Web-Security

This presentation is prepared from slides provided by Computer Networking : A Top Down Approach 6th edition by Jim Kurose, Keith Ross and slides form Henric Johnson

Blekinge Institute of Technology, Sweden

http://www.its.bth.se/staff/hjo/

henric.johnson@bth.se

Mohammad Homayoon Fayez, Zealand Institute of Business and Technology, Roskilde, Zealand Denmark

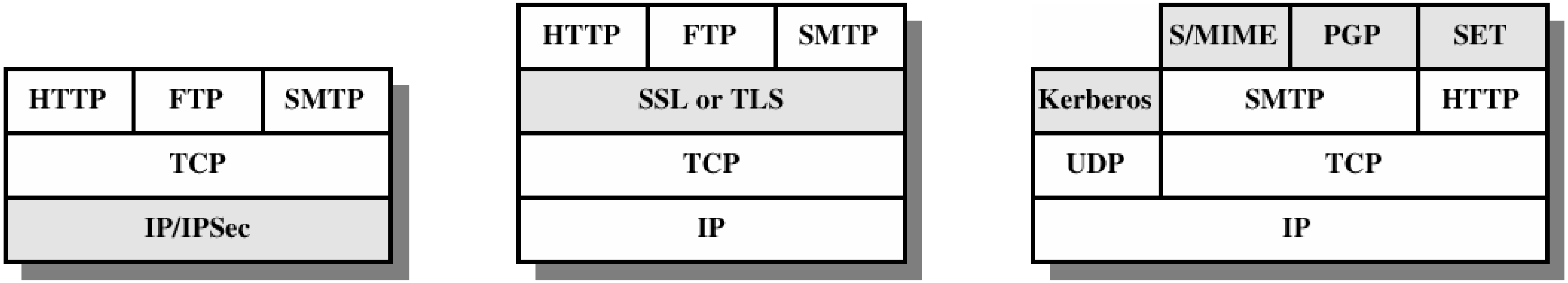
Outline

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

Web Security Considerations

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

Security facilities in the TCP/IP protocol stack



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

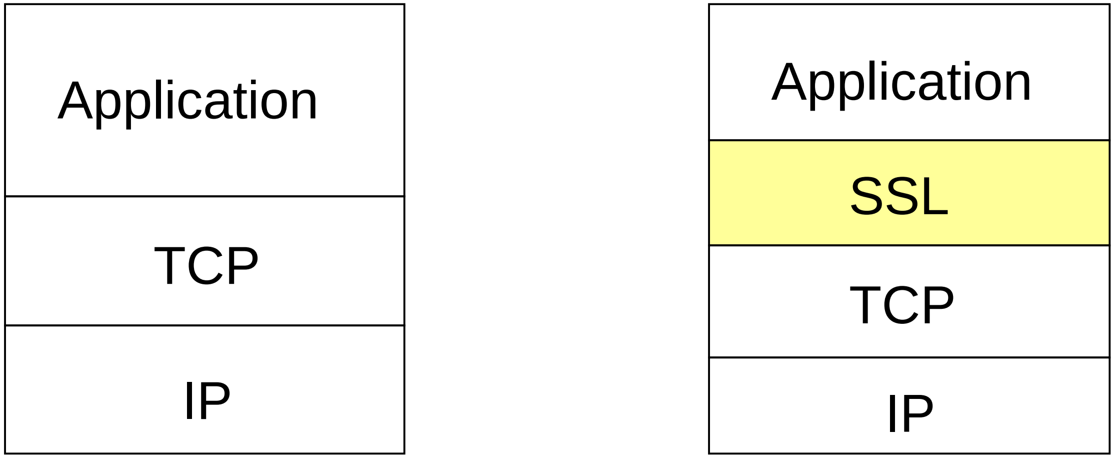
(c) Application Level

SSL: Secure Sockets Layer

- widely deployed security protocol
  - supported by almost all browsers, web servers
  - https
  - billions $/year over SSL
- mechanisms: [Woo 1994], implementation: Netscape
- variation -TLS: transport layer security, RFC 2246
- provides
  - *confidentiality*
  - *integrity*
  - *authentication*

- original goals:
  - Web e-commerce transactions
  - encryption (especially credit-card numbers)
  - Web-server authentication
  - optional client authentication
  - minimum hassle in doing business with new merchant
- available to all TCP applications
  - secure socket interface

