Chapter 7

WEB Security

Outline

- Web Security Considerations
- Secure Socket Layer (SSL) and Transport Layer Security (TLS)
- Secure Electronic Transaction (SET)

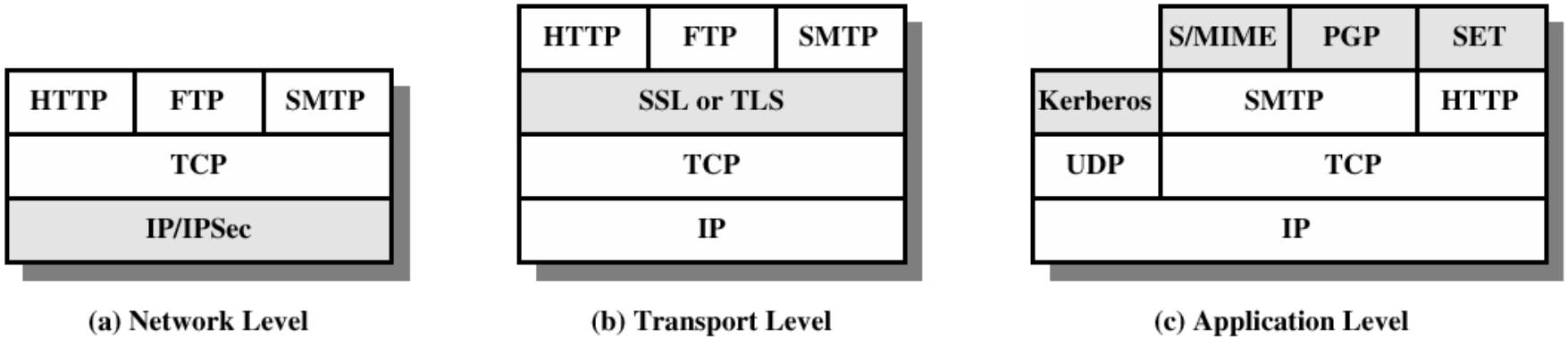
Web Security Considerations

- The WEB is very visible outlet.
- Complex software hides potential many security flaws.
- Web servers are easy to configure and manage.
- Users are not aware of the risks.

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

Security facilities in the TCP/IP protocol stack



*(b) Microsoft Explorer Browsers come equipped with SSLlll or TLS

SSL and TLS

- SSL was originated by Netscape
- TLS working group was formed within IETF
  - http://www.ietf.org/html.charters/tls-charter.html
- First version of TLS can be viewed as an SSLv3.1

SSL Architecture

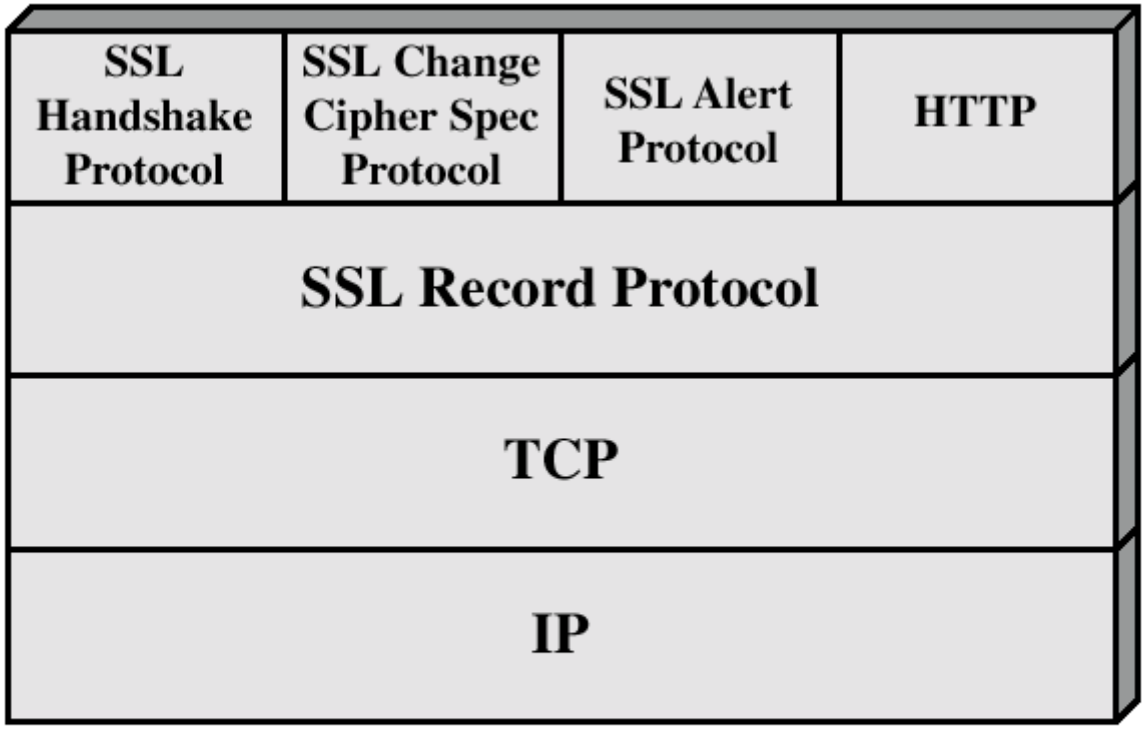


Figure 7.2 SSL Protocol Stack

SSL Connection

- SSL connection : a transport that provides a suitable type of service
- Parameters for connection state
  - Server and client random : byte sequence
  - Server write MAC secret: secret key used in MAC by the server
  - Client write MAC secret: secret key used in MAC by the client
  - Server write key: conventional encryption key for data encrypted by the server
  - Client write key: conventional encryption key for data encrypted by the client
  - 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
  - Sequence numbers

