Module V

Authentication Protocols: Remote user authentication principles, Kerberos, Federated Identity Management Email Security: Pretty Good Privacy (PGP), S/MIME.
IP Security: IP security overview, IP security Policy, malicious software, Firewalls: Need for firewalls, firewall characteristics, types of firewalls.

REMOTE USER-AUTHENTICATION PRINCIPLES

RFC 2828 defines user authentication as:

The process of verifying an identity claimed by or for a system entity. An authentication process consists of two steps:

• Identification step: Presenting an identifier to the security system. (Identifiers should be assigned carefully, because authenticated identities are the basis for other security services, such as access control service.)

• Verification step: Presenting or generating authentication information that corroborates the binding between the entity and the identifier.

There are four general means of authenticating a user's identity, which can be used alone or in combination:

• Something the individual knows: Examples include a password, a personal identification number (PIN), or answers to a prearranged set of questions.

• Something the individual possesses: Examples include cryptographic keys, electronic keycards, smart cards, and physical keys. This type of authenticator is referred to as a token.

• Something the individual is (static biometrics): Examples include recognition by fingerprint, retina, and face.

• Something the individual does (dynamic biometrics): Examples include recognition by voice pattern, handwriting characteristics, and typing rhythm.

All of these methods, properly implemented and used, can provide secure user authentication. However, each method has problems. An adversary may be able toguess or steal a password. Similarly, an adversary may be able to forge or steal a token. A user may forget a password or

lose a token. Furthermore, there is a significant administrative overhead for managing password and token information on systems and securing such information on systems. With respect to biometric authenticators, there are a variety of problems, including dealing with false positives and false negatives, user acceptance, cost, and convenience. For network-based user authentication, the most important methods involve cryptographic keys and something the individual knows, such as a password.

Mutual Authentication

Such protocols enable communicating parties to satisfy themselves mutually about each other's identity and to exchange session keys.

Central to the problem of authenticated key exchange are two issues: **confidentiality and timeliness.**

- To prevent masquerade and to prevent compromise of session keys, essential identification and session-key information must be communicated in encrypted form. This requires the prior existence of secret or public keys that can be used for this purpose.
- The second issue, timeliness, is important because of the threat of message replays. Such replays, at worst, could allow an opponent to compromise a session key or successfully impersonate another party

[GONG93] lists the following examples of replay attacks:

• Simple replay: The opponent simply copies a message and replays it later.
• Repetition that can be logged: An opponent can replay a timestamped message within the valid time window.
• Repetition that cannot be detected: This situation could arise because the original message could have been suppressed and thus did not arrive at its destination; only the replay message arrives.
• Backward replay without modification: This is a replay back to the message sender. This attack is possible if symmetric encryption is used and the sender cannot easily recognize the difference between messages sent and message received on the basis of content.

Two general approachesto coping with replay attacks are:

• Timestamps: Party A accepts a message as fresh only if the message contains a timestamp that, in A's judgment, is close enough to A's knowledge of current time. This approach requires that clocks among the various participants be synchronized.

- Challenge/response: Party A, expecting a fresh message from B, first sends B a nonce (challenge) and requires that the subsequent message (response) received from B contain the correct nonce value

It can be argued that the timestamp approach should not be used for connection-oriented applications because of the inherent difficulties with this technique:

- First, some sort of protocol is needed to maintain synchronization among the various processor clocks. This protocol must be both fault tolerant, to cope with network errors, and secure, to cope with hostile attacks
- Second, the opportunity for a successful attack will arise if there is a temporary loss of synchronization resulting from a fault in the clock mechanism of one of the parties
- Finally, because of the variable and unpredictable nature of network delays, distributed clocks cannot be expected to maintain precise synchronization

Therefore, any timestamp-based procedure must allow for a window of time sufficiently large to accommodate network delays yet sufficiently small to minimize the opportunity for attack

On the other hand, the challenge-response approach is unsuitable for a connectionless type of application, because it requires the overhead of a handshake before any connectionless transmission, effectively negating the chief characteristic of a connectionless transaction. For such applications, reliance on some sort of secure time server and a consistent attempt by each party to keep its clocks in synchronization may be the best approach.

One-Way Authentication

One application for which encryption is growing in popularity is electronic mail (e-mail). The very nature of electronic mail, and its chief benefit, is that it is not necessary for the sender and receiver to be online at the same time. Instead, the e-mail message is forwarded to the receiver's electronic mailbox, where it is buffered until the receiver is available to read it.

The "envelope" or header of the e-mail message must be in the clear, so that the message can be handled by the store-and-forward e-mail protocol, such as the Simple Mail Transfer Protocol (SMTP) or X.400. However, it is often desirable that the mail-handling protocol not require access to the plaintext form of the message, because that would require trusting the mail-handling mechanism. Accordingly, the e-mail message should be encrypted such that the mail-handling system is not in possession of the decryption key. A second requirement is that of authentication. Typically, the recipient wants some assurance that the message is from the alleged sender.

KERBEROS

Kerberos (In Greek mythology, Kerberos is a many headed dog, commonly three, the guardian of the entrance of Hades. Just as the Greek Kerberos/Cerberus has three heads, the modern Kerberos was intended to have three componentsto guard a network's gate: authentication, accounting, and audit. The last two heads were never implemented) is an authentication service developed as part of Project Athena at MIT.

The problem that Kerberos addresses is servers being able to restrict access toauthorized users and to be able to authenticate requests for service.In particular, the following three threats exist:

1. A user may gain access to a particular workstation and pretend to be another user operating from that workstation.
2. A user may alter the network address of a workstation so that the requests sentfrom the altered workstation appear to come from the impersonated workstation.
3. A user may eavesdrop on exchanges and use a replay attack to gain entranceto a server or to disrupt operations.

In any of these cases, an unauthorized user may be able to gain access to servicesand data that he or she is not authorized to access. Rather than building in elaborateauthentication protocols at each server, Kerberos provides a centralizedauthentication server whose function is to authenticate users to server's and servers to users.

Kerberos relies exclusively on symmetric encryption, making no use of public-keyencryption.

Two versions of Kerberos are in common use. Version 4 [MILL88, STEI88] implementations still exist. Version 5 [KOHL94] corrects some of the securitydeficiencies of version 4 and has been issued as a proposed Internet Standard (RFC 4120)

Motivation

If a set of users is provided with dedicated personal computers that have no networkconnections, then a user's resources and files can be protected by physically securingeach personal computer. When these users instead are served by a centralized timesharingsystem, the time-sharing operating system must provide the security. Theoperating system can enforce access-control policies based on user identity and usethe logon procedure to identify users.

Today, neither of these scenarios is typical. More common is a distributed architectureconsisting of dedicated user workstations (clients) and distributed or centralizedservers. In this environment, three approaches to security can be envisioned.

1. Rely on each individual client workstation to assure the identity of its user orusers and rely on each server to enforce a security policy based on user identification(ID).

2. Require that client systems authenticate themselves to servers, but trust the clientsystem concerning the identity of its user.
3. Require the user to prove his or her identity for each service invoked. Alsorequire that servers prove their identity to clients.

Kerberos supports this third approach. Kerberos assumes adistributed client/server architecture and employs one or more Kerberos servers toprovide an authentication service.
The first published report on Kerberos listed the followingrequirements.

- **Secure:** A network eavesdropper should not be able to obtain the necessaryinformation to impersonate a user.
- **Reliable:** Kerberos should be highly reliable and should employ adistributed server architecture with one system able to back up another.
- **Transparent:** The user should not be aware that authentication is takingplace beyond the requirement to enter a password.
- **Scalable:** The system should be capable of supporting large numbers of clients and server's i.e a modular, distributed architecture.

Steps to authenticate:

1. Client requests an authentication ticket (TGT) from the Key Distribution Center (KDC)
2. The KDC verifies the credentials and sends back an encrypted TGT and session key
3. The TGT is encrypted using the Ticket Granting Service (TGS) secret key
4. The client stores the TGT and when it expires the local session manager will request another TGT (this process is transparent to the user)

If the Client is requesting access to a service or other resource on the network, this is the process:

5. The client sends the current TGT to the TGS with the Service Principal Name (SPN) of the resource the client wants to access
6. The KDC verifies the TGT of the user and that the user has access to the service
7. TGS sends a valid session key for the service to the client
8. Client forwards the session key to the service to prove the user has access, and the service grants access.

KERBEROS REALMS AND MULTIPLE KERBERI

A full-service Kerberos environmentconsisting of a Kerberos server, a number of clients, and a number of applicationservers requires the following:

1. The Kerberos server must have the user ID and hashed passwords of all participatingusers in its database. All users are registered with the Kerberos server.

2. The Kerberos server must share a secret key with each server.All servers areregistered with the Kerberos server.

Such an environment is referred to as a **Kerberos realm**.

- A Kerberos realm is a set of managed nodes that share the same Kerberos database. The Kerberos database resides on the Kerberos master computer system, which should be kept in a physically secure room.
- A read-only copy of the Kerberos database might also reside on other Kerberos computer systems.
- However, all changes to the database must be made on the master computer system. Changing or accessing the contents of a Kerberos database requires the Kerberos master password.
- A related concept is that of a **Kerberos principal**, which is a service or user that is known to the Kerberos system. Each Kerberos principal is identified by its principal name.
- Principal names consist of three parts: a service or user name, an instance name, and a realm name networks of clients and servers under different administrative organizations typically constitute different realms.
- However, users in one realm may need access to servers in other realms, and some servers may be willing to provide service to users from other realms, provided that those users are authenticated.

