UNIT II
E-MAIL SECURITY & FIREWALLS

PGP – S/MIME – Internet Firewalls for Trusted System: Roles of Firewalls - Firewall related terminology- Types of Firewalls – Firewall designs – SET for E-Commerce Transactions.

Part-A

1.What is application level gateway?

An application level gateway also called a proxy server; act 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.

2. List the design goals of firewalls?

1. All traffic from inside to outside, and vise versa, must pass through the firewall.

2. Only authorized traffic, as defined by the local security policy, will be allowed to pass.

3. The firewall itself is immune to penetration.

3.What is mean by SET? What are the features of SET?

Secure Electronic Transaction (SET) is an open encryption and security specification designed to protect credit card transaction on the internet. Features are:

1. Confidentiality of information

2. Integrity of data

3. Cardholder account authentication

4. Merchant authentication

4. What are the steps involved in SET Transaction?

1. The customer opens an account

2. The customer receives a certificate

3. Merchants have their own certificate

4. The customer places an order.

5. The merchant is verified.

6. The order and payment are sent.

7. The merchant requests payment authorization.

8. The merchant confirm the order.

9. The merchant provides the goods or services.

10. The merchant requests payment.

5.Define S/MIME?

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

6. What are the headers fields define in MIME?

MIME version. Content type.

Content transfer encoding. Content id.

Content description.

7. What is MIME content type and explain?

It is used to declare general type of data. Subtype define particular format for that type of the data. It has 7 content type & 15 subtypes. They are,

1. Text type Plain text. Enriched.

2. Multipart type

Multipart/mixed. Multipart/parallel. Multipart/alternative. Multipart/digest.

3. Message type

Message/RFC822. Message/partial. Message/external.

4. Image type

JPEG. CIF.

5. Video type.

6. Audio type.

7. Application type

Post script. Octet stream.

8. What are the key algorithms used in S/MIME?

1.Digital signature standards.

2.Diffi Hellman.

3.RSA algorithm.

9. Give the steps for preparing envelope data MIME?

1. Generate Ks.
2. Encrypt Ks using recipient' s public key. RSA algorithm used for encryption. Prepare the 'recipient info block' .
3. Encrypt the message using Ks.

10.What are the services provided by PGP services

- Digital signature Message encryption Compression
- E-mail compatibility
- Segmentation

11. Explain the reasons for using PGP?

a) It is available free worldwide in versions that run on a variety of platforms, including DOS/windows, UNIX, Macintosh and many more.

b) 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, CAST-128, IDEA, 3DES for conventional encryption, SHA-1for hash coding.

c) It has a wide range of applicability from corporations that wish to select and enforce a standardized scheme for encrypting files and communication.

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

12. Why E-mail compatibility function in PGP needed?

Electronic mail systems only permit the use of blocks consisting of ASCII text. To accommodate this restriction PGP provides the service converting the row 8- bit binary stream to a stream of printable ASCII characters. The scheme used for this purpose is Radix-64 conversion.

13. Name any cryptographic keys used in PGP?

a) One-time session conventional keys. b) Public keys.

c) Private keys.

d) Pass phrase based conventional keys.

14.What is meant by S/MIME? (A/M-12)

S/MIME (Secure/Multipurpose Internet Mail Extensions) is a standard for public key encryption and signing of MIME data. S/MIME is on an IETF standards track and defined in a number of documents, most importantly RFCs (3369, 3370, 3850, 3851). S/MIME was originally developed by RSA Data Security Inc. The original specification used the IETF MIME specification with the de facto industry standard PKCS secure message format. Change control to S/MIME has since been vested in the IETF and the specification is now layered on cryptographic message syntax.

15.List out the types of firewalls.

1. Packet Filters
2. Circuit-Level Gateways
3. Application-Level Gateways

Part-B

1.Explain in detail about the PGP.

- PGP used data encryption software that ensures integrity, security, and privacy of data and messages sent over the internet.
- It uses two digital equivalents of physical keys: a public key used for 'locking' (encrypting) data that can be given by its owner to anyone who wants to send a secure transmission; and a private key used for 'unlocking' (decrypting) the data and known only to its owner.
- It is also used to digitally 'sign' an electronic document, thus authenticating its origin.

1.1 Confidentiality via Encryption

PGP provides confidentiality by encrypting messages to be transmitted or data files to be stored locally using a conventional encryption algorithm such as IDEA, 3DES or CAST- 128. In PGP, each symmetric key, known as a session key, is used only once. A new session key is generated as a random 128-bit number for each message. Since it is used only once, the session key is bound to the message and transmitted with it. To protect the key, it is encrypted with the receiver's public key. Figure 9.1 illustrates the sequence, which is described as follows:

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

- The session key is encrypted with RSA, using the recipient's public key.
- The sending PGP encrypts the message, using CAST-128 or IDEA or 3DES, with the session key. Note that the message is also usually compressed.
- The receiving PGP uses RSA with its private key to decrypt and recover the session key.
- The receiving PGP decrypts the message using the session key. If the message was compressed, it will be decompressed.

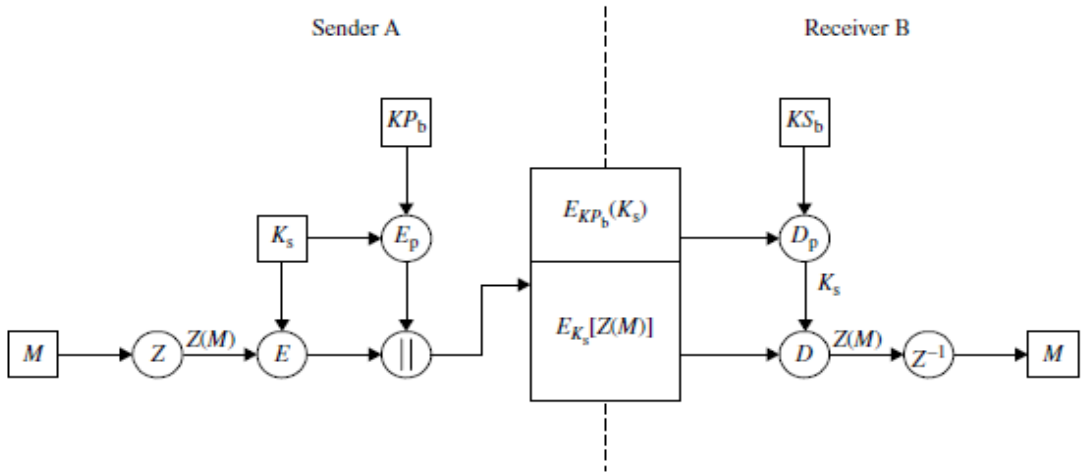


Figure 9.1 PGP confidentiality computation scheme with compression/decompression Algorithms.

Both digital signature and confidentiality services may be applied to the same message. First, a signature is generated from the message and attached to the message. Then the message plus signature are encrypted using a symmetric session key. Finally, the session key is encrypted using public-key encryption and prefixed to the encrypted block.

1.2 Authentication via Digital Signature

The digital signature uses a hash code of the message digest algorithm, and a public-key signature algorithm. Figure 9.2 illustrates the digital signature service provided by PGP.

The sequence is as follows:

- The sender creates a 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 a digital signature is produced.
- The binary signature is attached 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 received message and compares it with the decrypted hash code. If the two match, the message is accepted as authentic.

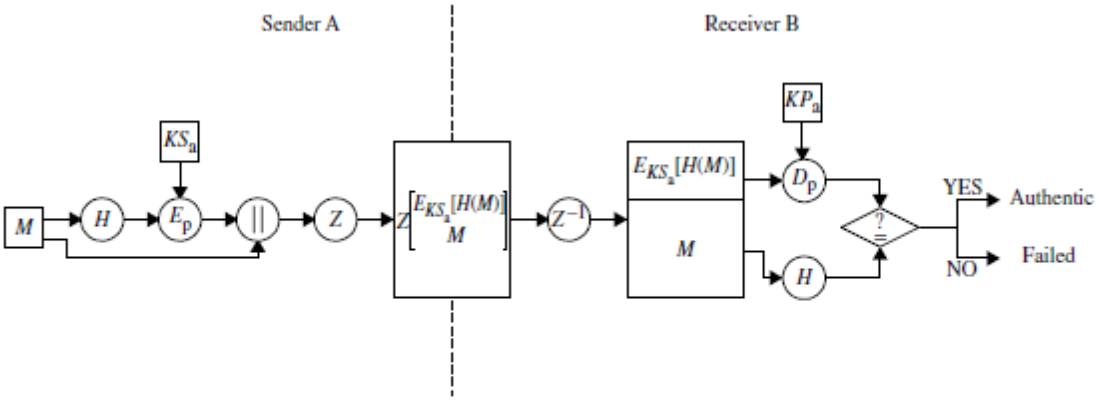


Figure 9.2 PGP authentication computation scheme using compression algorithm.

The combination of SHA-1 and RSA provides an effective digital signature scheme.
As an alternative, signatures can be generated using DSS/SHA-1.

1.3 Compression

- ✓ PGP compresses the message after applying the signature but before encryption.
- ✓ The placement of Z for compression and Z^{-1} for decompression is shown in Figures 9.1 and 9.2.
- ✓ This compression algorithm has the benefit of saving space both for e-mail transmission and for file storage.
- ✓ PGP makes use of a compression package called ZIP which is functionally equivalent to PKZIP developed by PKWARE, Inc. The zip algorithm is perhaps the most commonly used cross-platform compression technique.

Two main compression schemes, named after Abraham Lempel and Jakob Ziv, were

first proposed by them in 1977 and 1978, respectively. These two schemes for text compression (generally referred to as lossless compression) are broadly used because they are easy to implement and also fast.

- ✓ Huffman compression is a statistical data compression technique which reduces the average code length used to represent the symbols of an alphabet. Huffman code is an example of a code which is optimal when all symbols probabilities are integral powers of 1/2.
- ✓ A technique related to Huffman coding is Shannon–Fano coding. This coding divides the set of symbols into two equal or almost equal subsets based on the probability of occurrence of characters in each subset.
- ✓ The first subset is assigned a binary 0, the second a binary 1. Huffman encoding always generates optimal codes, but Shannon–Fano sometimes uses a few more bits.
- ✓ Decompression of LZ77-compressed text is simple and fast. Whenever a (position, length) pair is encountered, one goes to that *position* in that window and copies *length* bytes to the output.

1.4 Radix-64 Conversion

- ✓ When PGP is used, usually part of the block to be transmitted is encrypted. If only the signature service is used, then the message digest is encrypted (with the sender's private key).
- ✓ If the confidentiality service is used, the message plus signature (if present) are encrypted (with a one-time symmetric key).