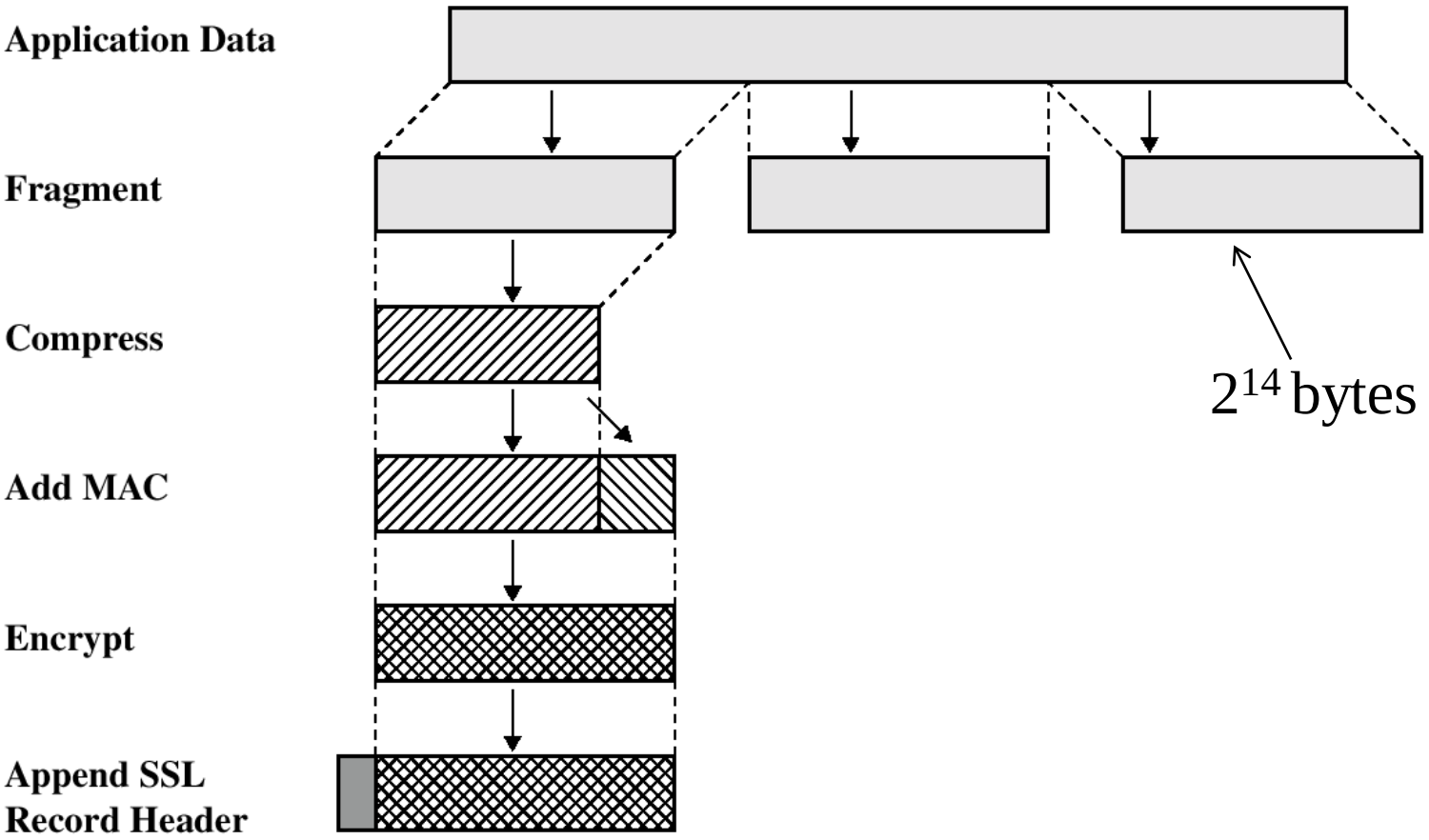
SSL Session

- SSL session : an association between a client and a server. Session state is defined by following parameters
  - Session id
  - Peer certificate: x.509.v3 certificate of the peer
  - Compression method
  - Cipher spec: encryption algorithm (null, AES, etc) and hash algorithm, (such as MD5 or SHA-1) used for MAC calculation. And defining cryptographic attributes such as the hash_size

SSL Record Protocol

- SSL Record Protocol provides two services for SSLL connections
  - Confidentiality: handshake protocol also defines a shared secret key that is used for conventional encryption of SSL payload
  - Message Integrity: handshake protocol also defines a shared secret key that is used to form a MAC

SSL Record Protocol Operation



MAC for SSL Record Protocol

hash(MAC_write_secret || pad_2||
hash(MAC_write_secret || pad_1 || seq_num ||
SSLCompressed.type ||
SSLCompressed.length ||SSLCompressed.fragment))

Where

MAC_write_secret = shared secret key

Hash = MD5 or SHA-1

Pad_1: the byte 0x36(0011 0110) repeated 48 times (384 bits) for MD5 and 40 times (320bits) for SHA-1