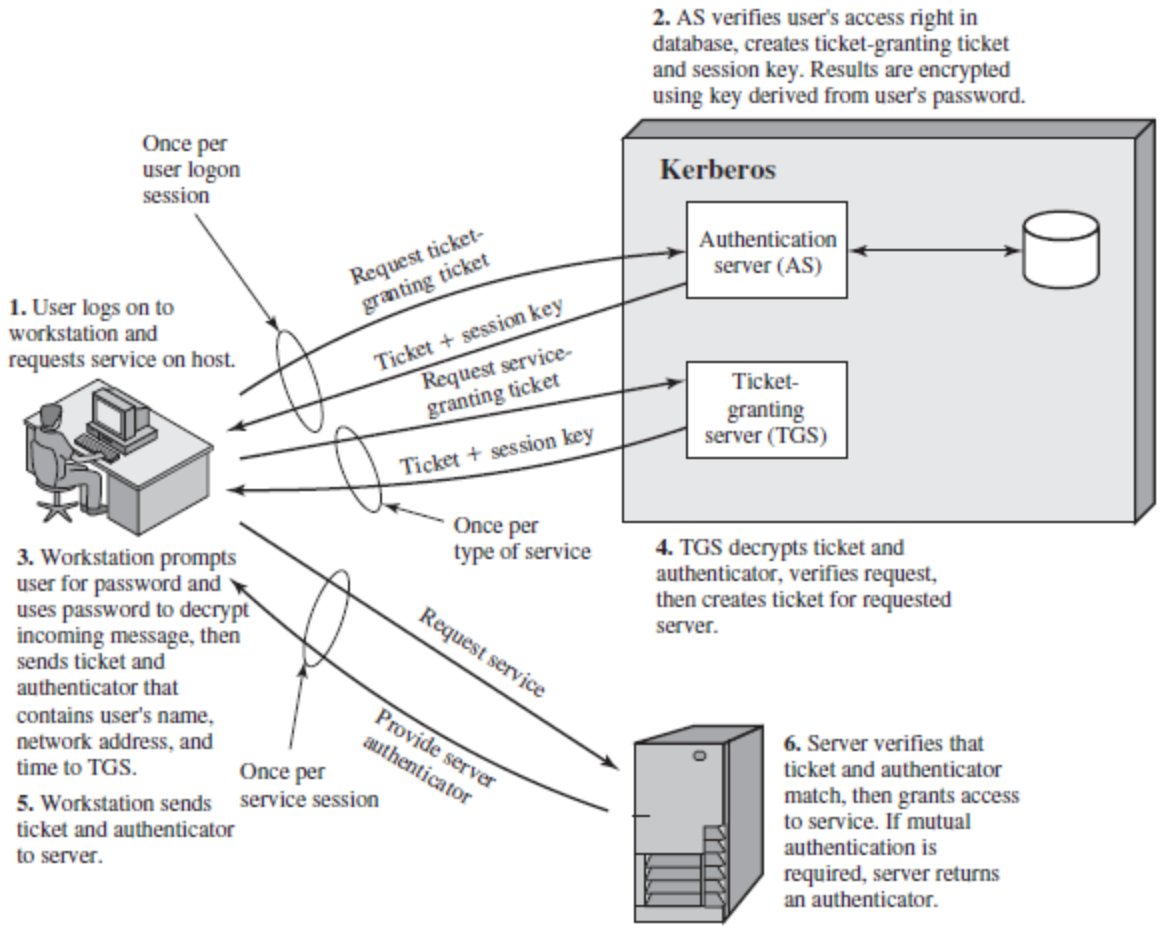


Figure 15.1 Overview of Kerberos

Kerberos provides a mechanism for supporting such inter-realm authentication.For two realms to support interrealm authentication, a third requirement is added:

3. The Kerberos server in each interoperating realm shares a secret key with theserver in the other realm.The two Kerberos servers are registered with each other.

The scheme requires that the Kerberos server in one realm trust the Kerberosserver in the other realm to authenticate its users. Furthermore, the participatingservers in the second realm must also be willing to trust the Kerberos server in thefirst realm.

- A user wishing service on a server in another realm needs a ticket forthat server.
- The user's client follows the usual procedures to gain access to the localTGS and then requests a ticket-granting ticket for a remote TGS (TGS in anotherrealm).
- The client can then apply to the remote TGS for a service-granting ticket forthe desired server in the realm of the remote TGS.

(1) C → AS: $ID_c \| ID_{tgs} \| TS_1$

(2) AS → C: $E(K_c, [K_{c,tgs} \| ID_{tgs} \| TS_2 \| Lifetime_2 \| Ticket_{tgs}])$

(3) C → TGS: $ID_{tgsrem} \| Ticket_{tgs} \| Authenticator_c$

(4) TGS → C: $E(K_{c,tgs}, [K_{c,tgsrem} \| ID_{tgsrem} \| TS_4 \| Ticket_{tgsrem}])$

(5) C → TGS_{rem}: $ID_{vrem} \| Ticket_{tgsrem} \| Authenticator_c$

(6) TGS_{rem} → C: $E(K_{c,tgsrem}, [K_{c,vrem} \| ID_{vrem} \| TS_6 \| Ticket_{vrem}])$

(7) C → V_{rem}: $Ticket_{vrem} \| Authenticator_c$

- The ticket presented to the remote server indicates the realm in whichthe user was originally authenticated. The server chooses whether to honor theremote request.
- One problem presented by the foregoing approach is that it does not scale well toomany realms. If there are realms, then there must be secure keyexchanges so that each Kerberos realm can interoperate with all other Kerberos realms.

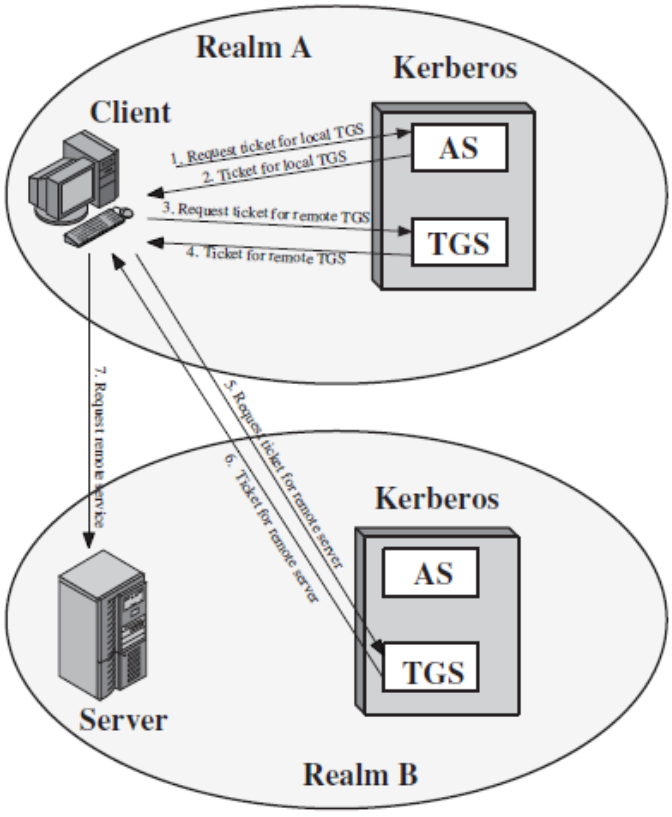


Figure 15.2 Request for Service in Another Realm

DIFFERENCES BETWEEN VERSIONS 4 AND 5

Environmental Shortcomings	**VERSION 4**	**VERSION 5**
Encryption system dependence	requires the use of DES	ciphertext is tagged with an encryption-type identifier so that any

		encryption technique may be used
Internet protocol dependence	requires the use of IP addresses. Other address types, such as the ISO network address, are not accommodated	network addresses are tagged with type and length, allowing any network addresses type to be used.
Message byte ordering	sender of message employs byte ordering of its own choosing and tags the message to indicate least significant byte or most significant byte in lowest address	all message structures are defined using Abstract Syntax Notation One (ASN.1) and Basic Encoding Rules (BER), which provide an unambiguous byte ordering.
Ticket lifetime	Lifetime values are encoded in an 8-bit quantity in units of five minutes. This may be inadequate for some applications.	tickets include an explicit start time and end time, allowing tickets with arbitrary lifetimes.
Authentication forwarding	does not allow credentials issued to one client to be forwarded to some other host and used by some other client.	allows credentials issued to one client to be forwarded to some other host and used by some other client.
Interrealm authentication	interoperability among realms requires on the order of Kerberos-to-Kerberos relationships	supports a method that requires fewer relationships, as described shortly.
Technical Deficiencies	VERSION 4	VERSION 5
PCBC encryption	Propagating cipher block chaining (PCBC) mode is vulnerable to an attack involving the interchange of ciphertext blocks.PCBC was intended to provide an integrity check as part of the encryption operation.	Provides explicit integrity mechanisms, allowing the standard CBC mode to be used for encryption. In particular, a checksum or hash code is attached to the message prior to encryption using CBC.
Session keys	Each ticket includes a session key that is used by the client to encrypt the authenticator sent to the service associated with that ticket.	it is possible for a client and server to negotiate a sub session key, which is to be used only for that one connection.
Password attacks	The message from the AS to the client includes material encrypted with a key based on the client's password. An opponent can capture this message and attempt to decrypt it by trying various passwords	Provides pre-authentication which should make password attacks more difficult, but it does not prevent them.