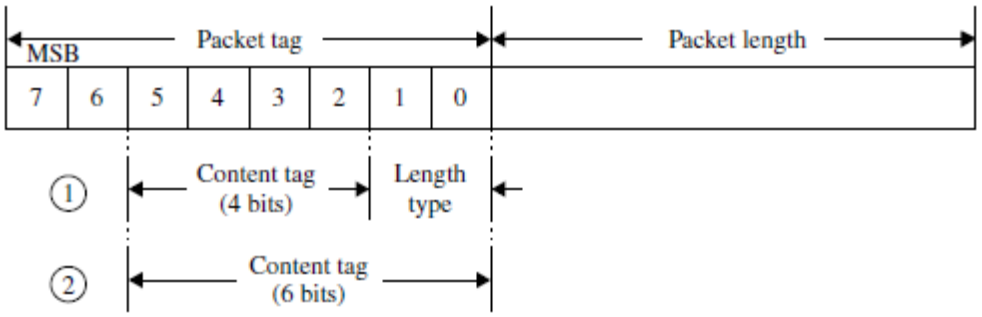
The scheme used for this purpose is radix-64 conversion. Each group of three octets of binary data is mapped into four ASCII characters. This format also appends a CRC to detect transmission errors. This radix-64 conversion is a wrapper around the binary PGP messages, and is used to protect the binary messages during transmission over non-binary channels, such as Internet e-mail.

1.5 Packet Headers

- ➢ A PGP message is constructed from a number of packets. A packet is a chunk of data which has a tag specifying its meaning. Each packet consists of a packet header of variable length, followed by the packet body.
- ➢ The first octet of the packet header is called the packet tag as shown in Figure 9.4. The MSB is 'bit 7' (the leftmost bit) whose mask is 0x80 (10000000) in hexadecimal. PGP 2.6.x only uses old format packets.

1.5.1 Packet Tags

The packet tag denotes what type of packet the body holds. The defined tags (in decimal) are:



(1) Old format packets: content tag (bits 5, 4, 3, 2); length type (bits 1,0)

(2) New format packets: content tag (bits 5, 4, 3, 2, 1, 0)

Figure 9.4 Packet header.

0–Reserved

1–Session key packet encrypted by public key

2–Signature packet

3–Session key packet encrypted by symmetric key

4–One-pass signature packet

5–Secret-key packet

6–Public-key packet

7–Secret-subkey packet

8–Compressed data packet

9–Symmetrically encrypted data packet

10–Marker packet

11–Literal data packet

12–Trust packet

13–User ID packet

14–Public subkey packet

60 ~ 63–Private or experimental values

1.6 PGP Packet Structure

A PGP file consists of a message packet, a signature packet and a session key packet.

1.6.1 Message Packet

This packet includes the actual data to be transmitted or stored as well as a header that includes control information generated by PGP such as a filename and a timestamp. A timestamp specifies the time of creation. The message component consists of a single literal data packet.

1.6.2.Signature Packet (Tag 2)

This packet describes a binding between some public key and some data. The most common signatures are a signature of a file or a block of text, and a signature that is a certification of a user ID.

Two versions of signature packets are defined. PGP 2.6.x only accepts version 3 signature. Version 3 provides basic signature information, while version 4 provides an expandable format with sub packets that can specify more information about the signature. It is reasonable to create a v3 signature if an implementation is creating an encrypted and signed message that is encrypted with a v3 key.

The signature includes the following components:

- ***Timestamp***
- ***Message digest (or hash code)***
- ***Leading two octets of hash code***
- ***Key ID of sender's public key***

Session Key Packets (Tag 1)

- This component includes the session key and the identifier of the receiver's public key that was used by the sender to encrypt the session key.
- A public-key-encrypted session key packet, *EKP*b (*K*s), holds the session key used to encrypt a message.
- The symmetrically encrypted data packets are preceded by one public-key-encrypted session key packet for each PGP 5.x key to which the message is encrypted.
- The message is encrypted with the session key, and the session key is itself encrypted and stored in the encrypted session key packet. The recipient of the message finds a session key that is encrypted to its public key, decrypts the session key, and then uses the session key to decrypt the message.

Table 9.2 Signature packet format of version 3 and version 4

Content	Length in octets	
	V3	V4
Version number: V3(3), V4(4)	1	1
Signature type	1	1
Creation time	4	
Signer's key ID	8	
Public-key algorithm	1	1
Hash algorithm	1	1
Field holding left 16 bits of signed hash value	2	2
One or more MPIs comprising the signature	Algorithm specific*	Algorithm specific
Scalar octet count for hashed subpacket data		2
Hashed subpacket data		Zero or more subpackets
Scalar octet count for all of the unhashed subpackets		2
Unhashed subpacket data		Zero or more subpackets

The body of this session key component consists of:

- ✓ A one-octet version number which is 3.
- ✓ An eight-octet key ID of the public key that the session key is encrypted to.
- ✓ A one-octet number giving the public key algorithm used.
- ✓ A string of octets that is the encrypted session key.

This string's contents are dependent on the public-key algorithm used:

– Algorithm-specific fields for RSA encryption: multiprecision integer (MPI) of RSA encrypted value *me*-mod *n*.

– Algorithm-specific fields for ElGamal encryption: MPI of ElGamal value *gk* mod *p*; MIP of ElGamal value *myk* mod *p*. The value '*m*' is derived from the session key.

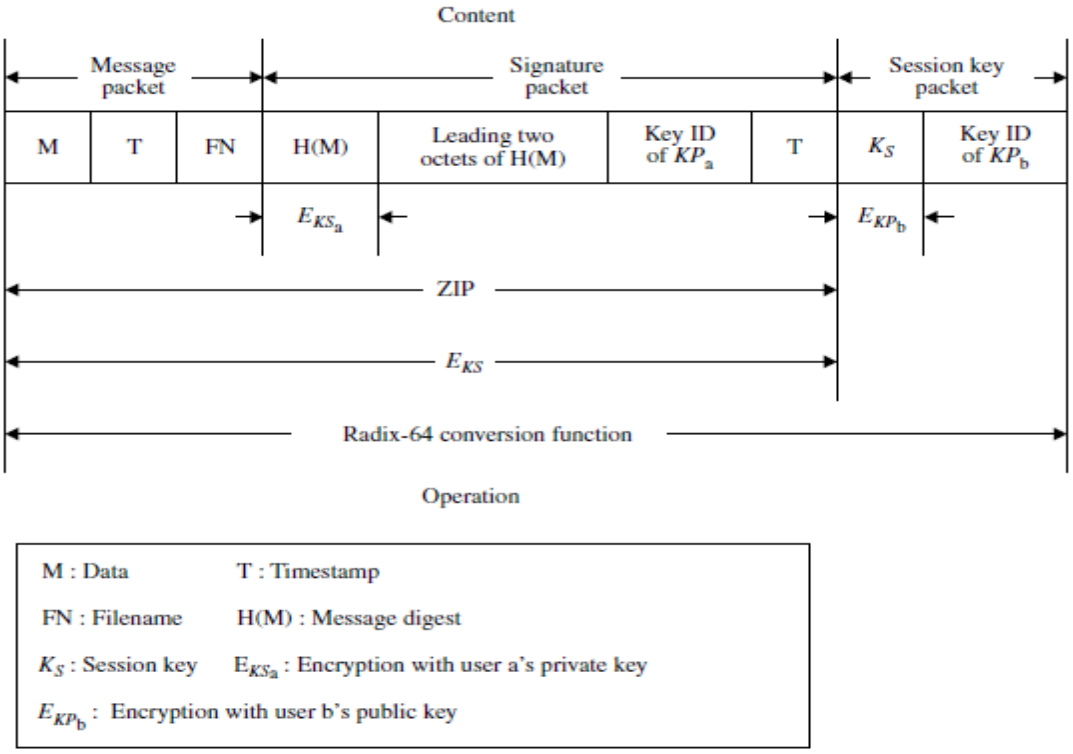


Figure 9.5 PGP message format.

Key Material Packet

A key material packet contains all the information about a public or private key. There are four variants of this packet type and two versions.

Key Packet Variants

There are:

· *Public-key packet (tag 6)*: This packet starts a series of packets that forms a PGP 5.x key.

· *Public subkey packet (tag 14)*: This packet has exactly the same format as a publickey packet, but denotes a subkey. One or more subkeys may be associated with a top-level key.

· *Secret-key packet (tag 5)*: This packet contains all the information that is found in a public-key packet, including the public-key materials, but also includes the secret-key material after all the public-key fields.

· *Secret-subkey packet (tag 7)*: A secret-subkey packet is the subkey analogous to the secret-key packet and has exactly the same format.

Public-key Packet Formats

There are two variants of version 3 packets and version 2 packets. Version 3 packets were originally generated by PGP 2.6. Version 2 packets are identical in format to version 3 packets, but are generated by PGP 2.5.

A v3 key packet contains:

· A one-octet version number (3).

· A four-octet number denoting the time that the key was created.

· A two-octet number denoting the time in days that this key is valid.

· A one-octet number denoting the public-key algorithm of this key.

· A series of multiprecision integers (MPIs) comprising the key material: an MPI of RSA public module *n*; an MPI of RSA public encryption exponent *e*.

Secret-key Packet Formats

The secret-key and secret-subkey packets contain all the data of public-key and publicsubkey packets in encrypted form, with additional algorithm-specific key data appended.

The secret-key packet contains:

· A public-key or public-subkey packet, as described above.

· One octet indicating string-to-key (S2K) usage conventions: 0 indicates that the secretkey data is not encrypted; 255 indicates that an S2K specifier is being given. Any other value specifies a symmetric-key encryption algorithm.

· If the S2K usage octet was 255, a one-octet symmetric encryption algorithm (optional).

· If the S2K usage octet was 255, an S2K specifier (optional). The length of the S2K specifier is implied by its type, as described above.

· If secret data is encrypted, an eight-octet IV (optional).

1.8 Algorithms for PGP 5.x

9.1.8.1 Public-Key Algorithms

ID	Algorithm
1	RSA (encrypt or sign)
2	RSA encryption only
3	RSA sign only
16	ElGamal (encrypt only)
17	DSA (DSS)
18	Reserved for elliptic curve
19	Reserved for ECDSA
20	ElGamal (encrypt or sign)
21	Reserved for Diffie–Hellman
100–110	Private/experimental algorithm

9.1.8.2 Symmetric-Key Algorithms

ID	Algorithm
0	Plaintext or unencrypted data
1	IDEA
2	Triple DES (DES–EDE)
3	CAST 5 (128-bit key)
4	Blowfish (128-bit key, 16 rounds)
5	SAFER-SK128 (13 rounds)
6	Reserved for DES/SK

ID	Algorithm
7	Reserved for AES (128-bit key)
8	Reserved for AES (192-bit key)
9	Reserved for ASE (256-bit key)
100–110	Private/experimental algorithm

9.1.8.3 Compression Algorithm

ID	Algorithm
0	Uncompressed
1	ZIP (RFC 1951)
2	ZLIB (RFC 1950)
100–110	Private/experimental algorithm

9.1.8.4 Hash Algorithms

ID	Algorithm
1	MD5
2	SHA-1
3	RIPE-MD/160
4	Reserved for double-width SAH (experimental)
5	MD2
6	Reserved for TIGER/192
7	Reserved for HAVAL (5 pass, 160-bit)
100–110	Private/experimental algorithm

These tables are not an exhaustive list. An implementation may utilise an algorithm not on these lists.