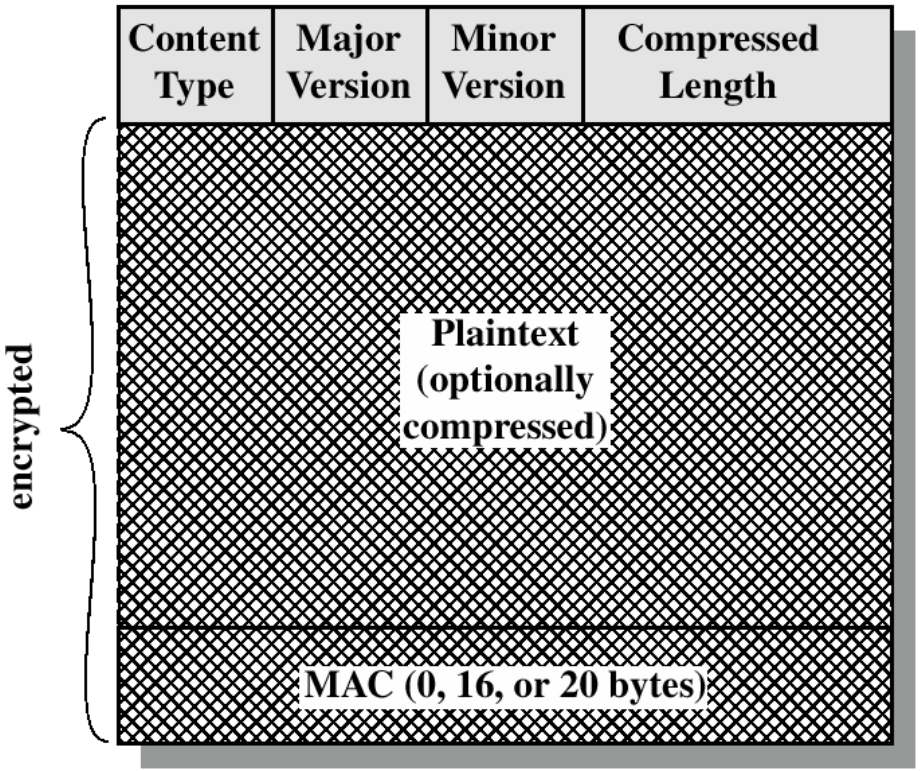
Pad_2 = the byte 0x5C (0101 1100) repeated 48 times for MD5 and 40 times for SHA-1

SSLCompressed.type = the higher-level protocol used to process this fragment

SSLCompressed.length = the length of the compressed fragment

SSLCompressed.fragment= the compressed fragment (if compression is not used, the plaintext fragment)

SSL Record Format



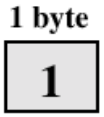
-Content type(8 bit): the higher layer protocol used to process the enclosed fragment

-Major version: for SSLv3, value:3,

-Minor version: for SSLv3, value:0

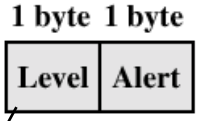
- Compressed length(16bits): the length in the bytes of the plaintext fragment; maximum value is $2^{14} + 2048$

Change Cipher Spec Protocol



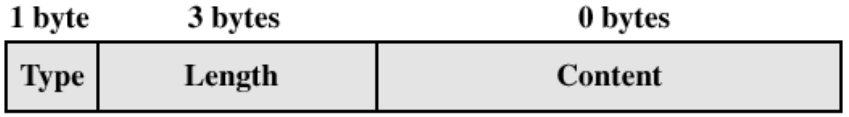
(a) Change Cipher Spec Protocol

To cause pending state to be copied into the current state

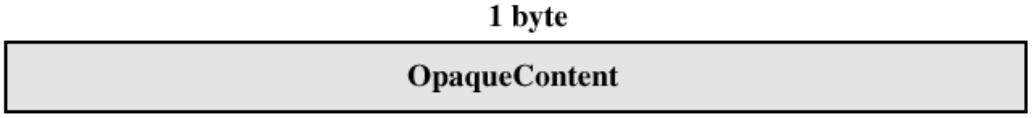


(b) Alert Protocol

warning(1) or fatal(2)



(c) Handshake Protocol



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

SSL Record Protocol Payload

Fatal 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

Handshake Protocol (1)

- The most complex part of SSL.
- Allows the server and client to authenticate each other.
- Negotiate encryption, MAC algorithm and cryptographic keys.
- Used before any application data are transmitted.

