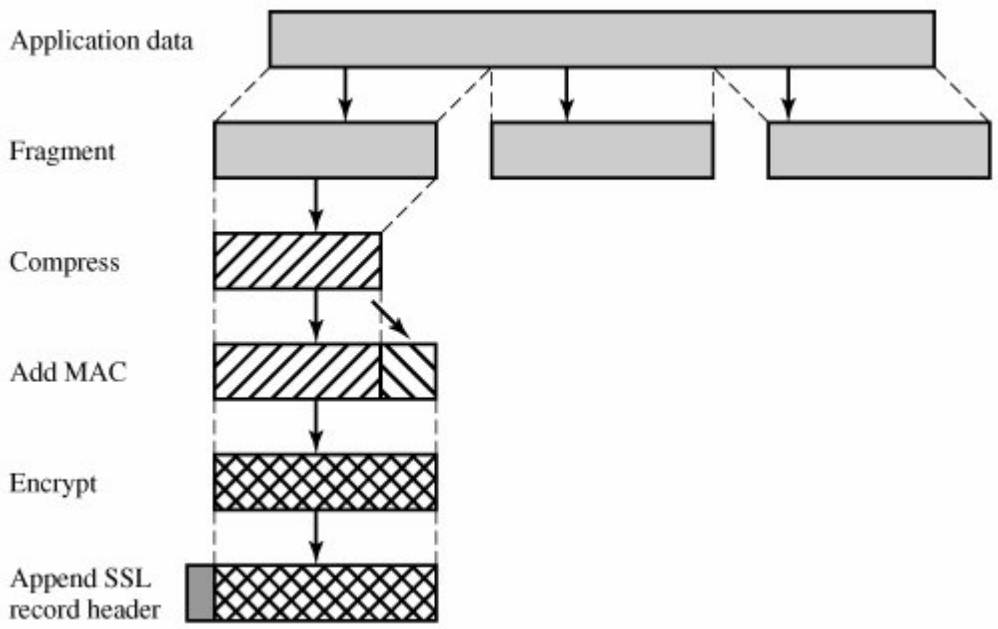
The SSL Record Protocol provides two services for SSL connections:
Confidentiality: The Handshake Protocol defines a shared secret key that is used for conventional encryption of SSL payloads.
Message Integrity: The Handshake Protocol also defines a shared secret key that is used to form a message authentication code (MAC).

The diagram indicates the overall operation of the SSL Record Protocol. The Record Protocol takes an application message to be transmitted, fragments the data into manageable blocks, optionally compresses the data, applies a MAC, encrypts, adds a header, and transmits the resulting unit in a TCP segment. Received data are decrypted, verified, decompressed, and reassembled and then delivered to higher-level users.
The first step is fragmentation. Each upper-layer message is fragmented into blocks of 2^{14} bytes (16384 bytes) or less. Next, compression is optionally applied. Compression must be lossless and may not increase the

content length by more than 1024 bytes. In SSLv3 (as well as the current version of TLS), no compression algorithm is specified, so the default compressionalgorithm is null.

The next step in processing is to compute a **message authentication code** over the compressed data.
The final step of SSL Record Protocol processing is to prepend a header, consisting ofthe following fields:

- **Content Type (8 bits):** The higher layer protocol used to process the enclosed fragment.
- **Major Version (8 bits):** Indicates major version of SSL in use. For SSLv3, the value is 3.
- **Minor Version (8 bits):** Indicates minor version in use. For SSLv3, the value is 0.
- **Compressed Length (16 bits):** The length in bytes of the plaintext fragment (orcompressed fragment if compression is used). The maximum value is 2_{14} + 2048.



SSL Record Protocol Operation

Change Cipher Spec Protocol

This protocol consists of a single message which consists of a single byte with the value

1. **Alert Protocol**

The Alert Protocol is used to convey SSL-related alerts to the peer entity.

2. **Handshake Protocol**

This protocol allows the server and client to authenticate each other and to negotiate an encryption and MAC algorithm and cryptographic keys to be used to protect data sent in an SSL record. The Handshake Protocol is used before any application data is transmitted. The Handshake Protocol consists of a series of messages exchanged by client and server.

SECURE ELECTRONIC TRANSACTION

SET is an open encryption and security specification designed to protect credit card transactions on the Internet. SET is not itself a payment system. Rather it is a set of security protocols and formats that enables users to employ the existing credit card payment infrastructure on an open network, such as the Internet, in a secure fashion.

SET provides three services:

- Provides a secure communications channel among all parties involved in a transaction
- Provides trust by the use of X.509v3 digital certificates
- Ensures privacy because the information is only available to parties in a transaction when and where necessary