2.Explain in detail about the MIME.

- Secure/Multipurpose Internet Mail Extension (S/MIME) provides a consistent means to send and receive secure MIME data.
- S/MIME, based on the Internet MIME standard, is a security enhancement to cryptographic electronic messaging.
- Further, S/MIME not only is restricted to e-mail, but can be used with any transport mechanism that carries MIME data, such as HTTP.

MIME

- MIME was defined to allow transmission of non-ASCII data through e-mail. MIME allows arbitrary data to be encoded in ASCII and then transmitted in a standard e-mail message.
- It is a supplementary protocol that allows non-ASCII data to be sent through SMTP.

- The MIME standard provides a general structure for the content type of Internet messages and allows extensions for new content-type applications.

MIME Description

- MIME transforms non-ASCII data at the sender's site to NVT ASCII data and delivers it to the client SMTP to be sent through the Internet. The server SMTP at the receiver's site receives the NVT ASCII data and delivers it to MIME to be transformed back to the original non-ASCII data.

MIME Header

MIME defines five headers that can be added to the original SMTP header section:

- MIME Version
- Content Type
- Content Transfer Encoding
- Content Id
- Content Description

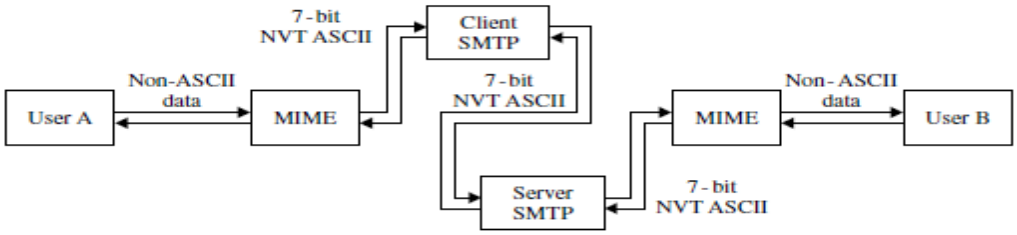


Figure 9.6 MIME showing a set of transforming functions.

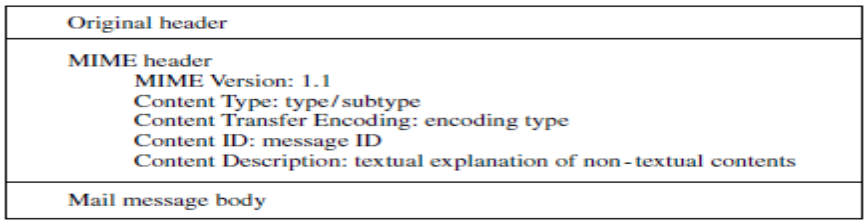


Figure 9.7 MIME header.

The MIMI header is shown in Figure 9.7 and described below.

MIME Version

This header defines the version of MIME used. The current version is 1.0.

Content Type

This header defines the type of data used in the message body. The content type and the content subtype are separated by a slash. MIME allows seven different types of data:

· ***Text*:** The original message is in 7-bit ASCII format.

· ***Multipart***: The body contains multiple, independent parts. The multipart header needs to define the boundary between each part. Each part has a separate content type and encoding.

Definition of multipart/signed:

– *MIME type name: multipart*

– *MIME subtype name: signed.*

– *Required parameters: boundary, protocol and micalg*

– *Optional parameters: none*

– *Security considerations: must be treated as opaque while in transit.*

Definition of multipart/encrypted:

– *MIME type name: multipart*

– *MIME subtype name: encrypted*

– *Required parameters: boundary and protocol*

– *Optional parameters: none*

– *Security considerations: none.*

Content Transfer Encoding

This header defines the method to encode the messages into ones and zeros for transport.

There are the five types of encoding: 7 bit, 8 bit, binary, Base64 and Quoted-printable. Table 9.3 describes the Content Transfer Encoding by the five types.

Table 9.3 Five types of encoding

Type	Description
7 bit	NVT ASCII characters and short lines
8 bit	Non-ASCII characters and short lines
Binary	Non-ASCII characters with unlimited-length lines
Base64	6-bit blocks of data encoded into 8-bit ASCII characters
Quoted-printable	Non-ASCII characters encoded as an equals sign followed by an ASCII code

Note that lines in the header identify the type of the data as well as the encoding used.

· ***7 bit*:** This is 7-bit NVT ASCII encoding. Although no special transformation is needed, the length of the line should not exceed 1000 characters.

· ***8 bit*:** This is 8-bit encoding. Non-ASCII characters can be sent, but the length of the

line still should not exceed 1000 characters. Since the underlying SMTP is able to transfer 8-bit non-ASCII characters, MIME does not do any encoding here.

· ***Binary*:** This is 8-bit encoding. Non-ASCII characters can be sent, and the length of the line can exceed 1000 characters. MIME does not do any encoding here; the underlying SMTP must be able to transfer binary data.

· ***Base64*** : This is a solution for sending data made of bytes when the highest bit is not necessarily zero.

· ***Quoted-printable*:** Base64 is a redundant encoding scheme. The 24-bit non-ASCII data becomes four characters consisting of 32 bits.

Content Id

This header uniquely identifies the whole message in a multiple message environment:

Content Id: id = <content id>

Content Description

This header defines whether the body is image, audio or video:

Content Description: <description>

MIME Security Multiparts

- ✓ The basic MIME by itself does not specify security protection.
- ✓ Accordingly, a MIME agent must provide security services by employing a security protocol mechanism, by defining two security subtypes of the MIME multipart content type: ***signed and encrypted***.
- ✓ The multipart/signed content type specifies how to support authentication and integrity services via digital signature. The multipart/singed content type contains exactly two body parts.
- ✓ The first body part is the one over which the digital signature was created, including its MIME headers.
- ✓ The second body part contains the control information necessary to verify the digital signature.

MIME Security with OpenPGP

PGP can generate either ASCII Armor or a stream of arbitrary 8-bit octets when encrypting data, generating a digital signature, or extracting public-key data. The ASCII Armor output is the required method for data transfer. When the data is to be transmitted in many parts, the MIME message/partial mechanism should be

used rather than the multipart ASCII Armor OpenPGP format. Before OpenPGP encryption, the data is written in MIME canonical format (body and headers).

When the OpenPGP digital signature is generated:

- The data to be signed must first be converted to its content-type specific canonical form.
- An appropriate Content Transfer Encoding is applied. In particular, line endings in the encoded data must use the canonical <CR><LF> sequence where appropriate.
- MIME content headers are then added to the body, each ending with the canonical <CR><LF> sequence.
- Any trailing white space must be removed from the signed material.
- The digital signature must be calculated over both the data to be signed and its set of content headers.
- The signature must be generated as detached from the signed data so that the process does not alter the signed data in any way.

3.Explain in detail about the S/MIME.

S/MIME

- S/MIME provides a way to send and receive 7-bit MIME data. S/MIME can be used with any system that transports MIME data.
- It can also be used by traditional mail user agents (MUAs) to add cryptographic security services to mail that is sent, and to interpret cryptographic security services in mail that is received.

The S/MIME agent represents user software that is a receiving agent, a sending agent, or both. S/MIME version 3 agents should attempt to have the greatest interoperability possible with S/MIME version 2 agents.

3.1 Cryptographic Message Syntax (CMS) Options

CMS allows for a wide variety of options in content and algorithm support. This subsection puts forth a number of support requirements and recommendations in order to achieve a base level of interoperability among all S/MIME implementations. CMS provides additional details regarding the use of the cryptographic algorithms.

DigestAlgorithmIdentifier

This type identifies a message digest algorithm which maps the message to the message digest. Sending and receiving agents must support SHA-1. Receiving agents shouldsupport MD5 for the purpose of providing backward compatibility with MD5-digested S/MIME v2 SignedData objects.

SignatureAlgorithmIdentifier

Sending and receiving agents must support id-dsa defined in DSS. Receiving agents should support rsaEncryption, defined in PRCS-1.

KeyEncryptionAlgorithmIdentifier

A key-encryption algorithm supports encryption and decryption operations. The encryption operation maps a key string to another encrypted key string under the control of a key encryption key. Sending and receiving agents must support Diffie–Hellman key exchange. Receiving agents should support rsaEncryption. The size of the private key is determined during key generation. Sending agents should support rsaEncryption.

General syntax

CMS defines multiple content types. Of these, only the data, signed data and enveloped data types are currently used for S/MIME.

· ***Data content type*:** This type is arbitrary octet strings, such as ASCII text files. Such strings need not have any internal structure. The data content type should have ASN.1 type Data:

Data ::= OCTET STRING

Sending agents must use the id-data content-type identifier to indicate the message content which has had security services applied to it.

***Signed-data content type*:** This type consists of any type and encrypted message digests of the content for zero or more signers. Any type of content can be signed by any number of signers in parallel. The encrypted digest for a signer is a digital signature on the content for that signer.

***Enveloped-data content type*:** An application/prcs7-mime subtype is used for the enveloped- data content type. This content type is used to apply privacy protection to a message. The type consists of encrypted content of any type and encrypted-content encryption keys for one or more recipients. The combination of encrypted content and encrypted content-encryption key for a recipient is called a *digital envelope* for that recipient.

Enhanced Security Services for S/MIME

- The security services described in this section are extensions to S/MIME version 3. Some of the features of each service use the concept of a *triple wrapped* message.
- A triple wrapped message is one that has been signed, then encrypted and then signed again.
- The S/MIME specification does not limit the number of nested encapsulations, so there may be more than three wrappings.

The inside signature is used for content integrity, non-repudiation with proof of origin, and binding attributes to the original content.

The outside signature provides authentication and integrity for information that is processed hop by hop, where each hop is an intermediate entity such as a mail list agent.

Triple Wrapped Message

The steps to create a triple wrapped message are as follows:

1. Start with the original content (a message body).

2. Encapsulate the original content with the appropriate MIME content-type headers.

3. Sign the inner MIME headers and the original content resulting from step 2.

4. Add an appropriate MIME construct to the signed message from step 3. The resulting message is called the *inside signature*.

5. Encrypt the step 4 result as a single block, turning it into an application/pkcs7- mime object.

6. Add the appropriate MIME headers: a content type of application/pkcs7-mime with parameters, and optional MIME headers such as Content-Transfer-Encoding and Content-Disposition.