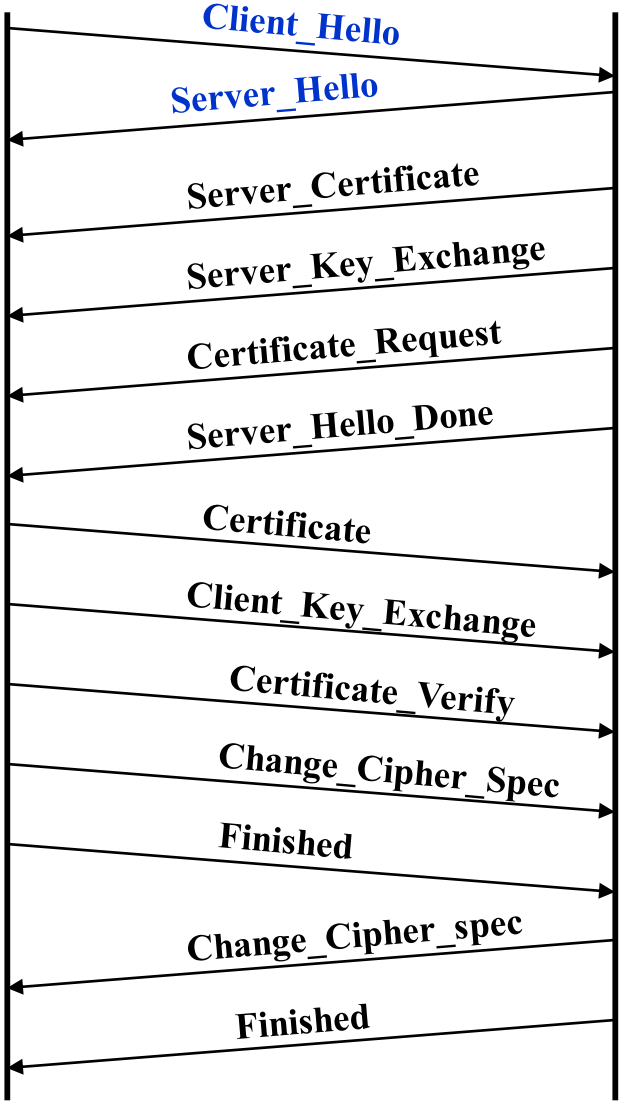
Handshake Protocol (2)

key exchange method, cipher spec

Table 7.2 SSL Handshake Protocol Message Types

Message Type	Parameters
hello_request	null
client_hello	version, random, session id, cipher suite, compression method
server_hello	version, random, session id, cipher suite, compression method
certificate	chain of X.509v3 certificates
server_key_exchange	parameters, signature
certificate_request	type, authorities
server_done	null
certificate_verify	signature
client_key_exchange	parameters, signature
finished	hash value

Handshake Protocol Action(1)



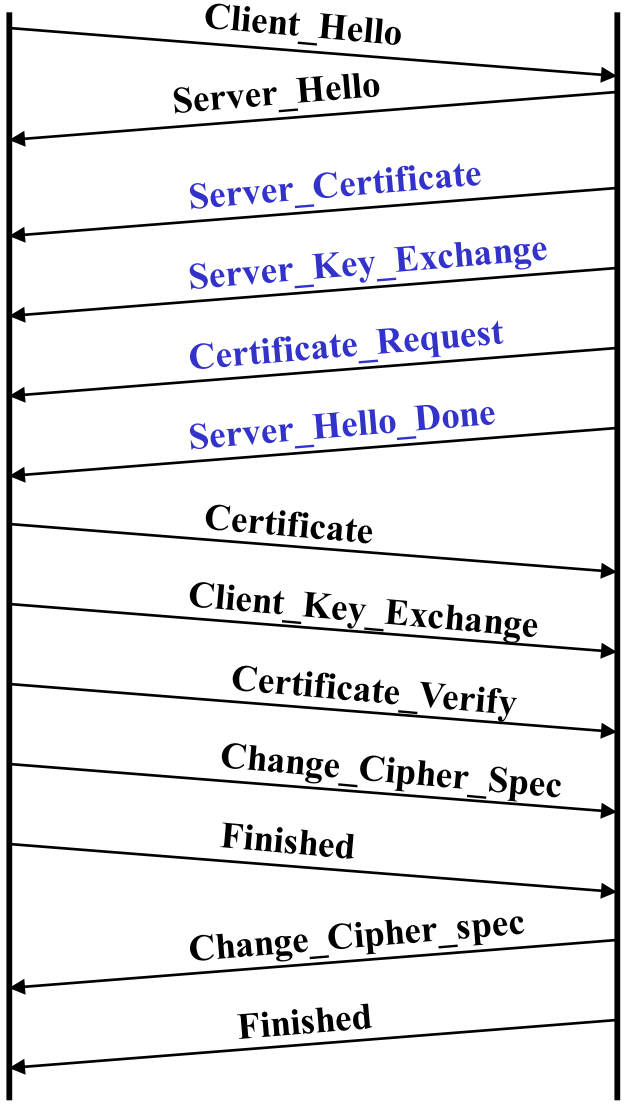
- **Phase1:** Create the Connection between the Client A and Server G and figure out what each entity can do!

$$1.A \rightarrow G : \{vers\#, r_A, SessID, CiphList, CompList\}$$

$$2.G \rightarrow A : \{vers\#, r_G, SessID, CiphChoice, CompChoice\}$$

- r_A is a nonce made of 4 bytes of timestamp and 28 bytes of random #. Similarly for r_G.
- SessID: 0 if new session, else is the session ID of an existing session (and the Handshake will update parameters)
- CiphList is a list of algorithms supported by the client in an order of decreasing preference (Key Exchange and Encryption Cipher)
- CiphChoice: The cipher suite chosen by the Server.

