Cryptography and Network Security

Dr. S.R. Shinde

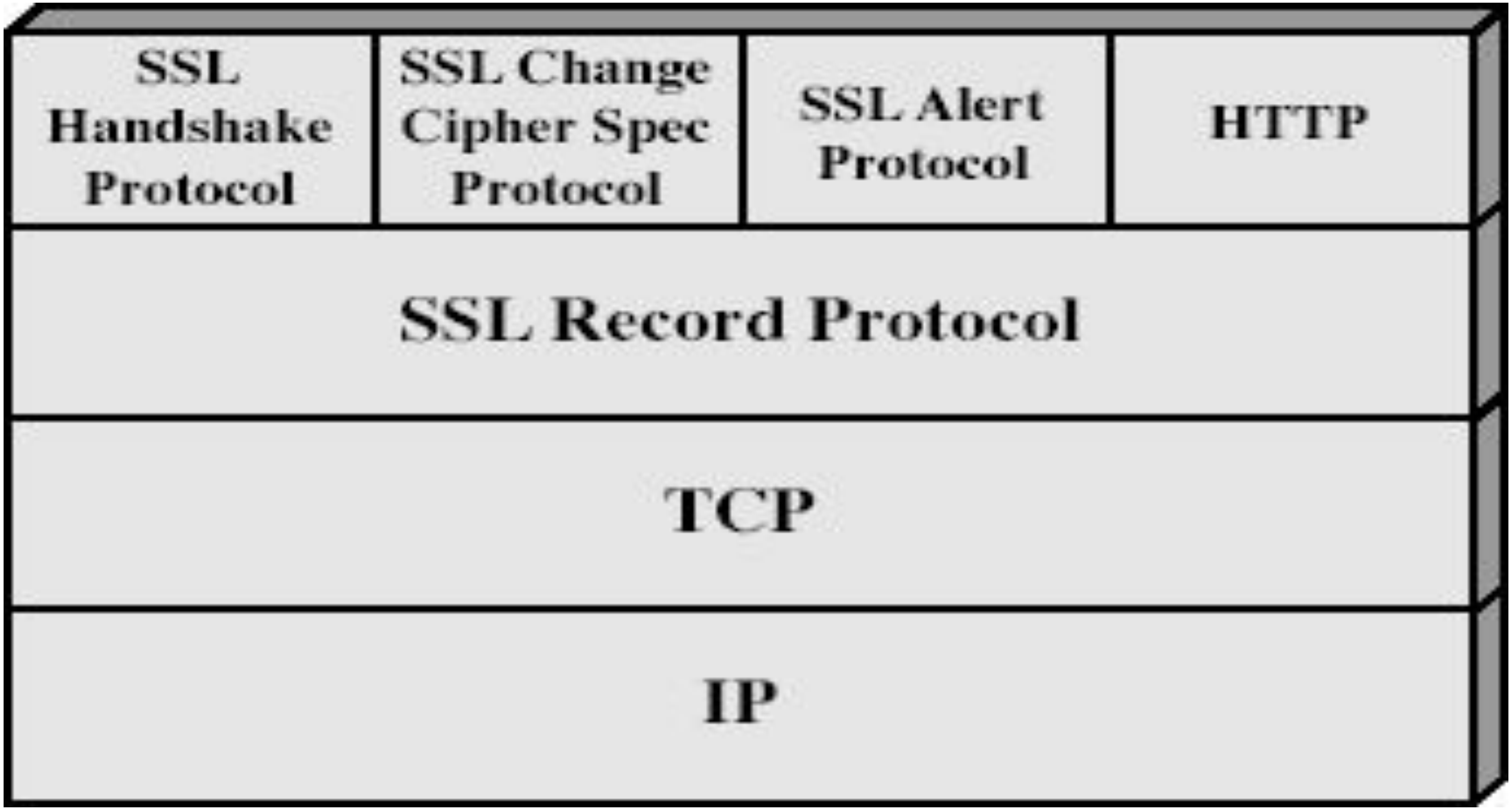
Web Security

- 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

SSL (Secure Socket Layer)

- transport layer security service
- originally developed by Netscape
- version 3 designed with public input
- subsequently became Internet standard known as TLS (Transport Layer Security)
- uses TCP to provide a reliable end-to-end service
- SSL has two layers of protocols

