Cryptography and Network Security

Unit – 4----- Part-1

Presentation by

Capt Dr Ravindra Babu Kallam

Topics to be Covered:

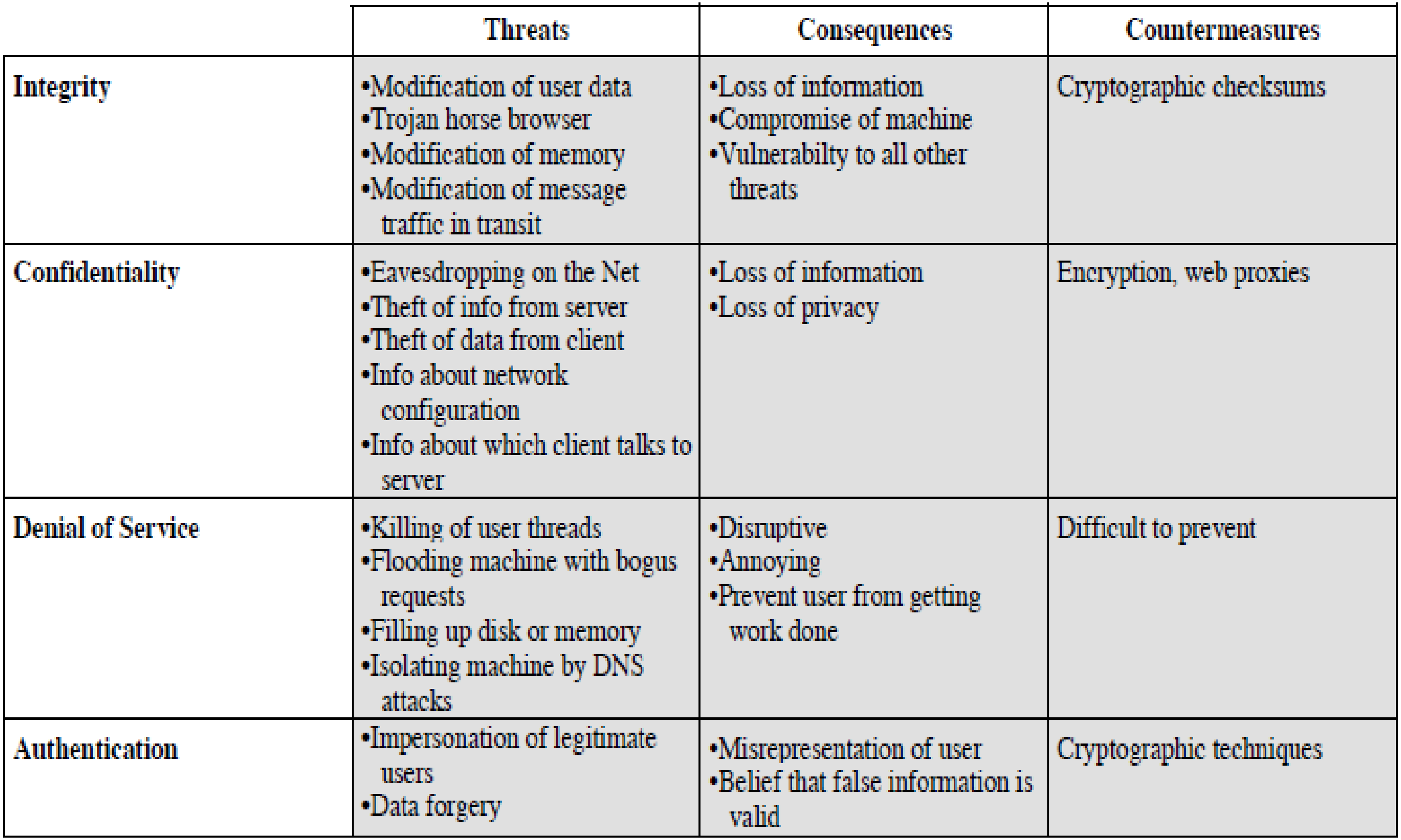
Web Security:

- Web Security considerations
- Security Socket Layer
- Transport layer security
- Secure electronic transaction
- HTTPS
- Secure Shell(SSH)

Web Security

- Web now widely used by business, government, individuals but Internet & Web are vulnerable
- have a variety of threats
  - Integrity
  - confidentiality
  - denial of service
  - authentication
- need added security mechanisms

Comparison of threats on the web



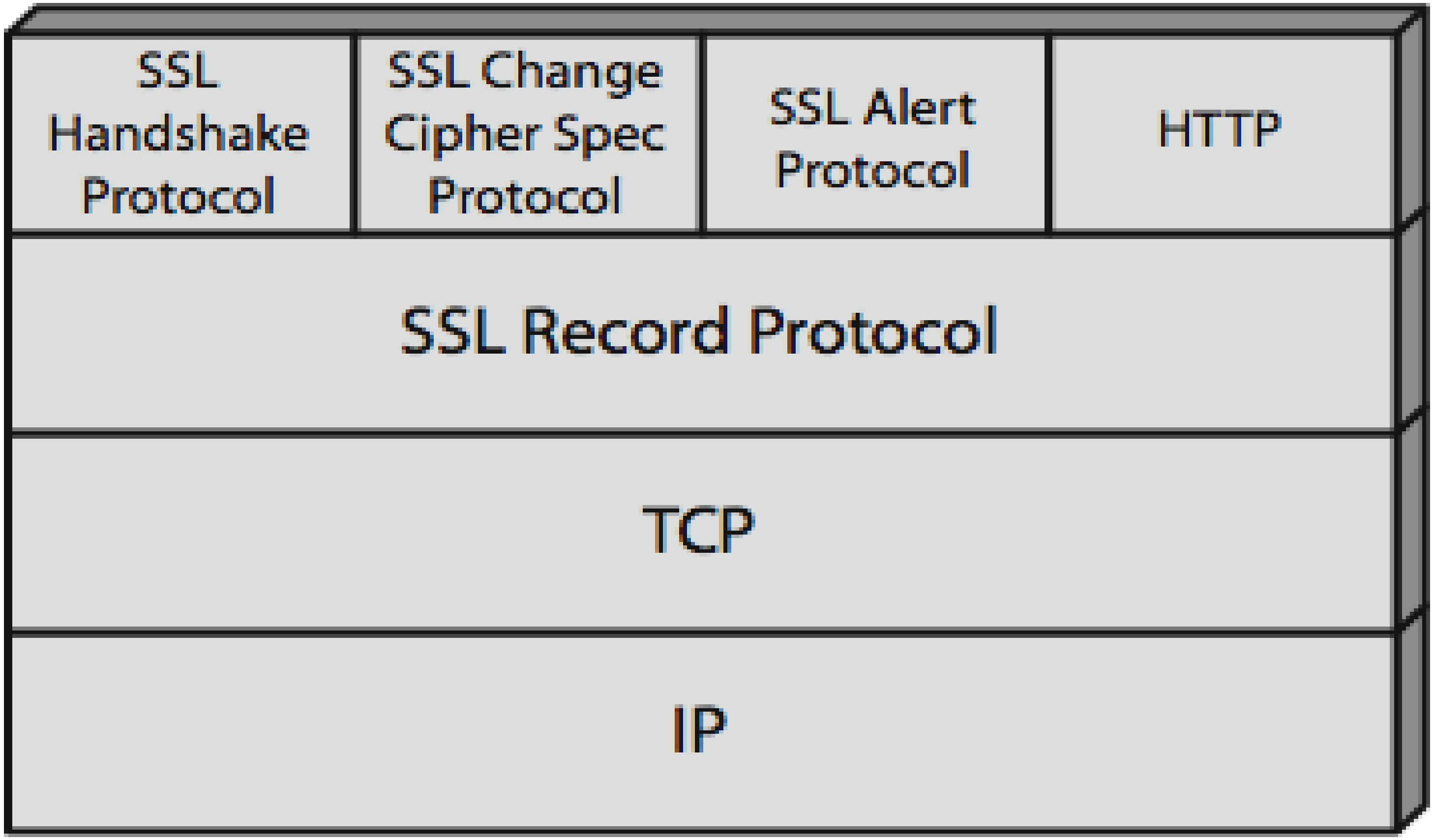
	Threats	Consequences	Countermeasures
Integrity	•Modification of user data •Trojan horse browser •Modification of memory •Modification of message traffic in transit	•Loss of information •Compromise of machine •Vulnerabilty to all other threats	Cryptographic checksums
Confidentiality	•Eavesdropping on 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