SSL and TCP/IP

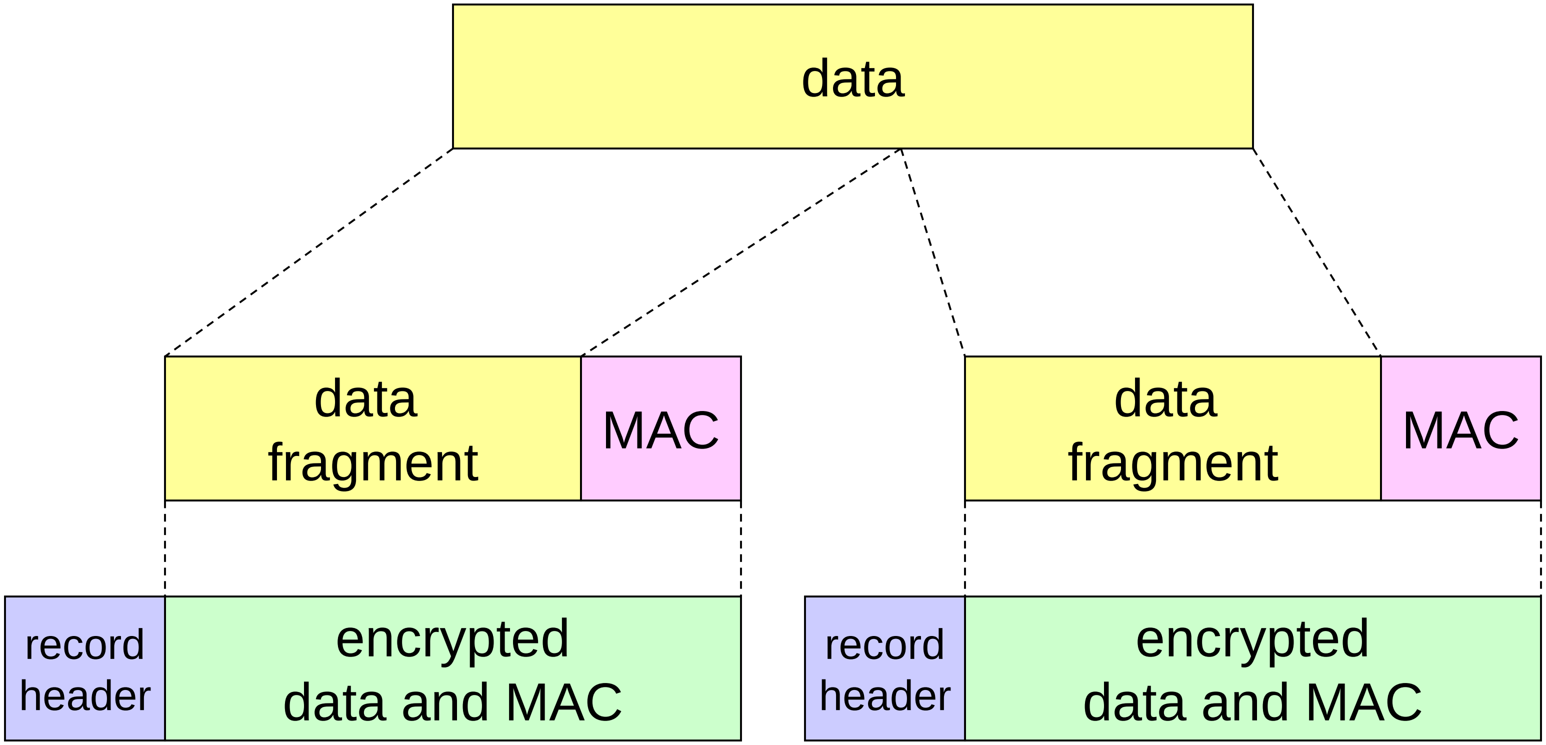


normal application

application with SSL

- SSL provides application programming interface (API) to applications
- C and Java SSL libraries/classes readily available