Key Features of SET

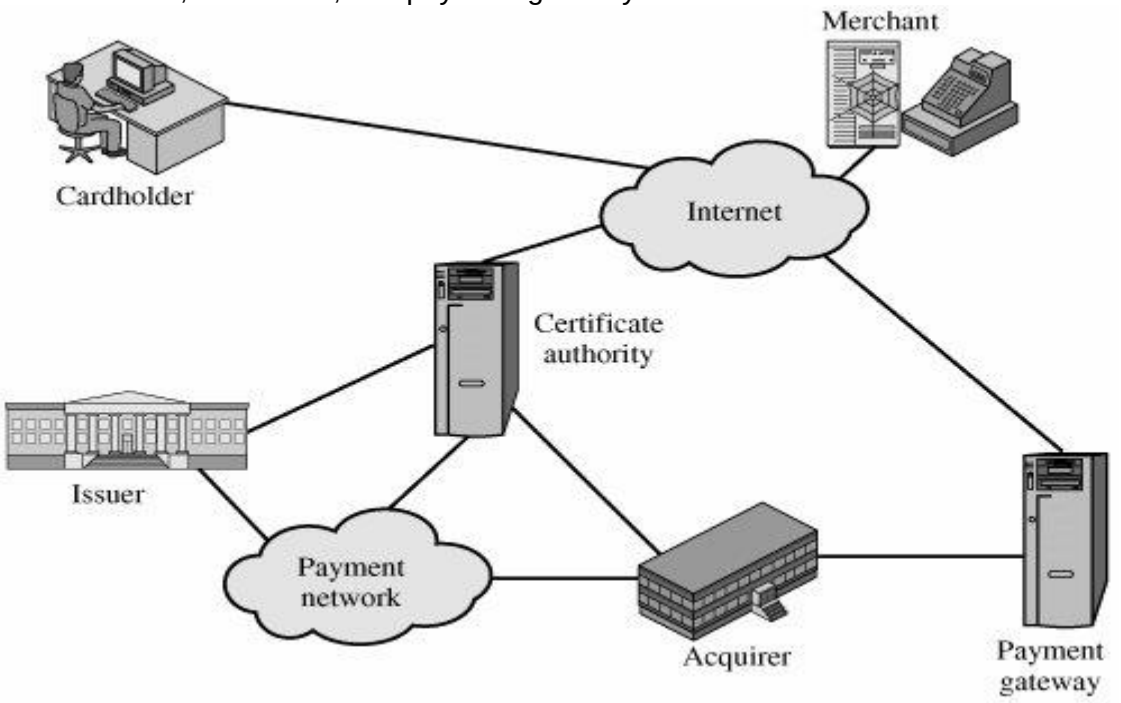
- Confidentiality of information
- Integrity of data
- Cardholder account authentication
- Merchant authentication

SET Participants

- **Cardholder:** A cardholder is an authorized holder of a payment card (e.g., MasterCard, Visa) that has been issued by an issuer.
- **Merchant:** A merchant is a person or organization that has goods or services to sell to the cardholder.
- **Issuer**: This is a financial institution, such as a bank, that provides the cardholder with the payment card.
- **Acquirer**: This is a financial institution that establishes an ccount with a merchant andprocesses payment card authorizations and payments.

Payment gateway: This is a function operated by the acquirer or a designated third party that processes merchant payment messages.

- **Certification authority (CA):** This is an entity that is trusted to issue X.509v3 public-key certificates for cardholders, merchants, and payment gateways.



Secure Electronic Commerce Components

5.5 SYSTEM SECURITY

INTRUDERS

One of the most publicized attacks to security is the intruder, generally referred to as hacker or cracker. Three classes of intruders are as follows:

- **Masquerader** – an individual who is not authorized to use the computer and who penetrates a system‟s access controls to exploit a legitimate user‟s account.
- **Misfeasor** – a legitimate user who accesses data, programs, or resources for which such access is not authorized, or who is authorized for such access but misuse his or her privileges.
- **Clandestine user** – an individual who seizes supervisory control of the system and uses this control to evade auditing and access controls or to suppress audit collection.

The masquerader is likely to be an outsider; the misfeasor generally is an insider; and the clandestine user can be either an outsider or an insider.Intruder attacks range from the benign to the serious. At the benign end of the scale, there are many people who simply wish to explore internets and see what is out there. At the serious end are individuals who are attempting to read privileged data, perform unauthorized modifications to data, or disrupt the system.

Benign intruders might be tolerable, although they do consume resources and may slow performance for legitimate users. However there is no way in advance to know whether an intruder will be benign or malign.

An analysis of previous attack revealed that there were two levels of hackers:

- The high levels were sophisticated users with a thorough knowledge of the technology.
- The low levels were the „foot soldiers‟ that merely use the supplied cracking programs with little understanding of how they work.

One of the results of the growing awareness of the intruder problem has been the establishment of a number of Computer Emergency Response Teams (CERT). These cooperative ventures collect information about system vulnerabilities and disseminate it to systems managers. Unfortunately, hackers can also gain access to CERT reports.

In addition to running password cracking programs, the intruders attempted to modify login software to enable them to capture passwords of users logging onto the systems.

Intrusion Techniques:

The objective of the intruders is to gain access to a system or to increase the range of privileges accessible on a system. Generally, this requires the intruders to acquire information that should be protected. In most cases, the information is in the form of a user password. Typically, a system must maintain a file that associates a password with each authorized user. If such a file is stored with no protection, then it is an easy matter to gain access to it. The password files can be protected in one of the two ways:

- **One way encryption** – the system stores only an encrypted form of user‟s password. In practice, the system usually performs a one way transformation (not reversible) in which the password is used to generate a key for the encryption function and in which a fixed length output is produced.
- **Access control** – access to the password file is limited to one or a very few accounts.

The following techniques are used for learning passwords.

1. Try default passwords used with standard accounts that are shipped with the system.Many administrators do not bother to change these defaults.
2. Exhaustively try all short passwords

Try words in the system‟s online dictionary or a list of likely passwords. Collect informationabout users such as their full names, the name of their spouse and children, pictures in their office and books in their office that are related to hobbies.

- Try user‟s phone number, social security numbers and room numbers.
- Try all legitimate license plate numbers.
- Use a Trojan horse to bypass restriction on access.
- Tap the line between a remote user and the host system.

Two principle countermeasures:

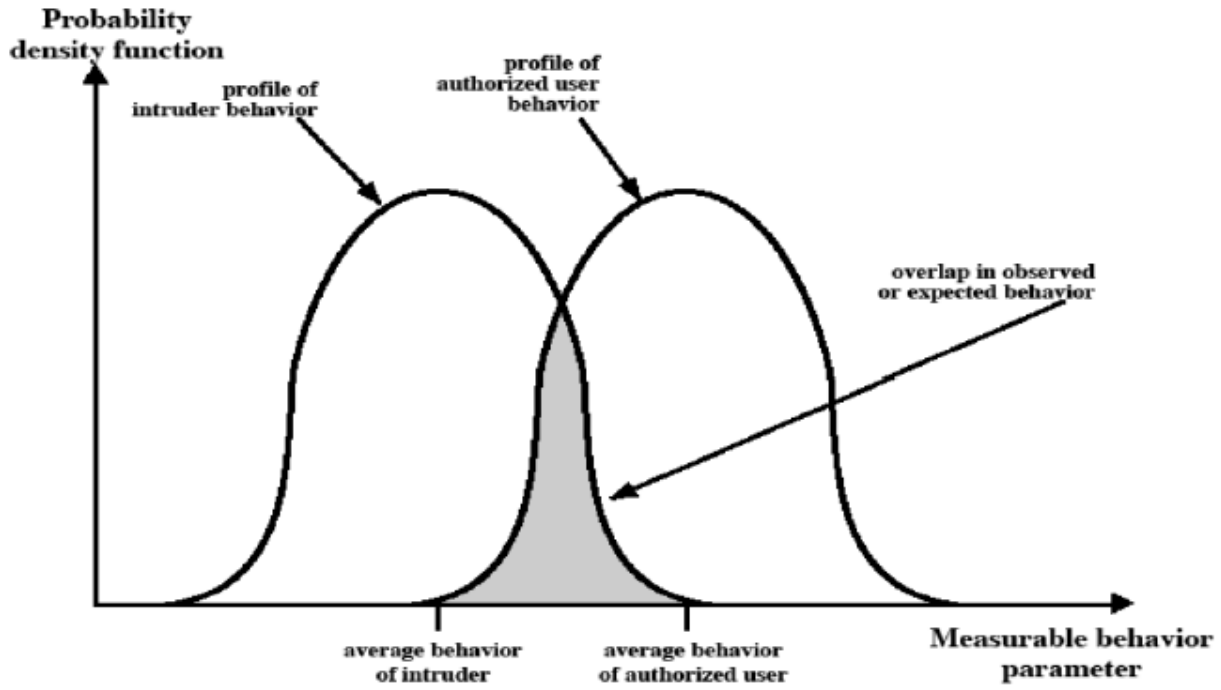
- **Detection** – concerned with learning of an attack, either before or after its success.
- **Prevention** – challenging security goal and an uphill bottle at all times.

Intrusion Detection

Inevitably, the best intrusion prevention system will fail. A system's second line of defense is intrusion detection, and this has been the focus of much research in recent years.
This interest is motivated by a number of considerations, including the following:
1. If an intrusion is detected quickly enough, the intruder can be identified and ejected from the system before any damage is done or any data are compromised.
2. An effective intrusion detection system can serve as a deterrent, so acting to prevent intrusions.
3. Intrusion detection enables the collection of information about intrusion techniques that can be used to strengthen the intrusion prevention facility.

Intrusion detection is based on the assumption that the behaviour of the intruder differs from that of a legitimate user in ways that can be quantified. Figure suggests, in very abstract terms, the nature of the task onfront the designer of an intrusion detection system. Although the typical behavior of an intruder differs from the typical behavior of an authorized user, there is an overlap in these behaviors. Thus, a loose interpretation of intruder behavior, which will catch more intruders, will also lead to a number of "false positives," or authorized users identified as intruders.

On the other hand, an attempt to limit false positives by a tight interpretation of intruder behavior will lead to an increase in false negatives, or intruders not identified as intruders. Thus, there is an element of Compromise and art in the practice of intrusion detection



Profiles of Behavior of Intruders and Authorized Users

Approaches to Intrusion Detection

The approaches to Intrusion detection are,

- Statistical anomaly detection
- Rule-based detection

1. Statistical anomaly detection: Involves the collection of data relating to the behavior of legitimate users over a period of time. Then statistical tests are applied to observed behavior to determine with a high level of confidence whether that behavior is not legitimate user behavior.

a. **Threshold detection**: This approach involves defining thresholds, independent of user, for the frequency of occurrence of various events.

b. **Profile based:** A profile of the activity of each user is developed and used to detect changes in the behavior of individual accounts.

2. Rule-based detection: Involves an attempt to define a set of rules that can be used to decide that a given behavior is that of an intruder.

Anomaly detection: Rules are developed to detect deviation from previous usage patterns

Penetration identification: An expert system approach is used which searches for suspicious behavior.

Distributed Intrusion Detection

The typical organization, however, needs to defend a distributed collection of hosts supported by a LAN points out the following major issues in the design of a distributed intrusion detection system.

A distributed intrusion detection system may need to deal with different audit record formats. In a heterogeneous environment, different systems will employ different native auditcollection systems and, if using intrusion detection, may employ different formats for security related audit records.

One or more nodes in the network will serve as collection and analysis points for the datafrom the systems on the network. Thus, either raw audit data or summary data must be transmitted across the network. Therefore, there is a requirement to assure the integrity and confidentiality of these data.

The below diagram shows the overall architecture, which consists of three main components:

- **Host agent module:** An audit collection module operating as a background process on a monitored system. Its purpose is to collect data on security-related events onthe host and transmit these to the central manager.
- **LAN monitor agent module:** Operates in the same fashion as a host agent module except that it analyzes LAN traffic and reports the results to the central manager.
- **Central manager module:** Receives reports from LAN monitor and host agents and processes and correlates these reports to detect intrusion.

