COMPUTER SECURITY AND CYBER LAW (CSCL)

LECTURER: ROLISHA STHAPIT/ DIKSHYA SINGH

UNIT 8

Web security and Email Security LH5

- Web security, Threats, SSL (Architecture, Handshake protocol, Handshake protocol action), overview of TLS and HTTPS, Secure Electronic Transaction overview, Dual Signature, Payment Processing, E-Mail, SMTP, PEM, PGP, Concept of Secure Email

- Web is now widely used by business, government, individuals but Internet & Web are vulnerable.
- The World Wide Web is fundamentally a client/server application running over the Internet and TCP/IP intranets.
- As such, the security tools and approaches discussed so far in this book are relevant to the issue of Web security.
- But, as pointed out in, the Web presents new challenges not generally appreciated in the context of computer and network security.

Web Security Threats

- Web Security Threats Table provides a summary of the types of security threats faced when using the Web.
- One way to group these threats is in terms of passive and active attacks.
- Passive attacks include eavesdropping on network traffic between browser and server and gaining access to information on a Web site that is supposed to be restricted.
- Active attacks include impersonating another user, altering messages in transit between client and server, and altering information on a Web site.
- Another way to classify Web security threats is in terms of the location of the threat: Web server, Web browser, and network traffic between browser and server.

- Issues of server and browser security fall into the category of computer system security but is also applicable to Web system security. Issues of traffic security fall into the category of network security.

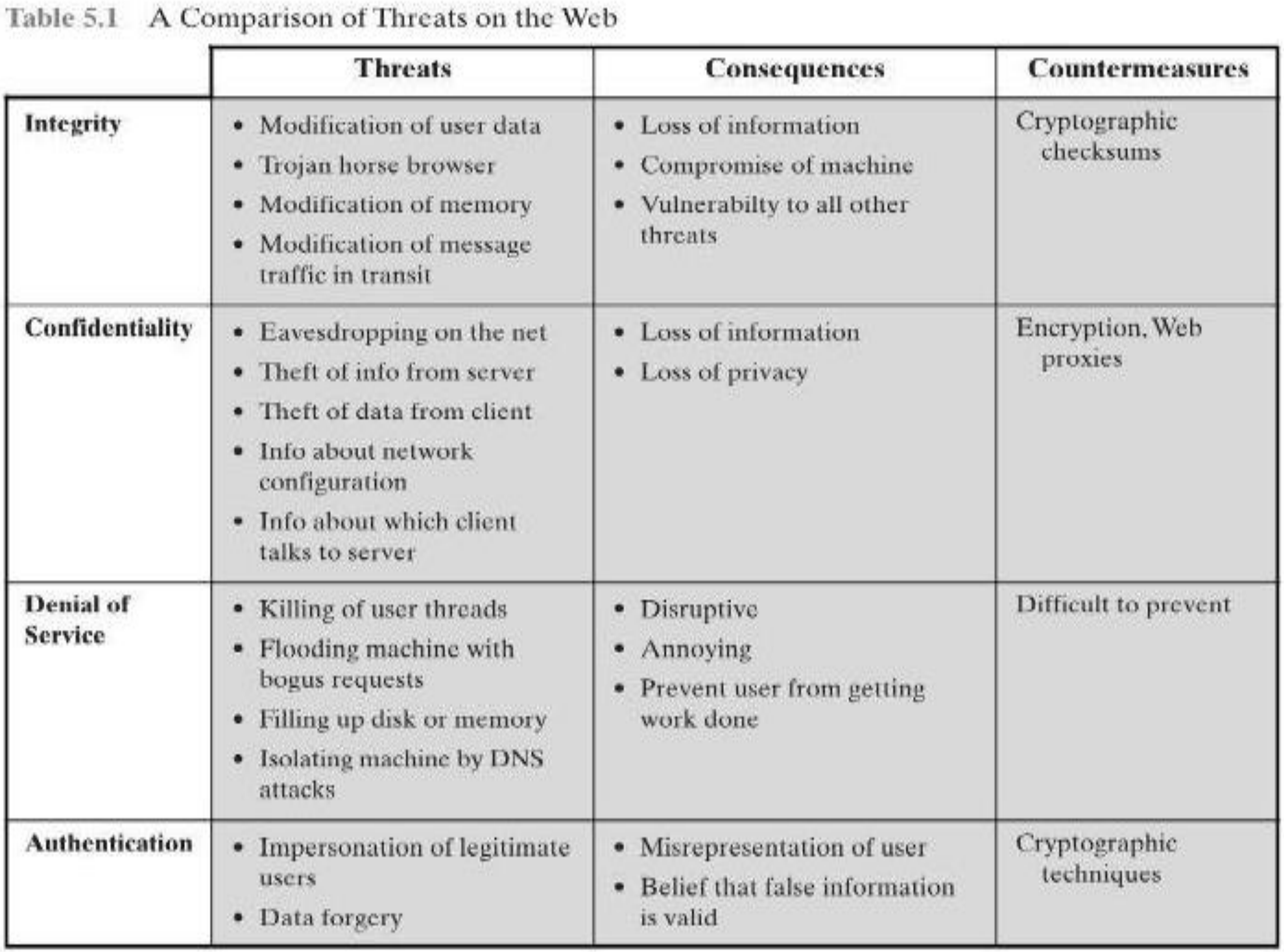
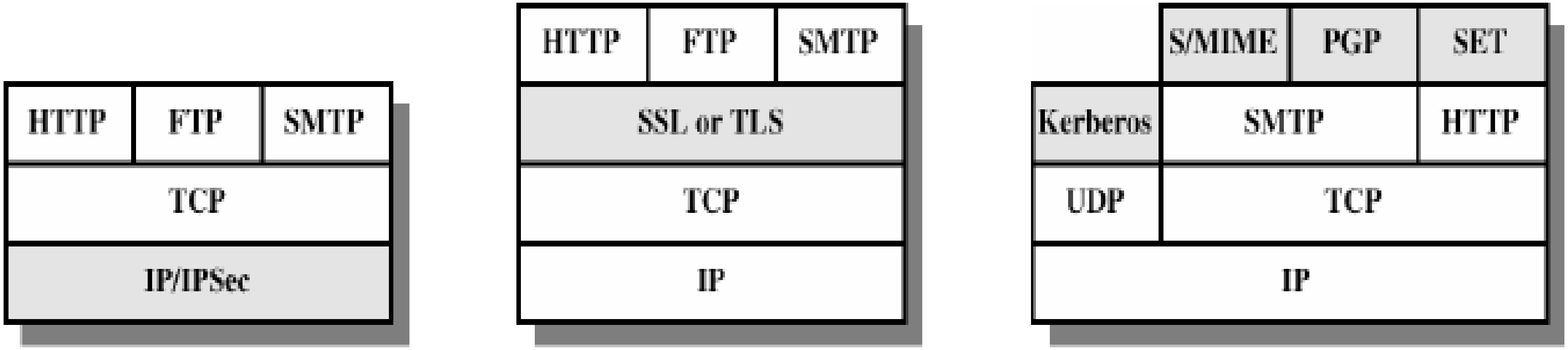


Table 5.1 A Comparison of Threats on the Web

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

Web Traffic Security Approaches

- A number of approaches providing web security are possible
  - □ Network level : IPSec
  - □ Transport level: SSL or TLS
  - □ Application level: SET