SSL Architecture



SSL Handshake Protocol

SSL Change Cipher Spec Protocol

SSL Alert Protocol

HTTP

SSL Record Protocol

TCP

IP

SSL Architecture

- **SSL session**
  - an association between client & server
  - created by the Handshake Protocol
  - define a set of cryptographic parameters
  - may be shared by multiple SSL connections
- **SSL connection**
  - a transient, peer-to-peer, communications link
  - associated with 1 SSL session

parameters defining session state

- Session identifier

  arbitrary byte sequence chosen by the server to identify an active or resumable session state

- Peer certificate

  X509.v3 certificate of the peer. This element of the state may be null

- Compression method

  The algorithm used to compress data prior to encryption.

parameters defining session state

- Cipher spec

  Specifies the bulk data encryption algorithm (such as null, DES, etc.) and hash algorithm (such as MD5 or SHA-l) used for MAC calculation. It also defines cryptographic attributes such as the hash_size.

- Master secret

  48-byte secret shared between the client and server.

- Is resumable

  flag indicating whether the session can be used to initiate new connections.

parameters defining connection state

- Server and client random

  Byte sequences that are chosen by the server and client for each connection.

- Server write MAC secret

  The secret key used in MAC operations on data sent by the server

- Client write MAC secret

  The secret key used in MAC operations on data sent by the client.

- Server write key

  The conventional encryption key for data encrypted by the server and decrypted by the client

parameters defining connection state

- Client write key

  The conventional encryption key for data encrypted by the client and decrypted by the server.

- Initialization vectors

  When a block cipher in CBC mode is used, an initialization vector (IV) is maintained for each key. This field is first initialized by the SSL Handshake Protocol. Thereafter the final ciphertext block from each record is preserved for use as the IV with the following record