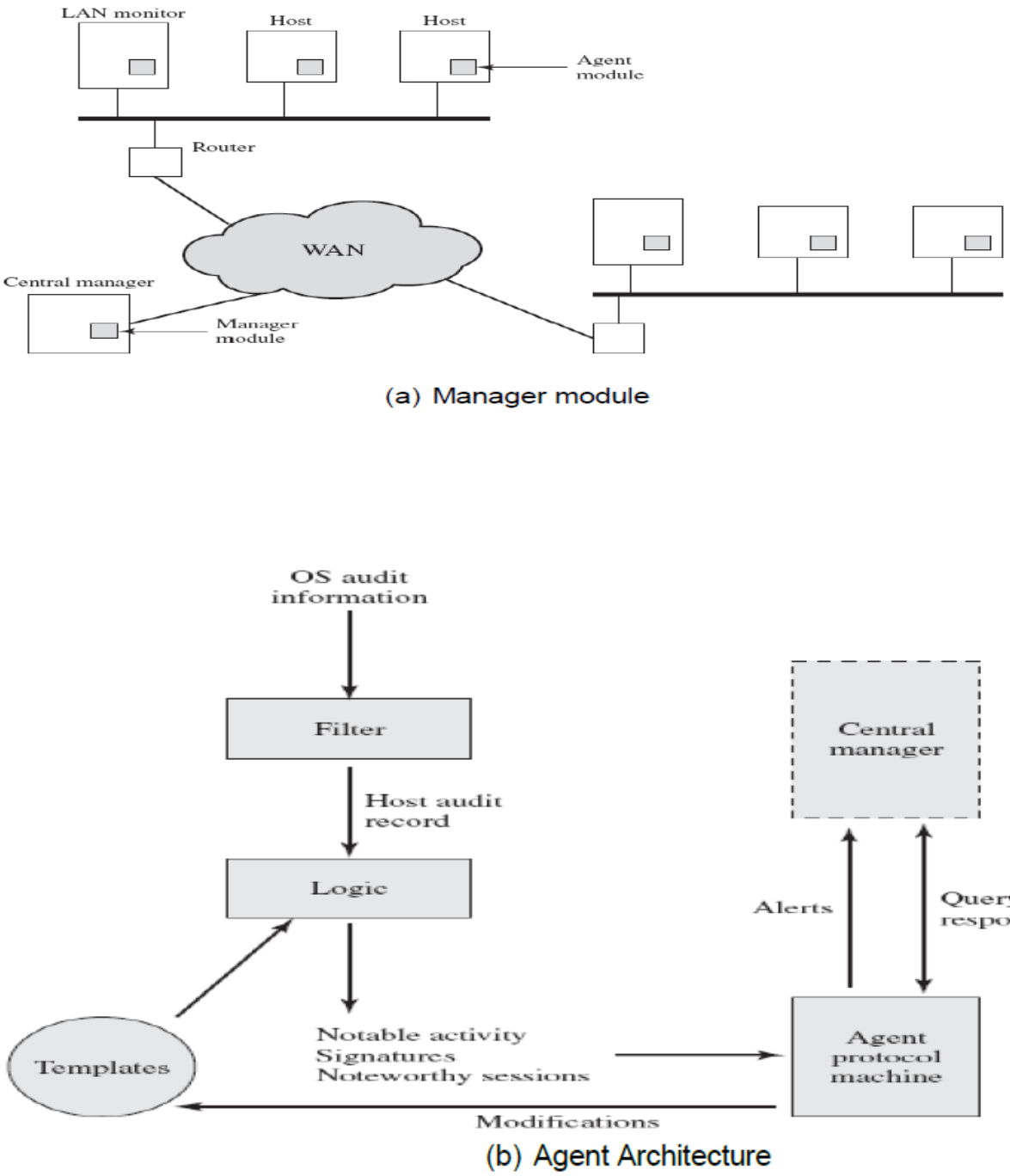
The scheme is designed to be independent of any operating system or system auditing
Implementation

.

- The agent captures each audit record produced by the native audit collection system
- A filter is applied that retains only those records that are of security interest.
- These records are then reformatted into a standardized format referred to as the host audit record (HAR).

Next, a template-driven logic module analyzes the records for suspicious activity.

- At the lowest level, the agent scans for notable events that are of interest independent of any past events.
- Examples include failed file accesses, accessing system files, and changing a file's access control.
- At the next higher level, the agent looks for sequences of events, such as known attack patterns (signatures).
- Finally, the agent looks for anomalous historical profile of that user, such as number of programs executed, number of files accessed, and the like.
- When suspicious activity is detected, an alert is sent to the central manager.
- The central manager includes an expert system that can draw inferences from received data.
- The manager may also query individual systems for copies of HARs to correlate with those from other agents.
- The LAN monitor agent also supplies information to the central manager.
- The LAN monitor agent audits host-host connections, services used, and volume of traffic.
- It searches for significant events, such as sudden changes in network load, the use of security-related services, and network activities such as rlogin.



Architecture of DIDS

5.6 MALICIOUS SOFTWARE

The words “Malicious Software” coin the word “Malware” and the meaning remains the same. Malicious Software refers to any malicious program that causes harm to a computer system or network. Malicious Malware Software attacks a computer or network in the form of viruses, worms, trojans, spyware, adware or rootkits. Their mission is often targeted at accomplishing unlawful tasks such as robbing protected data, deleting confidential documents or add software without the user consent.

Different Types of Malicious Software

- Computer Virus

A computer virus is a malicious software which self-replicates and attaches itself to other files/programs. It is capable of executing secretly when the host program/file is activated. The different types of Computer virus are Memory-Resident Virus, Program File Virus, Boot Sector Virus, Stealth Virus, Macro Virus, and Email Virus.

- Worms

A worm is a malicious software which similar to that of a computer virus is a self-replicating program, however, in the case of worms, it automatically executes itself. Worms spread over a network and are capable of launching a cumbersome and destructive attack within a short period.

- Trojan Horses

Unlike a computer virus or a worm – the trojan horse is a non-replicating program that appears legitimate. After gaining the trust, it secretly performs malicious and illicit activities when executed. Hackers make use of trojan horses to steal a user's password information, destroy data or programs on the hard disk. It is hard to detect!

- Spyware/Adware

Spyware secretly records information about a user and forwards it to third parties. The information gathered may cover files accessed on the computer, a user's online activities or even user's keystrokes.
Adware as the name interprets displays advertising banners while a program is running. Adware can also work like spyware, it is deployed to gather confidential information. Basically, to spy on and gather information from a victim's computer.

- Rootkit

A rootkit is a malicious software that alters the regular functionality of an OS on a computer in a stealthy manner. The altering helps the hacker to take full control of the system and the hacker acts as the system administrator on the victim's system. Almost all the rootkits are designed to hide their existence.