SSL record protocol

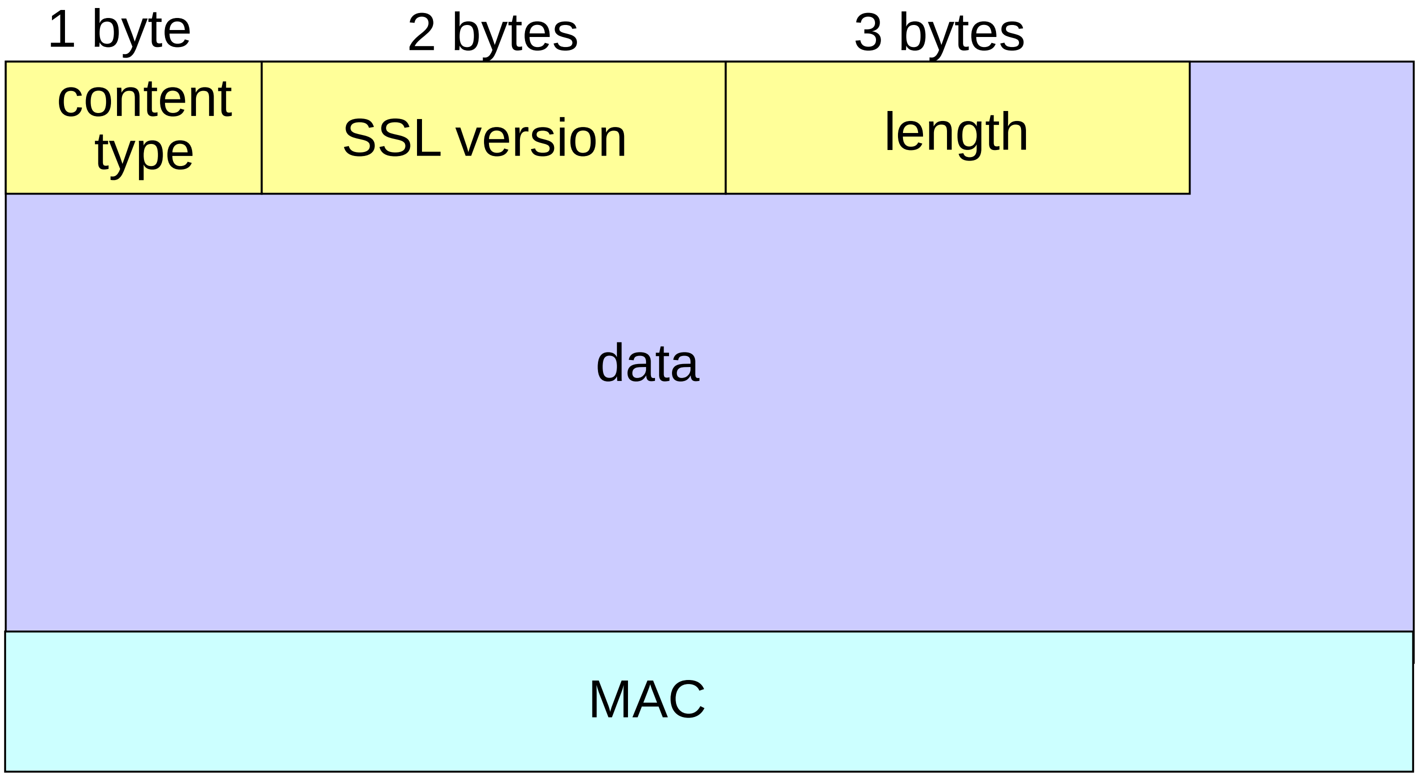


record header: content type; version; length

MAC: includes sequence number, MAC key M_x

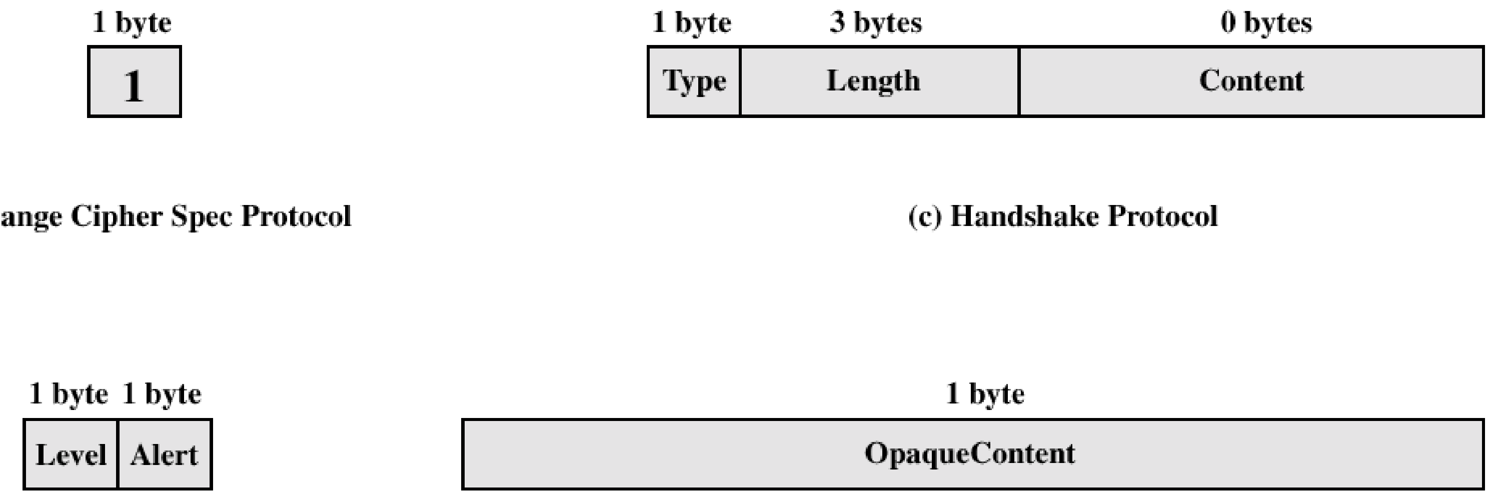
fragment: each SSL fragment 2^{14} bytes (~16 Kbytes)