HTTP | FTP | SMTP
TCP
IP/IPSec

(a) Network Level

HTTP | FTP | SMTP
SSL or TLS
TCP
IP

(b) Transport Level

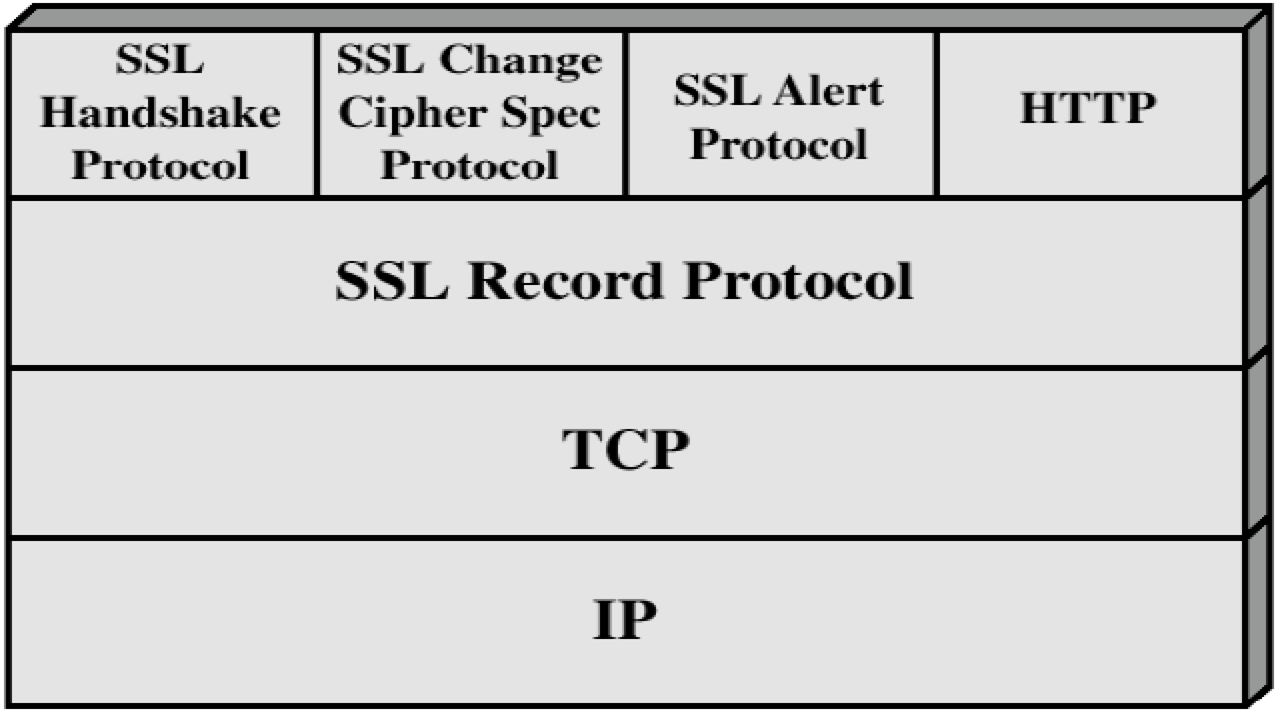
S/MIME | PGP | SET
Kerberos | SMTP | HTTP
UDP | TCP
IP

(c) Application Level

Security at the Transport Layer
Secured Socket Layer (SSL)

- It is a transport layer security service developed by Netscape, the version 3 designed with public review and from industry input and subsequently became Internet standard known as TLS (Transport Layer Security)
- It uses TCP to provide a reliable end-to-end service.

Figure: SSL Protocol Stack Architecture



SSL Handshake Protocol

SSL Change Cipher Spec Protocol

SSL Alert Protocol

HTTP

SSL Record Protocol

TCP

IP

- SSL is designed to make use of TCP to provide a reliable end-to-end secure service.
- SSL is not a single protocol but rather two layers of protocols, as illustrated in Figure above.
- The SSL Record Protocol provides basic security services to various higher- layer protocols.
- In particular, the Hypertext Transfer Protocol (HTTP), which provides the transfer service for Web client/server interaction, can operate on top of SSL.
- Three higher-layer protocols are defined as part of SSL**: the Handshake Protocol, The Change Cipher Spec Protocol, and the Alert Protocol.** These SSL specific protocols are used in the management of SSL exchanges.

- Two important SSL concepts are the SSL connection and the SSL session:

□ **Connection:** A connection is a network transport that provides a suitable type of service, such connections are transient, peer-to-peer relationships, associated with one session

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

- **SSL Record Protocol Services**

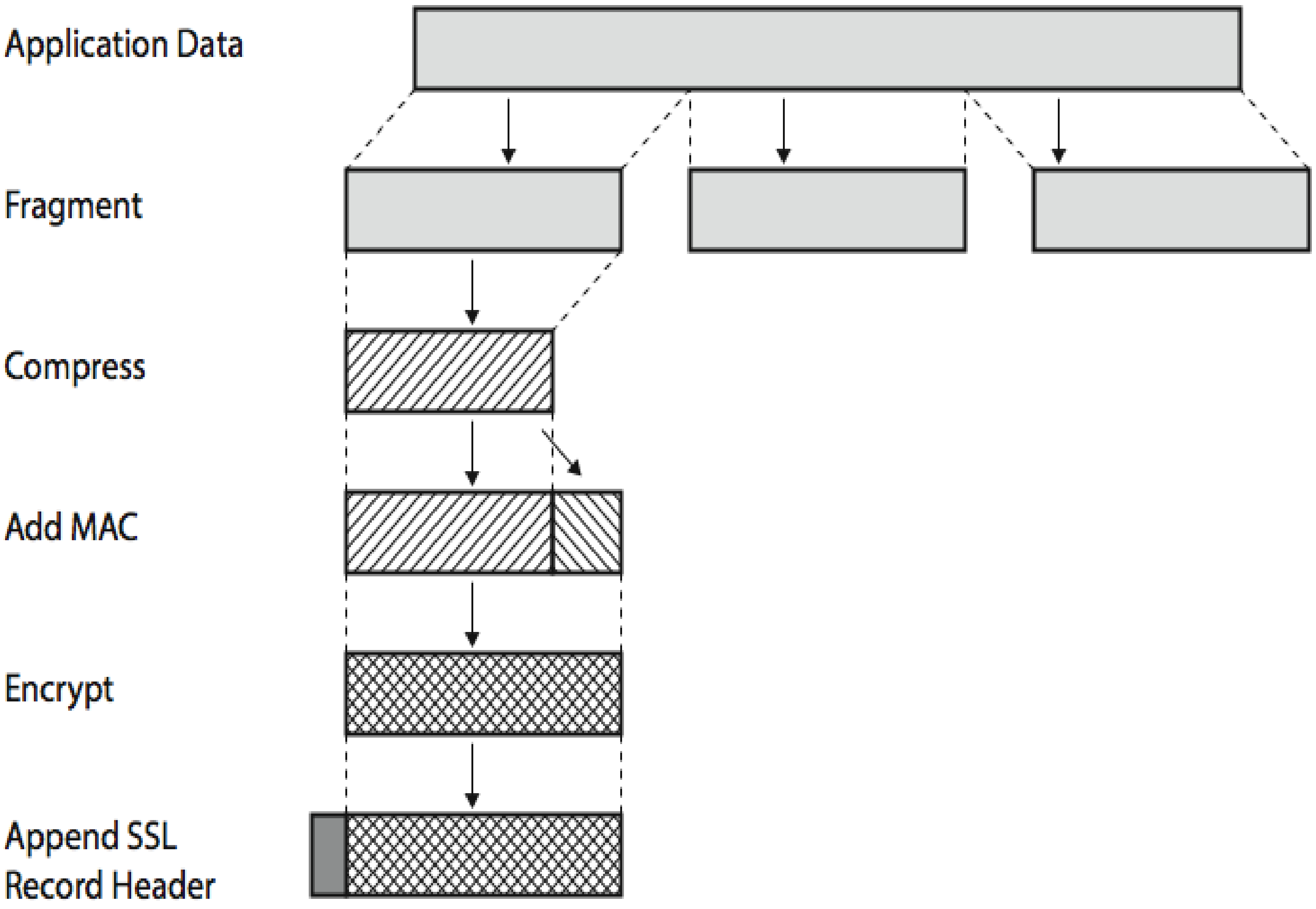
Message integrity:

- ➢ using a MAC with shared secret key
- ➢ similar to HMAC but with different padding

Confidentiality:

- ➢ using symmetric encryption with a shared secret key defined by Handshake Protocol
- ➢ AES, IDEA, RC2-40, DES-40, DES, 3DES, Fortezza, RC4-40, RC4-128
- ➢ message is compressed before encryption

SSL Record Protocol Operation



- The SSL Record Protocol takes an application to be transmitted , fragments the data into manageable blocks, optionally compresses the data, applies a MAC, encrypts , adds a header , and transmits the unit in a TCP segment.
- At the receiver, the received data are decrypted, verified, decompressed, and reassembled and then delivered to higher level users.