parameters defining connection state

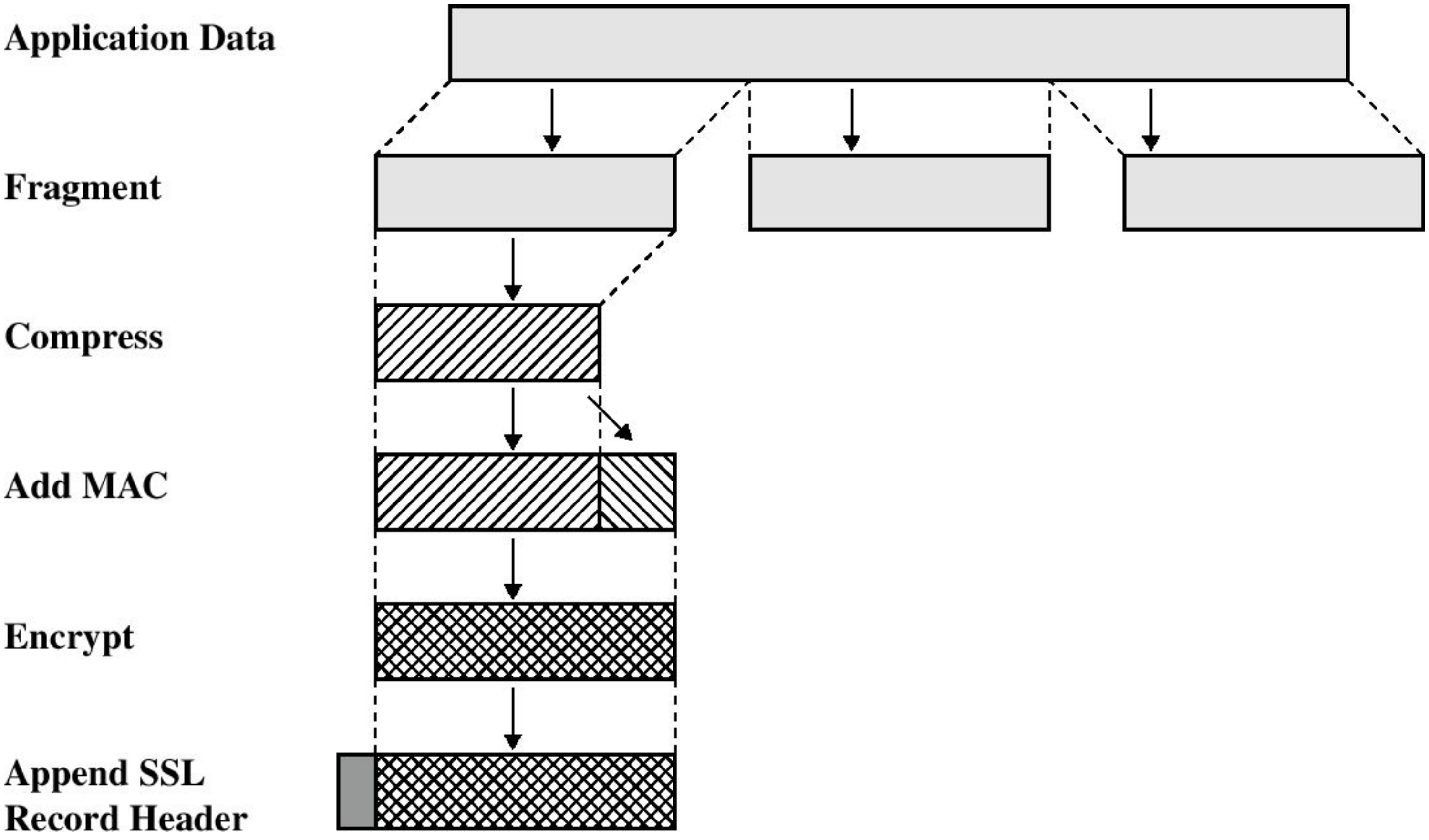
- Sequence numbers

  Each party maintains separate sequence numbers for transmitted and received messages for each connection. When a party sends or receives a change cipher spec message, the appropriate sequence number is set to zero. Sequence numbers may not exceed 2^{64}- 1.

SSL Record Protocol

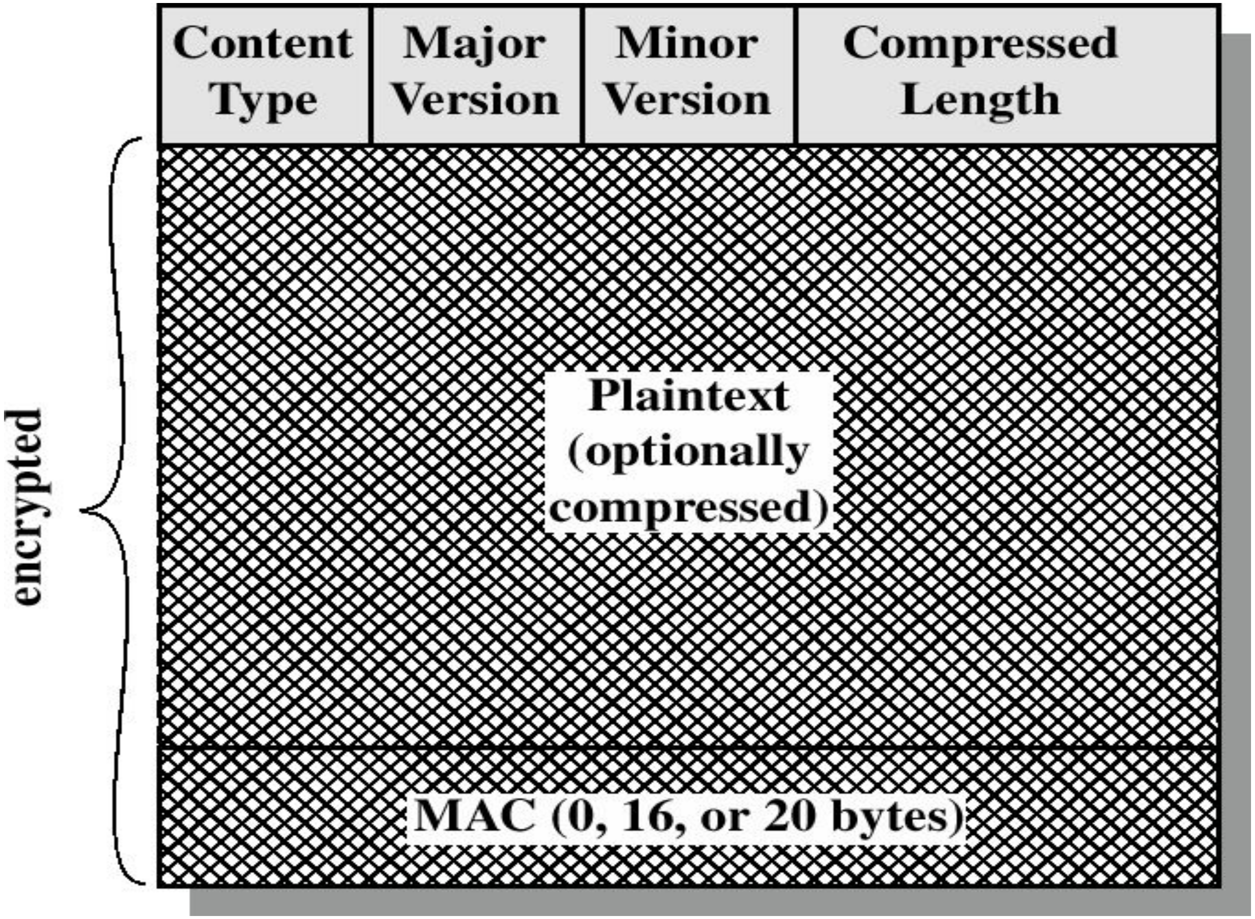
- **confidentiality**
  - using symmetric encryption with a shared secret key defined by Handshake Protocol
  - IDEA, RC2-40, DES-40, DES, 3DES, Fortezza, RC4-40, RC4-128
  - message is compressed before encryption
- **message integrity**
  - using a MAC with shared secret key
  - similar to HMAC but with different padding