THE VERSION 4 AUTHENTICATION DIALOGUE

Table 15.1 Summary of Kerberos Version 4 Message Exchanges

(1) $C \rightarrow AS \quad ID_c \| ID_{tgs} \| TS_1$

(2) $AS \rightarrow C \quad E(K_c, [K_{c,tgs} \| ID_{tgs} \| TS_2 \| Lifetime_2 \| Ticket_{tgs}])$

$$Ticket_{tgs} = E(K_{tgs}, [K_{c,tgs} \| ID_C \| AD_C \| ID_{tgs} \| TS_2 \| Lifetime_2])$$

(a) Authentication Service Exchange to obtain ticket-granting ticket

(3) $C \rightarrow TGS \quad ID_v \| Ticket_{tgs} \| Authenticator_c$

(4) $TGS \rightarrow C \quad E(K_{c,tgs}, [K_{c,v} \| ID_v \| TS_4 \| Ticket_v])$

$$Ticket_{tgs} = E(K_{tgs}, [K_{c,tgs} \| ID_C \| AD_C \| ID_{tgs} \| TS_2 \| Lifetime_2])$$

$$Ticket_v = E(K_v, [K_{c,v} \| ID_C \| AD_C \| ID_v \| TS_4 \| Lifetime_4])$$

$$Authenticator_c = E(K_{c,tgs}, [ID_C \| AD_C \| TS_3])$$

(b) Ticket-Granting Service Exchange to obtain service-granting ticket

(5) $C \rightarrow V \quad Ticket_v \| Authenticator_c$

(6) $V \rightarrow C \quad E(K_{c,v}, [TS_5 + 1])$ (for mutual authentication)

$$Ticket_v = E(K_v, [K_{c,v} \| ID_C \| AD_C \| ID_v \| TS_4 \| Lifetime_4])$$

$$Authenticator_c = E(K_{c,v}, [ID_C \| AD_C \| TS_5])$$

(c) Client/Server Authentication Exchange to obtain service

THE VERSION 5 AUTHENTICATION DIALOGUE

Table 15.3 Summary of Kerberos Version 5 Message Exchanges

(1) $C \rightarrow AS \quad Options \| ID_c \| Realm_c \| ID_{tgs} \| Times \| Nonce_1$

(2) $AS \rightarrow C \quad Realm_C \| ID_C \| Ticket_{tgs} \| E(K_c, [K_{c,tgs} \| Times \| Nonce_1 \| Realm_{tgs} \| ID_{tgs}])$

$$Ticket_{tgs} = E(K_{tgs}, [Flags \| K_{c,tgs} \| Realm_c \| ID_C \| AD_C \| Times])$$

(a) Authentication Service Exchange to obtain ticket-granting ticket

(3) $C \rightarrow TGS \quad Options \| ID_v \| Times \| Nonce_2 \| Ticket_{tgs} \| Authenticator_c$

(4) $TGS \rightarrow C \quad Realm_c \| ID_C \| Ticket_v \| E(K_{c,tgs}, [K_{c,v} \| Times \| Nonce_2 \| Realm_v \| ID_v])$

$$Ticket_{tgs} = E(K_{tgs}, [Flags \| K_{c,tgs} \| Realm_c \| ID_C \| AD_C \| Times])$$

$$Ticket_v = E(K_v, [Flags \| K_{c,v} \| Realm_c \| ID_C \| AD_C \| Times])$$

$$Authenticator_c = E(K_{c,tgs}, [ID_C \| Realm_c \| TS_1])$$

(b) Ticket-Granting Service Exchange to obtain service-granting ticket

(5) $C \rightarrow V \quad Options \| Ticket_v \| Authenticator_c$

(6) $V \rightarrow C \quad E_{K_{c,v}}[TS_2 \| Subkey \| Seq\#]$

$$Ticket_v = E(K_v, [Flag \| K_{c,v} \| Realm_c \| ID_C \| AD_C \| Times])$$

$$Authenticator_c = E(K_{c,v}, [ID_C \| Relam_c \| TS_2 \| Subkey \| Seq\#])$$

(c) Client/Server Authentication Exchange to obtain service

PRETTY GOOD PRIVACY

Notation Most of the notation used PGP is perhaps best way to summarize the following symbols are used

- KS=session key used in symmetric encryption scheme
- PRa= private key of user A, used in public-key encryption scheme
- PU=a public key of user A, used in public-key encryption scheme
- EP=public-key encryption
- DP =public-key decryption symmetric encryption
- DC= symmetric decryption
- H= hash function
- | | = concatenation
- Z= compression using ZIP algorithm
- R64= conversion to radix 64 ASCII format

Operational Description

The actual operation of PGP, as opposed to the management of keys, consists of four services: authentication, confidentiality, compression, and e-mail compatibility (Table 18.1).We examine each of these in turn.

AUTHENTICATION

The sequence is as follows.

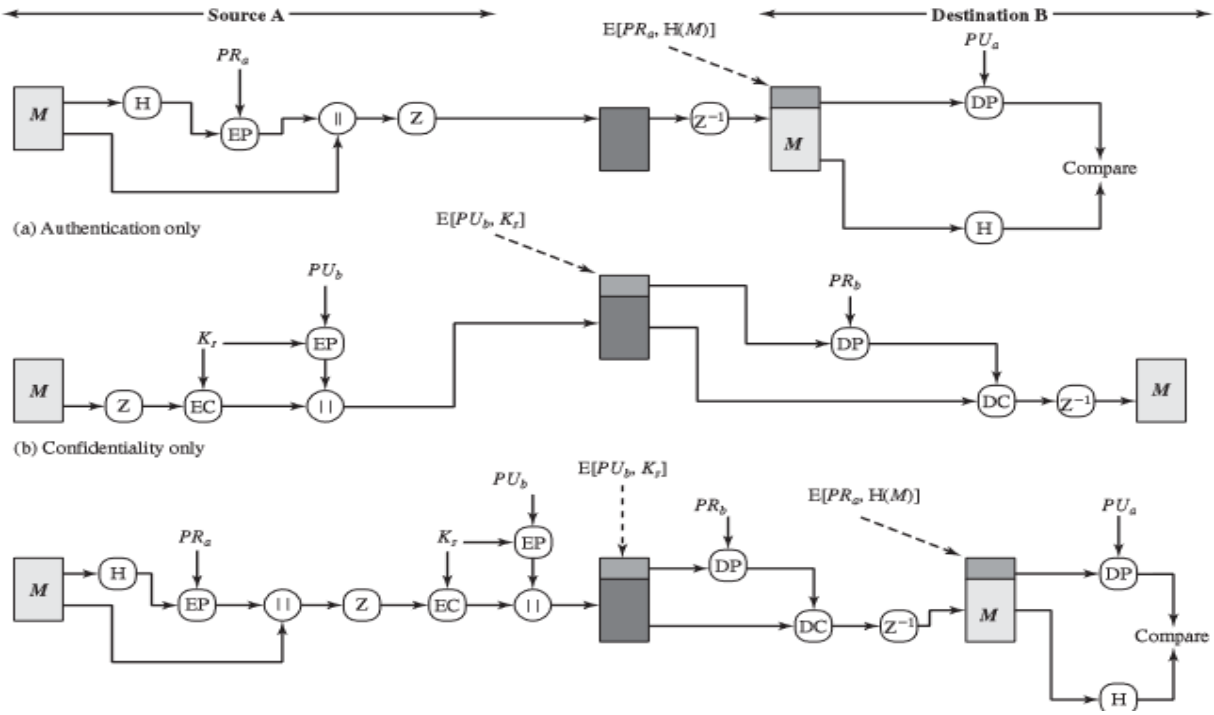
1. The sender creates a message.

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

3. The hash code is encrypted with RSA using the sender's private key, and the result is prepended to the message.

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

5. 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.



CONFIDENTIALITY Another basic service provided by PGP is confidentiality, which is provided by encrypting messages to be transmitted or to be stored locally as files.

Figure 18.1b illustrates the sequence, which can be described as follows.

1. The sender generates a message and a random 128-bit number to be used as a session key for this message only.

2. The message is encrypted using CAST-128 (or IDEA or 3DES) with the session key.

3. The session key is encrypted with RSA using the recipient's public key and is prepended to the message.

4. The receiver uses RSA with its private key to decrypt and recover the session key.

5. The session key is used to decrypt the message.

As an alternative to the use of RSA for key encryption, PGP provides an option referred to as Diffie-Hellman.

CONFIDENTIALITY AND AUTHENTICATION

As Figure 18.1c(above) illustrates, 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 message plus signature is encrypted and the session key is encrypted. It is generally more convenient to store a signature with a plaintext version of a message.

COMPRESSION

As a default, PGP compresses the message after applying the signature but before encryption.

The compression algorithm, indicated by Z for compression and Z^–1 for decompression in Figure above 18.1.