Handshake Protocol Action(2)



- **Phase2:** Server Authentication and Key Exchange
  - Server begins by sending its X.509 cert (and associated cert chain)
  - Next, a public key is sent (e.g. modulus and exponent, if RSA)
  - Server may Request a Cert from the Client
  - Server sends end phase 2 message

$$3.G \rightarrow A : \{G\_X509Cert\}$$

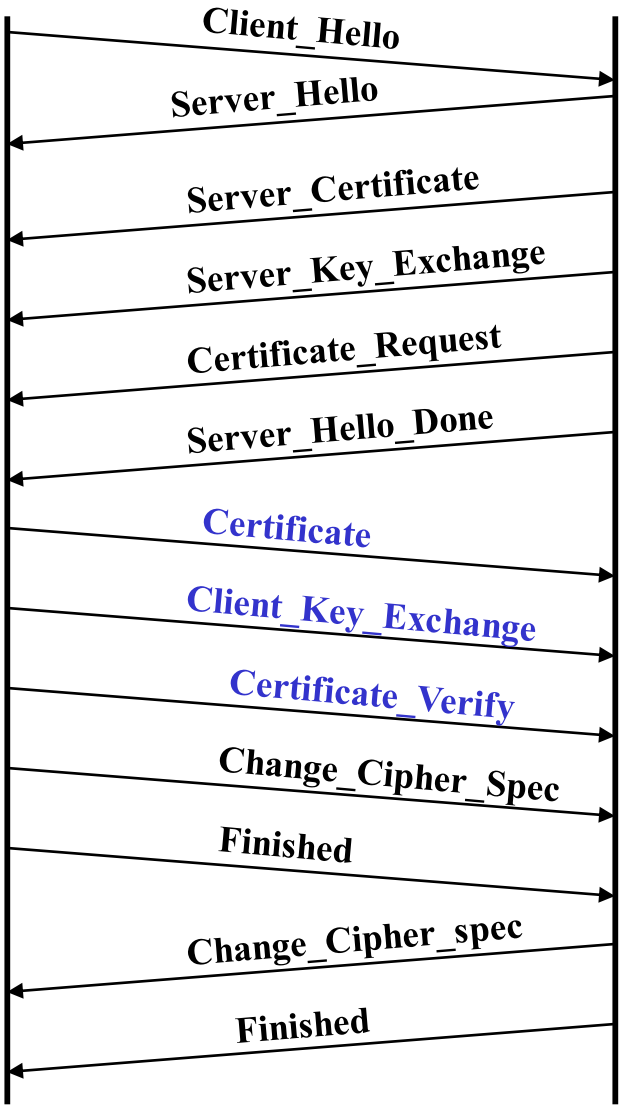
$$4.G \rightarrow A : \left\{(n_G, e_G) \,||\, E_{K_G}\left[\text{hash}\left(r_A \,||\, r_G \,||\, (n_G, e_G)\right)\right]\right\}$$

$$5.G \rightarrow A : \{\text{CertType} \,||\, \text{ValidCertA uthorities }\}$$

$$6.G \rightarrow A : \{\text{EndHello }\}$$

K_G is the private key, and hence E_{KG} is a signature operation by the Server
ValidCertAuthorities identifies the authorities the server will accept

Handshake Protocol Action(3)



- **Phase3:** Client Authentication and Key Exchange
  - Client verifies that the Server's Cert is valid, and checks that parameters sent are valid
  - If a cert was requested, then the Client sends one
  - Client generates a PreMasterSecret sPM

$$7.A \rightarrow G : \{A\_X509Cert\}$$

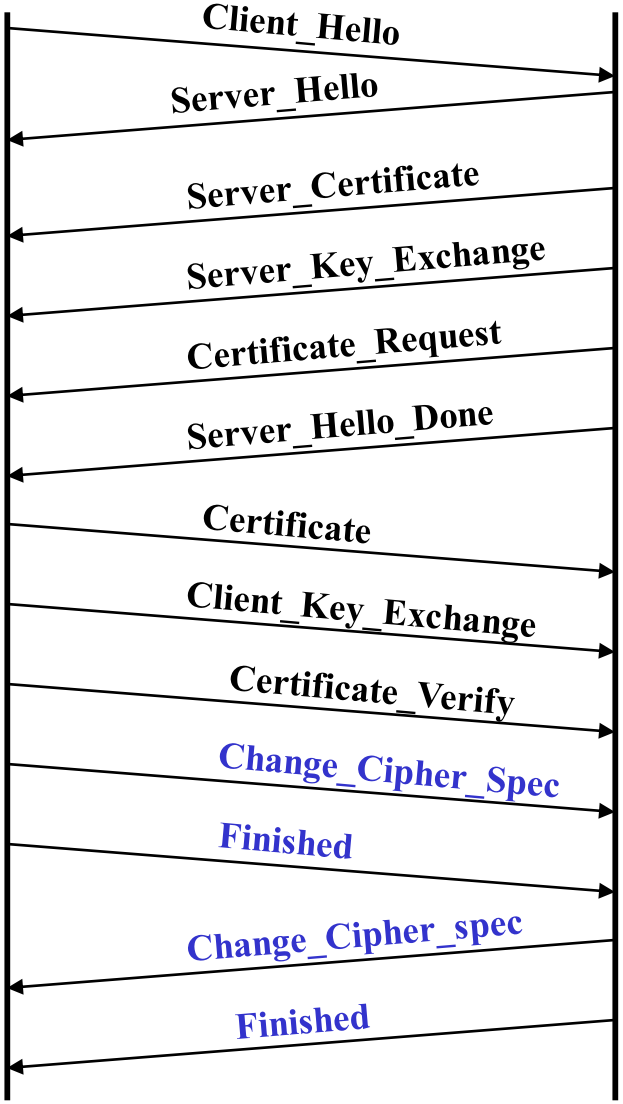
$$8.A \rightarrow G : \left\{E_{+K_G}[s_{PM}]\right\}$$

$$9.A \rightarrow G : \left\{hash(MS \,||\, r_G \,||\, hash(Messages1to8 \,||\, MS \,||\, r_A))\right\}$$

$$MS = MD5(s_{PM} \,||\, SHA1('A' \,||\, s_{PM} \,||\, r_A \,||\, r_G)) \,||$$
$$MD5(s_{PM} \,||\, SHA1('BB' \,||\, s_{PM} \,||\, r_A \,||\, r_G)) \,||$$
$$MD5(s_{PM} \,||\, SHA1('CCC' \,||\, s_{PM} \,||\, r_A \,||\, r_G))$$