SSL (Secure Socket Layer)

- SSL is most widely used Web security mechanism
- SSL provides security services between TCP and applications that use TCP.
- The Internet standard version is called transport layer service (TLS)
- SSL/TSL provides confidentiality using symmetric encryption and message integrity using a MAC.
- SSL/TSL includes protocol mechanisms to enable two TCP users to determine the security mechanisms and services they will use.
- SSL is designed to make use of TCP to provide a reliable end-to-end secure service.
- SSL has two layers of protocols

- **SSL 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
- **SSL session**
  - **an association between client & server**
  - **created by the Handshake Protocol**
  - **define a set of cryptographic parameters, which may be shared by multiple SSL connections**
  - **Sessions are used to avoid the expensive negotiation of new security parameters for each connection**

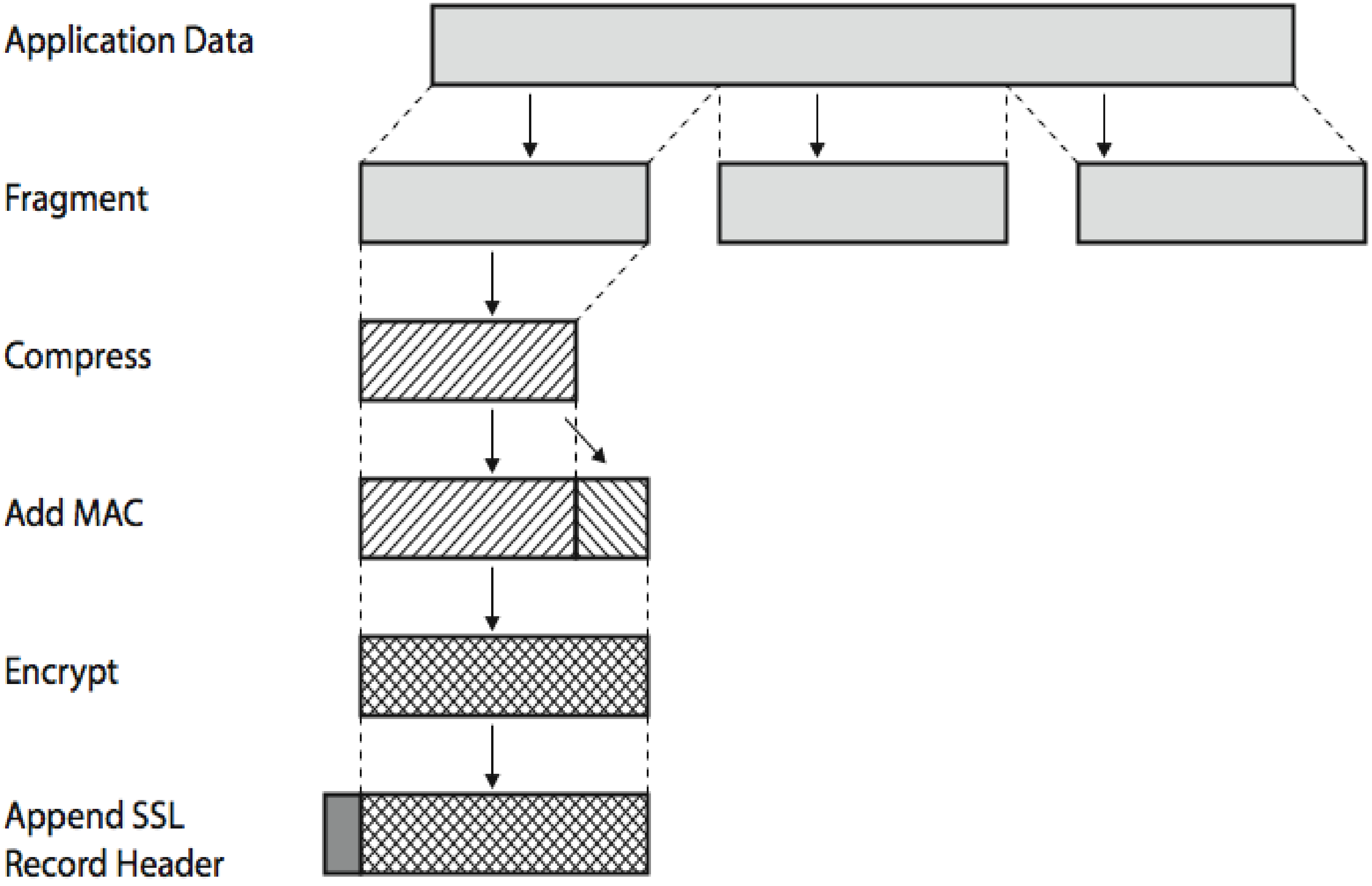
SSL Architecture



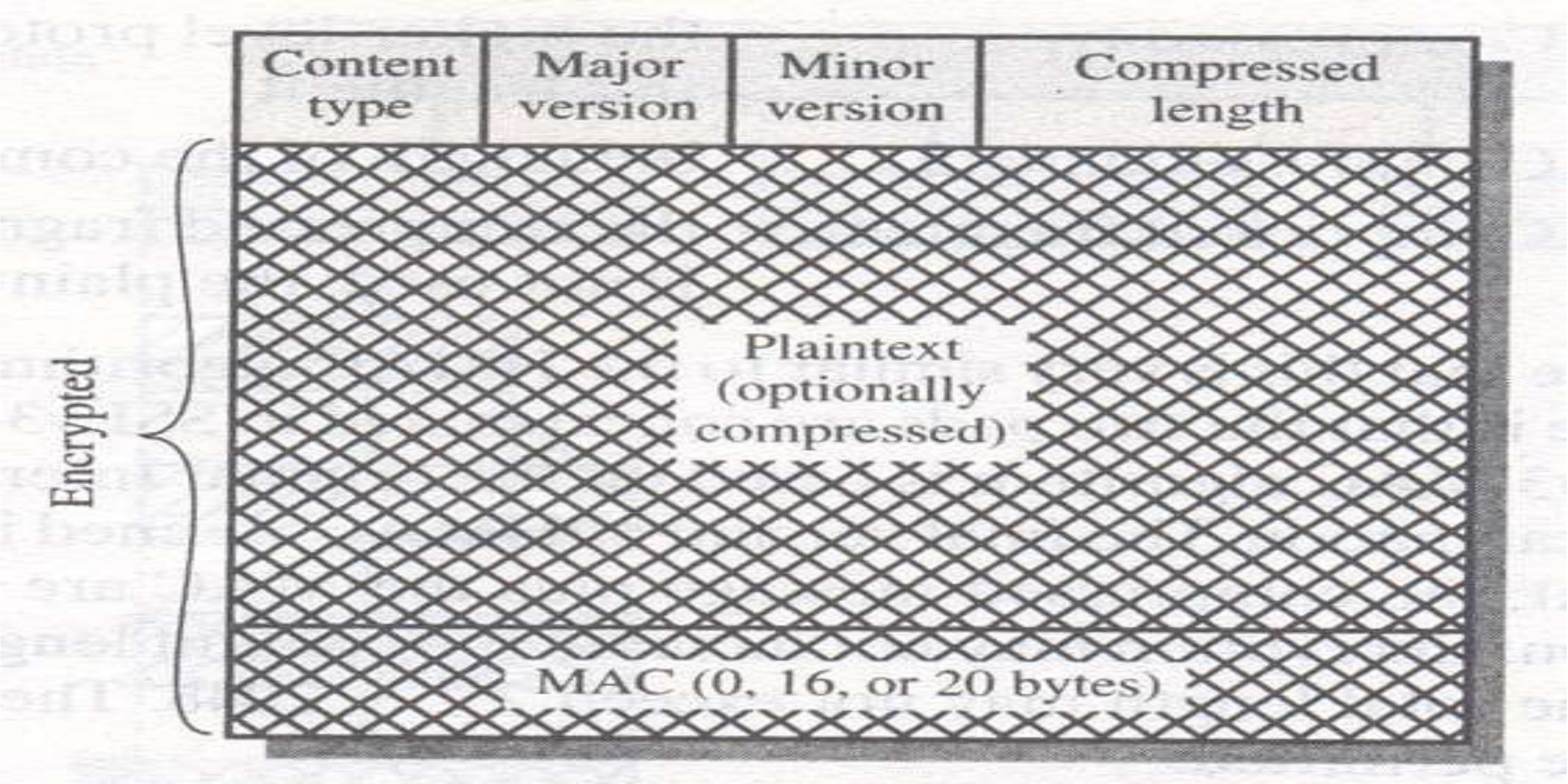
SSL Record Protocol Services

- **message integrity:** **It is provided with MAC.** The Handshake Protocol also defines a shared secret key that is used to form a message authentication code (MAC), which is similar to HMAC
- **confidentiality**
  - It is provided by using symmetric encryption with a shared secret key defined by Handshake Protocol
  - Ex: AES, IDEA, RC2, DES, 3DES
  - message is compressed before encryption