1. The signature is generated before compression for two reasons:

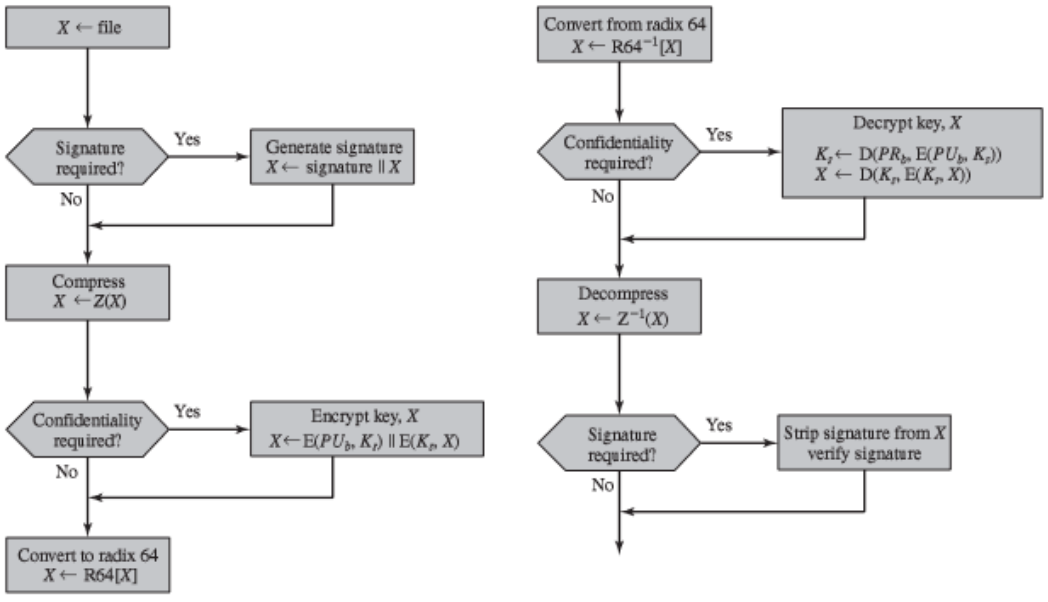
- It is preferable to sign an uncompressed message so that one can store only the uncompressed message together with the signature for future verification. If one signed a compressed document, then it would be necessary either to store a compressed version of the message for later verification or to recompress the message when verification is required.
- 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 compressed forms

2. 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.

E-MAIL COMPATIBILITY

When PGP is used, at least part of the block to be transmitted is encrypted. If the confidentiality service is used, the message plus signature (if present) are encrypted (with a one-time symmetric key). However, many electronic mail systems only permit the use of blocks consisting of ASCII text. 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.

One noteworthy aspect of the radix-64 algorithm is that it blindly converts the input stream to radix-64 format regardless of content, even if the input happens to be ASCII text. Thus, if a message is signed but not encrypted and the conversion is applied to the entire block, the output will be unreadable to the casual observer, which provides a certain level of confidentiality.



(a) Generic transmission diagram (from A)

(b) Generic reception diagram (to B)

Figure 18.2 Transmission and Reception of PGP Messages

Figure 18.2 shows the relationship among the four services so far discussed. On transmission (if it is required),a signature is generated using a hash code of the uncompressed plaintext. Then

the plaintext (plus signature if present) is compressed. Next, if confidentiality is required, the block (compressed plaintext or compressed signature plus plaintext) is encrypted and pretended with the public-key encrypted symmetric encryption key. Finally, the entire block is converted to radix-64 format.

Cryptographic Keys and Key Rings

PGP makes use of four types of keys: one-time session symmetric keys, public keys, private keys, and passphrase-based symmetric keys (explained subsequently).Three separate requirements can be identified with respect to these keys.

1. A means of generating unpredictable session keys is needed.

2. We would like to allow a user to have multiple public-key/private-key pairs. One reason is that the user may wish to change his or her key pair from time to time. When this happens, any messages in the pipeline will be constructed with an obsolete key.

3. Each PGP entity must maintain a file of its own public/private key pairs as well as a file of public keys of correspondents.

KEY IDENTIFIERS

As we have discussed, an encrypted message is accompanied by an encrypted form of the session key that was used for message encryption. The session key itself is encrypted with the recipient's public key. Hence, only the recipient will be able to recover the session key and therefore recover the message.

A key ID is also required for the PGP digital signature. Because a sender may use one of a number of private keys to encrypt the message digest, the recipient must know which public key is intended for use. Accordingly, the digital signature component of a message includes the 64-bit key ID of the required public key. When the message is received, the recipient verifies that the key ID is for a public key that it knows for that sender and then proceeds to verify the signature.

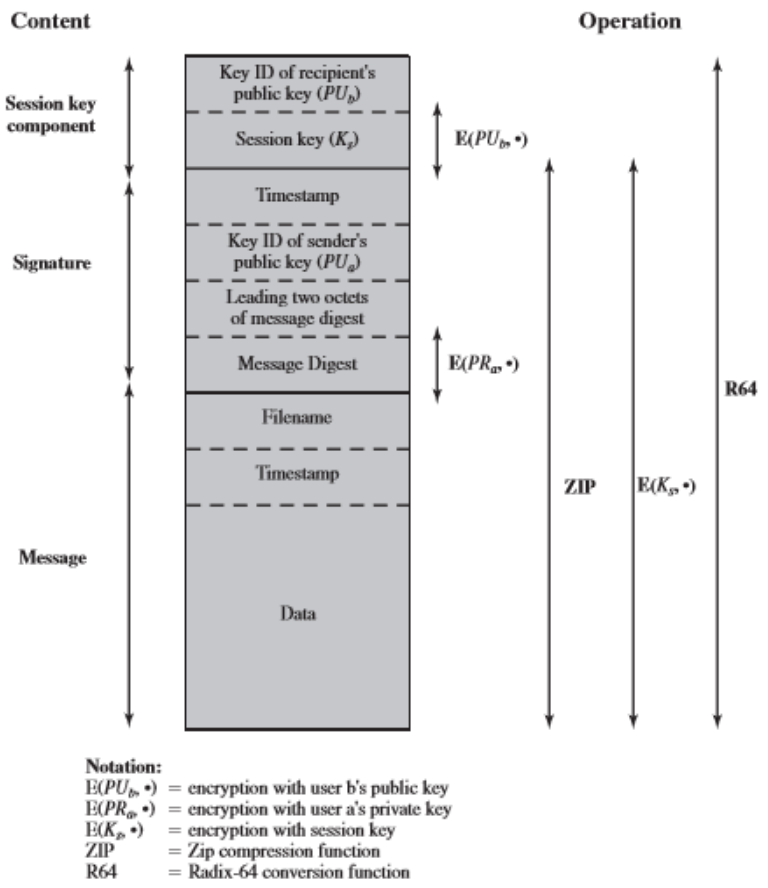


Figure 18.3 General Format PGP Message (from A to B)

Now that the concept of key ID has been introduced;

The message component includes the actual data to be stored or transmitted, as well as a filename and a timestamp that specifies the time of creation. The signature component includes the following.

• **Timestamp:** The time at which the signature was made.

• **Message digest:** The 160-bit SHA-1 digest encrypted with the sender's private signature key. The digest is calculated over the signature timestamp concatenated with the data portion of the message component. The inclusion of the signature timestamp in the digest insures against replay types of attacks.

Leading two octets of message digest: Enables the recipient to determine if the correct public key was used to decrypt the message digest for authentication by comparing this plaintext copy of the first two octets with the first two octets of the decrypted digest.

• **Key ID of sender's public key**: Identifies the public key that should be used to decrypt the message digest and, hence, identifies the private key that was used to encrypt the message digest.

KEY RINGS

The scheme used in PGP is to provide a pair of data structures at each node, one to store the public/private key pairs owned by that node and one to store the public keys of other users known at this node. These data structures are referred to, respectively, as the private-key ring and the public-key ring.

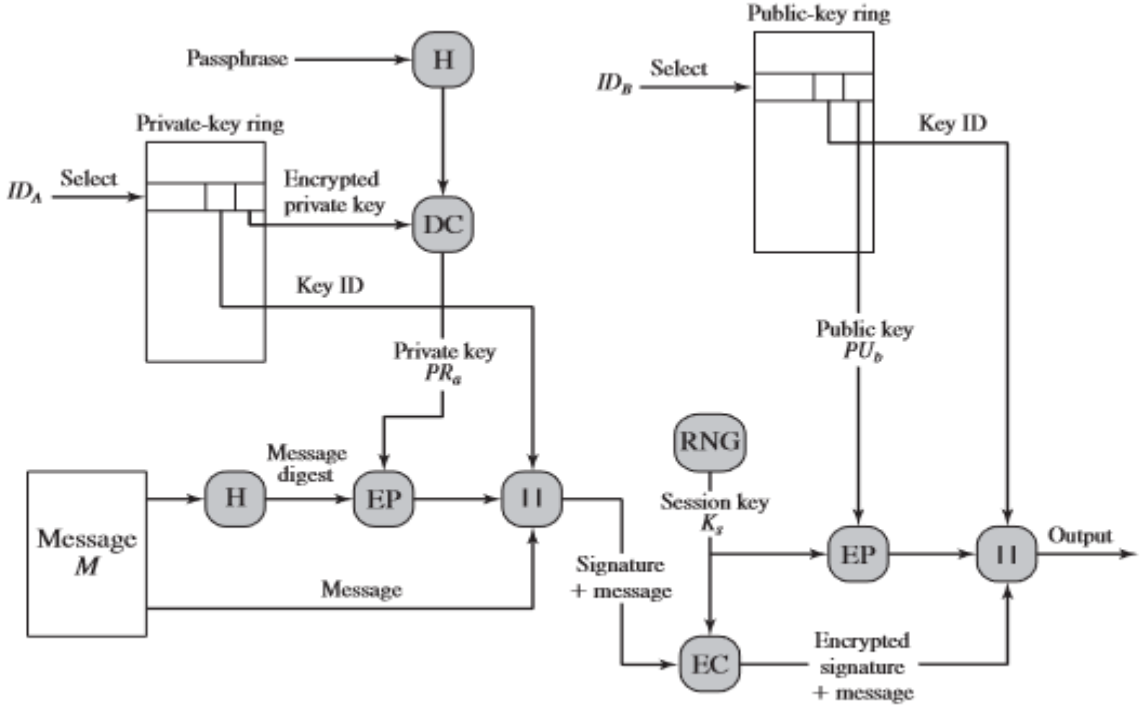


Figure 18.5 PGP Message Generation (from User A to User B: no compression or radix-64 conversion)

1. Signing the message:

a. PGP retrieves the sender's private key from the private-key ring using your user id as an index. If your user id was not provided in the command, the first private key on the ring is retrieved.

b. PGP prompts the user for the passphrase to recover the unencrypted private key.

c. The signature component of the message is constructed.