SSL - Record Protocol



SSL - record

fields of the header



Content Type

Major Version

Minor Version

Compressed Length

Plaintext (optionally compressed)

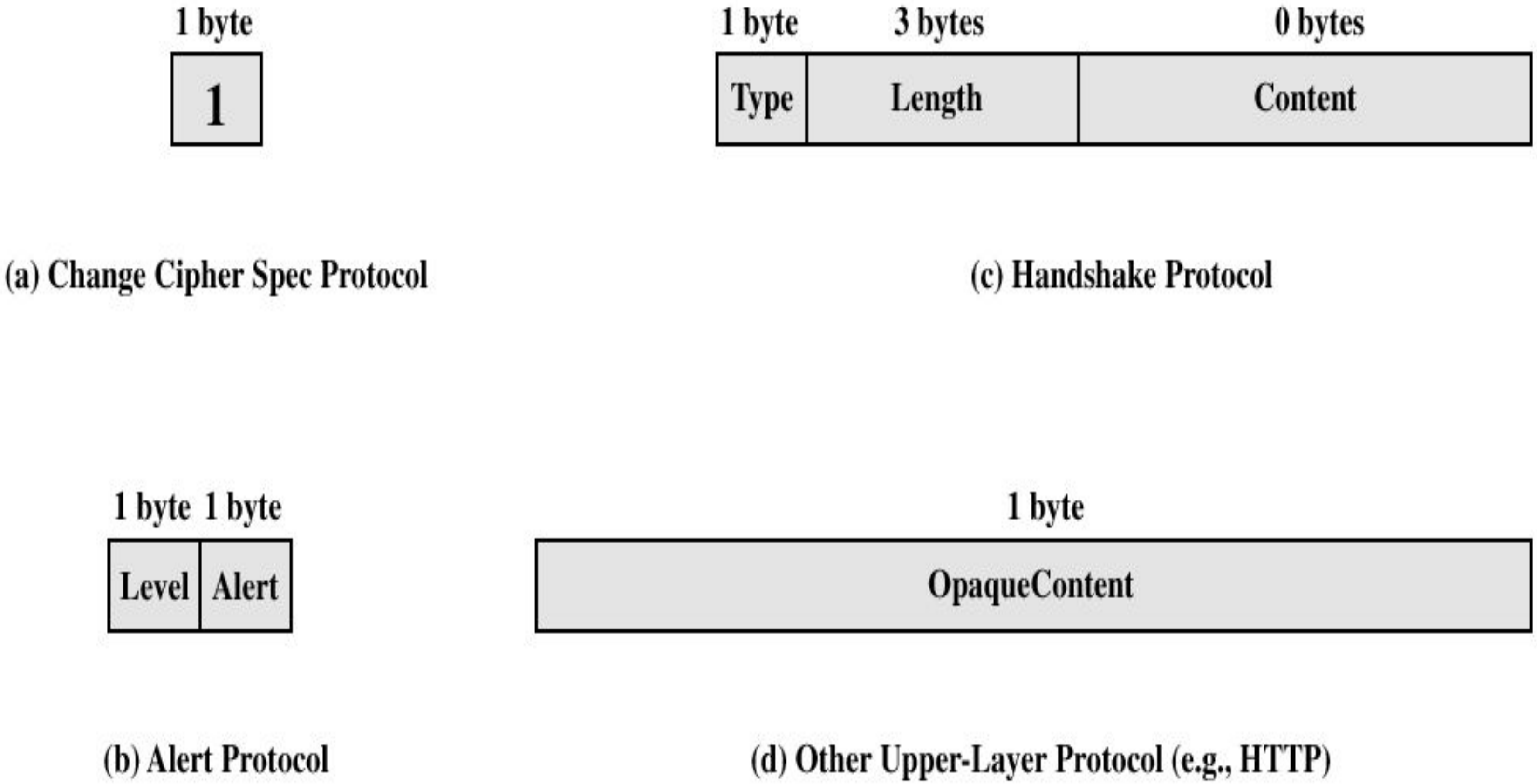
MAC (0, 16, or 20 bytes)