SSL record format



data and MAC encrypted (symmetric algorithm)

SSL Record Protocol Payload



1 byte

1

(a) Change Cipher Spec Protocol

1 byte | 3 bytes | 0 bytes

Type | Length | Content

(c) Handshake Protocol

1 byte | 1 byte

Level | Alert

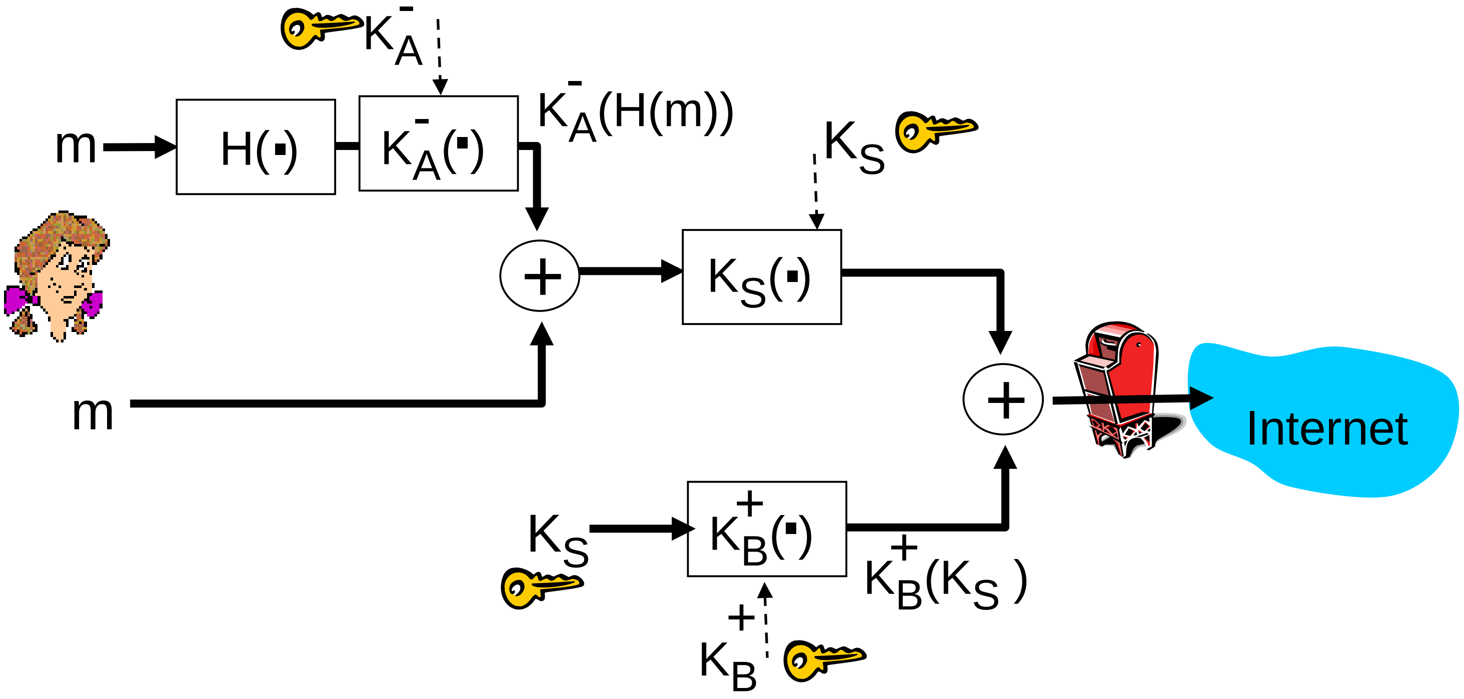
(b) Alert Protocol

1 byte

OpaqueContent

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

Could do something like PGP:



- but want to send byte streams & interactive data
- want set of secret keys for entire connection
- want certificate exchange as part of protocol: handshake phase

SSL cipher suite

- cipher suite
  - public-key algorithm
  - symmetric encryption algorithm
  - MAC algorithm
- SSL supports several cipher suites
- negotiation: client, server agree on cipher suite
  - client offers choice
  - server picks one

common SSL symmetric ciphers

- DES – Data Encryption Standard: block
- 3DES – Triple strength: block
- RC2 – Rivest Cipher 2: block
- RC4 – Rivest Cipher 4: stream

SSL Public key encryption

- RSA

Handshake Protocol

- 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.