2. Encrypting the message:

a. PGP generates a session key and encrypts the message.

b. PGP retrieves the recipient's public key from the public-key ring using her user id as an index.

c. The session key component of the message is constructed. The receiving PGP entity performs the following steps **(Figure 18.6).**

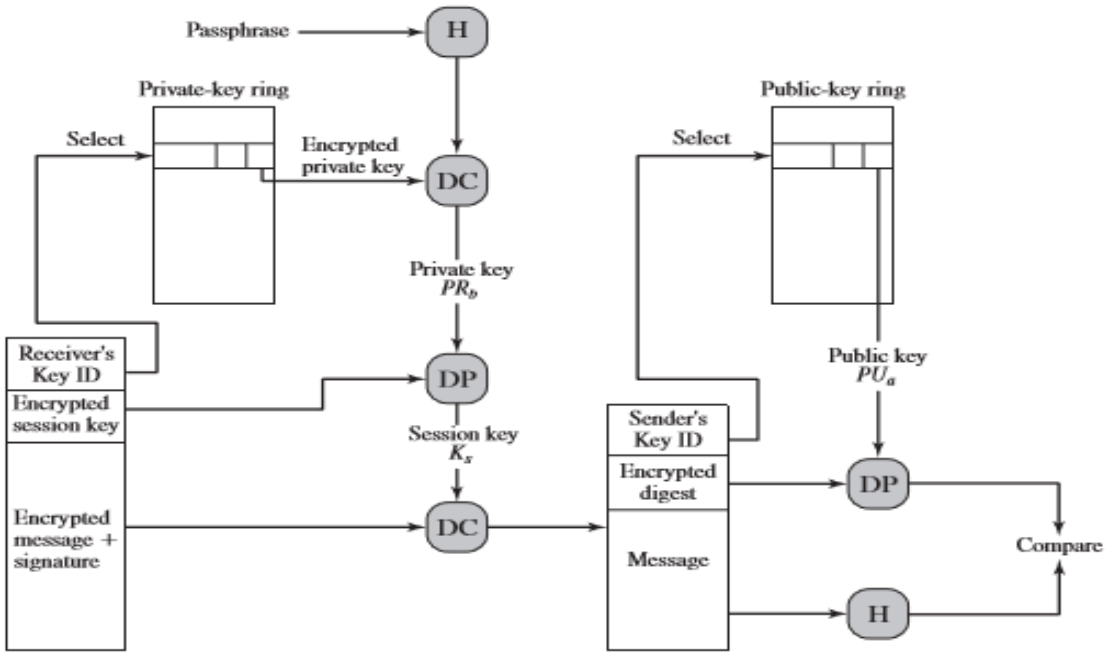


Figure 18.6 PGP Message Reception (from User A to User B; no compression or radix-64 conversion)

1. Decrypting the message:

a. 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.

b. PGP prompts the user for the passphrase to recover the unencrypted private key.

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

SESSION KEY GENERATION

Each session key is associated with a single message and is used only for the purpose of encrypting and decrypting that message. 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. These numbers are based on keystroke input from the user. Both the keystroke timing and the actual keys struck are used to generate the randomized stream. The random input is also combined with previous session key output from CAST-128 to form the key input to the generator. The result, given the effective scrambling of CAST-128, is to produce a sequence of session keys that is effectively unpredictable.

Public-Key Management

APPROACHES TO PUBLIC-KEY MANAGEMENT

The essence of the problem is this: User A must build up a public-key ring containing the public keys of other users to interoperate with them using PGP. Suppose that A's key ring contains a public key attributed to B, but in fact the key is owned by C. The result is that two threats now exist. First, C can send messages to A and forge B's signature so that A will accept the message as coming from B. Second, any encrypted message from A to B can be read by C.

A number of approaches are possible for minimizing the risk that a user's public-key ring contains false public keys.

The following are some approaches that could be used.

- Physically get the key from B.B could store the public key and hand it to A.A could then load the key into his system .This is a very secure method but has obvious practical limitations.
- Verify a key by telephone. If A can recognize B on the phone, A could call B and ask her to dictate the key, in radix-64 format, over the phone. As a more practical alternative, B could transmit her key in an e-mail message to A.
- Obtain B's public key from a mutual trusted individual D. For this purpose, the introducer, D, creates a signed certificate. The certificate includes B's public key, the time of creation of the key, and a validity period for the key. D generates an SHA-1 digest of this certificate, encrypts it with her private key, and attaches the signature to the certificate. Because only D could have created the signature, no one else can create a false public key and pretend that it is signed by D. The signed certificate could be sent directly to A by B or D, or it could be posted on a bulletin board.
- Obtain B's public key from a trusted certifying authority. Again, a public-key certificate is created and signed by the authority. A could then access the authority, providing a user name and receiving a signed certificate

THE USE OFTRUST

Although PGP does not include any specification for establishing certifying authorities or for establishing trust, it does provide a convenient means of using trust, associating trust with public keys, and exploiting trust information.

The basic structure is as follows. Associated with each such entry is a **key legitimacy field** that indicates the extent to which PGP will trust that this is a valid public key for this user; the higher the level of trust, the stronger is the binding of this user ID to this key. This field is computed by PGP. The key legitimacy field is derived from the collection of signature trust fields in the entry. Finally, each entry defines a public key associated with a particular owner, and an owner trust fieldis included that indicates the degree to which this public key is trusted to sign other public-

key certificates; this level of trust is assigned by the user. We can describe the operation of the trust processing as follows.

1. When A inserts a new public key on the public-key ring, PGP must assign a value to the trust flag that is associated with the owner of this public key. If the owner is A, and therefore this public key also appears in the private-key ring, then a value of ultimate trust is automatically assigned to the trust field.

Otherwise, PGP asks A for his assessment of the trust to be assigned to the owner of this key, and A must enter the desired level. The user can specify that this owner is unknown, untrusted, marginally trusted, or completely trusted.

2. When the new public key is entered, one or more signatures may be attached to it. More signatures may be added later. When a signature is inserted into the entry, PGP searches the public-key ring to see if the author of this signature is among the known public-key owners. If so, the **OWNERTRUS**T value for this owner is assigned to the **SIGTRUST** field for this signature. If not, an unknown user value is assigned.

3. The value of the key legitimacy field is calculated on the basis of the signature trust fields present in this entry. If at least one signature has a signature trust value of ultimate, then the key legitimacy value is set to complete. Otherwise, PGP computes a weighted sum of the trust values.

REVOKING PUBLIC KEYS

A user may wish to revoke his or her current public key either because compromise is suspected or simply to avoid the use of the same key for an extended period. The convention for revoking a public key is for the owner to issue a key revocation certificate, signed by the owner. This certificate has the same form as a normal signature certificate but includes an indicator that the purpose of this certificate is to revoke the use of this public key. Note that the corresponding private key must be used to sign a certificate that revokes a public key. The owner should then attempt to distribute this certificate as widely and as quickly as possible to enable potential correspondents to update their public-key rings.

IP SECURITY OVERVIEW

There is a need to secure the networkinfrastructure from unauthorized monitoring and control of network traffic and secure end-user-to-end-user traffic using authentication andencryption mechanisms which the IAB (Internet Architecture Board) deemed asnecessary security features in the next-generation IP i.e IPv6. The IPsec specification now exists as a set of Internetstandards.

Applications of IPsec

The principal feature of IPsec that enables it to support these varied application is that it can encrypt and/or authenticate all traffic at the IP level. Thus, all distributed Applications (including remote logon, client/server, e-mail, file transfer, Web access, and so on) can be secured.