Malicious Software History

Even before the internet became widespread, malicious software (virus) was infected on personal computers with the executable boot sectors of floppy disks. Initially, the computer viruses were written for the Apple II and Macintosh devices. After the IBM PC and MS-DOS system became more widespread they were also targeted in the similar fashion.
The first worms originated on multitasking Unix systems, they were the first network-borne infectious programs too. SunOS and VAX BSD systems were infected by the first well-known worm of the time called the Internet Worm of 1988. Ever since the advent of Microsoft Windows platform in the 1990s, the infectious codes were written in the macro language of Microsoft Word and similar programs.

Methods of protection against malicious software

Malicious Software is definitely a security threat for corporate users and individuals, thereby detecting and fighting malware remains on top of the agenda for many firms. Since the time BYOD culture started to flourish, Endpoint Security and Endpoint Protection have become the topics of discussion in many IT conference rooms. Many corporates today try to implement the best Endpoint Security or Endpoint Protection software to steer clear of the dangers.
Remember, if it is an individual system, it is essential to have an antivirus installed and if you already have one in place see to that it is updated at regular intervals. This approach will help you to remain safe during new breakouts. Comodo's Free Antivirus, Endpoint Security, Endpoint Protection Solutions are your best option for detecting and fighting malicious software. For more details visit our official page!

5.7 VIRUS AND RELATED THREATS

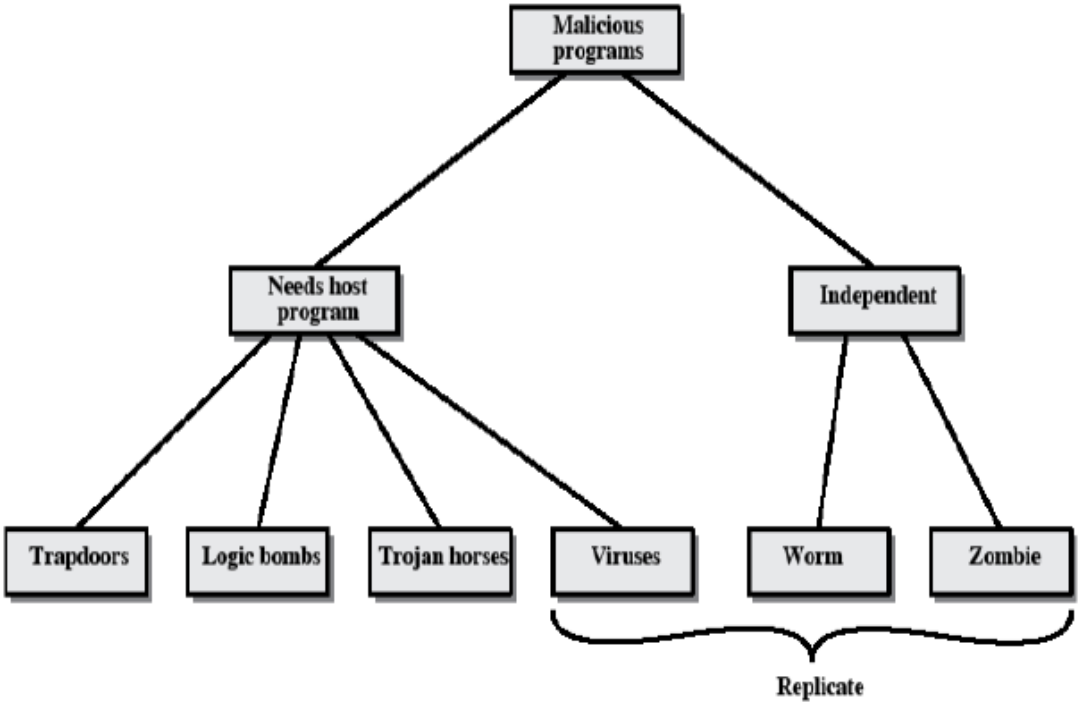
Malicious Programs:

Malicious software can be divided into **two categories**:

- Those that need a host program
- Those that is independent.

The former are essentially fragments of programs that cannot exist independently of some actual application program, utility, or system program. Viruses, logic bombs, and backdoors are examples.

The latter are self-contained programs that can be scheduled and run by the operating system. Worms and zombie programs are examples



TAXONOMY OF MALICIOUS PROGRAMS

The Nature of Viruses

A virus is a piece of software that can "infect" other programs by modifying them; the modification includes a copy of the virus program, which can then go on to infect other programs.

During its lifetime, a typical virus goes through the following **four phases**:

Dormant phase: The virus is idle. The virus will eventually be activated by some event, such as a date, the presence of another program or file, or the capacity of the disk exceeding some limit. Not all viruses have this stage.

Propagation phase: The virus places an identical copy of itself into other programs or into certain system areas on the disk. Each infected program will now contain a clone of the virus, which will itself enter a propagation phase.

Triggering phase: The virus is activated to perform the function for which it was intended. As with the dormant phase, the triggering phase can be caused by a variety of system events, including a count of the number of times that this copy of the virus has made copies of itself.

Execution phase: The function is performed. The function may be harmless, such as a message on the screen, or damaging, such as the destruction of programs and data files.