SSL Record Protocol Operation



SSL Record Format



Major version-3 & Minor version -0

- Content type: an 8 bit information indicates the type of content in the Msg. Ex: alert, application data, hand shake, etc
- Major Version(8bits): Indicates major version of SSL in use for SSLv3, the value is 3
- Minor Version(8bits): Indicates minor version is in use. For SSLv0 the value is 0.
- Compressed length (16bits): The length in bytes of the plaintext fragment after compression.

SSL Change Cipher Spec Protocol

- one of 3 SSL specific protocols which use the SSL Record protocol
- a single message of 1 byte
- causes pending state to be copied into the current state, which updates the cipher suite to be used on this connection.

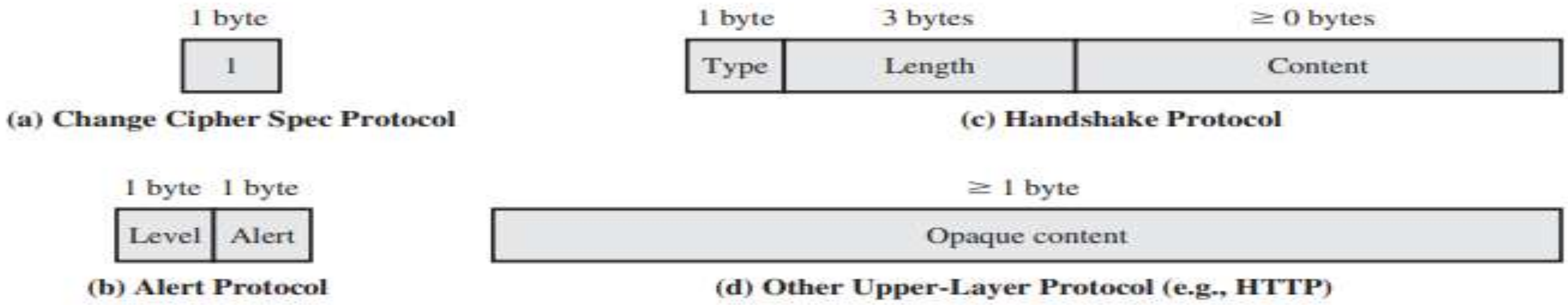


Figure 17.5 SSL Record Protocol Payload

SSL Alert Protocol

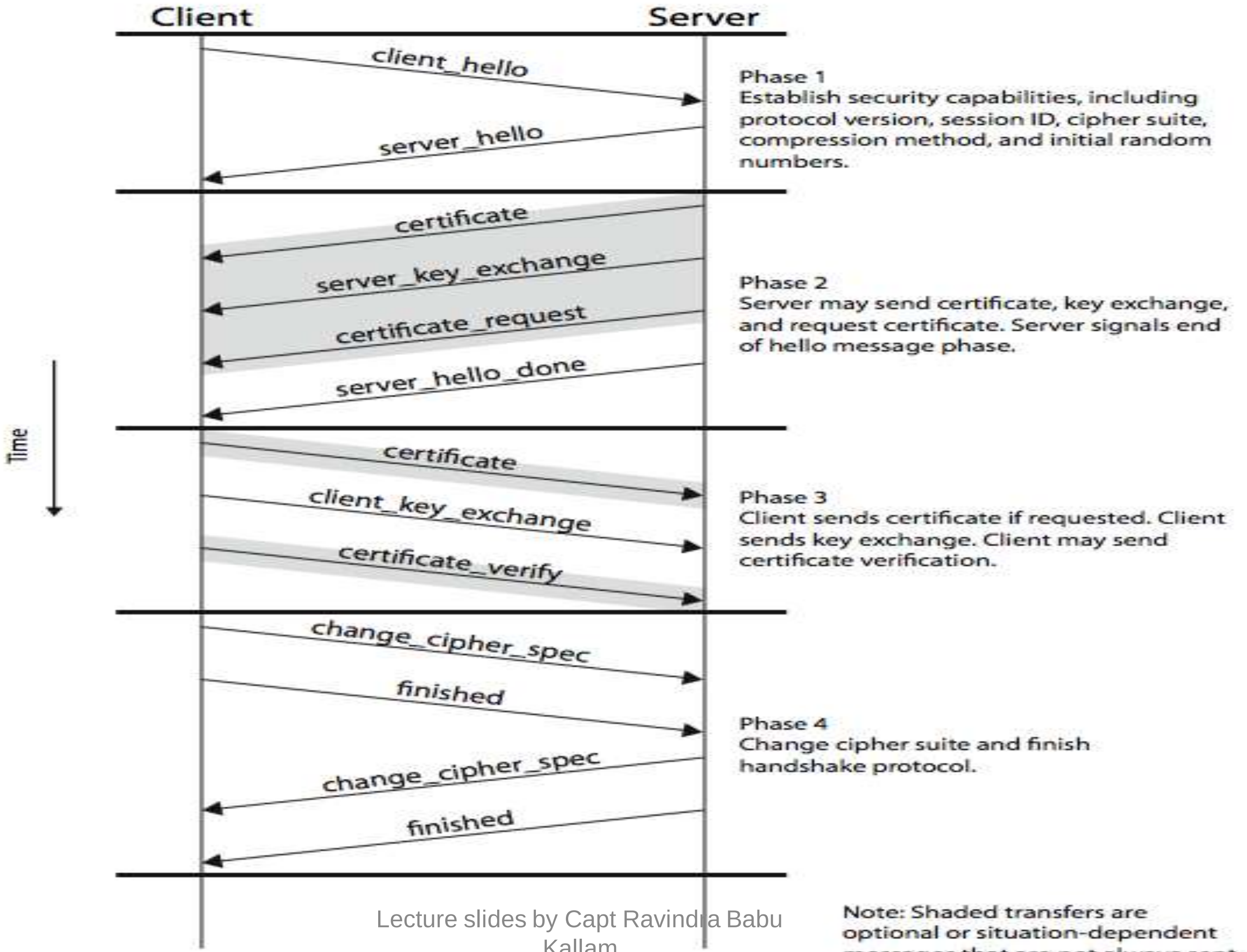
- conveys SSL-related alerts to peer entity
- It is of 2 bytes
- 1st byte shows the severity: warning or fatal
- 2nd byte shows: specific alert
  - fatal: SSL immediately terminates the connection. Other connections on the same session may continue, but no new connections on this session may be established.
    - unexpected message, bad record MAC, decompression failure, handshake failure, illegal parameter
  - warning: close notify, no certificate, bad certificate, unsupported certificate, certificate revoked, certificate expired, certificate unknown