- **SSL Alert Protocol**
- The Alert Protocol is used to conveys SSL-related alerts to peer entity.
- Each message in this protocol consists of two bytes: First byte takes the value warning(1) or fatal(2) to convey the severity of message. If the level is fatal SSL immediately terminate the connection
- Second byte contain the code that indicates the specific alert, some of the fatal of this alerts are:

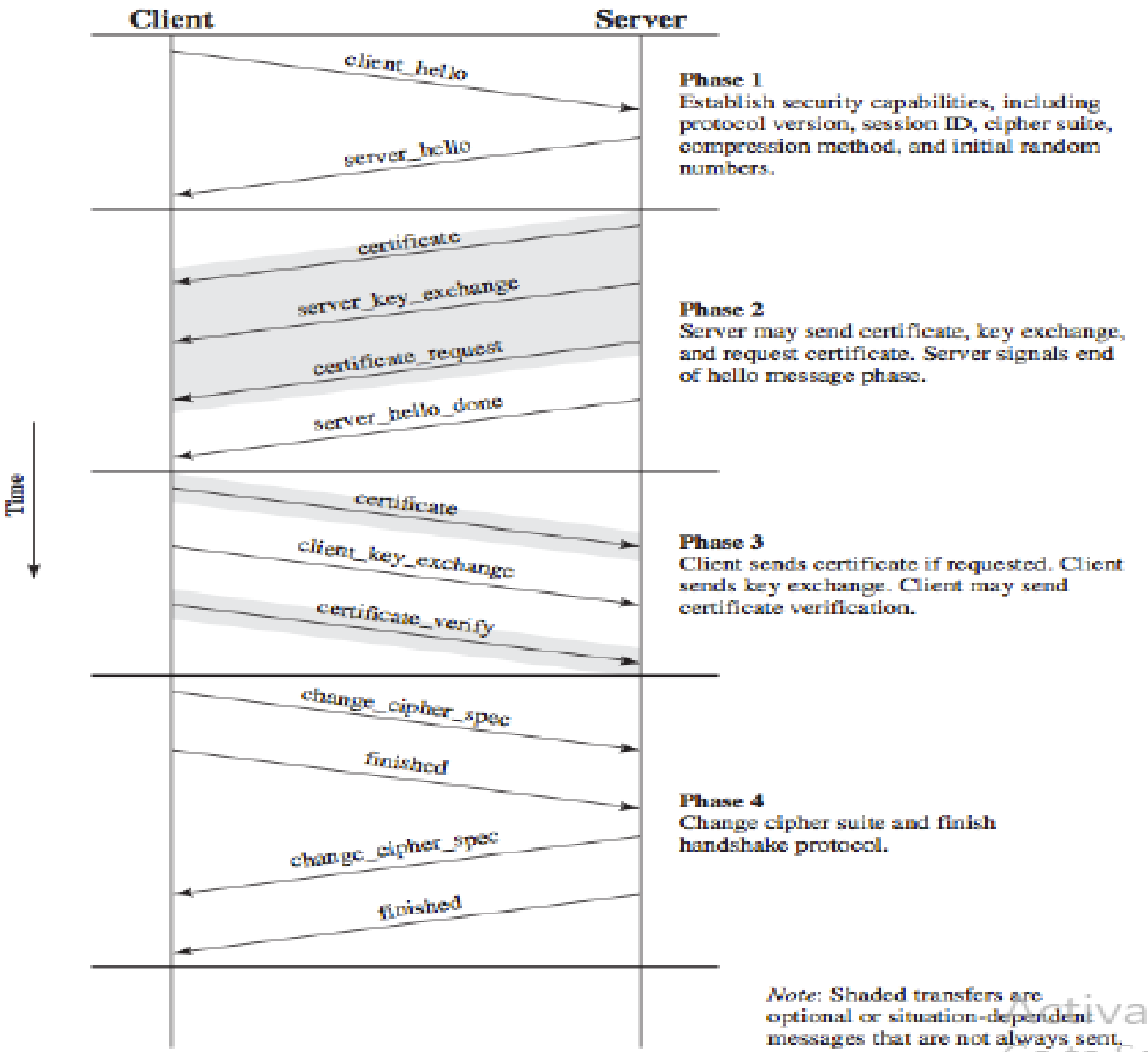
➢ Unexpected_message,bad_record_mac,decompression_failure, handshake_failure, illegal_parameter

➢ Close_notify,no_certificate,bad_certificate,unsupported_certifi cate,certificate_revoked,certificate_expired, certificate_unknown

Handshake Protocol

- The most complex part of SSL is the Handshake Protocol. This protocol allows the server and client to authenticate each other and to negotiate an encryption and MAC algorithm and cryptographic keys to be used to protect data sent in an SSL record.
- The Handshake Protocol is used before any application data is transmitted.
- The Handshake Protocol consists of a series of messages exchanged by client and server. All of these have the format. Each message has three fields:

- Type (1 byte): Indicates one of 10 messages.
- Length (3 bytes): The length of the message in bytes.
- Content (>=0bytes): The parameters associated with this message.



Phase 1
Establish security capabilities, including protocol version, session ID, cipher suite, compression method, and initial random numbers.

Phase 2
Server may send certificate, key exchange, and request certificate. Server signals end of hello message phase.

Phase 3
Client sends certificate if requested. Client sends key exchange. Client may send certificate verification.

Phase 4
Change cipher suite and finish handshake protocol.

Note: Shaded transfers are optional or situation-dependent messages that are not always sent.

Figure 5.5 Handshake Protocol Action

Phase 1: Establish Security Capabilities

- This phase is used to initiate a logical connection and to establish security capabilities.
- The exchange is initiated by client which sends client hello message with parameters such as protocol version, session Id, Cipher suite, and compression method
- After sending the client hello message the client waits for the server hello message with same parameters as the client hello message

Phase 2: Server Authentication and Key Exchange

- The Server begins this phase by sending its certificates, if it needs to be authenticated.
- The message contains one or chain of X.509 certificates.
- The server may send certificate, key exchange, and request certificate. The server signals end of hello message done

Phase 3: Client Authentication and Key Exchange

- In this phase of handshaking

➢ The client sends certificate if requested

➢ The client sends key exchange

➢ The client may send certificate verification

Phase 4: Finish

- This phase completes the setting up of a secure connection.
- The client sends a change_cipher_spec message
- Changes cipher suite and finish handshake protocol

Transport Layer Security(TLS)

- TLS is an IETF standardization initiative whose goal is to produce an Internet standard version of SSL.
- TLS is defined as a Proposed Internet Standard in RFC 5246. RFC 5246 is very similar to SSLv3.
- In this section, we highlight the differences.

Version Number

- The TLS Record Format is the same as that of the SSL Record Format and the fields in the header have the same meanings. The one difference is in version values. For the current version of TLS, the major version is 3 and the minor version is 3.

HTTPS

- HTTPS (HTTP over SSL) refers to the combination of HTTP and SSL to implement secure communication between a Web browser and a Web server. The HTTPS capability is built into all modern Web browsers.
- Its use depends on the Web server supporting HTTPS communication. For example, search engines do not support HTTPS.
- The principal difference seen by a user of a Web browser is that URL (uniform resource locator) addresses begin with https:// rather than http://. A normal HTTP connection uses port 80. If HTTPS is specified, port 443 is used, which invokes SSL.

- When HTTPS is used, the following elements of the communication are encrypted:

  - URL of the requested document

  - Contents of the document

  - Contents of browser forms (filled in by browser user)

  - Cookies sent from browser to server and from server to browser

  - Contents of HTTP header

Secure Electronic Transaction (SET)

- SET is an open encryption and security specification designed to protect credit card transactions on the Internet.
- This was emerged from a call for security standards by MasterCard and Visa.
- It is not itself a payment system but the set of security protocols and formats that enables users to employ the existing credit card payment infrastructure on an open network.
- SET protocol provides three services:

- Provides a secure communication channel among all parties involved in a transaction

- Provides trust by use of digital certificates

- Ensure privacy because the information is only available to parties in a transaction when and where necessary

SET Overview

Requirements: Following are few business requirements for secure payment processing with credit cards over the Internet or other networks:

- **Provide confidentiality of payment and ordering information:** It is necessary to assure cardholders that this information is safe and accessible only to the intended recipient. SET uses encryption to provide confidentiality.

- **Ensure the integrity of all transmitted data:** That is, ensure that no changes in content occur during transmission of SET messages. Digital signatures are used to provide integrity.

- **Provide authentication that a cardholder is a legitimate user of a credit card account:** A mechanism that links a cardholder to a specific account number reduces the incidence of fraud and overall cost of payment processing. Digital signatures and certificates are used to verify a legitimate card holder of valid account.

- **Provide authentication that a merchant can accept credit card transactions through its relationship with a financial institution:** This is complement to the preceding requirement. Cardholders need to be able to identify merchants with whom they can conduct secure transactions. Again, digital signatures and certificates are used.

- **Ensure the use of the best security practices and system design techniques to protect all legitimate parties in an electronic commerce transaction:** well tested on highly secure cryptographic algorithms and protocols are used.

- **Create a protocol that neither depends on transport security mechanisms nor prevents their use:** SET does not interfere with other security mechanisms like IPSec and SSL.

- **Facilitate and encourage interoperability among software and network providers:** SET protocols and formats are independent of hardware platform, operating system and web software.