$+K_G$ is the public key, and hence E_{+KG} is a encryption using the public key gained from the certificate
Messages1to8 is the concatenation of first 8 messages
MS is master secret and Step 9 is for verification

Handshake Protocol Action(4)



- **Phase 4:** Finish
  - Client tells Server to change cipher (via the Change Cipher Protocol).
  - Server responds with its own changed cipher message
  - Finished Message are hashes for verification

$$10. A \rightarrow G : \{ChangeCipher\}$$

$$11. A \rightarrow G : \{hash(MS \,||\, r_G \,||\, hash(Messages1to9 \,||\, Client \,||\, MS \,||\, r_A))\}$$

$$12. G \rightarrow A : \{CipherChanged\}$$

$$13. G \rightarrow A : \{hash(MS \,||\, r_G \,||\, hash(Messages1to9 \,||\, Server \,||\, MS \,||\, r_A))\}$$

Transport Layer Security

- The same record format as the SSL record format.
- Defined in RFC 4346 (April 2006); TLS 1.1
- Similar to SSLv3.
- Differences in the:
  - version number
  - message authentication code (HMAC)
  - pseudorandom function
  - alert codes
  - cipher suites
  - client certificate types
  - certificate_verify and finished message
  - cryptographic computations
  - padding

Transport Layer Security

- MAC: Use of HMAC algorithm in RFC2104
  - `HMAC_hash (MAC_write_secret, seq_num II TLSCompressed.type II TLSCompressed.version II TLSCompressed.length II TLSCompressed.fragment)`

Secure Electronic Transaction

- An open encryption and security specification.
- Protect credit card transaction on the Internet.
- Companies involved:
  - MasterCard, Visa, IBM, Microsoft, Netscape, RSA, Terisa and Verisign
- Not a payment system.
  - Set of security protocols and formats.