SSL Handshake Protocol

- allows server & client to:
  - authenticate each other
  - to negotiate encryption & MAC algorithms
  - to negotiate cryptographic keys to be used
- Exchanges have 4 phases:

1. Establish Security Capabilities
2. Server Authentication and Key Exchange
3. Client Authentication and Key Exchange
4. Finish

SSL Handshake Protocol



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.

HTTP

- 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
- A user of a Web browser will see URL addresses that begin with https:// rather than http:// for secure communication.
- If HTTPS is specified, port 443 is used, which invokes SSL
- There is no fundamental change in using HTTP over either SSL or TLS and both implementations are referred to as HTTPS

- **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**
  - **Cookies sent from browser to server and from server to browser**
  - **Contents of HTTP header**

TLS (Transport Layer Security)

- TLS is an IETF standard RFC 2246 similar to SSLv3
- TLS is very similar to SSL,with minor differences
  - in record format version number Major V-3,Minor V-1
  - uses HMAC for MAC
  - a pseudo-random function (PRF) expands secret values
  - has additional alert codes:
    - Decryption failed, record_ overflow, internal error, etc
  - change in certificate types, crypto computations & padding

TLS – Pseudorandom Function

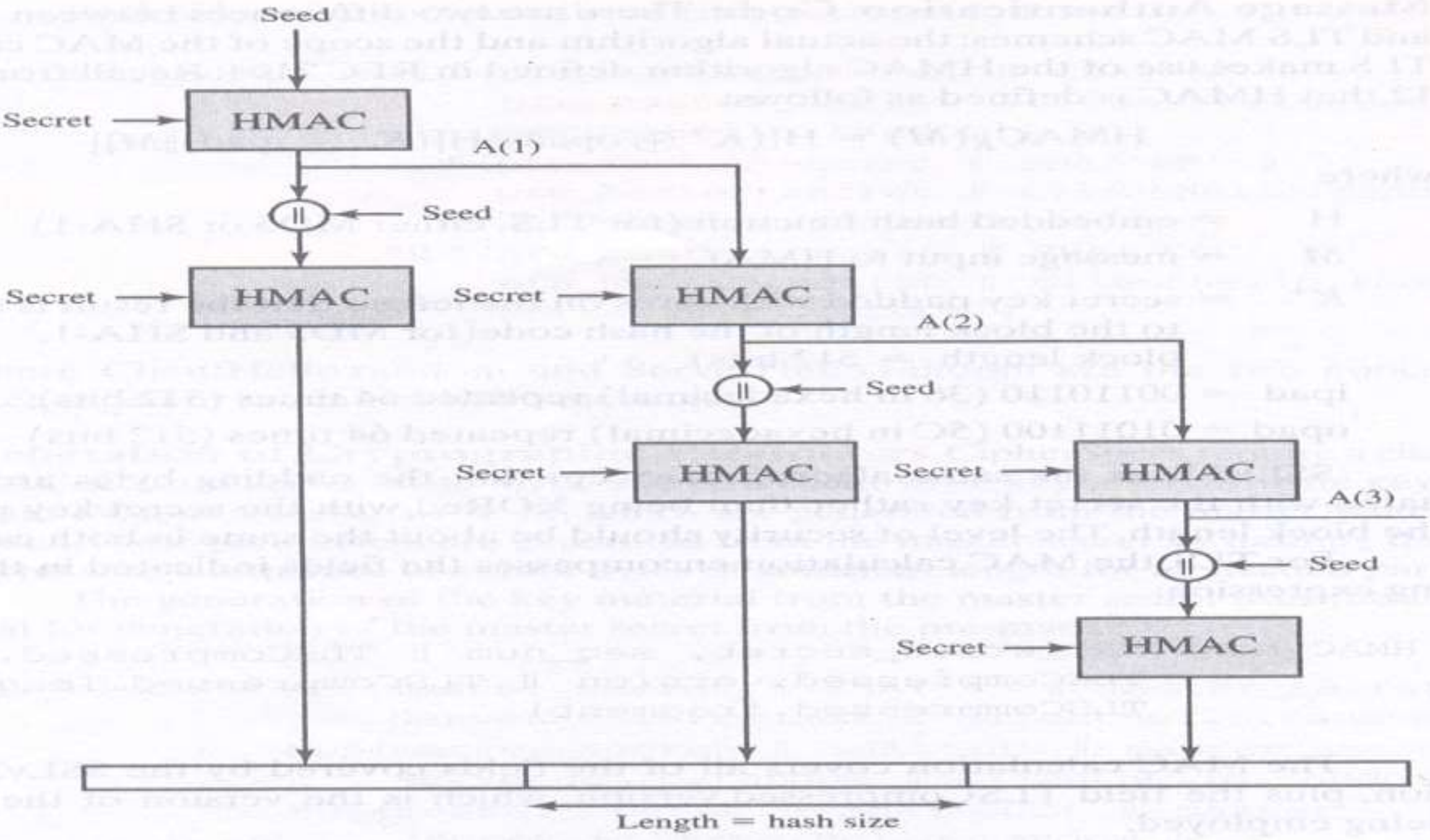
- The PRF is based on the following data expansion function:

  P_hash (secret,seed)= HMAC_hash(secret,A(1)|| seed)||
  HMAC_hash(secret,A(2)|| seed)||
  HMAC_hash(secret,A(3)|| seed)||……

  where A() is defined as : A(0) = seed
  A(i) = HMAC_hash (secret, A(i-1))

- The data expansion function makes use of the HMAC algorithm, with either MD5 or SHA-1 as the underlying hash function.
- To make PRF as secure as possible, it uses two hash algorithms in a way that should guarantee its security if either algorithm remains secure.