encrypted

fields

- Content Type (8 bits)
  - The higher layer protocol used to process the enclosed fragment (change_cipher_spec, alert, handshake, and application_data. The first three are the SSL-specific protocols; no distinction is made among the various applications (e.g., HTTP) that might use SSL)
- Major Version (8 bits)
  - Indicates major version of SSL in use. For SSLv3, the value is 3
- Minor Version (8 bits)
  - Indicates minor version in use. For SSLv3, the value is 0
- Compressed Length (16 its)
- The length in bytes of thbe plaintext fragment (or compressed fragment if compression is used).

SSL -



(a) Change Cipher Spec Protocol

(c) Handshake Protocol

(b) Alert Protocol

(d) Other Upper-Layer Protocol (e.g., HTTP)

SSL Change Cipher Spec Protocol

- one of 3 SSL specific protocols which use the SSL Record protocol
- a single message
- causes pending state to become current
- hence updating the cipher suite in use

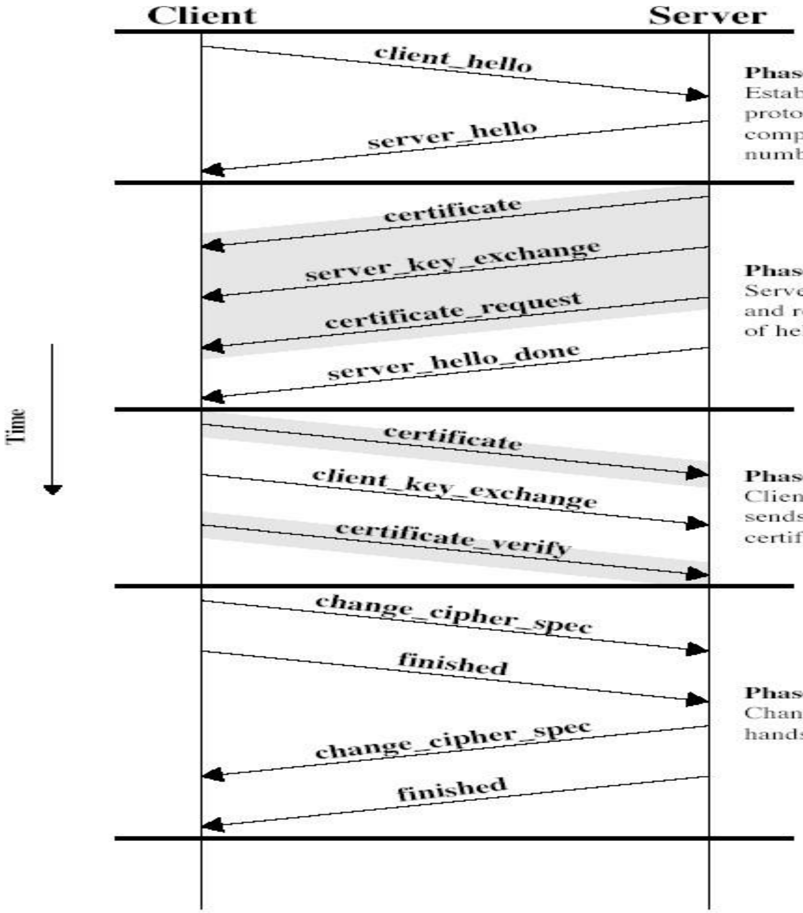
SSL Alert Protocol

- conveys SSL-related alerts to peer entity
- severity
    - warning or fatal
- specific alert
    - unexpected message, bad record mac, decompression failure, handshake failure, illegal parameter
    - close notify, no certificate, bad certificate, unsupported certificate, certificate revoked, certificate expired, certificate unknown
- compressed & encrypted like all SSL data

SSL Handshake Protocol

- allows server & client to:
  - authenticate each other
  - to negotiate encryption & MAC algorithms
  - to negotiate cryptographic keys to be used
- comprises a series of messages in phases
  - Establish Security Capabilities
  - Server Authentication and Key Exchange
  - Client Authentication and Key Exchange
  - 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.