Signed Receipts

The interaction steps in a typical transaction are:

1. Sender creates a signed message including a receipt request attribute.

2. Sender transmits the resulting message to the recipient(s).

3. Recipient receives message and determines if there are a valid signature and receipt request in the message.

4. Recipient creates a signed receipt.

5. Recipient transmits the resulting signed receipt message to the sender.

6. Sender receives the message and validates that it contains a signed receipt for the original message.

Receipt Request Creation

Multilayer S/MIME messages may contain multiple SignedData layers. Receipts are requested only for the innermost SignedData layer in a multilayer S/MIME message such as a triple wrapped message. Only one receipt request attribute can be included in the signedAttributes of SignerInfo.

4. Explain in detail about following topics of the Internet Firewalls for Trusted Systems:

1. Roles of Firewalls
2. Firewall related terminology

Roles of Firewalls

- The firewall imposes restrictions on packets entering or leaving the private network.
- All traffic from inside to outside, and vice versa, must pass through the firewall, but only authorised traffic will be allowed to pass.
- Firewalls create checkpoints (or choke points) between an internal private network and an untrusted Internet.
- The firewall may filter on the basis of IP source and destination addresses and TCP port number.
- Firewalls may block packets from the Internet side that claim a source address of a system on the intranet, or they may require the use of an access negotiation and encapsulation protocol like SOCKS to gain access to the intranet.
- The firewall also enforces logging, and provides alarm capacities as well. By placing logging services at firewalls, security administrators can monitor all access to and from the Internet.
- Firewalls may block TELNET or RLOGIN connections from the Internet to the intranet.
- The firewall provides protection from various kinds of IP spoofing and routing attacks.
- A firewall can limit network exposure by hiding the internal network systems and information from the public Internet.
- The firewall is a convenient platform for security-unrelated events such as a network address translator (which maps local addresses to Internet addresses) and has a network management function that accepts or logs Internet usage.
- The firewall certainly has some negative aspects: it cannot protect against internal threats such as an employee who cooperates with an external attacker;
- A firewall can effectively implement and control the traversal of IP multicast traffic.
- Some firewall mechanisms such as SOCKS are less appropriate for multicast because they are designed specifically for unicast traffic.

Firewall related terminology

1. Bastion Host
2. Proxy Server

3. SOCKS
4. Choke Point
5. De-militarised Zone (DMZ)
6. Logging and Alarms
7. VPN

Bastion Host

- A bastion host is a publicly accessible device for the network's security, which has a direct connection to a public network such as the Internet.
- The bastion host serves as a platform for any one of the three types of firewalls: packet filter, circuit-level gateway or application-level gateway.
- Bastion hosts must check all incoming and outgoing traffic and enforce the rules specified in the security policy.

The bastion host's role falls into the following three common types:

1. *Single-homed bastion host*
2. *Dual-homed bastion host*
3. *Multihomed bastion host*

Proxy Server

- Proxy servers are used to communicate with external servers on behalf of internal clients.
- A proxy service is set up and torn down in response to a client request, rather than existing on a static basis.
- The term proxy server typically refers to an application-level gateway, although a circuit-level gateway is also a form of proxy server.
- Each proxy is independent of other proxies on the bastion host.
- If there is a problem with the operation of any proxy, or if future vulnerability is discovered, it is easy to replace the proxy without affecting the operation of the proxy's applications.
- If the support of a new service is required, the network administrator can easily install the required proxy on the bastion host.
- A proxy generally performs no disk access other than to read its initial configuration file.
- This makes it difficult for an intruder to install Trojan horse sniffers or other dangerous files on the bastion host.

SOCKS

- ✓ The SOCKS protocol version 4 provides for unsecured firewall traversal for TCP-based client/server applications, including HTTP, TELNET and FTP.
- ✓ The new protocol extends the SOCKS version 4 model to include UDP, and allows the framework to include provision for generalized strong authentication schemes, and extends the addressing scheme to encompass domain name and IPv6 addresses.
- ✓ When a TCP-based client wishes to establish a connection to an object that is reachable only via a firewall, it must open a TCP connection to the appropriate SOCKS port on the SOCKS server system.
- ✓ The SOCKS service is conventionally located at TCP port 1080.
- ✓ If the connection request succeeds, the client enters negotiation for the authentication method to be used, authenticates with the chosen method, and then sends a relay request.
- ✓ The SOCKS server evaluates the request, and either establishes the appropriate connection or denies it.

Choke Point

- ✓ A choke point is the point at which a public internet can access the internal network.
- ✓ The most comprehensive and extensive monitoring tools should be configured on the choke points.
- ✓ Proper implementation requires that all traffic be funnelled through these choke points.
- ✓ Since all traffic is flowing through the firewalls, security administrators, as a firewall strategy, need to create choke points to limit external access to their networks.
- ✓ Once these choke points have been clearly established, the firewall devices can monitor, filter and verify all inbound and outbound traffic.
- ✓ Since a choke point is installed at the firewall, a prospective hacker will go through the choke point.

De-militarised Zone (DMZ)

- ✓ The DMZ is an expression that originates from the Korean War. It meant a strip of land forcibly kept clear of enemy soldiers.
- ✓ In terms of a firewall, the DMZ is a network that lies between an internal private network and the external public network.
- ✓ DMZ networks are sometimes called perimeter networks.
- ✓ A DMZ is used as an additional buffer to further separate the public network from the internal network.
- ✓ A gateway is a machine that provides relay services to compensate for the effects of a filter.
- ✓ The network inhabited by the gateway is often called the DMZ.
- ✓ A gateway in the DMZ is sometimes assisted by an internal gateway.

Logging and Alarms

- ➢ Logging is usually implemented at every device in the firewall, but these individual logs combine to become the entire record of user activity.
- ➢ Packet filters normally do not enable logging by default so as not to degrade performance. Packet filters as well as circuit-level gateways log only the most basic information. Since a choke point is installed at the firewall, a prospective hacker will go through the choke point.
- ➢ The user can then tell exactly what a hacker is doing, and have such information available for audit.
- ➢ The audit log is an essential tool for detecting and terminating intruder attacks.
- ➢ Many firewalls allow the user to preconfigure responses to unacceptable activities.
- ➢ The firewall should alert the user by several means. The two most common actions are for the firewall to break the TCP/IP connection, or to have it automatically set off alarms.

VPN

- ➢ VPNs are appropriate for any organization requiring secure external access to internal resources.
- ➢ All VPNs are tunneling protocols in the sense that their information packets or payloads are encapsulated or tunneled into the network packets.

- All data transmitted over a VPN is usually encrypted because an opponent with access to the Internet could eavesdrop on the data as it travels over the public network.
- The VPN encapsulates all the encrypted data within an IP packet. Authentication, message integrity and encryption are very important fundamentals for implementing a VPN.
- Without such authentication procedures, a hacker could impersonate anyone and then gain access to the network.
- Message integrity is required because the packets can be altered as they travel through the Internet. Without encryption, the information may become truly public. Several methods exist to implement a VPN.

5.Explain in detail about the Types of Firewalls.

4. Packet Filters
5. Circuit-Level Gateways
6. Application-Level Gateways

Packet Filters

- Packet filters are one of several different types of firewalls that process network traffic on a packet-by-packet basis.
- A packet filter's main function is to filter traffic from a remote IP host, so a router is needed to connect the internal network to the Internet.
- A packet filter is a device which inspects or filters each packet at a screening router for the content of IP packets.

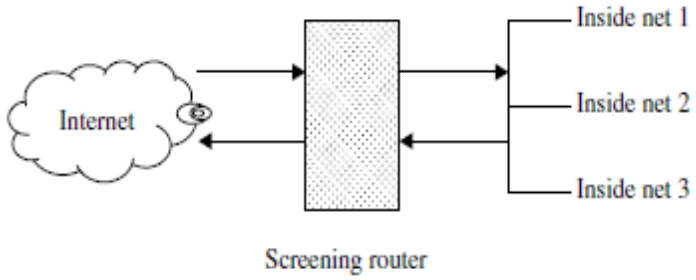


Figure 10.1 A screening router for packet filtering.

Packet filters typically set up a list of rules that are sequentially read line by line. Filtering rules can be applied based on source and destination IP addresses or network addresses, and TCP or UDP ports. Packet filters are read and then treated on

a rule-by-rule basis. A packet filter will provide two actions, forward or discard. If the action is in the forward process, the action takes place to route the packet as normal if all conditions within the rule are met.

Packet-Filtering Rules

- A packet filter applies a set of rules to each incoming IP packet and then forwards or discards the packet.
- The packet filter typically sets up a list of rules which may match fields in the IP or TCP header. If there is a match to one of the rules, that rule is able to determine whether to forward or discard the packet.
- If there is no match to any rule, then two default actions (forward and discard) will be taken.

TELNET packet filtering

- TELNET is a simple remote terminal access that allows a user to log onto a computer across an internet. TELNET establishes a TCP connection, and then passes keystrokes from the user's keyboard directly to the remote computer as if they had been typed on a keyboard attached to the remote machine.
- TELNET also carries output from the remote machine back to the user's screen. TELNET client software allows the user to specify a remote machine either by giving its domain name or IP address.
- TELNET can be used to administer a UNIX machine. Windows NT does not provide a TELNET serve with the default installation, but a third-party service can be easily added.
- TELNET sends all user names and passwords in plaintext. Experienced hackers can hijack a TELNET session in progress.
- TELNET should only be used when the user can verify the entire network connecting the client and server, not over the Internet.
- All TELNET traffic should be filtered at the firewall. TELNET runs on TCP port 23.

FTP packet filtering

- With FTP, two TCP connections are used: a control connection to set up the file transfer and a data connection for the actual file transfer.
- The data connection uses a different port number to be assigned for the transfer.

- Remember that most servers live on low-numbered ports, but most outgoing calls tend to use higher-numbered ports, typically above 1024.
- FTP is the first protocol for transferring or moving files across the Internet. Like many of the TCP/IP protocols, FTP was not designed with security in mind.
- Each FTP server has a *command channel*, where the requests for data and directory listings are issued, and a *data channel*, over which the requested data is delivered.
- FTP operates in two different modes (active and passive).
- In active mode, an FTP server receives commands on TCP/IP port 21 and exchanges data with the client.

SMTP packet filtering

- SMTP is a store/forward system, and such systems are well suited to firewall applications.
- SMTP receivers use TCP port 25; SMTP senders use a randomly selected port above 1023. Most e-mail messages are addressed with hostnames instead of IP addresses, and the SMTP server uses DNS (Directory and Naming Services) to determine the matching IP address.
- If the same machines handle internal and external mail delivery, a hacker who can spoof DNS information may be able to cause mail that was intended for internal destinations to be delivered to an external host.