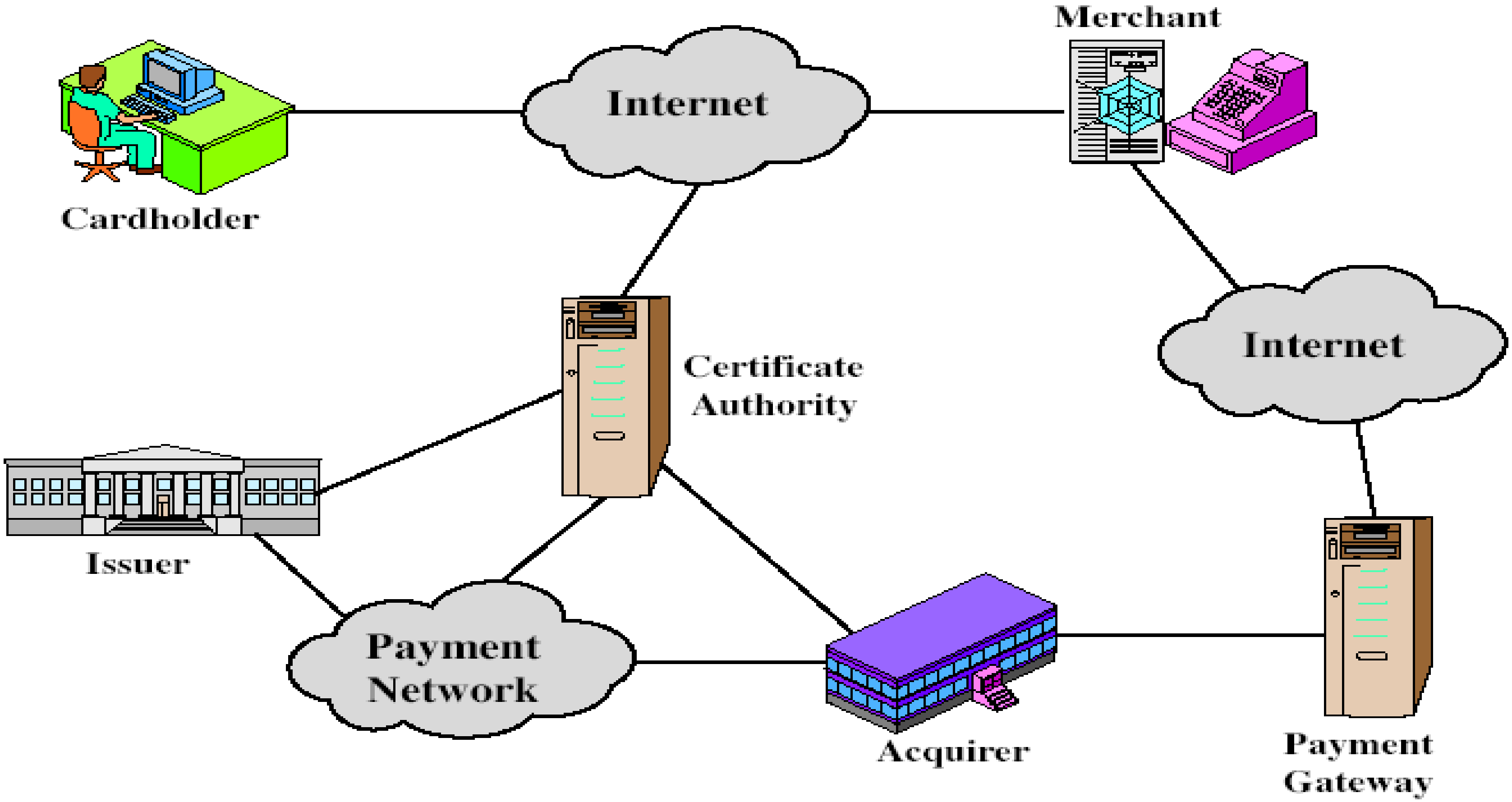
TLS Function P_hash (secret,seed)



Secure Electronic Transactions (SET)

- open encryption & security specification
- designed to protect Internet credit card transactions
- developed in 1996 by Master card, Visa etc
- not a payment system
- rather a set of security protocols & formats
  - secure communications amongst parties
  - trust from use of X.509v3 certificates
  - privacy by restricted info to those who need it

SET Components



- Cardholder: purchasers and interact with merchants from personal computers over the Internet
- Merchant: a person or organization that has goods or services to sell to the cardholder
- Issuer: a financial institution, such as a bank, that provides the cardholder with the payment card.
- Acquirer: a financial institution that establishes an account with a merchant and processes payment card authorizations and payments
- Payment gateway: a function operated by the acquirer or a designated third party that processes merchant payment messages
- Certification authority (CA): an entity that is trusted to issue X.509v3 public-key certificates for cardholders, merchants, and payment gateways

SET Transaction

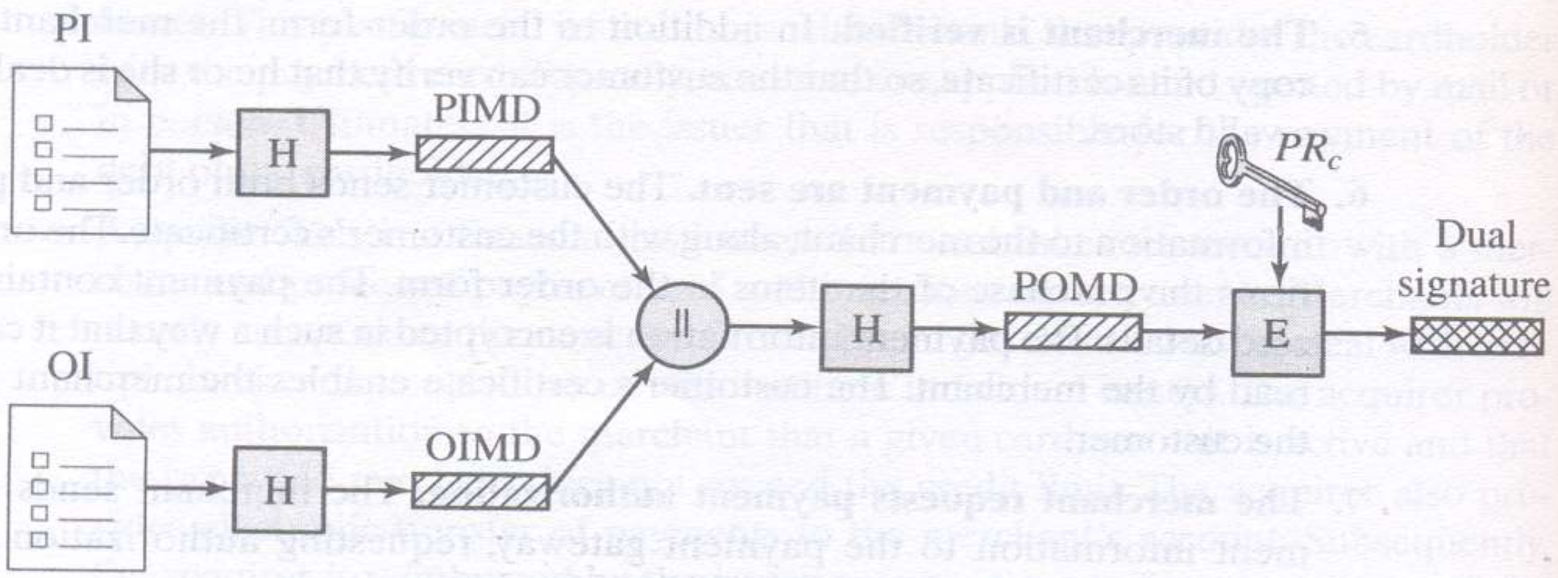
1. **Customer opens account**
2. **Customer receives a certificate**
3. **Merchants have their own certificates**
4. **Customer places an order and receives a form containing the list of items, their price, a total price, and an order number.**
5. **Merchant is verified by the customer through merchant certificate**
6. **Order and payment information are sent by the customer**
7. **Merchant requests payment authorization to the payment gateway**
8. **Merchant confirms order to the customer and provides goods or service to the customer**
10. **Merchant requests payment from payment gateway**