Virus Structure

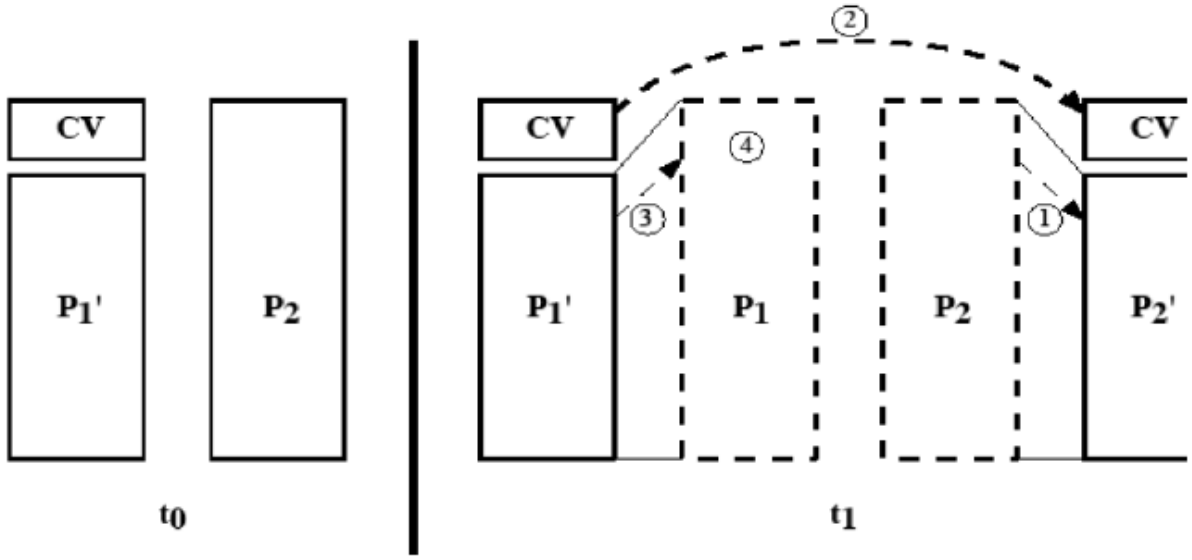
A virus can be pre pended or post pended to an executable program, or it can be embedded in some other fashion.

An infected program begins with the virus code and works as follows:

The first line of code is a jump to the main virus program. The second line is a special marker that is used by the virus to determine whether or not a potential victim program has already been infected with this virus. When the program is invoked, control is immediately transferred to the main virus program.

Finally, the virus transfers control to the original program. If the infection phase of the program is reasonably rapid, a user is unlikely to notice any difference between the execution of an infected and uninfected program. We assume that program P1 is infected with the virus CV. When this program is invoked, control passes to its virus, which performs the following steps

- For each uninfected file P2 that is found, the virus first compresses that file to produce P'2, which is shorter than the original program by the size of the virus.
- A copy of the virus is pre pended to the compressed program.
- The compressed version of the original infected program, P'1, is uncompressed.
- The uncompressed original program is executed



Compression Virus

Initial Infection:

Once a virus has gained entry to a system by infecting a single program, it is in a position to infect some or all other executable files on that system when the infected program executes.
Thus, viral infection can be completely prevented by preventing the virus from gaining entry in the first place. Unfortunately, prevention is extraordinarily difficult because a virus can be part of any program outside a system.

Types of Viruses:

Following categories as being among the most significant types of viruses:
Parasitic virus: The traditional and still most common form of virus. A parasitic virus attaches itself to executable files and replicates, when the infected program is executed, by finding other executable files to infect.
Memory-resident virus: Lodges in main memory as part of a resident system program. From that point on, the virus infects every program that executes.

Boot sector virus: Infects a master boot record or boot record and spreads when a system is booted from the disk containing the virus.
Stealth virus: A form of virus explicitly designed to hide itself from detection by antivirus software.
Polymorphic virus: A virus that mutates with every infection, making detection by the "signature" of the virus impossible.
Metamorphic virus: As with a polymorphic virus, a metamorphic virus with every infection. The difference is that a metamorphic virus rewrites itself completely at each iteration, increasing the difficulty of detection. Metamorphic viruses my change their behavior as well as their appearance.

Macro Viruses

Macro viruses are particularly threatening for a number of reasons:
1. A macro virus is platform independent. Virtually all of the macro viruses infect Microsoft Word documents. Any hardware platform and operating system that supports Word can be infected.
2. Macro viruses infect documents, not executable portions of code. Most of the information introduced onto a computer system is in the form of a document rather than a program.
3. Macro viruses are easily spread. A very common method is by electronic mail.

Macro viruses take advantage of a feature found in Word and other office applications such as Microsoft Excel, namely the macro. In essence, a macro is an executable program embedded in a word processing document or other type of file.

E-mail Viruses

A more recent development in malicious software is the e-mail virus. The first rapidly spreading e-mail viruses, such as Melissa, made use of a Microsoft Word macro embedded in an attachment. If the recipient opens the e-mail attachment, the Word macro is activated. Then
- The e-mail virus sends itself to everyone on the mailing list in the user's e-mail package
- The virus does local damage

Worms

A worm is a program that can replicate itself and send copies from computer to computer across network connections. Upon arrival, the worm may be activated to replicate and propagate again. Network worm programs use network connections to spread from system to system.
Once active within a system, a network worm can behave as a computer virus or bacteria, or it could implant Trojan horse programs or perform any number of disruptive or destructive actions. To replicate itself, a network worm uses some sort of network vehicle.
Examples include the following:
- **Electronic mail facility:** A worm mails a copy of itself to other systems.
- **Remote execution capability:** A worm executes a copy of itself on another system.
- **Remote login capability:** A worm logs onto a remote system as a user and then uses commands to copy itself from one system to the other.

The new copy of the worm program is then run on the remote system where, in addition to any functions that it performs at that system, it continues to spread in the same fashion.
A network worm exhibits the same characteristics as a computer virus: a dormant phase, a propagation phase, a triggering phase, and an execution phase. The propagation phase generally performs the following functions:
- Search for other systems to infect by examining host tables or similar repositories of remote system addresses
- Establish a connection with a remote system
- Copy itself to the remote system and cause the copy to be run As with viruses, network worms are difficult to counter.

The Morris Worm

The Morris worm was designed to spread on UNIX systems and used a number of different techniques for propagation.
1.It attempted to log on to a remote host as a legitimate user. In this method, the worm first attempted to crack the local password file, and then used the discovered passwords and corresponding user IDs. The assumption was that many users would use the same password on different systems.