Circuit-Level Gateways

- ✓ The circuit-level gateway represents a proxy server that statically defines what traffic will be forwarded.
- ✓ Circuit proxies always forward packets containing a given port number if that port number is permitted by the rule set.
- ✓ A circuit-leval gateway operates at the network level of the OSI model.
- ✓ This gateway acts as an IP address translator between the Internet and the internal system.
- ✓ The main advantage of a proxy server is its ability to provide Network Address Translation (NAT). NAT hides the internal IP address from the Internet.

Application-Level Gateways

- The application-level gateway represents a proxy server, performing at the TCP/IP application level, that is set up and torn down in response to a client request, rather than existing on a static basis.

- Application proxies forward packets only when a connection has been established using some known protocol.
- When the connection closes, a firewall using application proxies rejects individual packets, even if the packets contain port numbers allowed by a rule set.
- The application gateway analyses the entire message instead of individual packets when sending or receiving data.

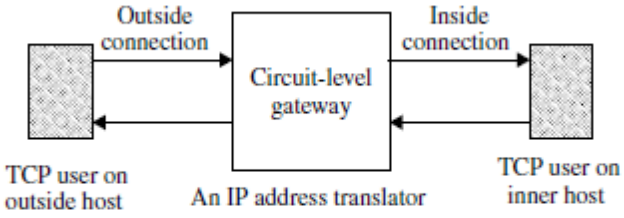


Figure 10.2 Circuit-level gateway for setting up two TCP connections.

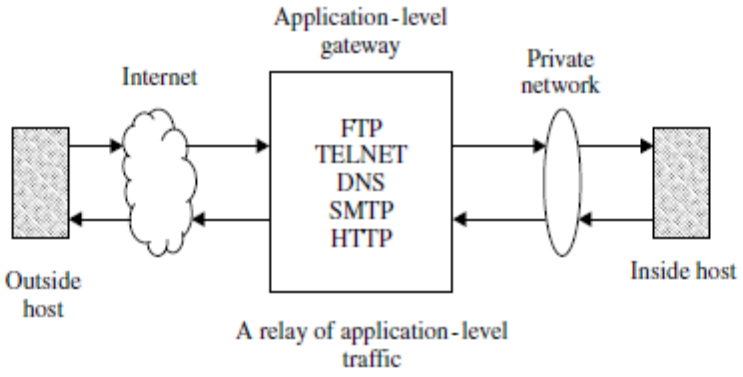


Figure 10.3 Application-level gateway for acting as a relay of application-level traffic.

When an inside host initiates a TCP/IP connection, the application gateway receives the request and checks it against a set of rules or filters. The application gateway (or proxy server) will then initiate a TCP/IP connection with the remote server.

6. Explain in detail about the Firewall Designs.

- The primary step in designing a secure firewall is obviously to prevent the firewall devices from being compromised by threats.
- To provide a certain level of security, the three basic firewall designs are considered:

- ***A single-homed bastion host,***
- ***A dual-homed bastion host***
- ***A screened subnet firewall.***

➢ The first two options are for creating a screened host firewall, and the third option contains an additional packet-filtering router to achieve another level of security.

Screened Host Firewall (Single-Homed Bastion Host)

✓ Single-homed bastion hosts can be configured as either circuit-level or application-level gateways. When using either of these two gateways, each of which is called a proxy server, the bastion host can hide the configuration of the internal network.

✓ NAT is essentially needed for developing an address scheme internally. It is a critical component of any firewall strategy.

✓ It translates the internal IP addresses to IANA registered addresses to access the Internet.

✓ The screened host firewall is designed such that all incoming and outgoing information is passed through the bastion host.

✓ The screening router is also configured to route outgoing traffic only if it originates from the bastion host.

✓ A single-homed implementation may allow a hacker to modify the router not to forward packets to the bastion host.

Screened Host Firewall (Dual-Homed Bastion Host)

The configuration of the screened host firewall using a dual-homed bastion host adds significant security, compared with a single-homed bastion host.

This firewall implementation is secure due to the fact that it creates a complete break between the internal network and the external Internet.

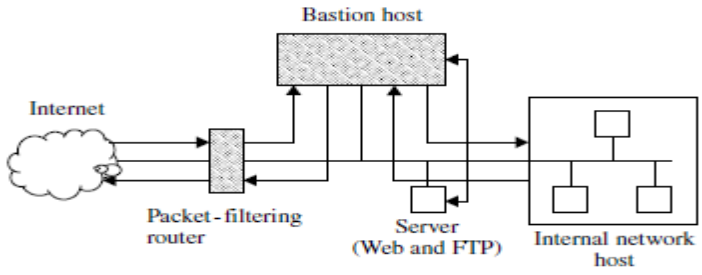


Figure 10.4 Screened host firewall system (single-homed bastion host).

As with the single-homed bastion, all external traffic is forwarded directly to the bastion host for processing. However, a hacker may try to subvert the bastion host and the router to bypass the firewall mechanisms. Even if a hacker could defeat either the screening router or the dual-homed bastion host, the hacker would still have to penetrate the other. Nevertheless, a dual-homed bastion host removes even this possibility. It is also possible to implement NAT for dual-homed bastion hosts.

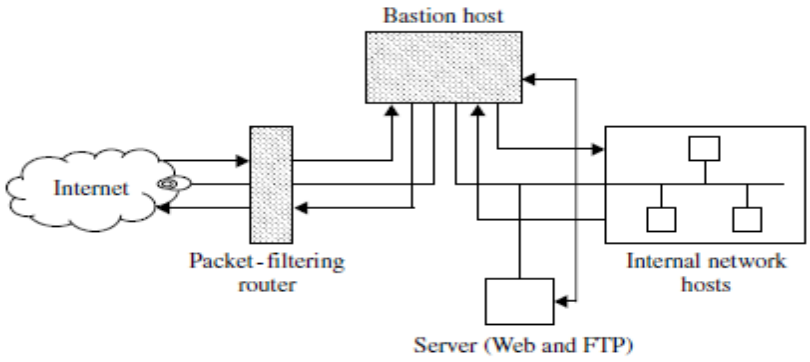


Figure 10.5 Screened host firewall system (dual-homed bastion host).

Screened Subnet Firewall

- ✓ The third implementation of a firewall is the screened subnet, which is also known as a DMZ.

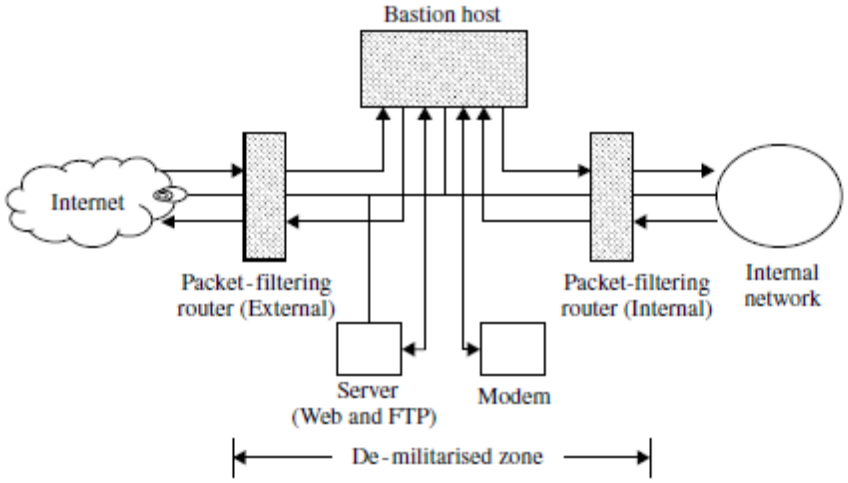


Figure 10.6 Screened subnet firewall system.

- ✓ This firewall is the most secure one among the three implementations, simply because it uses a bastion host to support both circuit- and application-level gateways.
- ✓ This DMZ then functions as a small isolated network positioned between the Internet and the internal network.
- ✓ The screened subnet firewall contains external and internal screening routers. Each is configured such that its traffic flows only to or from the bastion host. This arrangement prevents any traffic from directly traversing the DMZ subnetwork.
- ✓ The external screening router uses standard filtering to restrict external access to the bastion host, and rejects any traffic that does not come from the bastion host.
- ✓ This router also uses filters to prevent attacks such as IP spoofing and source routing.
- ✓ The internal screening router also uses rules to prevent spoofing and source routing.

7. Explain in detail about the SET for E-Commerce Transactions.

The Secure Electronic Transaction (SET) is a protocol designed for protecting credit card transactions over the Internet. It is an industry-backed standard that was formed by MasterCard and Visa (acting as the governing body) in February 1996.

Business Requirements for SET

This section describes the major business requirements for credit card transactions by means of secure payment processing over the Internet. They are listed below:

1. *Confidentiality of information (provide confidentiality of payment and order information)*:
2. *Integrity of data (ensure the integrity of all transmitted data)*:
3. *Cardholder account authentication (provide authentication that a cardholder is a legitimate customer of a branded payment card account)*:
4. *Merchant authentication (provide authentication that a merchant can accept credit card transactions through its relationship with an acquiring financial institution)*:

5. *Security techniques (ensure the use of the best security practices and system design techniques to protect all legitimate parties in an electronic commerce transaction)*:
6. *Creation of brand-new protocol (create a protocol that neither depends on transport security mechanisms nor prevents their use)*:
7. *Interoperability (facilitate and encourage interoperability among software and network providers)*:

SET System Participants

A discrepancy is found between an SET transaction and a retail or mail order transaction: in a face-toface retail transaction, electronic processing begins with the merchant or the acquirer, but, in an SET transaction, the electronic processing begins with the cardholder.

1. *Cardholder*:
2. *Issuer*:
3. *Merchant*:
4. *Acquirer*:
5. *Payment gateway*:
6. *Certification Authority*:

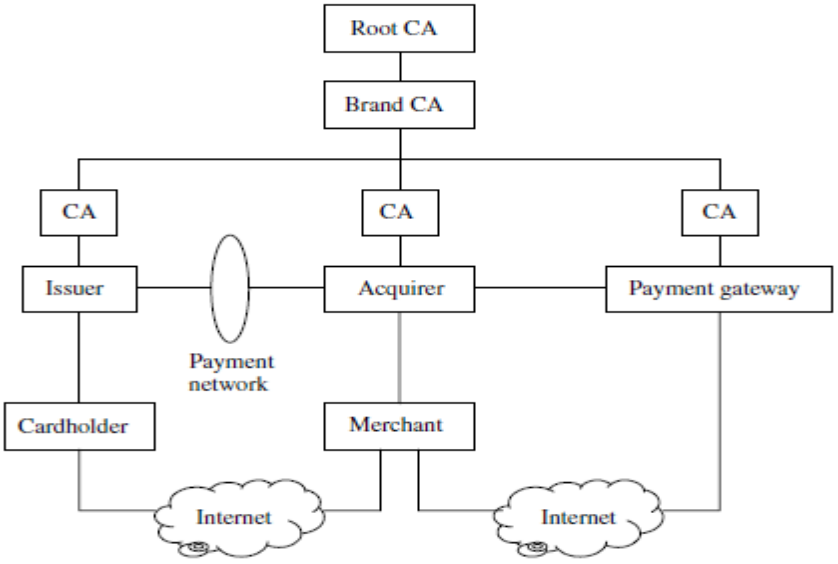


Figure 11.1 The SET hierarchy indicating the relationships between the participants.