Dual Signature

- customer creates dual messages
  - order information (OI) for merchant
  - payment information (PI) for bank
- neither party needs details of other
- but **must** know they are linked
- use a dual signature for this
  - signed concatenated hashes of OI & PI

$$DS=E(PR_c, [H(H(PI)||H(OI))])$$

Construction of dual signature



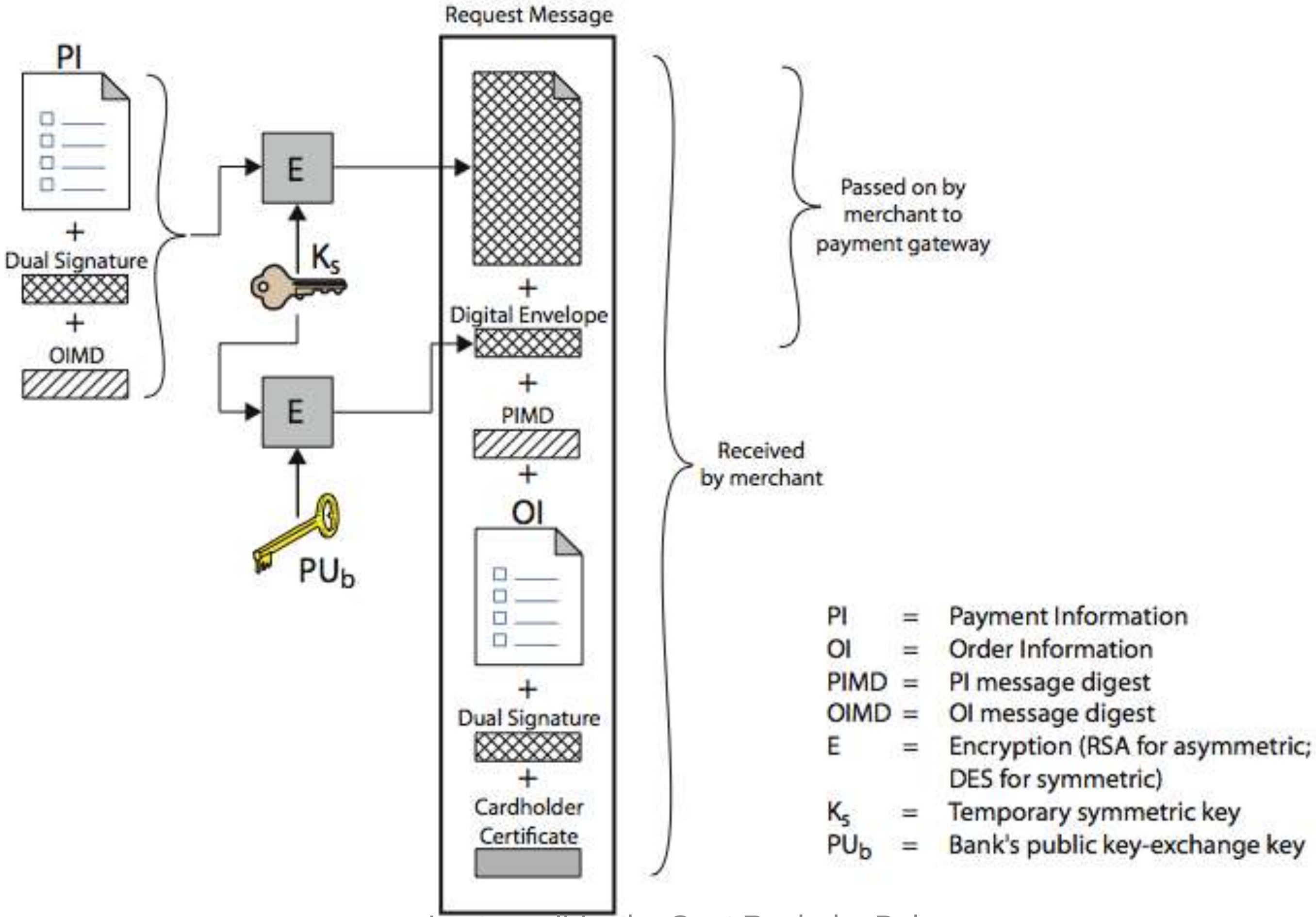
PI	= Payment information	PIMD	= PI message digest
OI	= Order information	OIMD	= OI message digest
H	= Hash function (SHA-1)	POMD	= Payment order message digest
||	= Concatenation	E	= Encryption (RSA)
		PR_c	= Customer's private signature key

SET Purchase Request

- SET purchase request exchange consists of four messages

1. Initiate Request - get certificates
2. Initiate Response - signed response
3. Purchase Request - of OI & PI
4. Purchase Response - ack order