Key Features of SET

To meet the requirements outlined above, SET incorporates the following features;

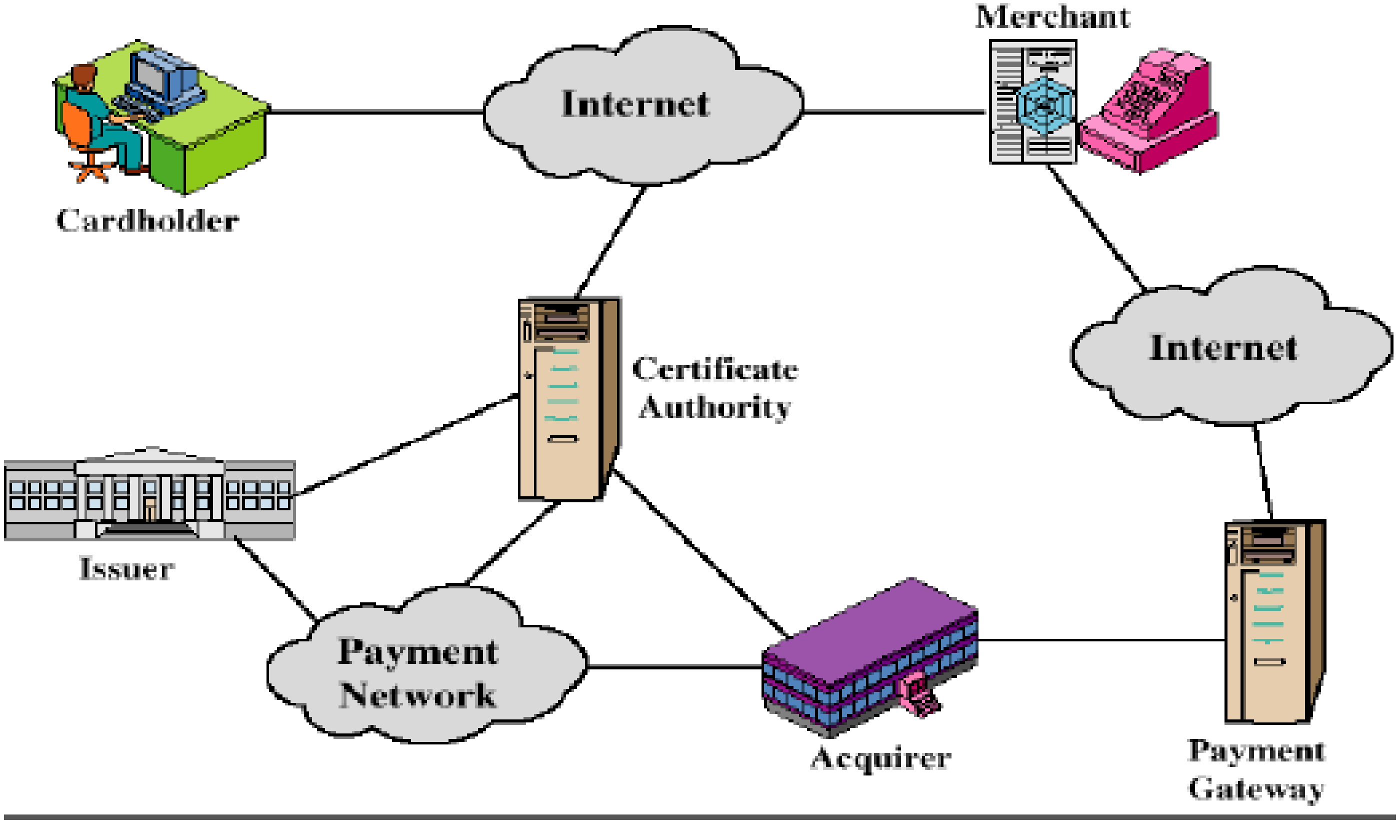
- **Confidentiality of information:** Cardholder account and payment information is secured as it travels across the network. An interesting and important feature if SET is that it prevents the merchant from learning the cardholder‘s credit card number; this is only provided to the issuing bank. Encryption is used to provide confidentiality.

- **Integrity of data:** Payment information sent from cardholders to merchants include order information, personal data, and payment instructions. SET guarantees that these message contents are not altered in transit. Digital signatures provide message integrity.

- **Cardholder account authentication:** SET enables merchants to verify that a cardholder is legitimate user of a valid card account number. Digital certificates are used for this purpose.

- **Merchant authentication:** SET enables cardholders to verify that a merchant has a relationship with a financial institution allowing it to accept payment cards. Digital certificates are used for this purpose.

SET Participants



- **Cardholder:** A cardholder is an authorized holder of a payment card (e.g., MasterCard, Visa) that has been issued by an issuer.
- **Merchant:** A merchant is a person or organization that has goods or services to sell to the cardholder. Typically, these services are offered via a website or by electronic mail. The merchant that accepts payment cards must have a relationship with an acquirer.
- **Issuer:** This is a financial institution, such as bank, that provides the card holder with the payment card and is responsible for payment of the debt of the cardholder.
- **Acquirer:** This is a financial institution that establishes an account with a merchant and processes payment card authorization and payments. Merchants usually accept more than one card, but they do not want to be associated with multiple issuers. In this case acquirer provides authorization to the merchant that a given card account is active and that the proposed purchase does not exceed the credit limit. The acquirer also provides electronic transfer of payments to the merchant‘s account. Later, the acquirer is reimbursed by the issuer over some sort of payment network for electronic fund transfer.

- **Payment gateway:** This is a function operated by an acquirer or designated third party that processes merchant payment messages. It interfaces between SET and the existing bankcard payment networks for authorization and payment functions.
- **Certification authority:** This is a trusted party that issues X.509v3 public key certificates for card holders, merchants and payment gateways. The success of SET will depend on the existence of a CA infrastructure available for this purpose. A hierarchy of CA is used so that participants need not be directly certified by a root authority.

SET Transaction

1. **The customer opens an account.** The customer obtains a credit card account with a bank that supports electronic payment and SET.

2. **The customer receives a certificate.** After suitable verification of identity, the customer receives X.509v3 digital certificate, which is signed by the bank. The certificate verifies the customers RSA public key and its expiration date. It also establishes a relationship, guaranteed by the bank, between the customer‘s key pair and his/her credit card.

3. **Merchants have their own certificates.** A merchant accepting card must possess two certificates for two public keys owned by the merchant: one for signing messages and the other for key exchange. The merchant also needs a copy of payment gateway‘s public key certificate.

4. **The customer places an order**. This is a process that may involve the customer first browsing through the merchant‘s web site to select items and determine the price. The customer sends a list of the items to be purchased to the merchant, who returns an order form containing the list of items, their price, a total price, and an order number.

5.**The merchant is verified.** The merchant sends a copy of its certificate so that the customer can verify that it's a valid store.