Figure 11.1 illustrates the SET hierarchy which reflects the relationships between the participants in the SET system, described in the preceding paragraphs. In the SET environment, there exists a hierarchy of CAs. The SET protocol specifies a method of *trust chaining* for entity authentication. This trust chain method entails the exchange of digital certificates and verification of the public keys by validating the digital signatures of the issuing CA.

Cryptographic Operation Principles

SET is the Internet transaction protocol providing security by ensuring confidentiality, data integrity, authentication of each party and validation of the participant's identity. To meet these requirements, SET incorporates the following cryptographic principles:

- ***Confidentiality*:**
- ***Integrity*:**
- ***Authentication*:**

Dual Signature and Signature Verification

SET introduced a new concept of digital signature called *dual signatures*. A dual signature is generated by creating the message digest of two messages: order digest and payment digest.

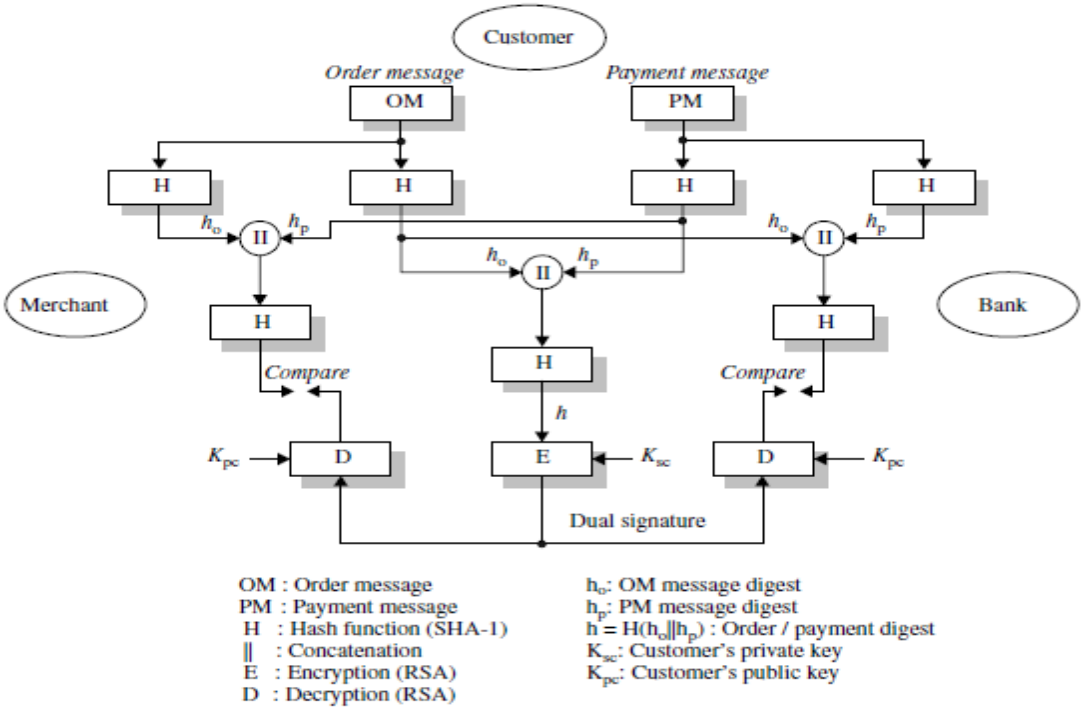


Figure 11.2 Dual signature and order/payment message authentication.

Referring to Figure 11.2, the customer takes the hash codes (message digests) of both the order message and payment message by using the SHA-1 algorithm. ***Computation of the dual signature (DS) is shown as follows:***

DS = EKsc(h)

where h = H(H(OM)||H(PM))

= H(ho||hp)

EKsc (= dc) is the customer's private signature key.

Authentication and Message Integrity

When user A wishes to sign the plaintext information and send it in an encrypted message (ciphertext) to user B, the entire encryption process is as configured in Figure 11.4. The encryption/decryption processes for message integrity consist of the following steps.

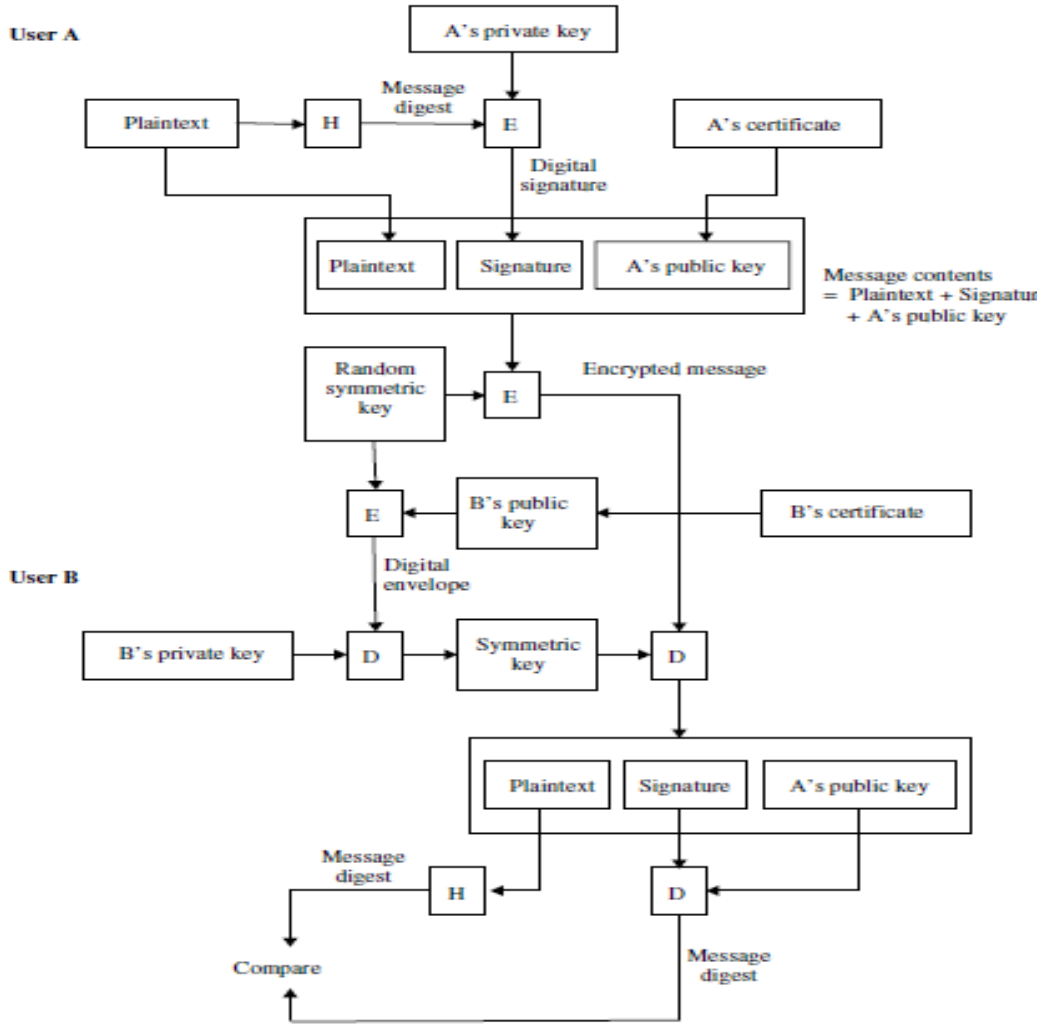


Figure 11.4 Encryption/Decryption overview for message integrity.

1. Encryption process:

- ✓ User A sends the plaintext through a hash function to produce the message digest that is used later to test the message integrity.
- ✓ A then encrypts the message digest with his or her private key to produce the digital signature.
- ✓ Next, A generates a random symmetric key and uses it to encrypt the plaintext, A's signature and a copy of A's certificate, which contains A's public key. To decrypt the plaintext later, user B will require a secure copy of this temporary symmetric key.
- ✓ B's certificate contains a copy of his or her public key. To ensure secure transmission of the symmetric key, A encrypts it using B's public key. The encrypted key, called the digital envelope, is sent to B along with the encrypted message itself.
- ✓ A sends a message to B consisting of the DES-encrypted plaintext, signature and A's public key, and the RSA-encrypted digital envelope.

2. Decryption process:

- ➢ B receives the encrypted message from A and decrypts the digital envelope with his or her private key to retrieve the symmetric key.
- ➢ B uses the symmetric key to decrypt the encrypted message, consisting of the plaintext, A's signature and A's public key retrieved from A's certificate.
- ➢ B decrypts A's digital signature with A's public key that is acquired from A's certificate. This recovers the original message digest of the plaintext.
- ➢ B runs the plaintext through the same hash function used by A and produces a new message digest of the decrypted plaintext.
- ➢ Finally, B compares his or her message digest to the one obtained from A's digital signature.