SSL: handshake (1)

Purpose

1. server authentication
2. negotiation: agree on crypto algorithms
3. establish keys
4. client authentication (optional)

SSL: handshake (2)

1. client sends list of algorithms it supports, along with client nonce
2. server chooses algorithms from list; sends back: choice + certificate + server nonce
3. client verifies certificate, extracts server's public key, generates pre_master_secret, encrypts with server's public key, sends to server
4. client and server independently compute encryption and MAC keys from pre_master_secret and nonces
5. client sends a MAC of all the handshake messages
6. server sends a MAC of all the handshake messages

SSL: handshaking (3)

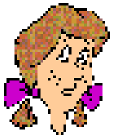
last 2 steps protect handshake from tampering

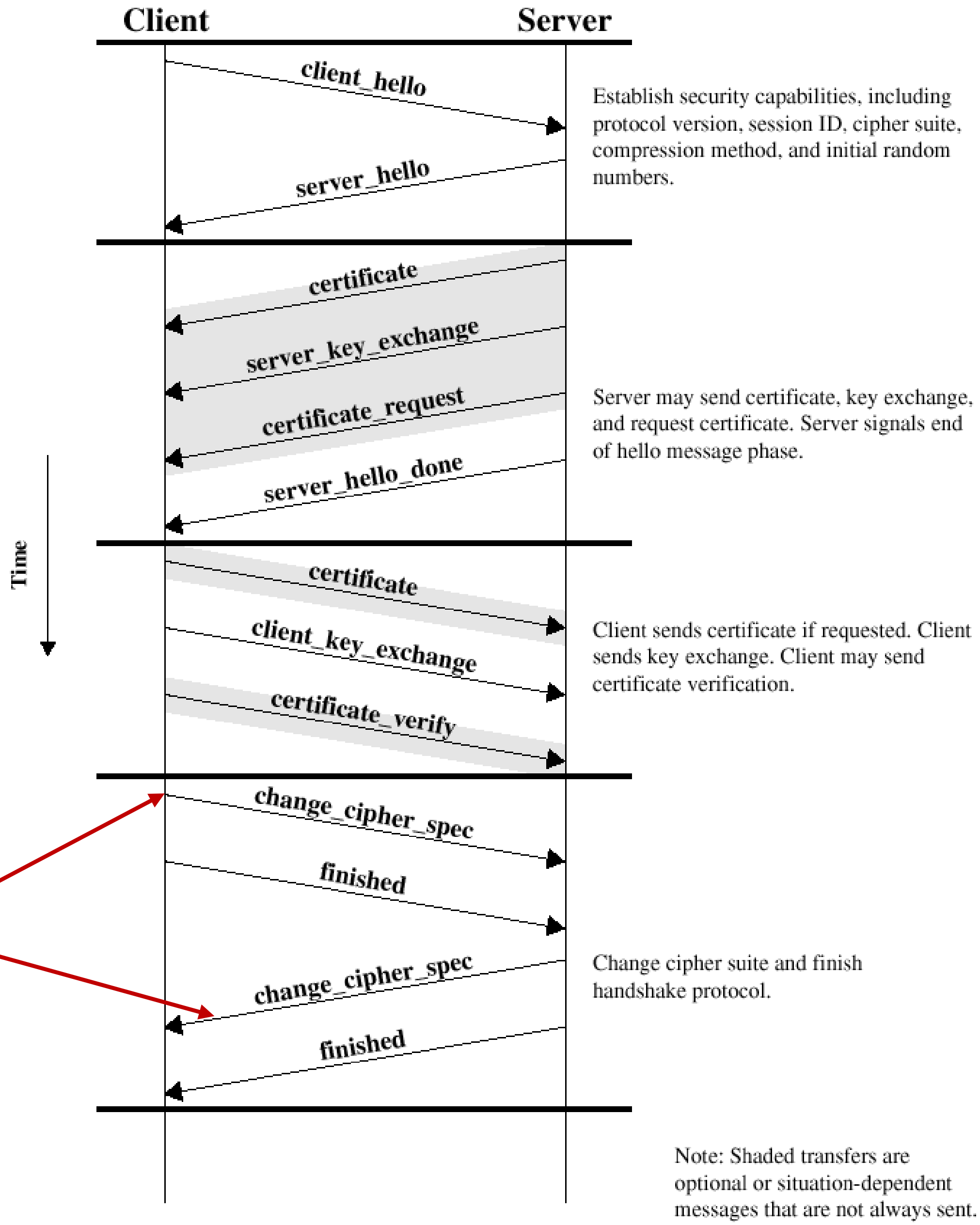
- client typically offers range of algorithms, some strong, some weak
- man-in-the middle could delete stronger algorithms from list
- last 2 steps prevent this
  - last two messages are encrypted

SSL: handshaking (4)

- why two random nonces?
- suppose Trudy sniffs all messages between Alice & Bob
- next day, Trudy sets up TCP connection with Bob, sends exact same sequence of records
  - Bob (Amazon) thinks Alice made two separate orders for the same thing
  - solution: Bob sends different random nonce for each connection. This causes encryption keys to be different on the two days
  - Trudy's messages will fail Bob's integrity check

SSL connection





Client

Server

Time

client_hello

server_hello

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

certificate

server_key_exchange

certificate_request

server_hello_done

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

certificate

client_key_exchange

certificate_verify

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

change_cipher_spec

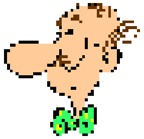
finished

change_cipher_spec

finished

Change cipher suite and finish handshake protocol.