Purchase Request – Customer

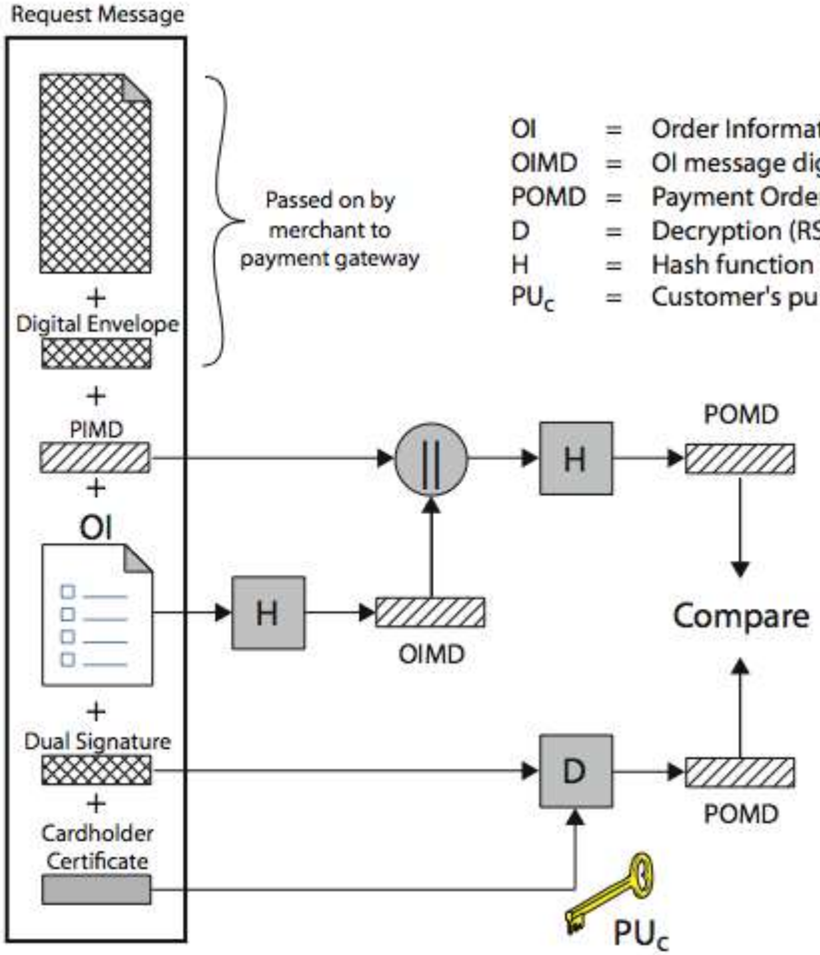


PI	=	Payment Information
OI	=	Order Information
PIMD	=	PI message digest
OIMD	=	OI message digest
E	=	Encryption (RSA for asymmetric; DES for symmetric)
K_s	=	Temporary symmetric key
PU_b	=	Bank's public key-exchange key

Purchase Request – Merchant

1. verifies cardholder certificates using CA sigs
2. verifies dual signature using customer's public signature key to ensure order has not been tampered with in transit & that it was signed using cardholder's private signature key
3. processes order and forwards the payment information to the payment gateway for authorization (described later)
4. sends a purchase response to cardholder

Purchase Request – Merchant



OI	=	Order Information
OIMD	=	OI message digest
POMD	=	Payment Order message digest
D	=	Decryption (RSA)
H	=	Hash function (SHA-1)
PU_c	=	Customer's public signature key

Payment Gateway Authorization

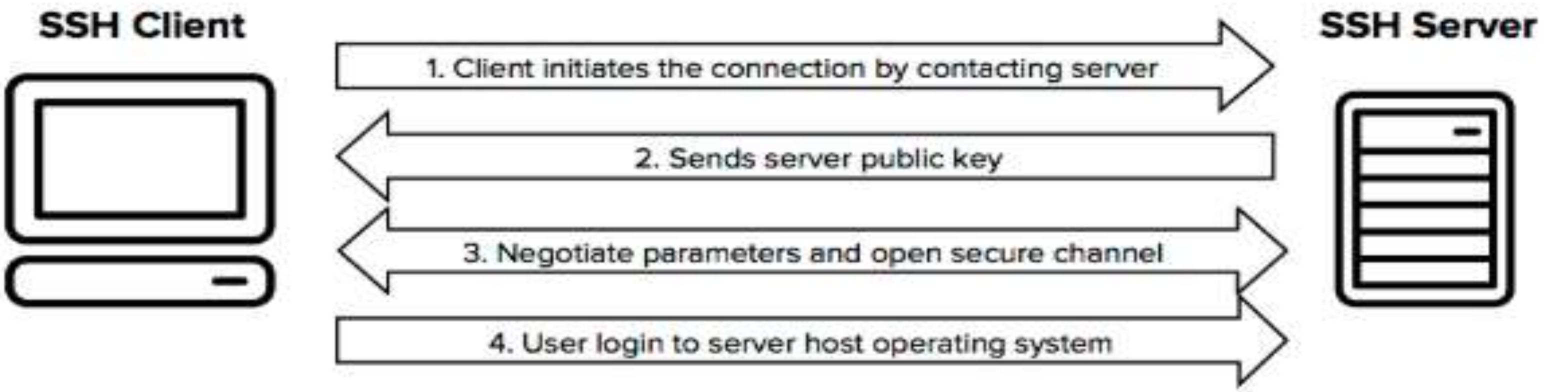
1. verifies all certificates
2. decrypts digital envelope of authorization block to obtain symmetric key & then decrypts authorization block
3. verifies merchant's signature on authorization block
4. decrypts digital envelope of payment block to obtain symmetric key & then decrypts payment block
5. verifies dual signature on payment block
6. verifies that transaction ID received from merchant matches that in PI received (indirectly) from customer
7. requests & receives an authorization from issuer
8. sends authorization response back to merchant

Payment Capture

- merchant sends payment gateway a payment capture request
- gateway checks request
- Transfers funds to the merchants account
- notifies merchant using capture response

Secure Shell (SSH)

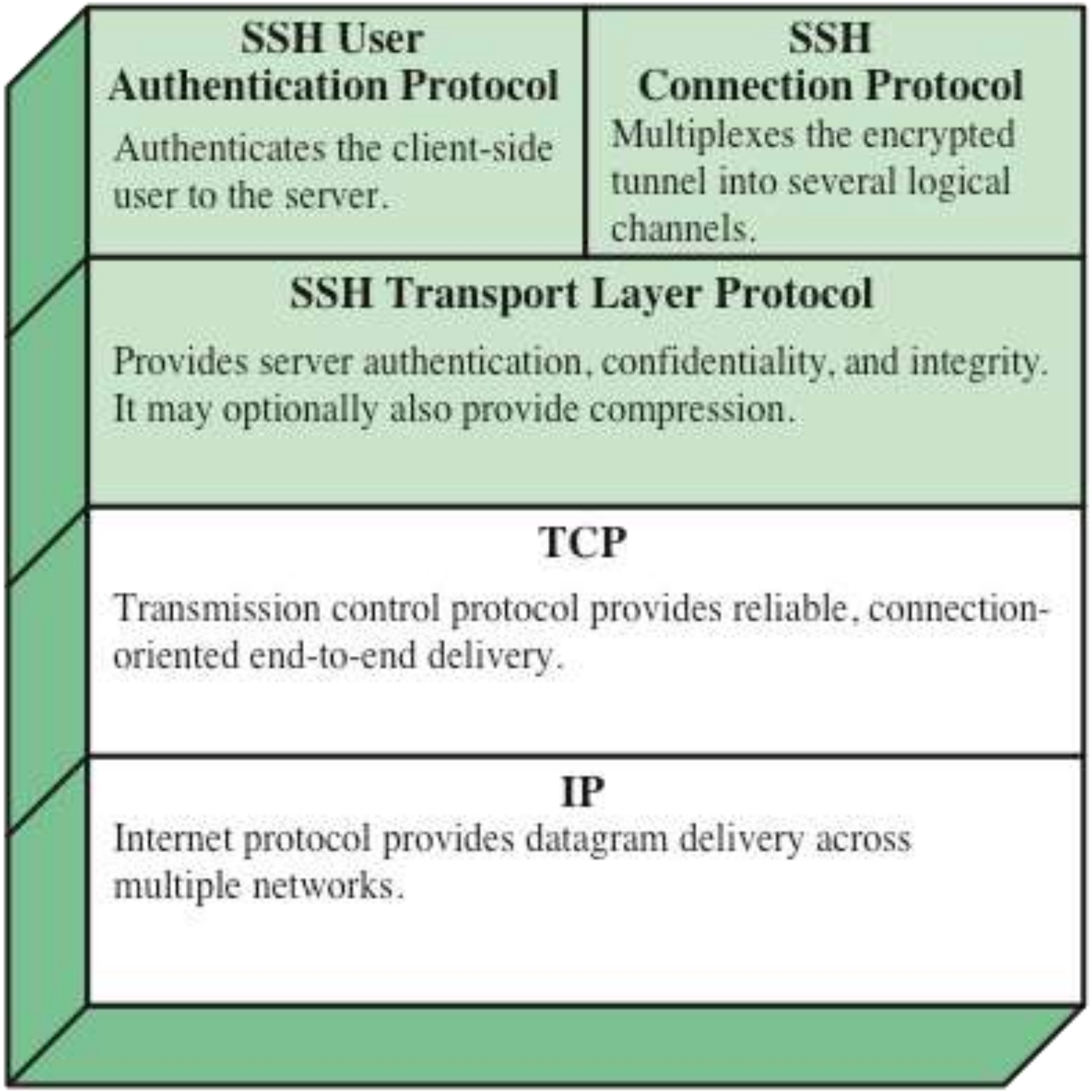
- Secure Shell (SSH) is a protocol for secure network communications designed to be relatively simple and inexpensive to implement.
- The SSH protocol uses encryption to secure the connection between a client and a server. All user authentication, commands, output, and file transfers are encrypted to protect against attacks in the network.