Payment Processing

1. ***Cardholder Registration***
2. ***Merchant Registration***
3. ***Purchase Request***
4. ***Payment Authorization***
5. ***Payment Capture***

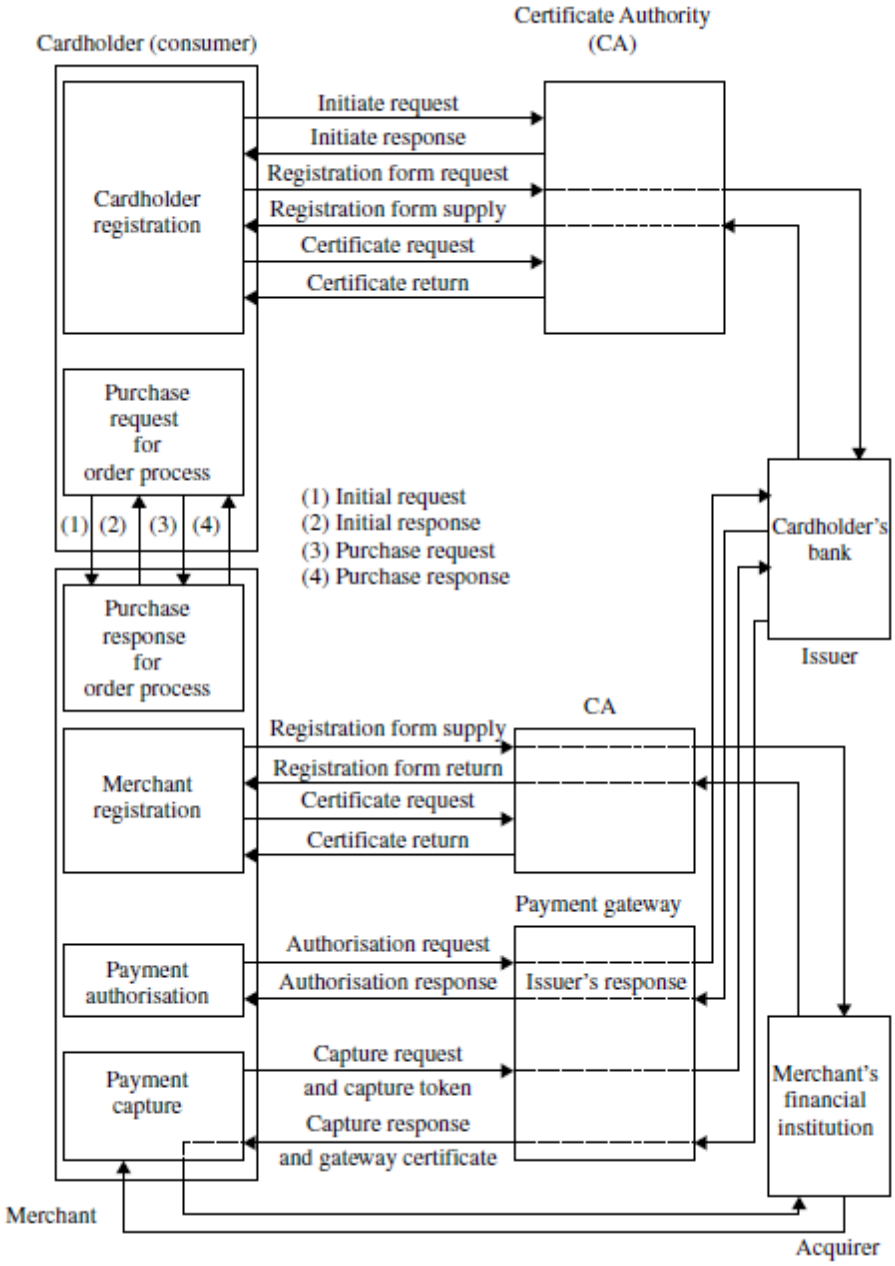


Figure 11.6 Overall picture of payment processing.

Cardholder Registration

The cardholder must register with a CA before sending SET messages to the merchant. The cardholder needs a public/private-key pair for use with SET. The scenario of cardholder registration is described in the following.

1. Registration request/response processes:

The registration process can be started when the cardholder requests a copy of the

CA certificate. When the CA receives the request, it transmits its certificate to the cardholder. The cardholder verifies the CA certificate by traversing the trust chain to the root key. The cardholder holds the CA certificate to use later during the registration process.

- The cardholder sends the *initiate request* to the CA.
- Once the initiate request is received from the cardholder, the CA generates the response and digitally signs it by generating a message digest of the response and encrypting it with the CA's private key.
- The CA sends the *initiate response* along with the CA certificate to the cardholder.
- The cardholder receives the initiate response and verifies the CA certificate by traversing the trust chain to the root key.
- The cardholder verifies the CA certificate by decrypting it with the CA's public key and comparing the result with a newly generated message digest of the initiate response.

2. Registration form process:

- The cardholder generates the registration form request.
- The cardholder encrypts the SET message with a random symmetric key (No. 1).
- The DES key, along with the cardholder's account number, is then encrypted with the CA's public key.
- The cardholder transmits the encrypted registration form request to the CA.
- The CA decrypts the symmetric DES key (No. 1) and cardholder's account number with the CA's private key. The CA then decrypts the registration form request using the symmetric DES key (No. 1).
- The CA determines the appropriate registration form and digitally signs it by generating a message digest of the registration form and encrypting it with the CA's private key.
- The CA sends the registration form and the CA certificate to the cardholder.
- The cardholder receives the registration form and verifies the CA certificate by traversing the trust chain to the root key.
- The cardholder verifies the CA's signature by decrypting it with the CA's public key and comparing the result with a newly generated message digest

of the registration form. The cardholder then completes the registration form.

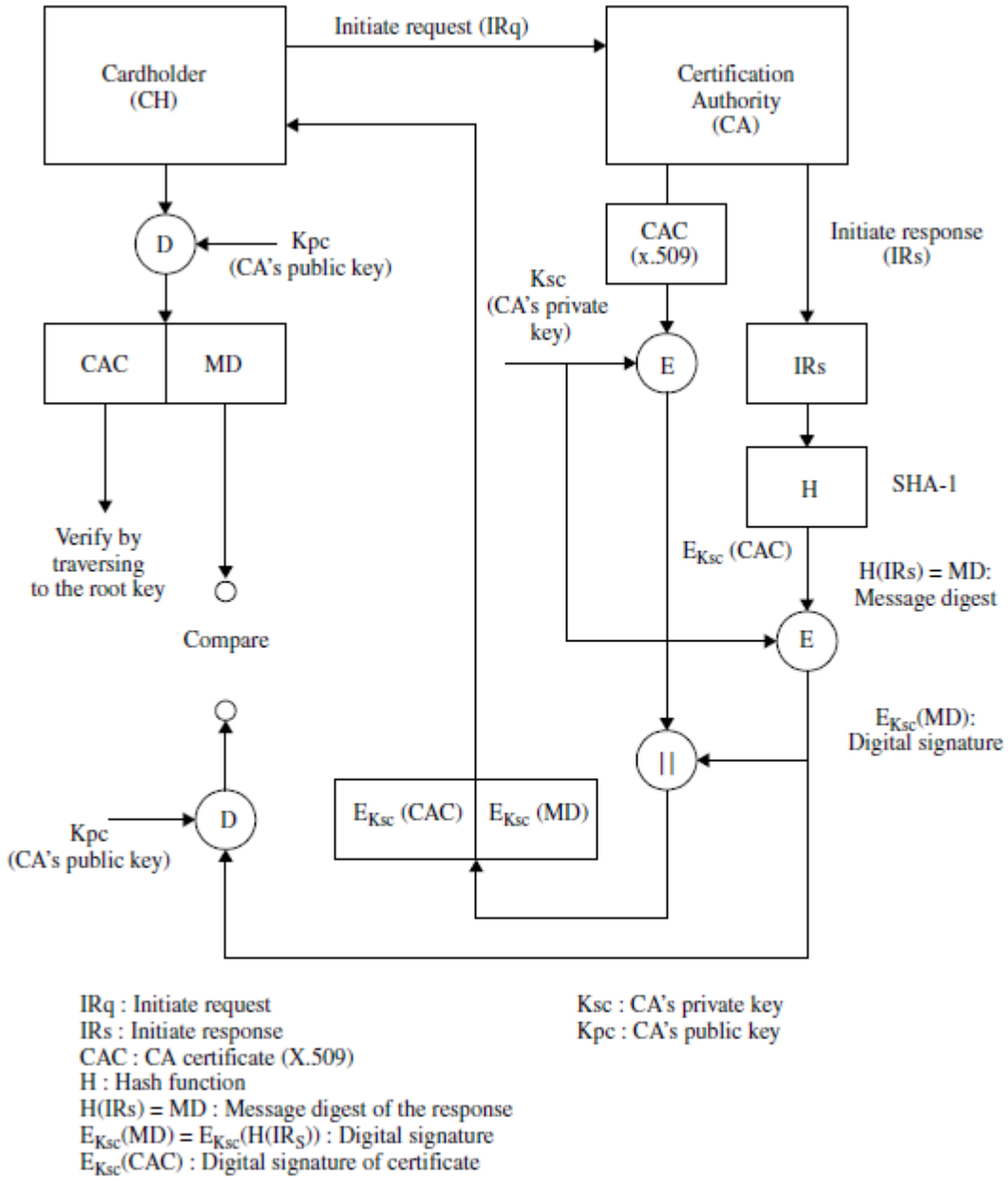


Figure 11.7 Registration request/response processes.

3. *Certificate request/response processes*:

- ✓ The cardholder generates the certificate request, including the information entered into the registration form.

- ✓ The cardholder creates a message with request, the cardholder's public key and a newly generated symmetric key (No. 2), and digitally signs it by generating a message digest of the cardholder's private key.
- ✓ The cardholder encrypts the message with a randomly generated symmetric key (No. 3). This symmetric key, along with the cardholder's account information, is then encrypted with the CA's public key.
- ✓ The cardholder transmits the encrypted certificated request messages to the CA.
- ✓ The CA decrypts the No. 3 symmetric key and cardholder's account information with the CA's private key, and then decrypts the certificate request using this symmetric key.
- ✓ The CA verifies the cardholder's signature by decrypting it with the cardholder's public key and comparing the result with a newly generated message digest of the certificate requested.
- ✓ The CA verifies the certificate request using the cardholder's account information and information from the registration form.
- ✓ The CA generates the certificate response and digitally signs it by generating a message digest of the response and encrypting it with the CA's private key.

Merchant Registration

Merchants must register with a CA before they can receive SET payment instructions from cardholders. In order to send SET messages to the CA, the merchant must have a copy of the CA's public key which is provided in the CA certificate.

1. Registration form process:

The registration process starts when the merchant requests the appropriate registration form.

- o The merchant sends the initiate request of the registration form to the CA.
- o To register, the merchant fills out the registration form with information such as the merchant's name, address and ID.
- o The CA receives the initiate request.
- o The CA selects an appropriate registration form and digitally signs it by generating a message digest of the registration form and encrypting it with the CA's private key.

- The CA sends the registration form along with the CA certificate to the merchant.
- The merchant verifies the CA's signature by decrypting it with the CA's public key and comparing the result with a newly computed message digest of the registration form.
- The merchant creates two public/private-key pairs for use with SET: key encryption and signature.

2. Certificate request/create process:

- The merchant generates the certificate request.
- The merchant creates the message with request and both merchant public keys and digitally signs it by generating a message digest of the certificate request and encrypting it with the merchant's private key.
- The merchant encrypts the message with a random symmetric key (No. 1). This symmetric key, along with the merchant's account data, is then encrypted with the CA's public key.
- The merchant transmits the encrypted certificate request message to the CA.
- The CA decrypts the symmetric key (No. 1) and the merchant's account data with the CA's private key, and then decrypts the message using the symmetric key (No. 1).
- The CA verifies the merchant's signature by decrypting it with the merchant's public key and comparing the result with a newly computed message digest of the certificate request.
- The CA confirms the certificate request using the merchant information.
- Upon verification, the CA creates the merchant certificate digitally signing the certificate with the CA's private key.
- The CA generates the certificate response and digitally signs it by generating a message digest of the response and encrypting it with the CA's private key.
- The CA transmits the certificate response to the merchant.
- The merchant receives the certificate response from the CA. The merchant decrypts the digital envelope to obtain the symmetric key. This key is used to decrypt the registration response containing the certificates.