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



everything henceforth is encrypted

TCP FIN follows

Key derivation

- client nonce, server nonce, and pre-master secret input into pseudo random-number generator.
  - produces master secret
- master secret and new nonces input into another random-number generator: “key block”
  - because of resumption: TBD
- key block sliced and diced:
  - client MAC key
  - server MAC key
  - client encryption key
  - server encryption key
  - client initialization vector (IV)
  - server initialization vector (IV)

Transport Layer Security

- The same record format as the SSL record format.
- Defined in RFC 2246.
- Similar to SSLv3.
- Differences in the:
  - version number
  - message authentication code
  - pseudorandom function
  - alert codes
  - cipher suites
  - client certificate types
  - certificate_verify and finished message
  - cryptographic computations
  - padding

Secure Electronic Transactions

- 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.

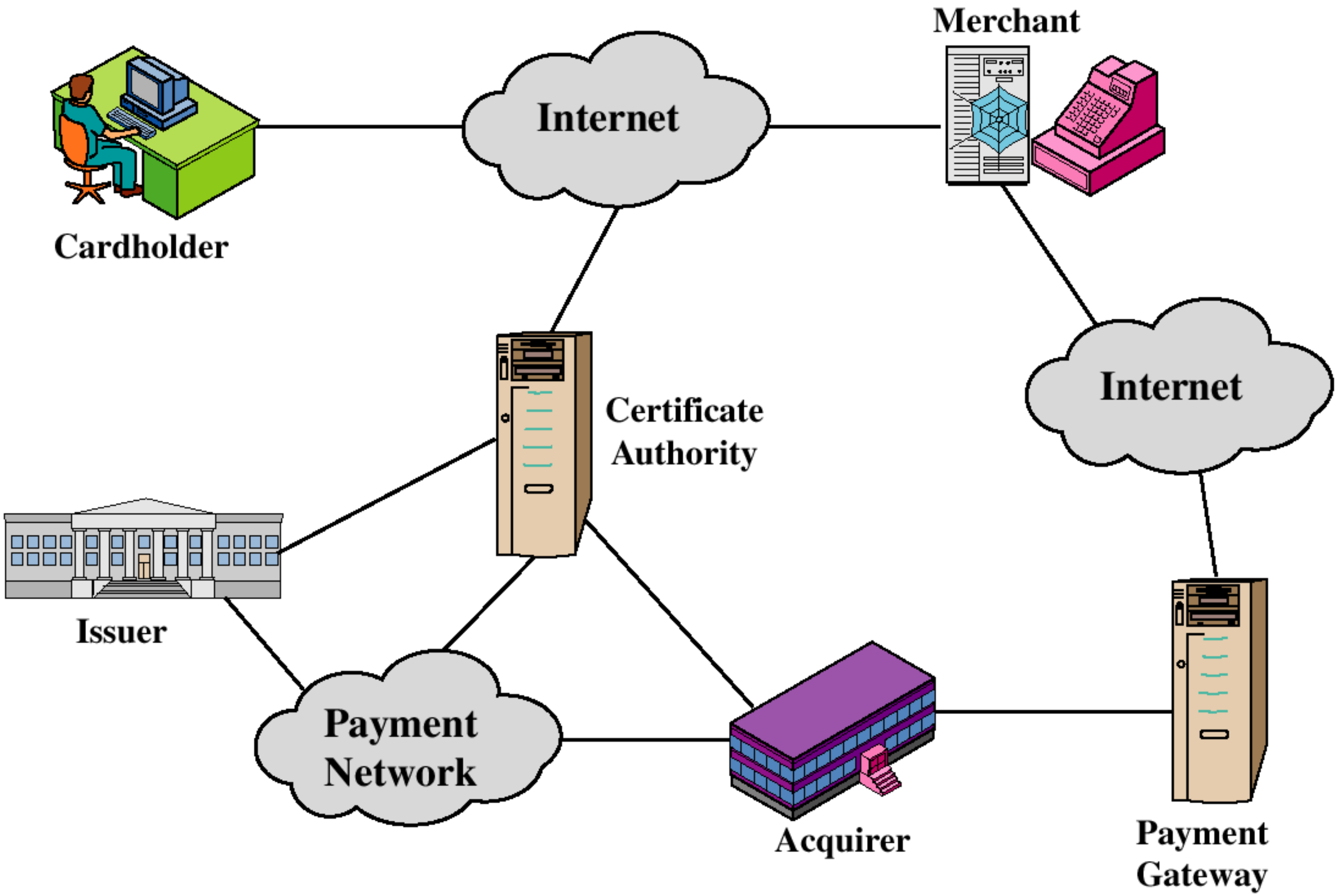
SET Services

- 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
  - Cardholder account authentication
  - Merchant authentication