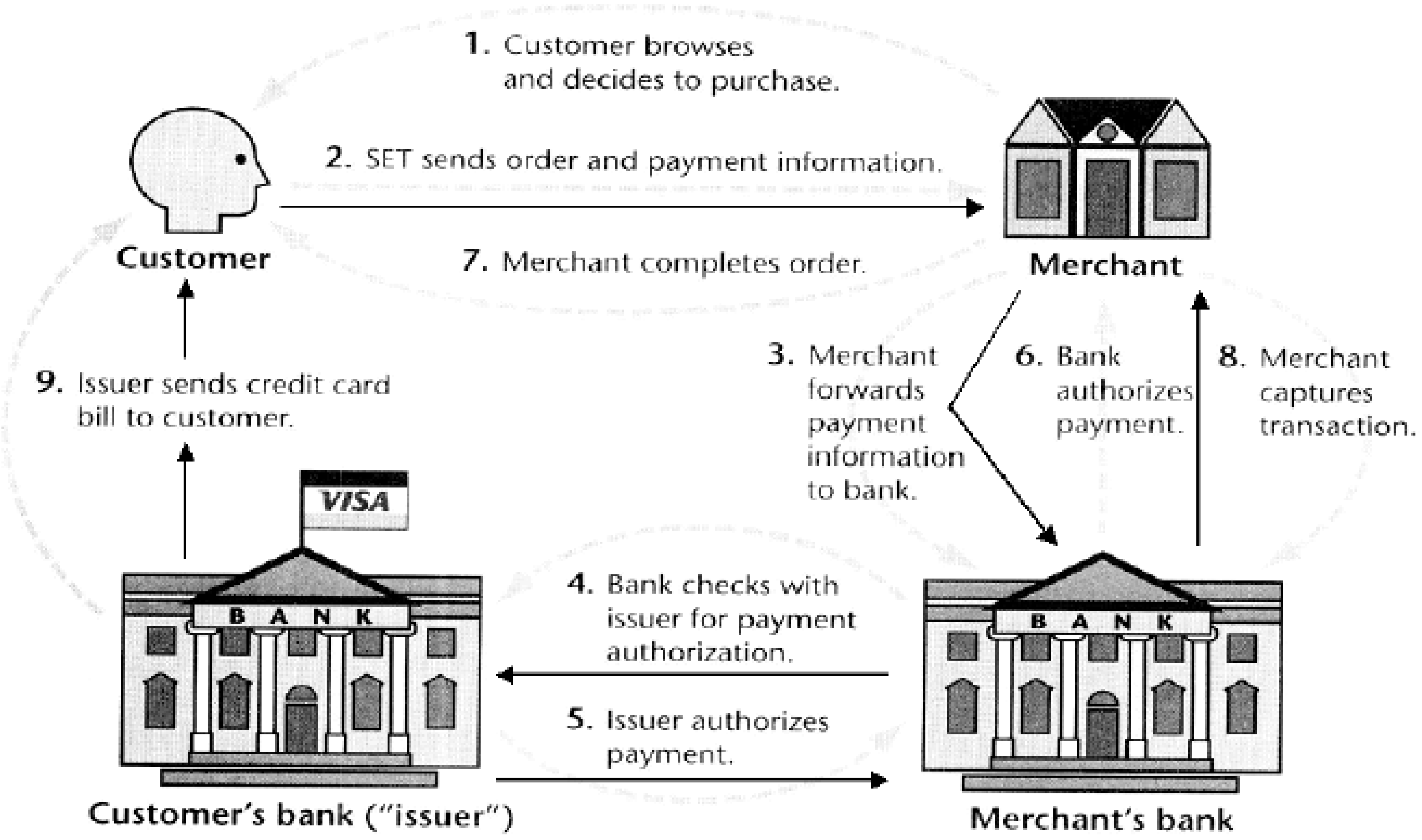
6. **The order and payment are sent.** The customer sends both order and payment information to the merchant along with his/her certificate. The order confirms the purchase of the items in the order form. The payment contains credit card details. The payment information is encrypted in such a way that the merchant cannot read it. The customer‘s certificate enables the merchant to verify the customer.

7. **The merchant requests payment authorization.** The merchant sends the payment information to the payment gateway, requesting authorization that the customer‘s available credit is sufficient for this purchase

8. **The merchant confirms the order**. The merchant sends confirmation of the order to the customer.

9. **The merchant ships the goods or provides the service to the customer**. **The merchant requests payment.** This information is sent to the payment gateway.

Fig: SET Transaction


1. Customer browses and decides to purchase.
Customer
2. SET sends order and payment information.
Merchant
7. Merchant completes order.
3. Merchant forwards payment information to bank.
6. Bank authorizes payment.
8. Merchant captures transaction.
9. Issuer sends credit card bill to customer.
VISA
BANK
4. Bank checks with issuer for payment authorization.
5. Issuer authorizes payment.
BANK
Customer's bank ("issuer")
Merchant's bank

Dual Signature

- The purpose of the dual signature is to link two messages that are intended for two different recipients but allows only one party to read each.

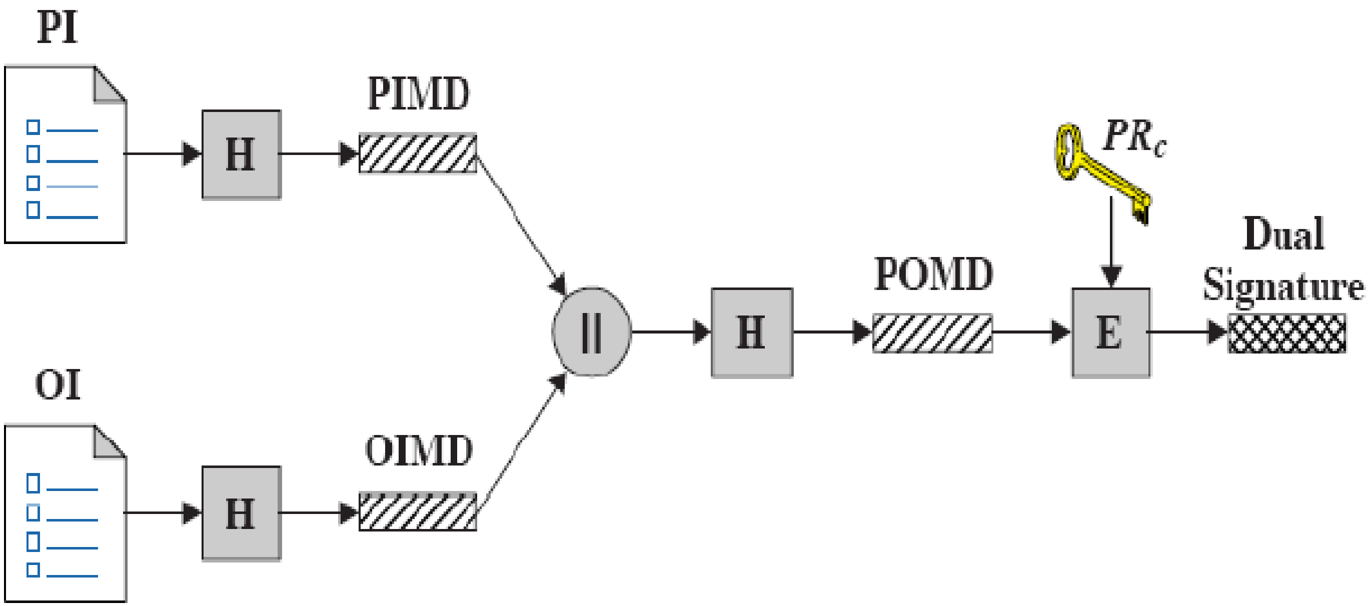
Dual Signature in SET

- The concept is to link two messages intended for two different receivers and they **are Order Information (OI)** from Customer to Merchant and **Payment Information (PI)** from Customer to Bank.
- The goal here is to limit information to a “Need-to-Know” basis i.e. merchant does not need credit card number, bank does not need details of customer order and afford the customer extra protection in terms of privacy by keeping these items separate.
- And again this link is needed to prove that payment is intended for this order and not some other one.

Need for Dual Signature

- Suppose that customers send the merchant two messages: The signed order information (OI) and the signed payment information (PI).
- In addition, the merchant passes the payment information (PI) to the bank. If the merchant can capture another order information (OI) from this customer, the merchant could claim this order goes with the payment information (PI) rather than the original.

Dual Signature Operation



PI	=	Payment Information	PIMD	=	PI message digest
OI	=	Order Information	OIMD	=	OI message digest
H	=	Hash function (SHA-1)	POMD	=	Payment Order message digest
||	=	Concatenation	E	=	Encryption (RSA)
			PR_c	=	Customer's private signature key

- The operation for dual signature is as follows:
  - Take hash (SHA-1) of payment and order information.
  - These two hash values are concatenated [H(PI) || H(OI)] and then the result is hashed.
  - Customer encrypts the final hash with a private key creating the dual signature.

$$DS= E_{KRC}[H(H(PI) || H(OI))]$$

Digital Signature Verification

- **By Merchant:** The merchant has the public key of the customer obtained from the customer‘s certificate. Now, the merchant can compute two values:

H(PIMD || H(OI)) and DKUC[DS] and they should be equal.

- **By Bank:** The bank is in possession of DS, PI, the message digest for OI, and the customer‘s public key, then the bank can compute the following:

H(H(PI) || OIMD) and DKUC [DS] and they must be same.

Here, merchant has received OI & verified the signature, bank has received PI & verified the signature and the customer has linked the OI and PI and can prove the linkage.

Payment Processing

- This process is broken down into two steps: **payment authorization and payment capture.**
- **Purchase Request:** Before the Purchase Request exchange begins, the cardholder has completed browsing, selecting, and ordering. The end of this preliminary phase occurs when the merchant sends a completed order form to the customer. The purchase request exchange involves four messages: Initiate Request, Initiate Response, Purchase Request and Purchase Response.

Payment Authorization

- The merchant sends an authorization request message to the payment gateway so as to ensure the receipt of the payment by the merchant. The payment authorization message consists of two messages**: authorization request and authorization response.**
- **Authorization Request:** Merchant sends this message to the payment gateway with:

Purchase-related information: this information was obtained from the customer with: PI, dual signature calculated over the PI & OI and signed with customer‘s private key, OI message digest (OIMD) and digital envelop.