To obtain the passwords, the worm ran a passwordcracking program that tried

- Each user's account name and simple permutations of it
- A list of 432 built-in passwords that Morris thought to be likely candidates
- All the words in the local system directory

2.It exploited a bug in the finger protocol, which reports the location of a remote user.
3.It exploited a trapdoor in the debug option of the remote process that receives and sends mail.

5.8 FIREWALLS

Internet connectivity is no longer an option for most organizations. However, while internet access provides benefits to the organization, it enables the outside world to reach and interact with local network assets.
This creates the threat to the organization. While it is possible to equip each workstation and server on the premises network with strong security features, such as intrusion protection, this is not a practical approach.
The alternative, increasingly accepted, is the firewall. The firewall is inserted between the premise network and internet to establish a controlled link and to erect an outer security wall or perimeter.

The aim of this perimeter is to protect the premises network from internet based attacksand to provide a single choke point where security and audit can be imposed.

The firewall can be a single computer system or a set of two or more systems that cooperate to perform the firewall function.

Firewall Characteristics

- All traffic from inside to outside, and vice versa, must pass through the firewall. This is achieved by physically blocking all access to the local network except via the firewall.
- Only authorized traffic, as defined by the local security policy, will be allowed to pass.Various types of firewalls are used, which implement various types of security policies.
- The firewall itself is immune to penetration. This implies that use of a trusted system with a secure operating system. This implies that use of a trusted system with asecure operating system.

Four techniques that firewall use to control access and enforce the site"s security policy is as follows:

- **Service control** – determines the type of internet services that can be accessed, inbound or outbound. The firewall may filter traffic on this basis of IP address and TCP port number; may provide proxy software that receives and interprets each service request before passing it on; or may host the server software itself, such as web or mail service.
- **Direction control** – determines the direction in which particular service request may be initiated and allowed to flow through the firewall.
- **User control** – controls access to a service according to which user is attempting to access it.
- **Behavior control** – controls how particular services are used.

Capabilities of Firewall

A firewall defines a single choke point that keeps unauthorized users out of the protected network, prohibits potentially vulnerable services from entering or leaving the network, and provides protection from various kinds of IP spoofing and routing attacks.

- A firewall provides a location for monitoring security related events. Audits and alarms can be implemented on the firewall system.
- A firewall is a convenient platform for several internet functions that are not security related.
- A firewall can serve as the platform for IPsec.

Limitations of Firewall

- The firewall cannot protect against attacks that bypass the firewall.
- The firewall does not protect against internal threats.
- The firewall cannot protect against the transfer of virus-infected programs or files.

Because of the variety of operating systems and applications supported inside the perimeter, it would be impractical and perhaps impossible for the firewall to scan all incoming files, e-mail, and messages for viruses.

TYPES OF FIREWALLS

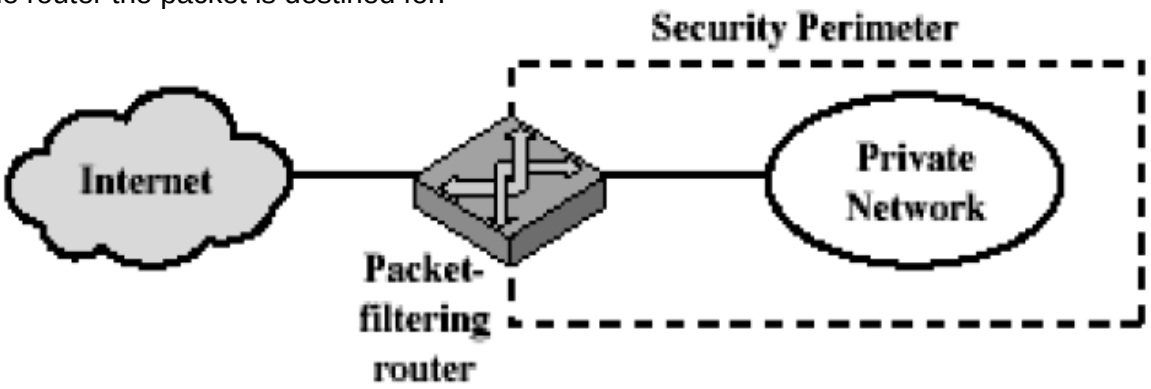
There are 3 common types of firewalls.

- Packet filters
- Application-level gateways
- Circuit-level gateways

Packet Filtering Router

A packet filtering router applies a set of rules to each incoming IP packet and then forwards or discards the packet. The router is typically configured to filter packets going in both directions. Filtering rules are based on the information contained in a network packet:

- **Source IP address** – IP address of the system that originated the IP packet.
- **Destinations IP address** – IP address of the system, the IP is trying to reach.
- **Source and destination transport level address** – transport level port number.
- **IP protocol field** – defines the transport protocol
- **Interface** – for a router with three or more ports, which interface of the router the packet come from or which interface of the router the packet is destined for.



Packet Filtering Router

The packet filter is typically set up as a list of rules based on matches to fields in the IP or TCP header. If there is a match to one of the rules, that rule is invoked to determine whether to forward or discard the packet. If there is no match to any rule, then a default action is taken.

Two default policies are possible:

- Default = discard: That which is not expressly permitted is prohibited.
- Default = forward: That which is not expressly prohibited is permitted.

Advantages of packet filter router

- Simple
- Transparent to users
- Very fast

Weakness of packet filter firewalls

- Packet filter firewalls do not examine upper-layer data; They cannot prevent attacks that employ application specific vulnerabilities or functions.
- As limited information is available to the firewall, the logging functionality present in packet filter firewall is limited.
- It does not support advanced user authentication schemes.
- They are generally vulnerable to attacks such as layer address spoofing.

Attacks on Packet Filtering Routers
Some of the attacks that can be made on packet filtering routers and the appropriate counter measures are the following:

- **IP address spoofing** – the intruders transmit packets from the outside with a source IP address field containing an address of an internal host.

Countermeasure: to discard packet with an inside source address if the packet arrives on an external interface.

Source routing attacks – the source station specifies the route that a packet should take as it crosses the internet; i.e., it will bypass the firewall.