Handshake protocol

4 steps

1. Hello: determine security capabilities
2. Server sends certificate, asks for certificate and starts exchange session keys
3. Client sends certificate and continues exchanges of keys
4. End of handshake protocol: encoded methods changes

Note: some requests are optional

clear separation between handshake and the rest (to avoid attacks)

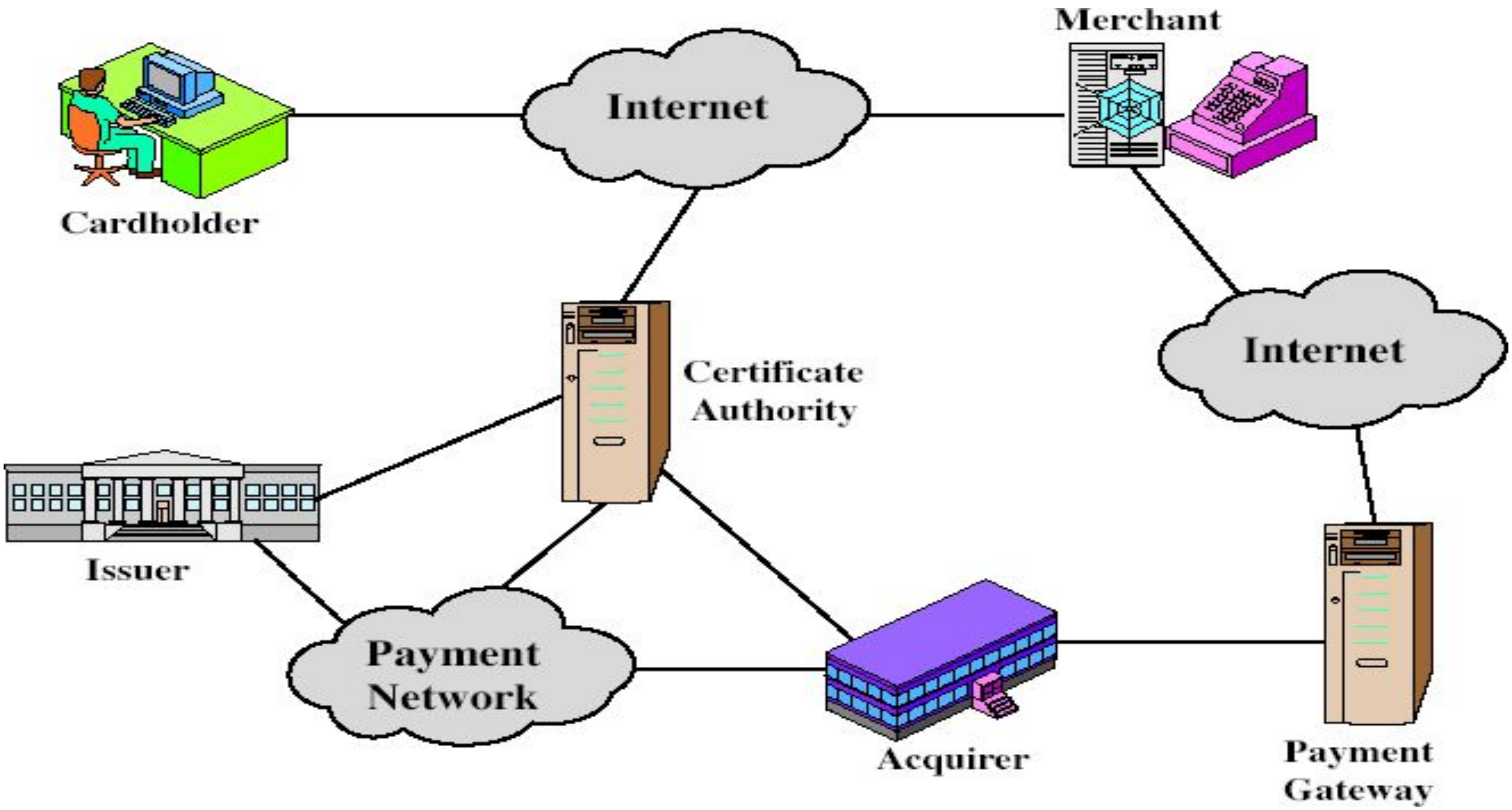
TLS (Transport Layer Security)