Authorization-related information: This message is generated by merchant with: An authorization block having transaction ID - signed with merchant‘s private key and encrypted with one-time session key generated by merchant and digital envelope formed by encrypting the one time key with the payment gateway‘s public key.

Certificates: the merchants includes the cardholder‘s signature key certificate (for verifying the dual signature), the merchant‘s signature key certificate (for verifying the merchant‘s signature), and the merchant‘s key exchange certificate (needed in the payment gateway‘s response).

The payment gateway performs the following tasks

1. Verifies all certificates.

2. Decrypts authorization block digital envelope to obtain symmetric key and then decrypts the authorization block.

3. Verifies the merchant signature on authorization block

4. Decrypts payment block digital envelope to obtain symmetric key and decrypt block.

5. Verifies dual signature on payment block

6. Verifies that the transaction ID received from the merchant matches the PI received from the customer.

7. Requests and receives issuer‘s authorization.

- **Authorization Response:** After receiving the authorization from the issuer, the payment gateway returns this message to the merchant with:

Authorization-related Information: It includes an authorization block, signed with the gateway‘s private signature key and encrypted with a one time symmetric key generated by the gateway. It also includes a digital envelope with encrypted, one time key, using merchant‘s public key exchange key.

Capture Token Information: It is used to effect the payment later. It has same form as of above message. It is not processed but returned as is with the payment gateway.

Certificate: The gateways signature key certificate.

Payment Capture

- The payment gateway is engaged in payment capture transaction with capture request and capture response messages.
- **Capture Request:** For this message, the merchant generates, signs, and encrypts a capture request block that consists of: payment amount, transaction ID and also the encrypted capture token received in the authorization response for this transaction as well as the merchant‘s signature key and key –exchange key certificates.

Once the payment gateway receives the capture request message, it decrypts and verifies the capture request block and decrypts and verifies the capture token block and then checks the consistency among them. It then creates the clearing request that is sent to the issuer over the private payment network. The gateway then notifies the merchant of payment in capture response message.

- **Capture Response:** This message includes a capture response block that the gateway signs and encrypts. It also includes gateway‘s signature key certificate. The merchant software stores the capture response to be used for reconciliation with payment received from the acquirer.

SET Overhead

Simple purchase transaction using SET has

– Four messages between merchant and customer

– Two messages between merchant and payment gateway

– 6 digital signatures

– 9 RSA encryption/decryption cycles

– 4 DES encryption/decryption cycles

– 4 certificate verifications

– Multiple servers need copies of all certificates

E-mail (Electronic-mail)

- Many of the electronic mail systems today are already connected together in networks, so that users can send mail to each other, regardless of which mail system each of them is connected to.
- In the future, almost all systems will be connected in this way. This means that all the electronic mail systems, when connected, behave as one large system.
- This large system may eventually be comparable in size and complexity to the world-wide international telephone network, but will have more advanced technical functions, and will be more of a data-processing system than the telephone network.

- Electronic mail, as defined, has the following properties:

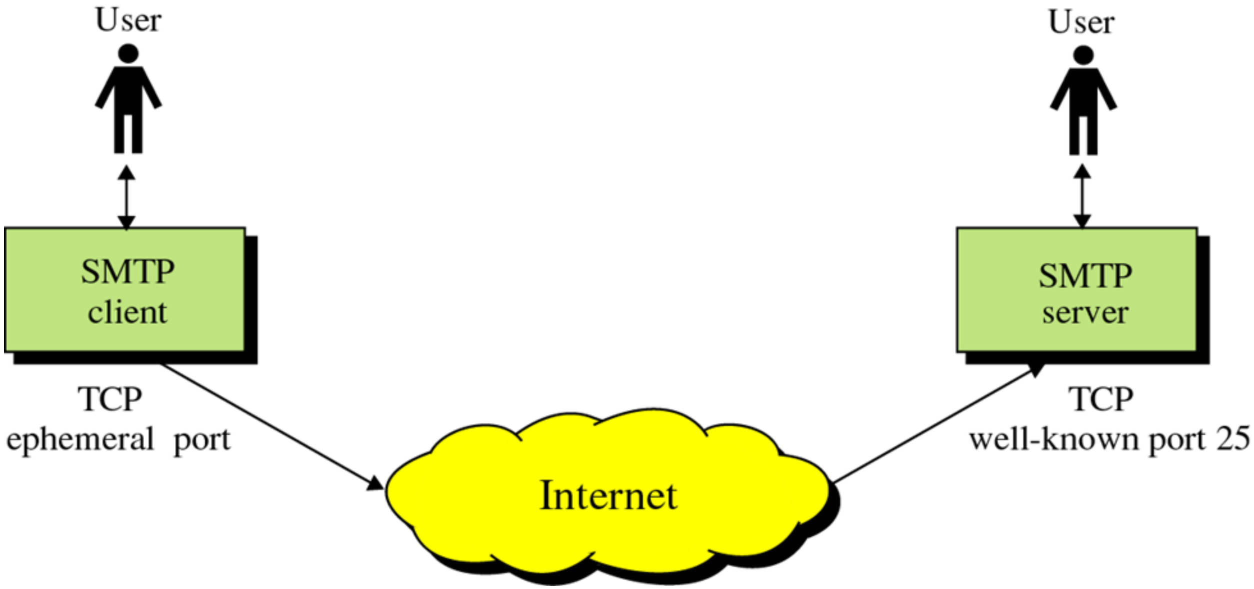
- The user produces, sends, and usually also receives mail at a computer screen, a terminal, or a personal computer.

- The messages sent have a data structure, which can be handled by a computer. This structure can be more or less advanced: it can, for example, allow the user to ask his computer to find the last received letter from person N about the subject XYZ, or to find the outgoing message, to which a certain incoming message replies.

Simple Mail Transfer Protocol (SMTP):

- SMTP is an Internet standard for electronic mail (e-mail) transmission across Internet Protocol (IP) networks originated in 1982 (rfc0821, Jon Postel).
- Its main goal is to transfer mail reliably and efficiently.
- SMTP was first defined by RFC 821 and last updated by RFC 5321 which includes the extended SMTP (ESMTP) additions, and is the protocol in widespread use today.
- SMTP uses TCP port 25. The protocol for new submission is effectively the same as SMTP, but it uses port 587 instead. SMTP connections secured by SSL are known by the shorthand SMTPS, though SMTPS is not a protocol in its own right.

- While electronic mail servers and other mail transfer agents use SMTP to send and receive mail messages, user-level client mail applications typically only use SMTP for sending messages to a mail server for relaying.
- For receiving messages, client applications usually use either the Post Office Protocol (POP) or the Internet Message Access Protocol (IMAP) or a proprietary system to access their mail box accounts on a mail server.



Problem: SMTP doesn't support inherent security (Authentication, Encryption). It only uses NVT (Network Virtual Terminal) 7-bit ASCII format. It doesn't support audio, video etc.

RFC 822

- RFC 822 defines a format for text messages that are sent using electronic mail.
- It has been the standard for Internet-based text mail message and remains in common use.
- In the RFC 822 context, messages are viewed as having an envelope and contents. The envelope contains whatever information is needed to accomplish transmission and delivery.
- The contents compose the object to be delivered to the recipient. The RFC 822 standard applies only to the contents.
- However, the content standard includes a set of header fields that may be used by the mail system to create the envelope, and the standard is intended to facilitate the acquisition of such information by programs.

- The overall structure of a message that conforms to RFC 822 is very simple.
- A message consists of some number of header lines (the header) followed by unrestricted text (the body).
- The header is separated from the body by a blank line.
- Put differently, a message is ASCII text, and all lines up to the first blank line are assumed to be header lines used by the user agent part of the mail system.

A header line usually consists of a keyword, followed by a colon, followed by the keyword's arguments; the format allows a long line to be broken up into several lines. The most frequently used keywords are From, To, Subject, and Date. Here is an example message:

- Date: Tue, 16 Jan 1998 10:37:17 (EST)
- From: "William Stallings" <ws@shore.net>
- Subject: The Syntax in RFC 822
- To: Smith@Other-host.com
- Cc: Jones@Yet-Another-Host.com

- Hello. This section begins the actual message body, which is delimited from the message heading by a blank line.

Another field that is commonly found in RFC 822 headers is Message-ID. This field contains a unique identifier associated with this message.