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

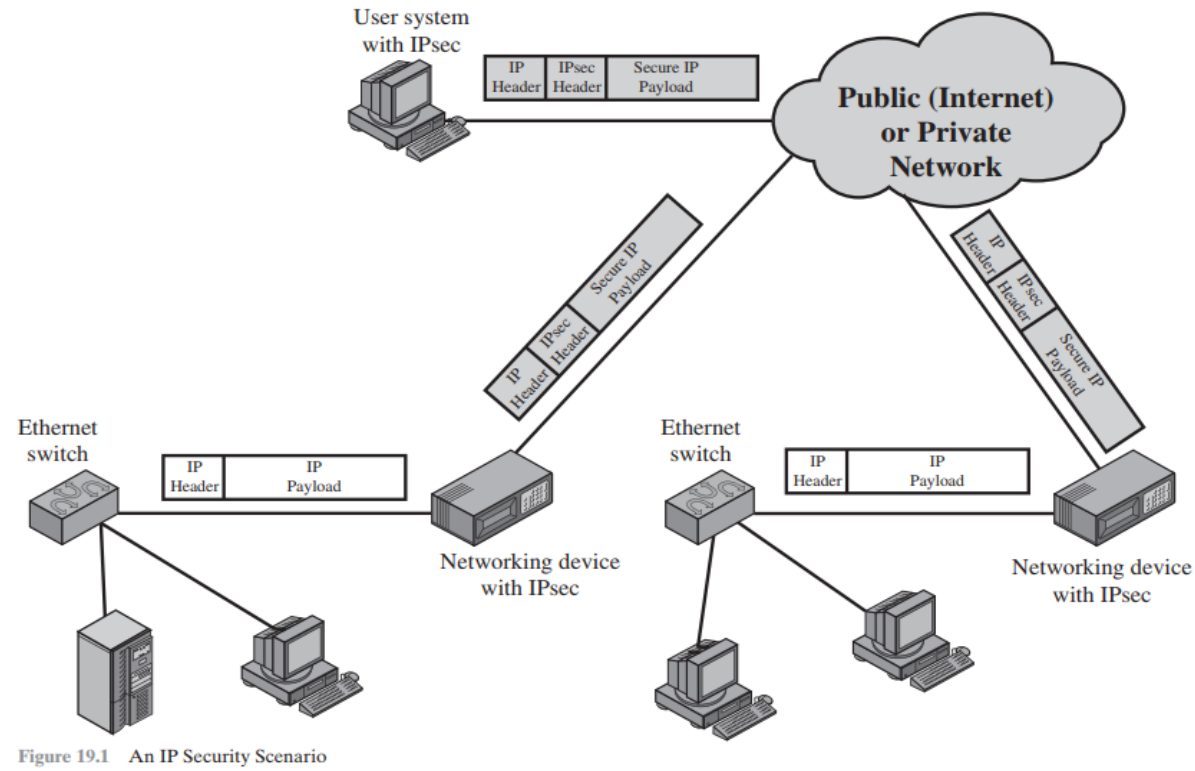


Figure 19.1 An IP Security Scenario

Benefits of IPsec

- Provides strong security that can be applied to all traffic crossing the perimeter when applied in a firewall or router.
- IPsec in a firewall is resistant to bypass if all traffic from the outside must use IP.
- IPsec is below the transport layer (TCP, UDP) and 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.
- Provides security for individual users if needed - useful for offsite workers and for setting up a secure virtual sub network within an organization for sensitive applications.

Routing Applications: IPsec can assure that

- A router advertisement comes from an authorized router.
- A neighbor advertisement comes from an authorized router.
- A redirect message comes from router to which initial IP packet was sent.
- A routing update is not forged. Without such security measures, an opponent can disrupt communications or divert some traffic.

IPsec Documents:

IPsec encompasses three functional areas: authentication, confidentiality, and key management. The totality of the IPsec specification is scattered across dozens of that can be categorized into the following groups:

- **Architecture**: Covers the general concepts, security requirements, definitions, and mechanisms defining IPsec technology.
- **Authentication Header (AH):**is an extension header to provide message authentication.
- **Encapsulating Security Payload (ESP):**consists of an encapsulating header and trailer used to provide encryption or combined encryption/authentication.
- **Internet Key Exchange (IKE):**is a collection of documents describing the key management schemes for use with IPsec.
- **Cryptographic algorithms:**encompasses a large set of documents that define and describe cryptographic algorithms for encryption, message authentication, pseudorandom functions (PRFs), and cryptographic key exchange.
- **Other:**There are a variety of other IPsec-related RFCs, including those dealing with security policy and management information base (MIB) content.

IPsec Services

IPsec provides security services at the IP layer by enabling a system to select required security protocols, determine the algorithms to use for the services, and put in place any cryptographic keys required to provide the requested services. Two protocols are used to provide security:

- **Authentication Header (AH)** - an authentication protocol designated by the header of the protocol
- **Encapsulating Security Payload (ESP)** -a combined encryption/ authentication protocol designated by the format of the packet for that protocol.

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

TRANSPORT MODE:

- Transport mode provides protection primarily for upper-layer protocols which extends to the payload of an IP packet. Example: TCP or UDP segment or an ICMP packet.
- ESP in transport mode encrypts and optionally authenticates the IP payload but not the IP header.
- AH in transport mode authenticates the IP payload and selected portions of the IP header.

TUNNEL MODE:

- Tunnel mode provides protection to the entire IP packet. 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, 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.

Table 19.1 Tunnel Mode and Transport Mode Functionality

	Transport Mode SA	Tunnel Mode SA
AH	Authenticates IP payload and selected portions of IP header and IPv6 extension headers.	Authenticates entire inner IP packet (inner header plus IP payload) plus selected portions of outer IP header and outer IPv6 extension headers.
ESP	Encrypts IP payload and any IPv6 extension headers following the ESP header.	Encrypts entire inner IP packet.
ESP with Authentication	Encrypts IP payload and any IPv6 extension headers following the ESP header. Authenticates IP payload but not IP header.	Encrypts entire inner IP packet. Authenticates inner IP packet.

IP SECURITY POLICY

IPsec policy consists of two databases:

- Security association database (SAD)
- Security policy database (SPD)

Security Associations (SA): An association is a one-way logical connection between a sender and a receiver that affords security services to the traffic carried on it. If a peer relationship is needed for two-way secure exchange, then two security associations are required. Security services are afforded to an SA for the use of AH or ESP, but not both.

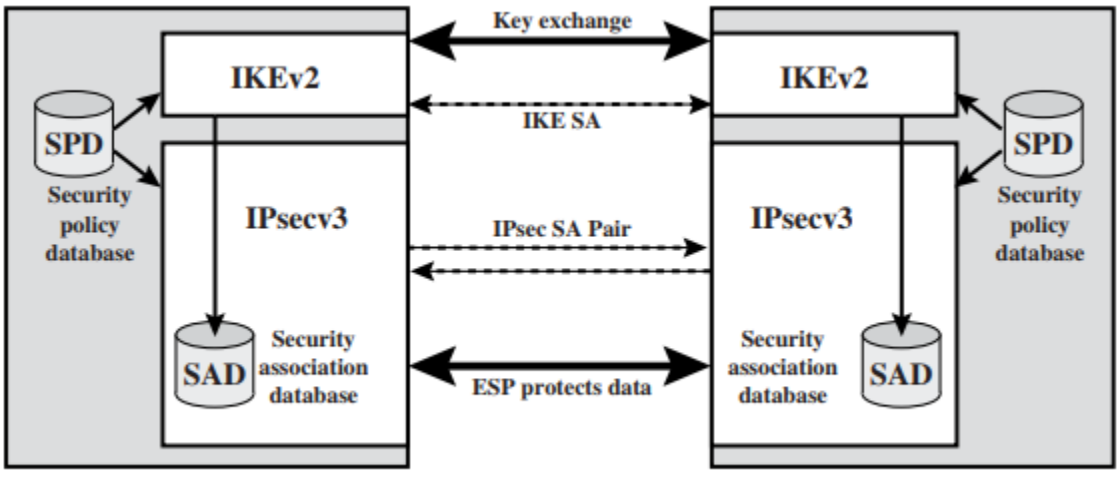


Figure 19.2 IPsec Architecture

A security association is uniquely identified by three parameters.

- **Security Parameters Index (SPI):** A bit string assigned to a SA which has local significance only. Helps thereceiving system to select the SA under which a received packet will beprocessed.
- **IP Destination Address:** address of the destination endpoint of the SA, which may be an end-user system or a network system i.e firewall or router.
- **Security Protocol Identifier**: indicates whether the association is an AH or ESP security association.

Security Association Database (SAD)

A security association is normally defined by the following parameters in an SAD entry.

- **Security Parameter Index:** A 32-bit value selected by the receiving end of an SA to uniquely identify the SA. In an SAD entry for an outbound SA, the SPI is used to construct the packet's AH or ESP header. In an SAD entry for an inbound SA, the SPI is used to map traffic to the appropriate SA.
- **Sequence Number Counter:** A 32-bit value used to generate the Sequence Number field in AH or ESP headers
- **Sequence Counter Overflow:** A flag indicating whether overflow of theSequence Number Counter should generate an auditable event and prevent further transmission of packets on this SA (required for all implementations).
- **Anti-Replay Window:** Used to determine whether an inbound AH or ESP packet is a replay
- **AH Information:** Authentication algorithm, keys, key lifetimes, and related parameters being used with AH (required for AH implementations).
- **ESP Information:** Encryption and authentication algorithm, keys, initialization values, key lifetimes, and related parameters being used with ESP(required for ESP implementations).
- **Lifetime of this Security Association:** A time interval or byte count afterwhich an SA must be replaced with a new SA (and new SPI) or terminated,plus an indication of which of these actions should occur (required for allimplementations).
- **IPsec Protocol Mode**: Tunnel, transport, or wildcard.
- **Path MTU**: Any observed path maximum transmission unit (maximum sizeof a packet that can be transmitted without fragmentation) and agingvariables (required for all implementations).

Security Policy Database (SPD)

An SPD contains entries, each of which defines a subset of IP traffic andpoints to an SA for that traffic.Each SPD entry is defined by a set of IP and upper-layer protocol field values,called selectors. In effect, these selectors are used to filter outgoing traffic in order to map it into a particular SA.

Outbound processing obeys the following general sequence for each IP packet.

1. Compare the values of the appropriate fields in the packet (the selector fields)against the SPD to find a matching SPD entry, which will point to zero or more SAs.
2. Determine the SA if any for this packet and its associated SPI.
3. Do the required IPsec processing (i.e., AH or ESP processing).

The following selectors determine an SPD entry:

- **Remote IP Address**: may be a single IP address, an enumerated list or range of addresses, or a wildcard (mask) address.
- **Local IP Address:** may be a single IP address, an enumerated list or range of addresses, or a wildcard (mask) address
- **Next Layer Protocol**: The IP protocol header includes a field that designates the protocol operating over IP. If AH or ESP is used, then this IP protocol header immediately precedes the AH or ESP header in the packet.
- **Name:** A user identifier from the operating system.
- **Local and Remote Ports:** These may be individual TCP or UDP port values, an enumerated list of ports, or a wildcard port.