SET Participants

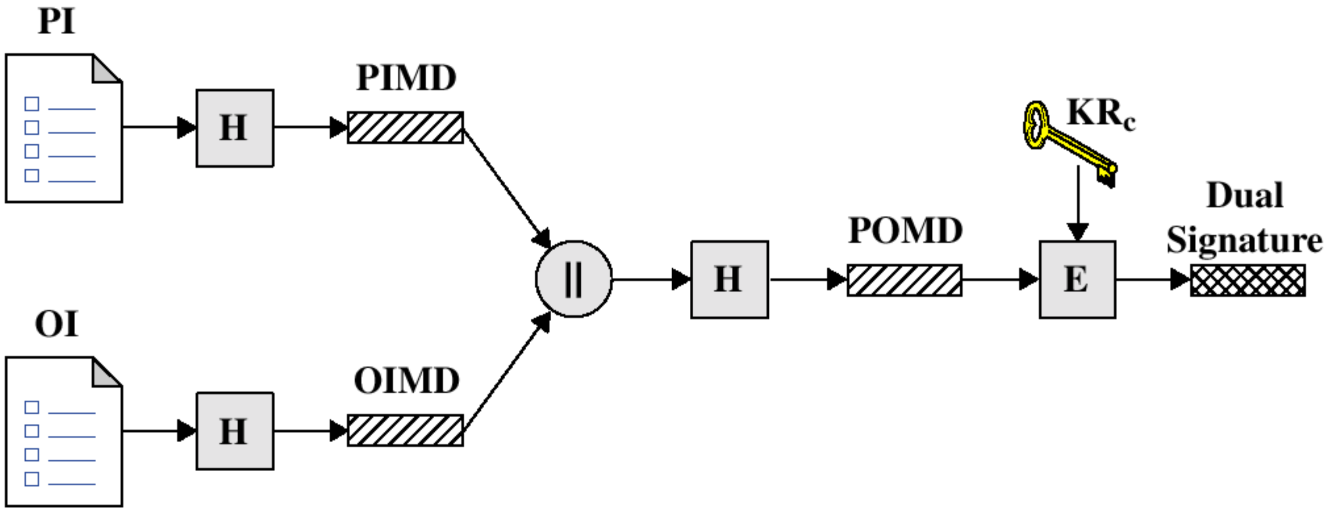


Sequence of events for transactions

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.