Pretty Good Privacy (PGP):

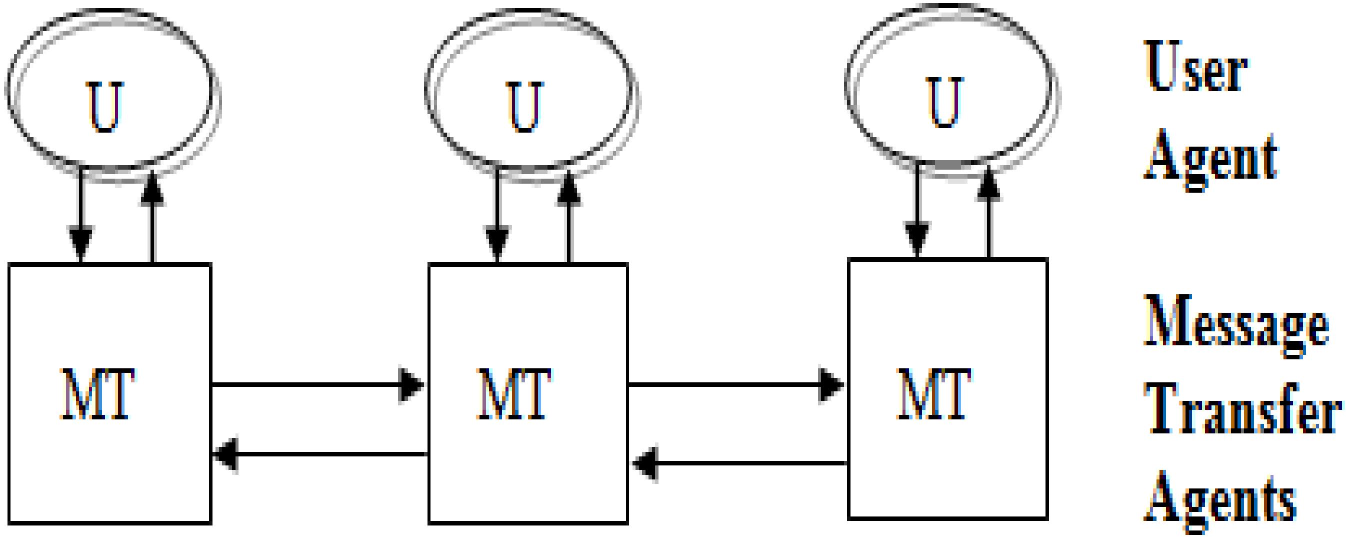
- PGP is a public key encryption package to protect e-mail and data files.
- It lets you communicate securely with people you've never met, with no secure channels needed for prior exchange of keys.
- It's well featured and fast, with sophisticated key management, digital signatures, data compression, and good ergonomic design.
- The actual operation of PGP is based on five services: authentication, confidentiality, compression, e-mail compatibility, and segmentation.

- PGP provides authentication via a digital signature scheme.
- PGP provides confidentiality by encrypting messages before transmission
- PGP compresses the message after applying the signature and before encryption. The idea is to save space.
- PGP encrypts a message together with the signature (if not sent separately) resulting into a stream of arbitrary 8-bit octets. But since many e-mail systems permit only use of blocks consisting of ASCII text, PGP accommodates this by converting the raw 8-bit binary streams into streams of printable ASCII characters using a radix-64 conversion scheme. On receipt, the block is converted back from radix-64 format to binary.

- To accommodate e-mail size restrictions, PGP automatically segments email messages that are too long. However, the segmentation is done after all the housekeeping is done on the message, just before transmitting it. So the session key and signature appear only once at the beginning of the first segment transmitted. At receipt, the receiving PGP strips off all e-mail headers and reassembles the original mail

Privacy Enchanced Mail (PEM):

- PEM is an Internet standard for providing security services to electronic mail.
- It uses cryptographic techniques to provide message integrity checking, originator authentication, and confidentiality.
- It lets you know that a message hasn't been changed, who it's from, and, optionally, allows you to keep it secret from all but the intended recipients.
- The figure below shows a typical network mail service.
- The U (user agent) interacts directly with the sender.
- When the message is composed, the U hands it to the MT (message transport, or transfer, agent).
- The MT transfers the message to its destination host, or to another MT, which in turn transfers the message further.
- At the destination host, the MT invokes a user agent to deliver the message.


U
U
U
User
Agent
MT
MT
MT
Message
Transfer
Agents

- An attacker can read electronic mail at any of the computers on which MTs handling the message reside, as well as on the network itself.
- An attacker could also modify the message without the recipient detecting the change.
- Because authentication mechanisms are minimal and easily evaded, a sender could forge a letter from another and inject it into the message handling system at any MT, from which it would be forwarded to the destination.
- Finally, a sender could deny having sent a letter, and the recipient could not prove otherwise to a disinterested party.
- These four types of attacks (violation of confidentiality, authentication, message integrity, and nonrepudiation) make electronic mail nonsecure.
- So IETF with the goal of e-mail privacy develop electronic mail protocols that would provide the following services.

1. Confidentiality, by making the message unreadable except to the sender and recipient(s)

2. Origin authentication, by identifying the sender precisely

3. Data integrity, by ensuring that any changes in the message are easy to detect

4. Nonrepudiation of origin (if possible)

The protocols were named Privacy-Enhanced Electronic Mail (or PEM).

Design Principles

Design goals of PEM were as follows:

1. Not to redesign existing mail system or protocols

2. To be compatible with a range of MTAs, UAs, and other computers. The protocols must work with a wide range of software, including software in all environments that connect to the Internet.

3. To make privacy enhancements available separately, so they are not required. A new protocol provides specific services, the user should be able to use the services desired, which may (or may not) be all the ones that the protocol provides.

4. To enable two parties to use the protocol to communicate without prearrangement. Arranging a communications key out of band (such as in person or over the telephone) can be time-consuming and prone to error. Furthermore, callers must authenticate themselves to the recipients. This is difficult and is another error-prone operation.

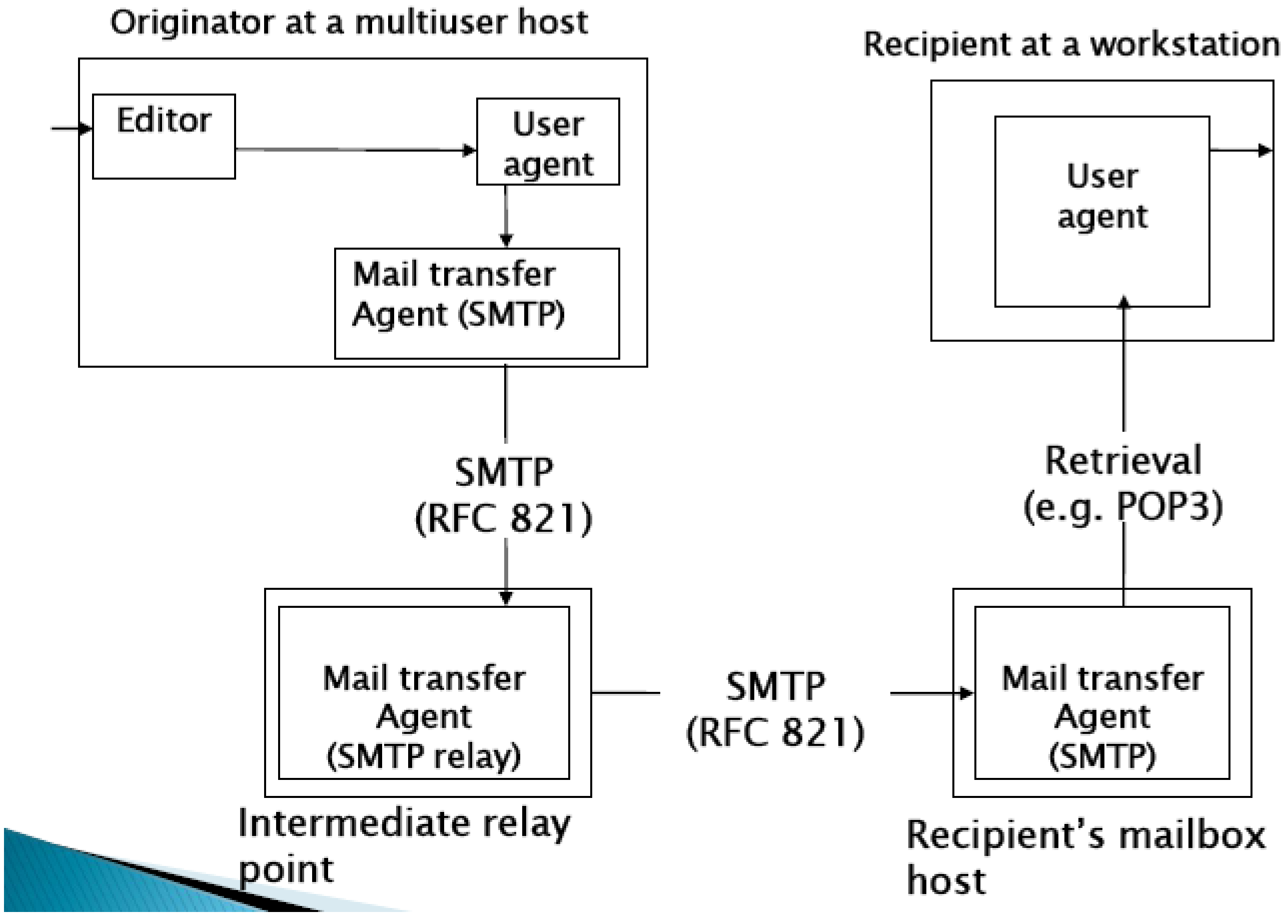
PEM vs. PGP

– **Use of different ciphers:** PGP uses IDEA cipher but PEM uses DES in CBC mode.