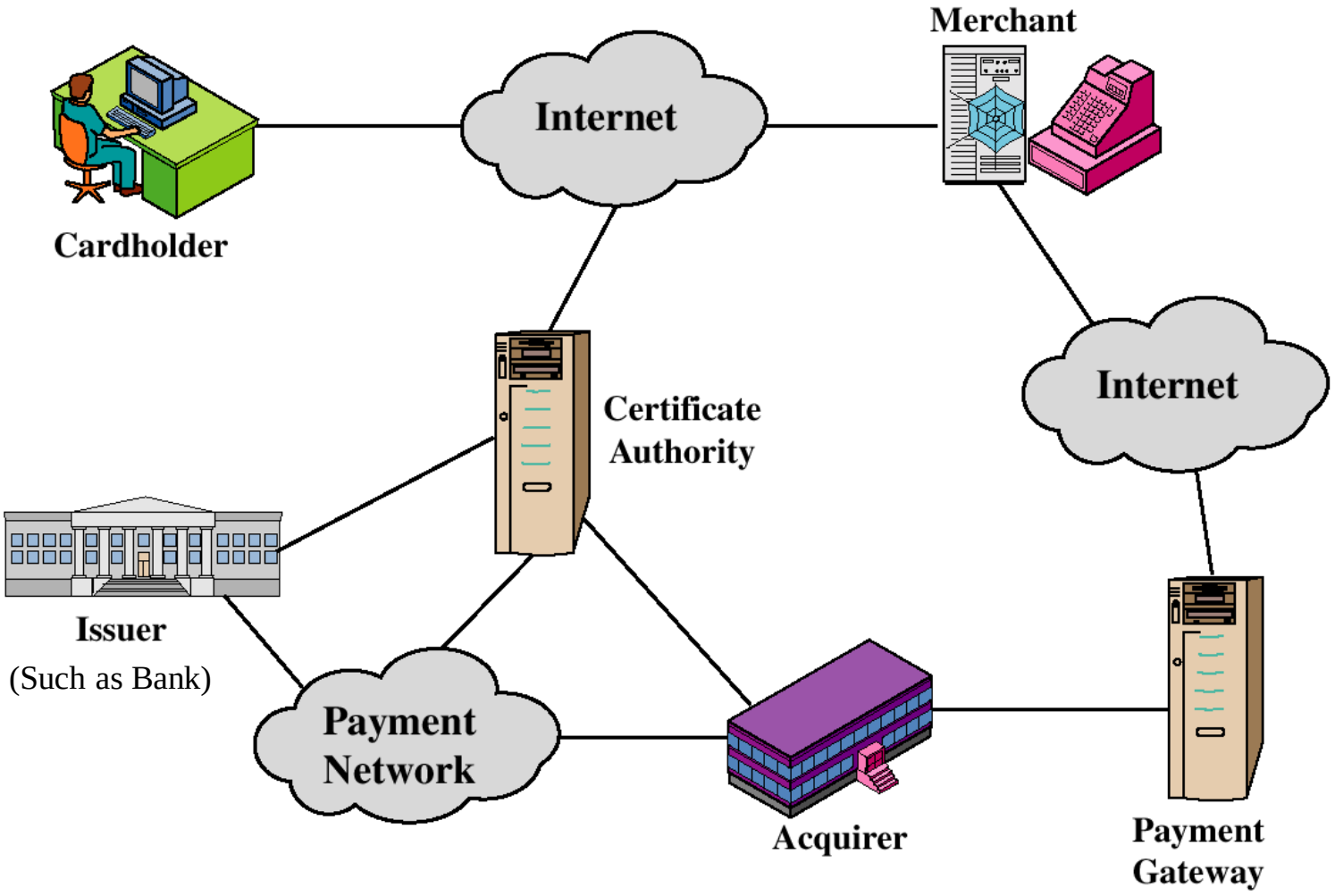
SET Services

- Provides a secure communication channel in a transaction.
- Provides tust by the use of X.509v3 digital certificates.
- Ensures privacy.

SET Overview

- Key Features of SET:
  - Confidentiality of information
  - Integrity of data (using SHA-1)
  - Cardholder account authentication
  - Merchant authentication

SET Participants

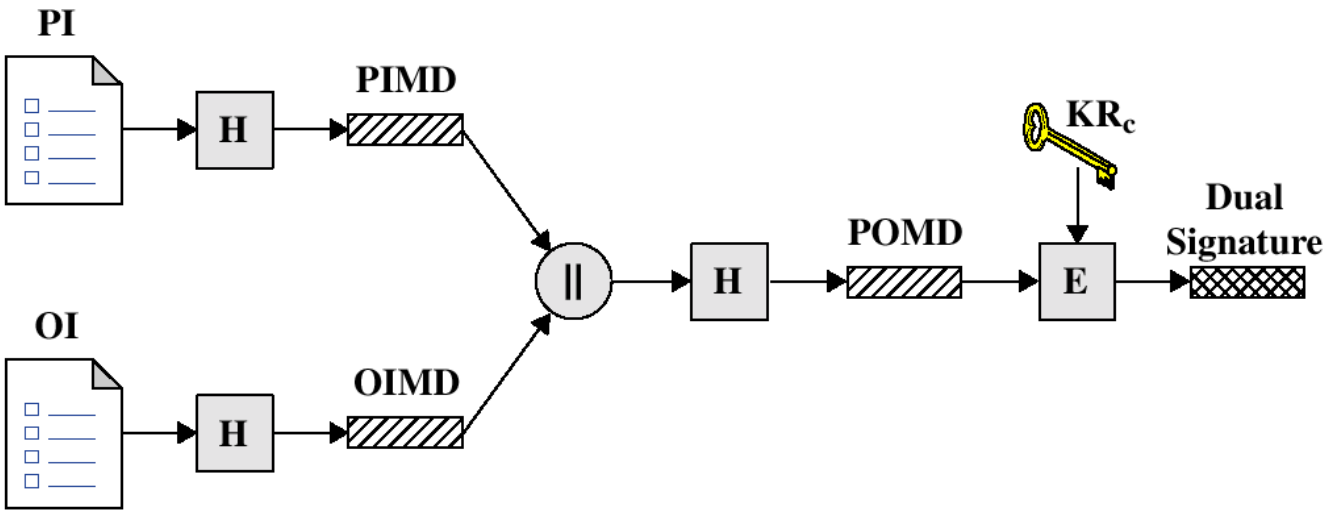


Sequence of events for transactions

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

Dual Signature

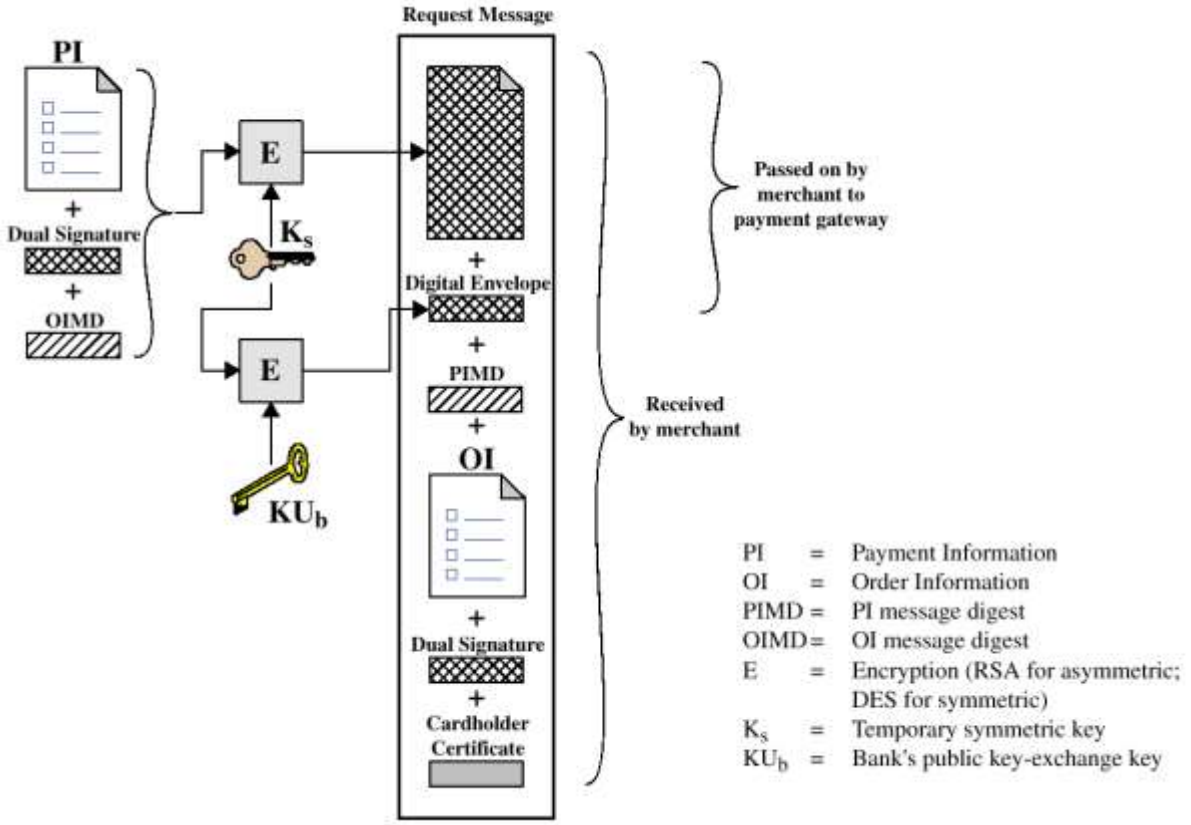
$$DS = E_{KR_c}[H(H(PI) \| \mathrm{H(OI)})]$$