- IETF standard RFC 2246 similar to SSLv3
- with minor differences
  - in record format version number
  - uses HMAC for MAC
  - a pseudo-random function expands secrets
  - has additional alert codes
  - some changes in supported ciphers
  - changes in certificate negotiations
  - changes in use of padding

Secure Electronic Transactions (SET)

- open encryption & security specification
- to protect Internet credit card transactions
- developed in 1996 by Mastercard, 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



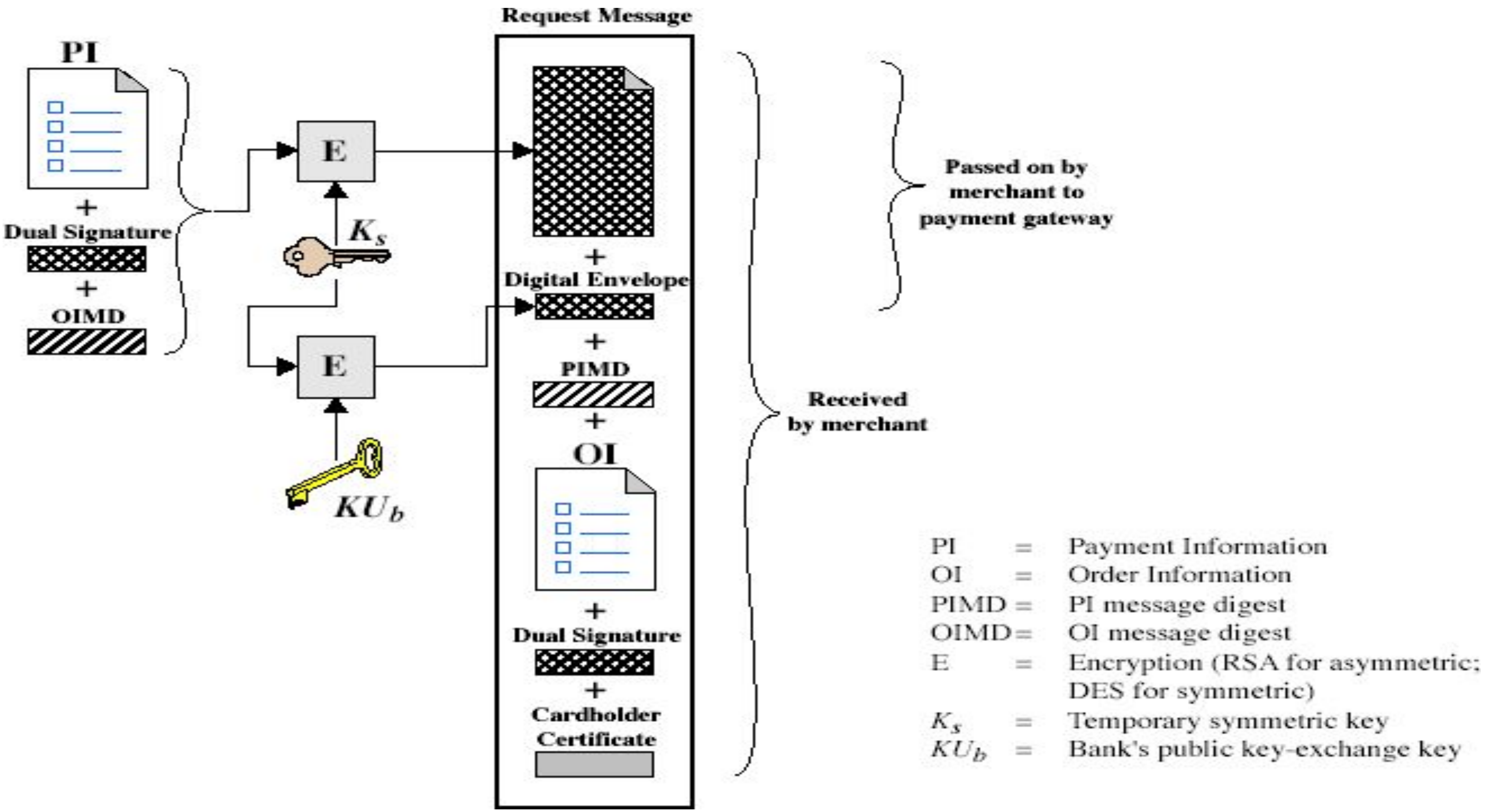
SET Transaction

1. customer opens account
2. customer receives a certificate
3. merchants have their own certificates
4. customer places an order
5. merchant is verified
6. order and payment are sent
7. merchant requests payment authorization
8. merchant confirms order
9. merchant provides goods or service
10. merchant requests payment