– **Use of certificate models:** PGP uses general —web of trust‖ but PEM uses hierarchical certification structure

– **Handling end of line:** PGP remaps end of line if message tagged —text, but leaves them alone if message tagged —binary whereas PEM always remaps end of line.

Integration Of PEM Into Existing Mail System


Originator at a multiuser host
Editor
User agent
Mail transfer Agent (SMTP)
SMTP (RFC 821)
Mail transfer Agent (SMTP relay)
Intermediate relay point
SMTP (RFC 821)
Mail transfer Agent (SMTP)
Recipient's mailbox host
Retrieval (e.g. POP3)
Recipient at a workstation
User agent

MIME

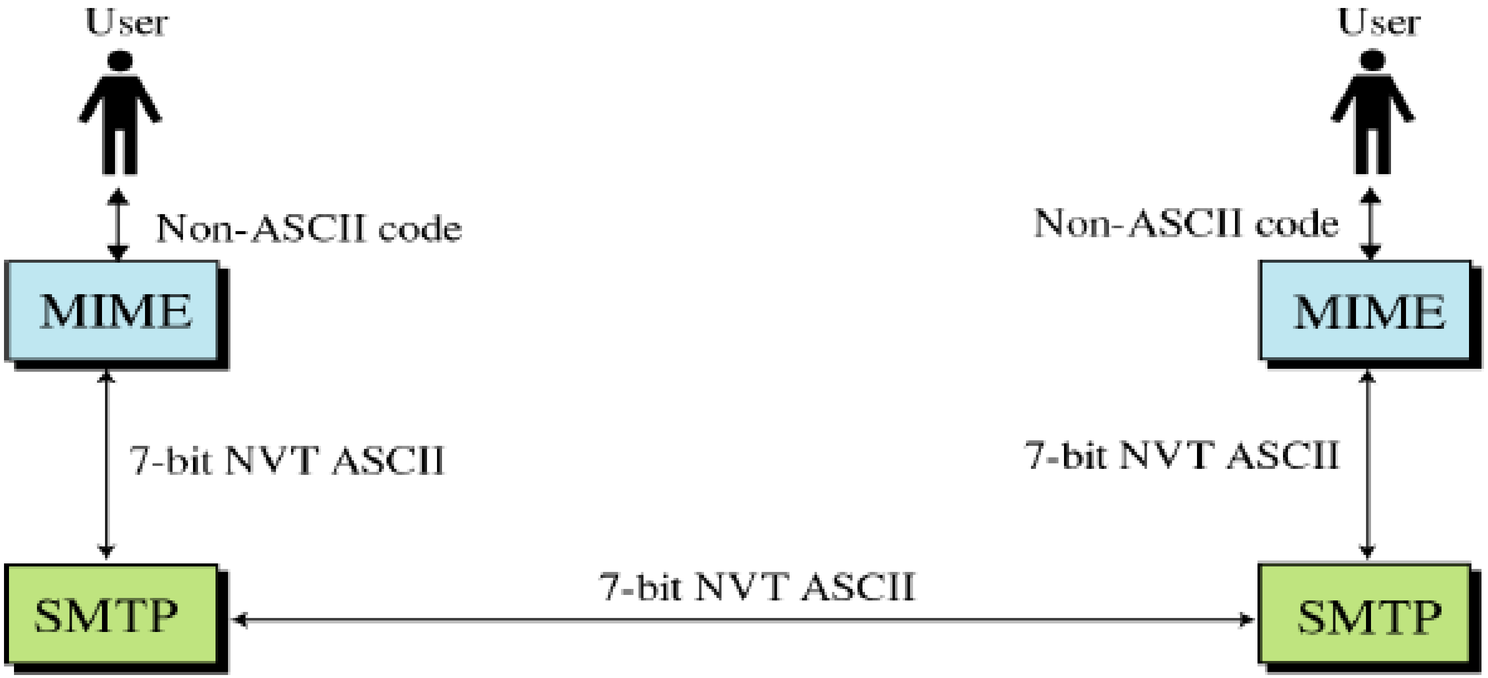
- MIME (Multi-Purpose Internet Mail Extensions) is an extension of the original Internet e-mail protocol that lets people use the protocol to exchange different kinds of data files on the Internet: audio, video, images, application programs, and other kinds, as well as the ASCII text handled in the original protocol, the Simple Mail Transport Protocol (SMTP).
- Servers insert the MIME header at the beginning of any Web transmission. Clients use this header to select an appropriate "player" application for the type of data the header indicates.
- Multipurpose Internet Mail Extension (MIME) is an extension to the RFC 5322 framework that is intended to address some of the problems and limitations of the use of Simple Mail Transfer Protocol (SMTP), defined in RFC 821, or some other mail transfer protocol and RFC 5322 for electronic mail.

- The MIME specification includes the following elements.

1. Five new message header fields are defined, which may be included in an RFC 5322 header. These fields provide information about the body of the message.

2. A number of content formats are defined, thus standardizing representations that support multimedia electronic mail.

3. Transfer encodings are defined that enable the conversion of any content format into a form that is protected from alteration by the mail system.



User

Non-ASCII code

MIME

7-bit NVT ASCII

SMTP

7-bit NVT ASCII

User

Non-ASCII code

MIME

7-bit NVT ASCII

SMTP

S/MIME(Secure/Multipurpose Internet Mail Extension)

- S/MIME (Secure/Multipurpose Internet Mail Extension) is a security enhancement to the MIME Internet e-mail format standard, based on technology from RSA Data Security.
- Although both PGP and S/MIME are on an IETF standards track, it appears likely that S/MIME will emerge as the industry standard for commercial and organizational use, while PGP will remain the choice for personal e-mail security for many users.
- S/MIME is defined in a number of documents, most importantly RFCs 3369, 3370, 3850 and 3851.

S/MIME Functionality

- In terms of general functionality, S/MIME is very similar to PGP.
- Both offer the ability to sign and/or encrypt messages.

Functions

S/MIME provides the following functions:

Enveloped data: This consists of encrypted content of any type and encrypted-content encryption keys for one or more recipients.

Signed data: A digital signature is formed by taking the message digest of the content to be signed and then encrypting that with the private key of the signer. The content plus signature are then encoded using base64 encoding. A signed data message can only be viewed by a recipient with S/MIME capability.

- Clear-signed data: As with signed data, a digital signature of the content is formed. However, in this case, only the digital signature is encoded using base64. As a result, recipients without S/MIME capability can view the message content, although they cannot verify the signature

Concept of Secure Email

- Email security refers to the collective measures used to secure the access and content of an email account or service.
- It allows an individual or organization to protect the overall access to one or more email addresses/accounts.
- An email service provider implements email security to secure subscriber email accounts and data from hackers - at rest and in transit.

- Email security is a broad term that encompasses multiple techniques used to secure an email service. From an individual/end user standpoint, proactive email security measures include:
    - Strong passwords
    - Password rotations
    - Spam filters
    - Desktop-based anti-virus/anti-spam applications

- Similarly, a service provider ensures email security by using strong password and access control mechanisms on an email server; encrypting and digitally signing email messages when in the inbox or in transit to or from a subscriber email address. It also implements firewall and software-based spam filtering applications to restrict unsolicited, untrustworthy and malicious email messages from delivery to a user’s inbox.

ASSIGNMENT

- Explain the SET transaction with diagram.
- Explain the SET participants with diagram.
- Explain dual signature with diagram.
- Explain SSL handshake protocol with diagram.
- Explain SSL architecture.
- Explain PEM and PGP.