Table 19.2 Host SPD Example

Protocol	Local IP	Port	Remote IP	Port	Action	Comment
UDP	1.2.3.101	500	*	500	BYPASS	IKE
ICMP	1.2.3.101	*	*	*	BYPASS	Error messages
*	1.2.3.101	*	1.2.3.0/24	*	PROTECT: ESP intransport-mode	Encrypt intranet traffic
TCP	1.2.3.101	*	1.2.4.10	80	PROTECT: ESP intransport-mode	Encrypt to server
TCP	1.2.3.101	*	1.2.4.10	443	BYPASS	TLS: avoid double encryption
*	1.2.3.101	*	1.2.4.0/24	*	DISCARD	Others in DMZ
*	1.2.3.101	*	*	*	BYPASS	Internet

IP Traffic Processing

IPsec is executed on a packet-by-packet basis. When IPsec is implemented, each outbound IP packet is processed by the IPsec logic before transmission, and each inbound packet is processed by the IPsec logic after reception and before passing the packet contents on to the next higher layer (e.g., TCP or UDP).

OUTBOUND PACKETS

A block of data from a higher layer, such as TCP, is passed down to the IP layer and an IP packet is formed, consisting of an IP header and an IP body. Then the following steps occur:

1. IPsec searches the SPD for a match to this packet.
2. If no match is found, then the packet is discarded and an error message is generated.
3. If a match is found, further processing is determined by the first matching entry in the SPD. If the policy for this packet is DISCARD, then the packet is discarded. If the policy is BYPASS, then there is no further IPsec processing; the packet is forwarded to the network for transmission.
4. If the policy is PROTECT, then a search is made of the SAD for a matching entry. If no entry is found, then IKE is invoked to create an SA with the appropriate keys and an entry is made in the SA.
5. The matching entry in the SAD determines the processing for this packet. Either encryption, authentication, or both can be performed, and either transport or tunnel mode can be used. The packet is then forwarded to the network for transmission.

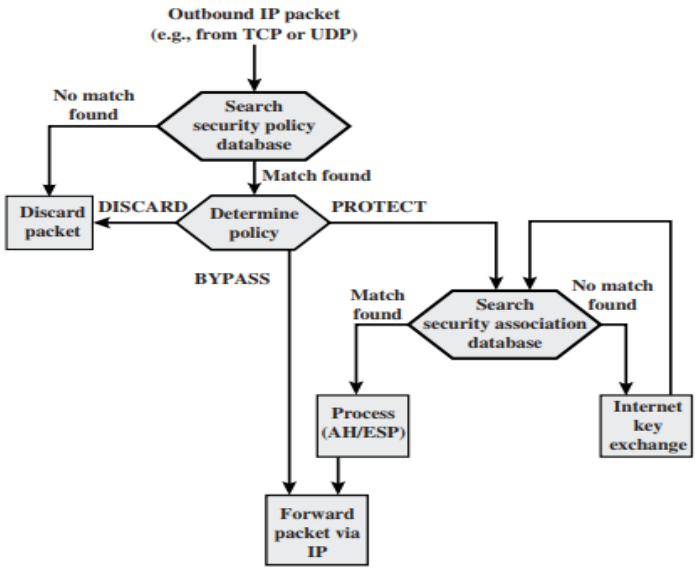


Figure 19.3 Processing Model for Outbound Packets

INBOUND PACKETS

An incoming IP packet triggers the IPsec processing. The following steps occur:

1. IPsec determines whether this is an unsecured IP packet or one that has ESP or AH headers/trailers, by examining the IP Protocol field (IPv4) or Next Header field (IPv6).
2. If the packet is unsecured, IPsec searches the SPD for a match to this packet.If the first matching entry has a policy of BYPASS, the IP header is processed and stripped off and the packet body is delivered to the next higher layer, such as TCP. If the first matching entry has a policy of PROTECT or DISCARD, or if there is no matching entry, the packet is discarded.
3. For a secured packet, IPsec searches the SAD. If no match is found, the packet is discarded. Otherwise, IPsec applies the appropriate ESP or AH processing. Then, the IP header is processed and stripped off and the packet body is delivered to the next higher layer, such as TCP.

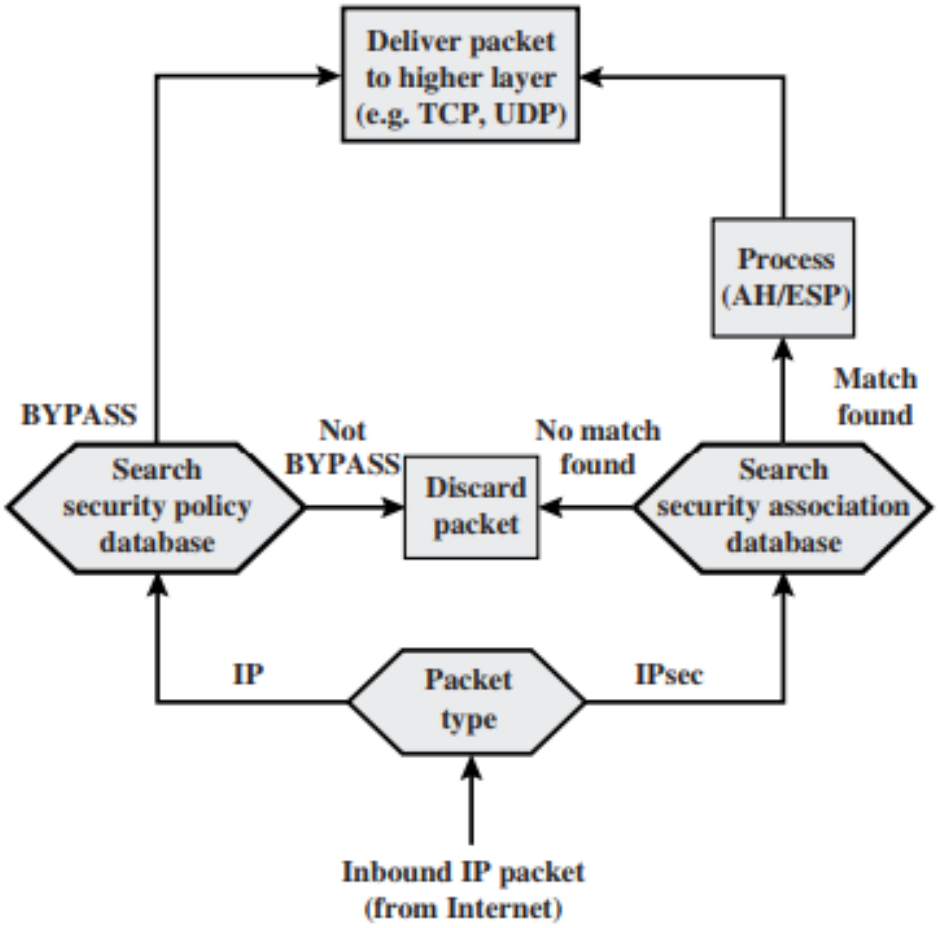


Figure 19.4 Processing Model for Inbound Packets

MALICIOUS SOFTWARE

Malicious Software

Malicious software or malware is software that is intentionally included or inserted in a system for a harmful purpose.

Types of Malicious Software

1.Parasitic Malware or Dependent Malware

Parasitic malware, 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

2.Independent Malware

Independent malware is a self-contained program that can be scheduled and run by the operating system. Worms and bot programs are examples.

Looking At Each Malware In Detail

Backdoor

Any mechanism that bypasses a normal security check; it may allow unauthorized access to functionality

Logic Bomb

A program inserted into software by an intruder. A logic bomb lies dormant until a predefined condition is met; the program then triggers an unauthorized act.

Spyware

Software that collects information from a computer and transmits it to another system

Adware

Advertising that is integrated into software. It can result in pop-up ads or redirection of a browser to a commercial site.

Trojan War

A computer program that appears to have a useful function, but also has a hidden and potentially malicious function that evades security mechanisms, sometimes by exploiting legitimate authorizations of a system entity that invokes the Trojan horse program

Mobile Code

Software (e.g., script, macro, or other portable instruction) that can be shipped unchanged to a heterogeneous collection of platforms and execute with identical semantics.

Worm : A computer program that can run independently and can propagate a complete working version of itself onto other hosts on a network.

VIRUSES

Definition

A computer virus is a piece of software that can "infect" other programs by modifying them; the modification includes injecting the original program with a routine to make copies of the virus program, which can then go on to infect other programs

A computer virus has three parts :

• Infection mechanism: The means by which a virus spreads, enabling it to replicate. The mechanism is also referred to as the infection vector.

• Trigger: The event or condition that determines when the payload is activated or delivered.

• Payload: What the virus does, besides spreading. The payload may involve damage or may involve benign but noticeable activity

Sample Virus Code

```
program V :=

{

goto main;

 1234567;

 subroutine infect-executable :=
```

```
{loop:
file := get-random-executable-file;
if (first-line-of-file = 1234567)
then goto loop
else prepend V to file;
}
subroutine do-damage :=
{whatever damage is to be done}
subroutine trigger-pulled := {return true if some condition holds}

main: main-program :=
{infect-executable;
if trigger-pulled then do-damage;
goto next;}
next: }
```

Counter Measures

Antivirus Approach

Any good antivirus software must possess the following features

Detection: Once the infection has occurred, determine that it has occurred and locate the virus

• **Identification:** Once detection has been achieved, identify the specific virus that has infected a program.