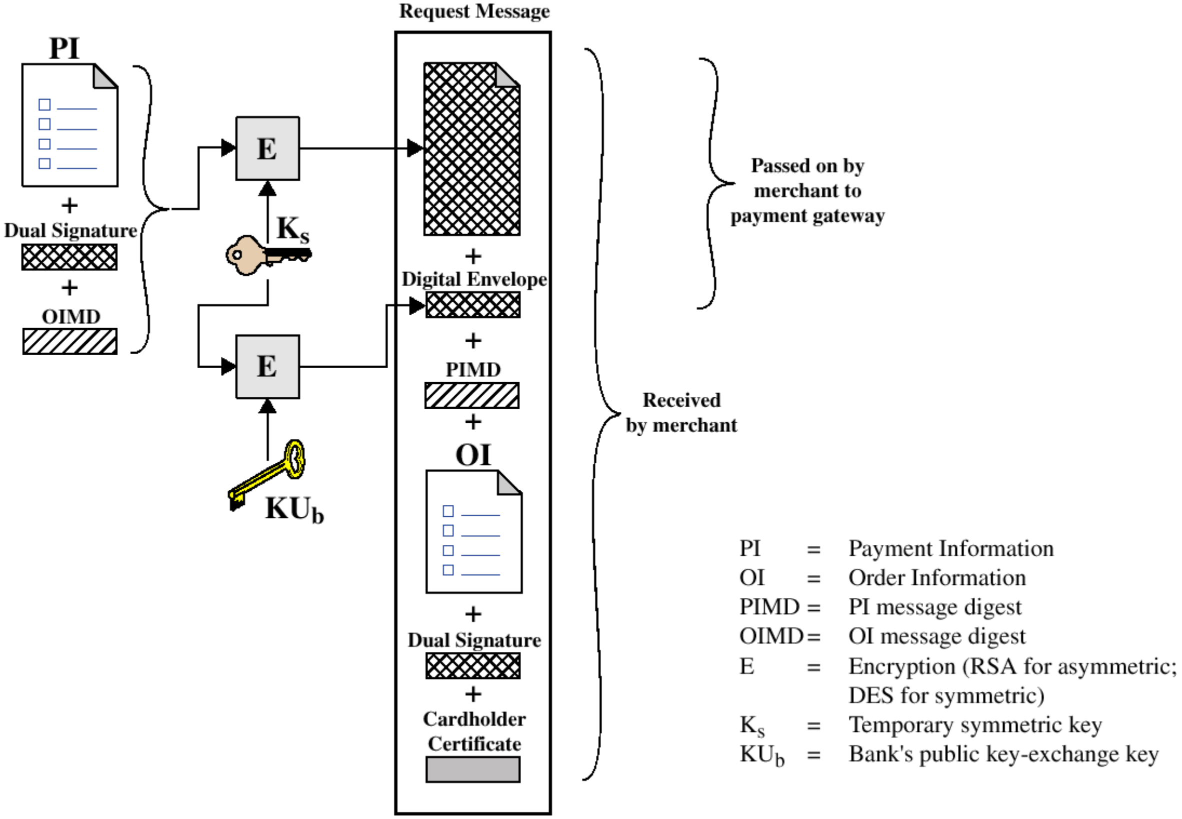
Dual Signature

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



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)
		KR_c =	Customer's private signature key

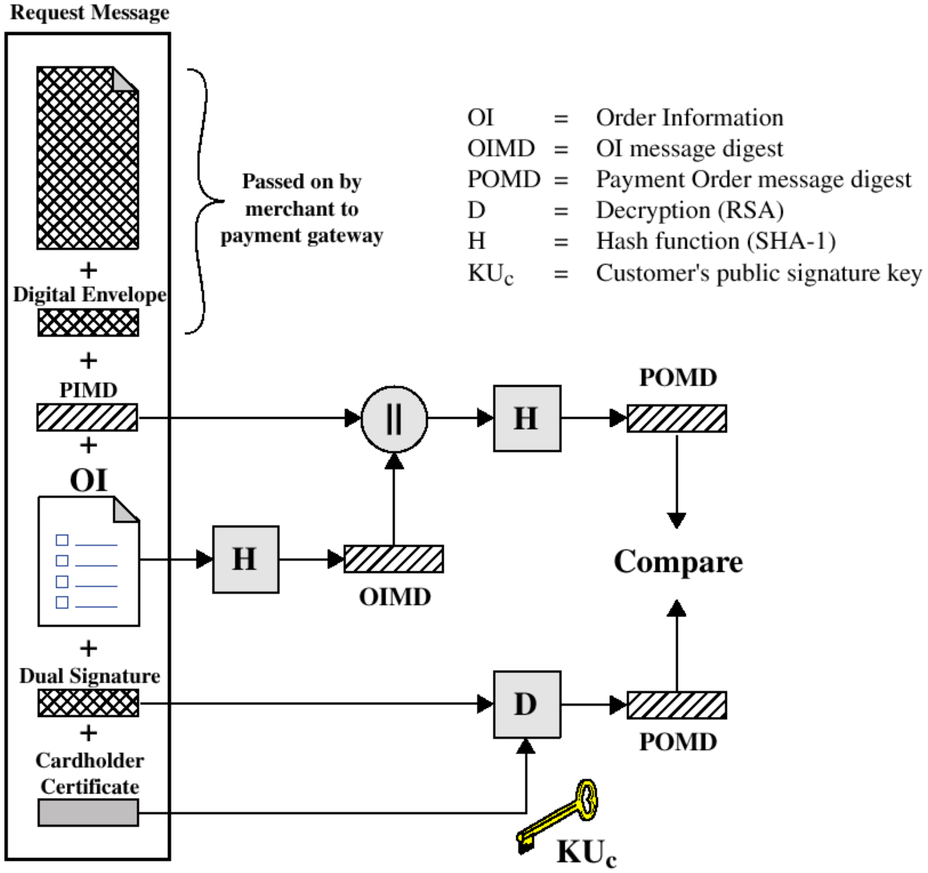
Payment processing



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

Cardholder sends Purchase Request

Payment processing



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

Merchant Verifies Customer Purchase Request

Payment processing

- Payment Authorization:
  - Authorization Request
  - Authorization Response
- Payment Capture:
  - Capture Request
  - Capture Response

Recommended Reading and WEB sites

- Drew, G. *Using SET for Secure Electronic Commerce*. Prentice Hall, 1999
- Garfinkel, S., and Spafford, G. Web Security & Commerce. O'Reilly and Associates, 1997
- MasterCard SET site
- Visa Electronic Commerce Site
- SETCo (documents and glossary of terms)