Countermeasure: to discard all packets that uses this option.

➢**Tiny fragment attacks** – the intruder create extremely small fragments and force the TCP header information into a separate packet fragment. The attacker hopes that only the first fragment is examined and the remaining fragments are passed through.

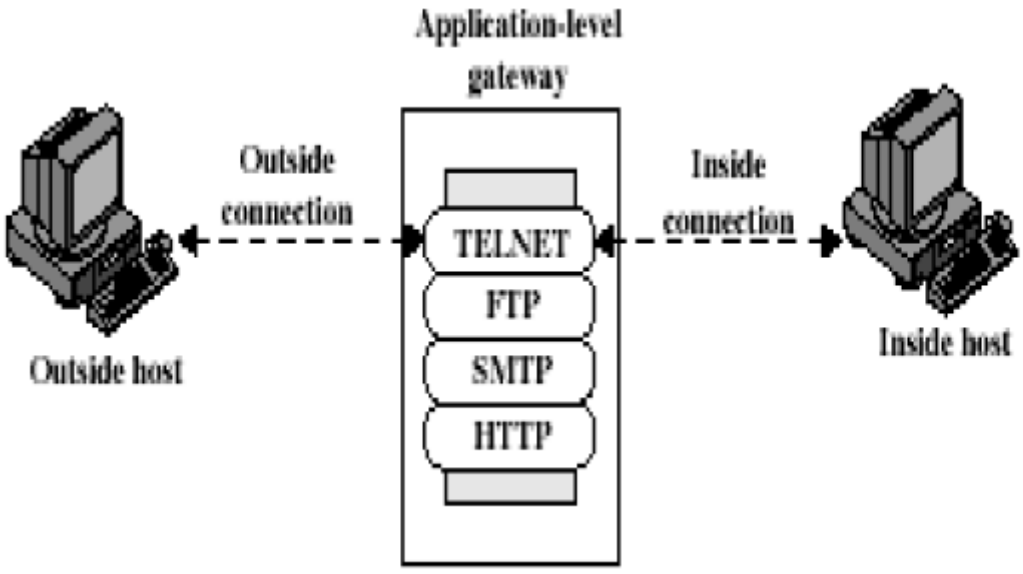
Countermeasure: to discard all packets where the protocol type is TCP and the IP fragment offset is equal to 1.

Application Level Gateway

An Application level gateway also called a proxy server, acts as a relay of application level traffic. The user contacts the gateway using a TCP/IP application, such as Telnet or FTP, and the gateway asks the user for the name of the remote host to be accessed.

When the user responds and provides a valid user ID and authentication information, the gateway contacts the application on the remote host and relays TCP segments containing the application data between the two endpoints.

Application level gateways tend to be more secure than packet filters. It is easy to log and audit all incoming traffic at the application level. A prime disadvantage is the additional processing overhead on each connection.



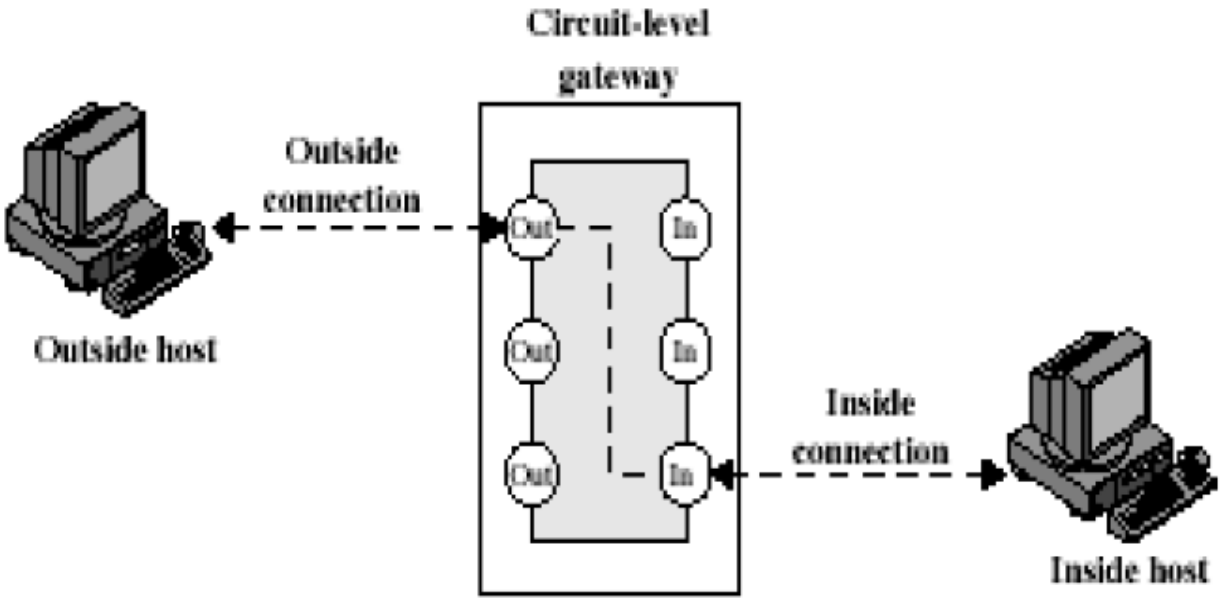
Application Level Gateway

Circuit Level Gateway

Circuit level gateway can be a stand-alone system or it can be a specified function performed by an application level gateway for certain applications. A Circuit level gateway does not permit an end-to-end TCP connection; rather, the gateway sets up two TCP connections, one between itself and a TCP user on an inner host and one between itself and a TCP user on an outer host.

Once the two connections are established, the gateway typically relays TCP segments from one connection to the other without examining the contents. The security function consists of determining which connections will be allowed.

A typical use of Circuit level gateways is a situation in which the system administrator trusts the internal users. The gateway can be configured to support application level or proxy service on inbound connections and circuit level functions for outbound connections.



Circuit Level Gateway