Dual Signature

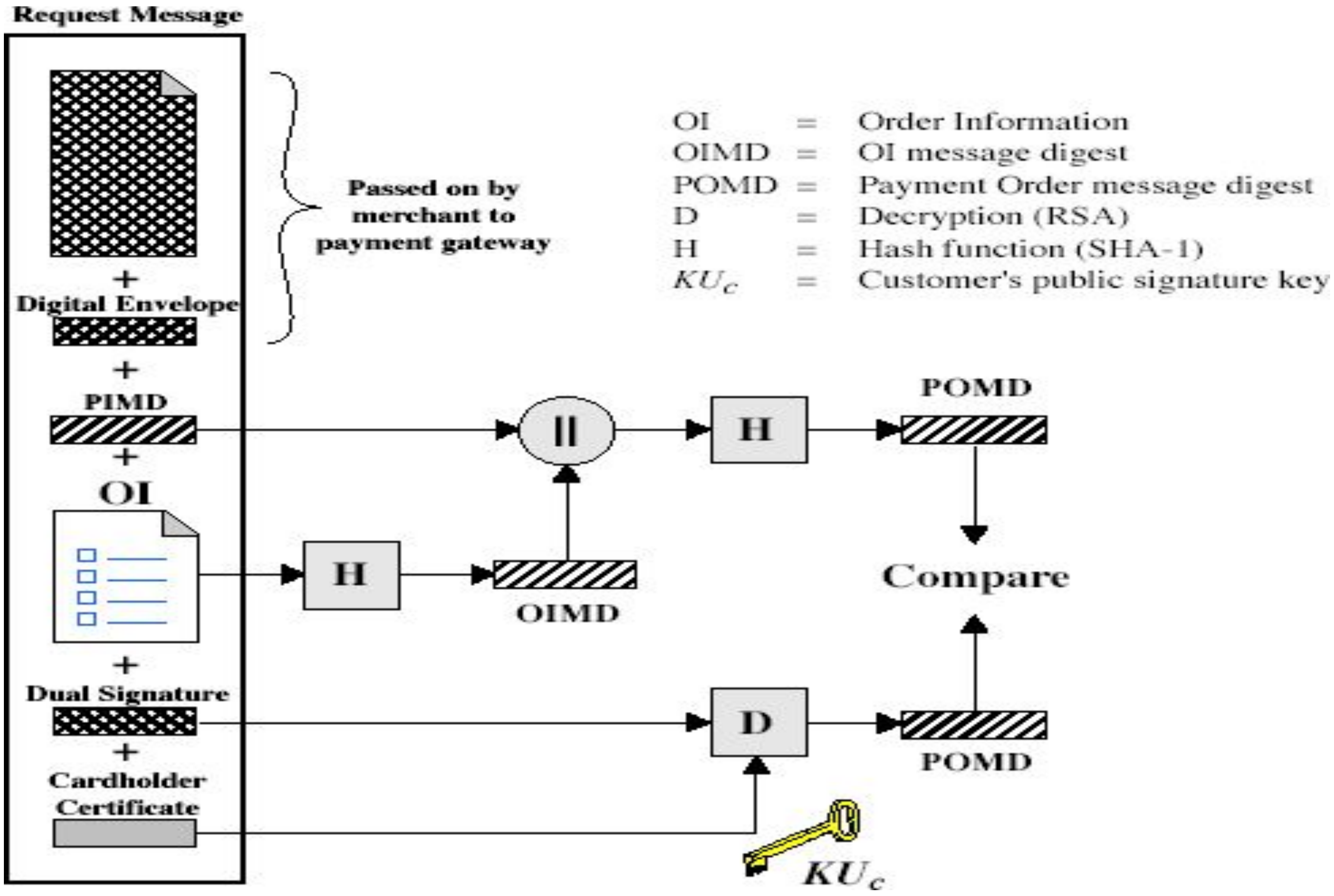
- customer creates dual messages
  - order information (OI) for merchant
  - payment information (PI) for bank
- neither party needs details of other
- but **must** know they are linked
- use a dual signature for this
  - signed concatenated hashes of OI & PI

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
KU_b	=	Bank's public key-exchange key

Purchase Request – Merchant



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

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
- then causes funds to be transferred to merchants account
- notifies merchant using capture response

Summary

- have considered:
  - need for web security
  - SSL/TLS transport layer security protocols
  - SET secure credit card payment protocols