• **Removal:** Once the specific virus has been identified, remove all traces of the virus from the infected program and restore it to its original state. Remove the virus from all infected systems so that the virus cannot spread further.

Advanced Antivirus Approach

Generic Decryption:
Generic decryption (GD) technology enables the antivirus program to easily detect even the most complex polymorphic viruses while maintaining fast scanning speeds . It contains the following elements:

• **CPU emulator:** A software-based virtual computer. Instructions in an executable file are interpreted by the emulator rather than executed on the underlying processor. The emulator includes software versions of all registers and other processor hardware, so that the underlying processor is unaffected by programs interpreted on the emulator.

• **Virus signature scanner:** A module that scans the target code looking for known virus signatures.

• **Emulation control module:** Controls the execution of the target code.

FIREWALLS

- A firewall forms a barrier through which the traffic going in each direction must pass. A firewall security policy dictates which traffic is authorized to pass in each direction.
- A firewall may be designed to operate as a filter at the level of IP packets, or may operate at a higher protocol layer.

Firewalls can be an effective means of protecting a local system or network of systems from network-based security threats while at the same time affording access to the outside world via wide area networks and the Internet.

The need for Firewalls

The computers in the organisations or in our houses are a small electronic devices that do a countable services. They have undergone a steady evolution.

• Centralized data processing system, with a central mainframe supporting a number of directly connected terminals

• Local area networks (LANs) interconnecting PCs and terminals to each other and the mainframe

• Premises network, consisting of a number of LANs, interconnecting PCs, servers, and perhaps a mainframe or two

• Enterprise-wide network, consisting of multiple, geographically distributed premises networks interconnected by a private wide area network (WAN)

• Internet connectivity, in which the various premises networks all hook into the Internet and may or may not also be connected by a private WAN

The point above state that systems have come to a point where we can do most of the things like sharing or manipulating information which might not always be secure when there is

connectivity to and fro to your system. Firewall is a filter that stops unwanted network from passing through it.

Characteristics of Firewalls

Design goals of firewall.

1. All the traffic to and from the system should pass through a firewall. This is achieved by physically blocking all access to the local network except the firewall.
2. Only authorized traffic, as defined by the local security policy, will be allowed to pass the firewall.
3. The firewall itself is immune to penetration. Trusted computer systems are suitable for hosting a firewall and often required in government applications

The techniques that firewall uses to control access and enforce the site security policy.

1. Service control- Determines the types of Internet services that can beaccessed, inbound or outbound. Applies filtering based on services requested inbound or outbound.
2. Direction control- Determines the direction in which particular service requests may be initiated and allowed to flow through the firewall
3. User control- Controls access to a service according to which user is attempting to access it.
4. Behaviour control- Controls how particular services are used. For example, the firewall may filter e-mail to eliminate spam, or it may enable external access to only a portion of the information on a local Web server.

Limitations of firewalls.

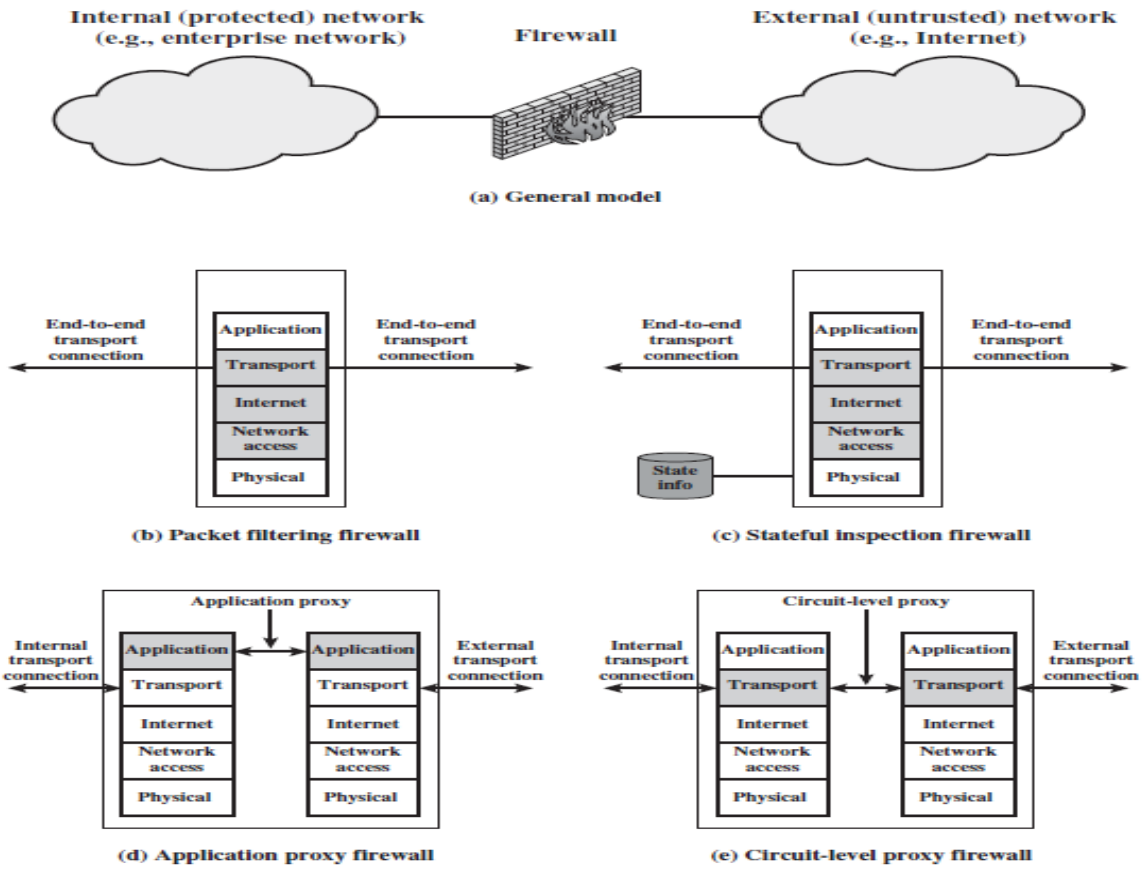
1. The firewall cannot protect against attacks that bypass the firewall.
2. The firewall may not protect fully against internal threats, such as a disgruntled employee or an employee who unwittingly cooperates with an external attacker.
3. A laptop, PDA, or portable storage device may be used and infected outsidethe corporate network, and then attached and used internally.
4. The firewall cannot guard against wireless communications between local systems on different sides of the internal firewall.

Types of Firewall

Depending on the type of firewall, it may examine one or more protocol headers in each packet. It can act as either positive or negative filter, i.eallowing only the packets that meet the criterion or vice versa.
Given below are some principal types of firewalls.

- Packet filtering firewall
- Stateful inspection firewall

- Application proxy firewall
- Circuit level proxy firewall



Packet filtering firewall-

Packet filtering firewall applies a set of rules to each incoming and outgoing IP packet and then forwards or discards the packet. Filtering rules are based on information contained in a network packet; source IP address, Destination IP address, source and destination transport-level address, IP protocol field, Interface.

The packet filter is typically set up as a list of rules based on matches to fieldsin the IP or TCP header. This is a policy likely to be preferred by businesses and government organizations. The one advantage of a packet filtering firewall is its simplicity. Also , packet filters typically are transparent to users and are very fast.

Weakness of packet filter firewall:

- Because packet filter firewalls do not examine upper-layer data, they cannotprevent attacks that employ application-specific vulnerabilities or functions.
- Because of the limited information available to the firewall, the logging functionality present in packet filter firewalls is limited.
- Most packet filter firewall do not support advanced user authentication schemes.
- Due to the small number of variables used in access control decisions, packet filter firewalls are susceptible to security breaches caused by improper configurations.

Stateful Inspection Firewall-

Stateful inspection, also known as dynamic packet filtering, is a firewall technology that monitors the state of active connections and uses this information to determine which network packets to allow through the firewall.

Stateful inspection has largely replaced an older technology, static packet filtering. In static packet filtering, only the headers of packets are checked -- which means that an attacker can sometimes get information through the firewall simply by indicating "reply" in the header. Stateful inspection, on the other hand, analyses packets down to the application layer. By recording session information such as IP addresses and port numbers, a dynamic packet filter can implement a much tighter security posture than a static packet filter can.

Stateful inspection monitors communications packets over a period of time and examines both incoming and outgoing packets. Outgoing packets that request specific types of incoming packets are tracked and only those incoming packets constituting a proper response are allowed through the firewall.

In a firewall that uses stateful inspection, the network administrator can set the parameters to meet specific needs. In a typical network, ports are closed unless an incoming packet requests connection to a specific port and then only that port is opened. This practice prevents port scanning, a well-known hacking technique.

Application level gateway-

An application-level gateway, also called an application proxy, 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 applicationon the remote host and relays TCP segments containing the application databetween the two endpoints. If the gateway does not implement the proxy code for a specific application, the service is not supported and cannot be forwarded across the firewall. Further, the gateway can be configured to support only specific features of an application that the network administrator considers acceptable while denying all other features.

Application-level gateways tend to be more secure than packet filters. A prime disadvantage of this type of gateway is the additional processing overhead on each connection.

Circuit-level Gateway-

This kind of firewall is a stand-alone system or it can be a specialized function performed by an application-level gateway for certain applications. As with an application gateway, 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 outside 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 administratortrusts the internal users.The gateway can be configured to support application-level or proxy service on inbound connections and circuit-level functions foroutbound connections. In this configuration, the gateway can incur the processingoverhead of examining incoming application data for forbidden functions but doesnot incur that overhead on outgoing data.