PI = Payment Information
OI = Order Information
H = Hash function (SHA-1)
|| = Concatenation

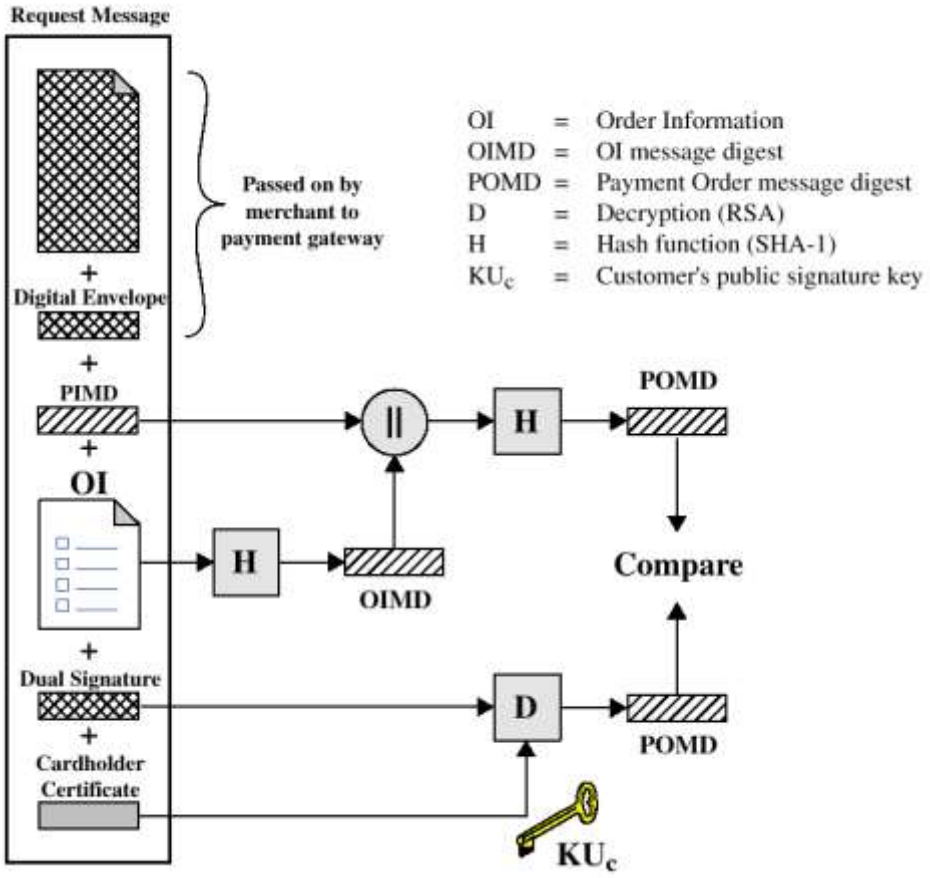
PIMD = PI message digest
OIMD = OI message digest
POMD = Payment Order message digest
E = Encryption (RSA)
KR_c = Customer's private signature key

Payment processing



Cardholder sends Purchase Request

Payment processing



Merchant Verifies Customer Purchase Request

Payment processing

- Payment Authorization
  (mechant - payment G/W - issuer)
  - Authorization Request
  - Authorization Response
- Payment Capture (mechant – Payment G/W)
  - Capture Request
  - Capture Response

Summary

- Secure socket layer (SSL) provides security services between TCP and application that use TCP. The Internet standard version is called transport layer service (TLS)
- SSL/TLS provides confidentiality using symmetric encryption and message integrity using a message authentication code
- SSL/TLS includes protocol mechanism to enable two two TCP users to determine the security mechanisms and services they will use
- Secure electronic transaction (SET) is an open encryption and security specification design to protect credit and transactions on the Internet