- The merchant verifies the certificates by traversing the trust chain to the root key.
- The merchant verifies the CA's signature by decrypting it with the CA's public key and comparing the result with a newly computed message digest of the certificate response.
- The merchant stores the certificates and information from the response for use in future e-commerce transactions.

Purchase Request

The purchase request exchange should take place after the cardholder has completed browsing, selecting and ordering. Before the end of this preliminary phase occurs, the merchant sends a completed order form to the cardholder (customer).

1. Initiate request:

· The cardholder sends the initiate request to the merchant.

· The merchant receives the initiate request.

· The merchant generates the response and digitally signs it by generating a message digest of the response and encrypting it with the merchant's private key.

· The merchant sends the response along with the merchant and payment gateway certificates to the cardholder.

2. Initiate response:

· The cardholder receives the initiate response and verifies the certificates by traversing the trust chain to the root key.

· The cardholder verifies the merchant's signature by decrypting it with the merchant's public key and comparing the result with a newly computed message digest of the response.

. The cardholder creates the order message (OM) using information from the shopping phase and payment message (PM). At this step the cardholder completes payment instructions.

3. Purchase request:

- The cardholder generates a dual signature for the OM and PM by computing the message digests of both, concatenating the two digests, computing the message digest of the result and encrypting it using the cardholder's private key.

- ✓ The cardholder generates a random symmetric key (No. 1) and uses it to encrypts the PM. The cardholder then encrypts his or her account number as well as the random symmetric key used to encrypt the PM in a digital envelope using the payment gateway's key.
    - o The cardholder transmits the OM and the encrypted PM to the merchant.
- ✓ The merchant verifies the cardholder certificate by traversing the trust chain to the root key.
    - o The merchant verifies the cardholder's dual signature on the OM by decrypting it with the cardholder's public key and comparing the result with a newly computed message digest of the concatenation of the message digests of the OM and PM.
- ✓ The merchant processes the request, including forwarding the PM to the payment gateway for authorisation.

4. Purchase response:

- ✓ The merchant creates the purchase response including the merchant signature certificate and digitally signs it by generating a message digest of the purchase response and encrypting it with the merchant's private key.
    - o The merchant transmits the purchase response to the cardholder.
    - o If the transaction was authorised, the merchant fulfils the order to the cardholder.
- ✓ The cardholder verifies the merchant signature certificate by traversing the trust chain to the root key.
    - o The cardholder verifies the merchant's digital signature by decrypting it with the merchant's public key and comparing the result with a newly computed message digest of the purchase response.
    - o The cardholder stores the purchase response.

Payment Authorisation

1. Authorisation request:

- o The merchant creates the authorisation request.
- ✓ The merchant digitally signs an authorisation request by generating a message digest of the authorisation request and encrypting it with the merchant's private key.
- ✓ The merchant encrypts the authorisation request using a random symmetric key (No. 2), which in turn is encrypted with the payment gateway public key.

- The merchant transmits the encrypted authorisation request and the encrypted PM from the cardholder purchase request to the payment gateway.
- The gateway verifies the merchant certificate by traversing the trust chain to the root key.
- The payment gateway decrypts the digital envelope of the authorisation request to obtain the symmetric encryption key (No. 2) with the gateway private key. The gateway then decrypts the authorisation request using the symmetric key (No. 2).
- The gateway verifies the merchant's digital signature by decrypting it with the merchant's public key and comparing the result with a newly computed message digest of the authorisation request.
- The gateway verifies the cardholder's certificate by traversing the trust chain to the root key.
- The gateway decrypts the symmetric key (No. 1) and the cardholder account information with the gateway private key. It uses the symmetric key to decrypt the PM.
- The gateway verifies the cardholder's dual signature on the PM by decrypting it with the cardholder's public key and comparing the result with a newly computed message digest of the concatenation of the message digest of the OM and the PM.
- The gateway ensures consistency between the merchant's authorisation request and the cardholder's PM.
- The gateway sends the authorisation request through a financial network to the cardholder's financial institution (issuer).

2. **Authorisation response:**

- The gateway creates the authorisation response message and digitally signs it by generating a message digest of the authorisation response and encrypting it with the gateway's private key.
- The gateway encrypts the authorisation response with a new randomly generated symmetric key (No. 3). This key is then encrypted with the merchant's public key.

- The gateway creates the capture token and digitally signs it by generating a message digest of the capture token and encrypting it with the gateway's private key.
    - The gateway encrypts the capture token with a new symmetric key (No. 4). This key and the cardholder account information are then encrypted with the gateway's public key.
    - The gateway transmits the encrypted authorisation response to the merchant.
- The merchant verifies the gateway certificate by traversing the trust chain to the root key.
- The merchant decrypts the symmetric key (No. 3) with the merchant's private key and then decrypts the authorisation response using the symmetric key (No. 3).

1. Capture request:

• The merchant creates the capture request.

• The merchant embeds the merchant certificate in the capture request and digitally signs it by generating a message digest of the capture request and encrypting it with the merchant's private key.

• The merchant encrypts the capture request with a randomly generated symmetric key (No. 5). This key is then encrypted with the payment gateway's public key.

• The merchant transmits the encrypted capture request and encrypted capture token previously stored from the authorisation response to the payment gateway.

• The gateway verifies the merchant certificate by traversing the trust chain to the root key.

• The gateway decrypts the symmetric key (No. 5) with the gateway's private key and then decrypts the capture request using the symmetric key (No. 5).

• The gateway verifies the merchant's digital signature by decrypting it with the merchant's public key and comparing the result with a newly computed message digest of the capture request.

• The gateway decrypts the symmetric key (No. 4) with the gateway's private key and then decrypts the capture token using the symmetric key (No. 4).

• The gateway ensures consistency between the merchant's capture request and the capture token.

• The gateway sends the capture request through a financial network to the cardholder's issuer (financial institution).

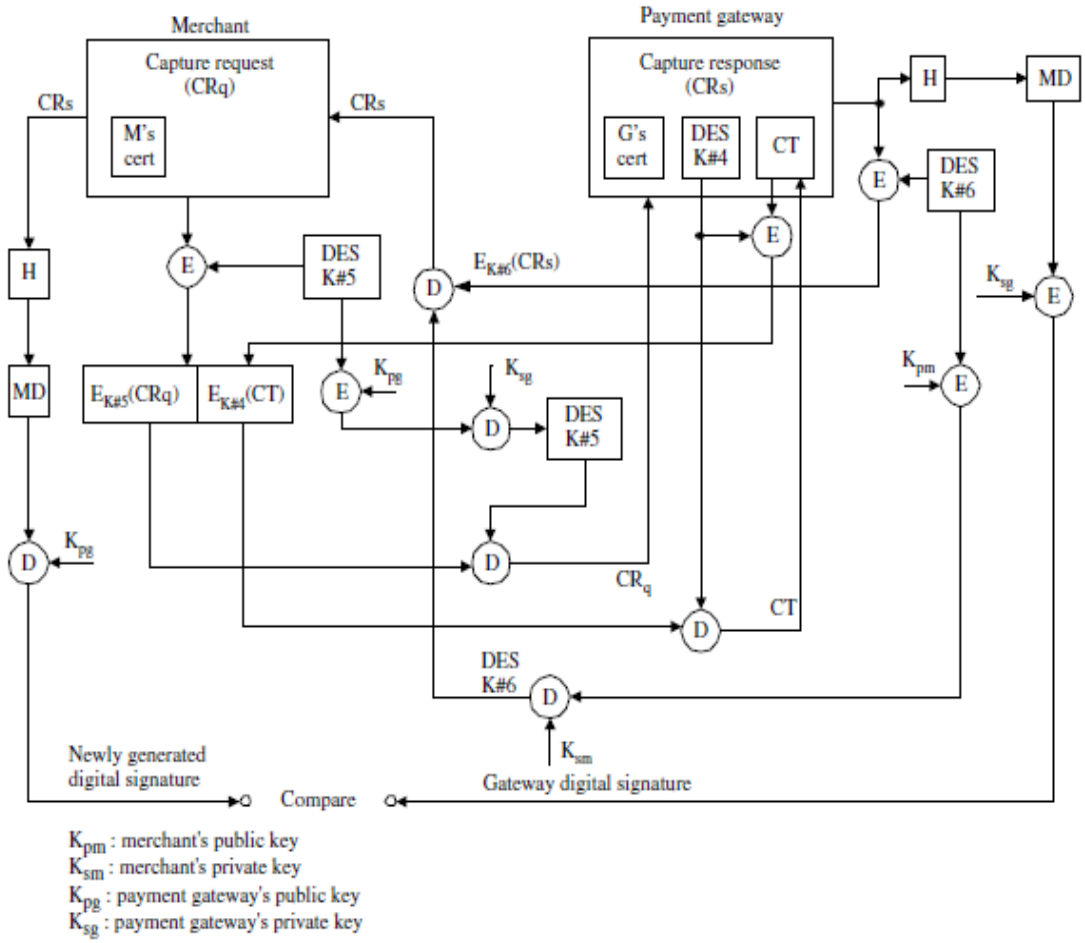


Figure 11.9 Payment capture process.

2. Capture response:

• The gateway creates the capture response message, including the gateway signature certificate, and digitally signs it by generating a message digest of the capture response and encrypting it with the gateway's private key.

• The gateway encrypts the capture response with a newly generated symmetric key (No. 6). This key is then encrypted with the merchant's public key.

• The gateway transmits the encrypted capture response to the merchant.

• The merchant verifies the gateway certificate by traversing the trust chain to the root key.

• The merchant decrypts the symmetric key (No. 6) with the merchant's private key and then decrypts the capture response using the symmetric key (No. 6).

Important Questions

Part-A

1.What is application level gateway?

2. List the design goals of firewalls?

3.What is mean by SET? What are the features of SET?

4. What are the steps involved in SET Transaction?

5.Define S/MIME?

6. What are the headers fields define in MIME?

7. What is MIME content type and explain?

8. What are the key algorithms used in S/MIME?

9. Give the steps for preparing envelope data MIME?

10.What are the services provided by PGP services

11. Explain the reasons for using PGP?

12. Why E-mail compatibility function in PGP needed?

13. Name any cryptographic keys used in PGP?

14.What is meant by S/MIME? (A/M-12)

15.List out the types of firewalls.

Part-B

1.Explain in detail about the PGP.

2.Explain in detail about the S/MIME.

3.Explain in detail about the Types of Firewalls in Internet Firewalls for Trusted System.

4. Explain in detail about the Firewall related terminology in Internet Firewalls for Trusted System.

5.Explain in detail about the SET for E-Commerce Transactions.