SSH Client
SSH Server
1. Client initiates the connection by contacting server
2. Sends server public key
3. Negotiate parameters and open secure channel
4. User login to server host operating system

- The initial version, SSH1 was focused on providing a secure remote logon facility to replace TELNET and other remote logon schemes that provided no security.
- SSH also provides a more general client/server capability and can be used for network functions as file transfer and e-mail.
- SSH client and server applications are widely available for most operating systems.
- It has become the method of choice for remote login and is rapidly becoming one of the most pervasive applications for encryption technology outside of embedded systems.

SSH is organized as three protocols that typically run on top of TCP

- Transport Layer Protocol: Provides server authentication, data confidentiality, and data integrity. It may optionally provide compression.
- User Authentication Protocol: Authenticates the user to the server.
- Connection Protocol: Multiplexes multiple logical communications channels over a single, underlying SSH connection.



SSH User Authentication Protocol

Authenticates the client-side user to the server.

SSH Connection Protocol

Multiplexes the encrypted tunnel into several logical channels.

SSH Transport Layer Protocol

Provides server authentication, confidentiality, and integrity. It may optionally also provide compression.

TCP

Transmission control protocol provides reliable, connection-oriented end-to-end delivery.

IP

Internet protocol provides datagram delivery across multiple networks.

Figure 17.8 SSH Protocol Stack

SSH Transport Layer Cryptographic Algorithms

Encryption:

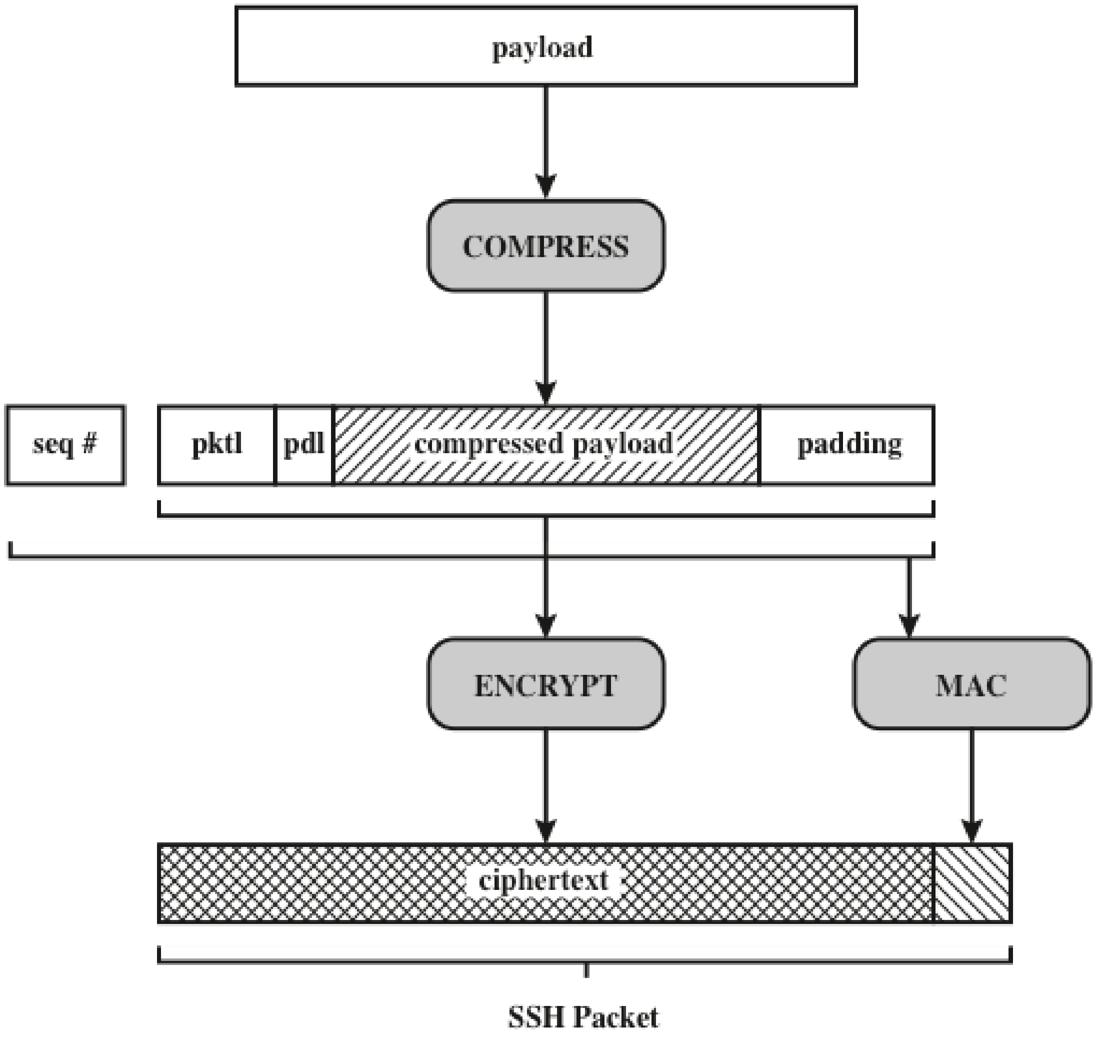
3DES in CBC mode
Blowfish in CBC mode
AES with 128/192 bit Key
RC4 with 128 bit key
CAST-128 in CBC mode

MAC Algorithms:

HMAC –SHA1
HMAC-MD5

Compression Algorithm:

Zlib Algorithm



pktl = packet length
pdl = padding length

Figure 17.10 SSH Transport Layer Protocol Packet Formation

- Packet length: Length of the packet in bytes, not including the packet length and MAC fields.

- Padding length: Length of the random padding field.

- Payload: Useful contents of the packet. Prior to algorithm negotiation, this field is uncompressed. If compression is negotiated, then in subsequent packets, this field is compressed.

- Random padding: Once an encryption algorithm has been negotiated, this field is added. It contains random bytes of padding so that that total length of the packet (excluding the MAC field) is a multiple of the cipher block size or 8 bytes for a stream cipher.

- Message authentication code (MAC): If message authentication has been negotiated, this field contains the MAC value. The MAC value is computed over the entire packet plus a sequence number, excluding the MAC field. Once an encryption algorithm has been negotiated, the entire packet (excluding the MAC field) is encrypted after the MAC value is calculated.

End of Part 1 of 4th Unit