E-commerce security: SSL/TLS, SET and others.

4.1

Universidad Carlos III de Madrid
Grupo SeTI · Dpto. Informática

Electronic payment systems

Purpose:

facilitate the safe and secure transfer of monetary value electronically between multiple parties

Participating parties:

- Buyer (payer)
- Merchant (payee)
- Issuer (bank interacting with payer)
- Acquirer (bank interacting with payee)
- Arbiter
- Other entities (card associations, clearinghouses…)

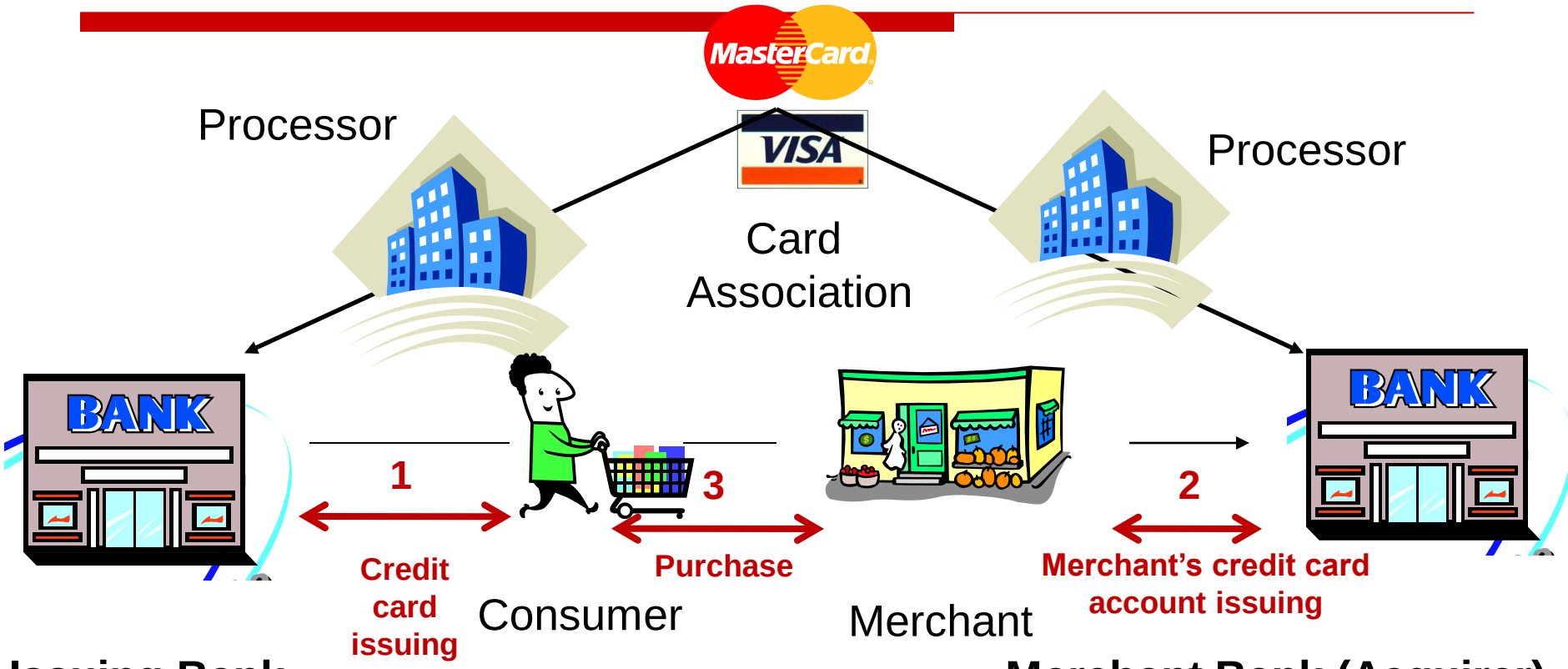
Electronic Payment Systems: Classification

- Simile to real-world payment systems
  - Cash
  - Check
  - **Credit card**
  - Stored Value (e.g., debit cards, pre-paid cards, smart cards)
  - Accumulating Balance

Electronic Payment Systems: Credit Card

- Represents an account that extends credit to consumers, permitting consumers to purchase items while deferring payment, and allows consumers to make payments to multiple vendors at one time
- Credit card associations – Nonprofit associations (Visa, MasterCard) that set standards for issuing banks
- Issuing banks – Issue cards and process transactions
- Processing centers (clearinghouses) – Handle verification of accounts and balances

Credit card - Participants



Issuing Bank

- Issues card
- Extends credit
- Assumes risk of card
- Cardholder reporting

Merchant Bank (Acquirer)

- Sets up merchant
- Extends credit
- Assumes risk of merchant
- Funds merchant

Credit card – Process flow

- Purchase at merchant's shop using a POS (Point of Sale) terminal
  1. Authorization
  2. Batching
  3. Clearing and settlement
  4. Funding

Credit card-based electronic payment systems

- **Problem:** communicate credit card # and purchasing data securely through Internet (at least)
  - Authentication of buyer and merchant
  - Confidential transmissions
- Systems vary by
  - type of public-key encryption
  - type of symmetric encryption
  - message digest algorithm
  - number of parties having private keys
  - number of parties having certificates

SOURCE: MICHAEL I. SHAMOS

Credit card-based electronic payment systems

- **SSL** (Netscape, 1994)
  - 1 or 2 parties have private keys
- **TLS** (IETF, 1999-2008)
  - IETF version of SSL - stronger algorithms
  - The Transport Layer Security (TLS) Protocol Version 1.2 (RFC 5246), 2008
- CyberCash (CyberCash Corp., 1995)
- SEPP (MasterCard, IBM, Netscape,1995)
- STT (VISA, Microsoft, 1995)
- **SET** (Visa+Mastercard, 1996)
  - All parties have digital certificates
- **3-D Secure** (Visa, 2002)
  - On-line authentication

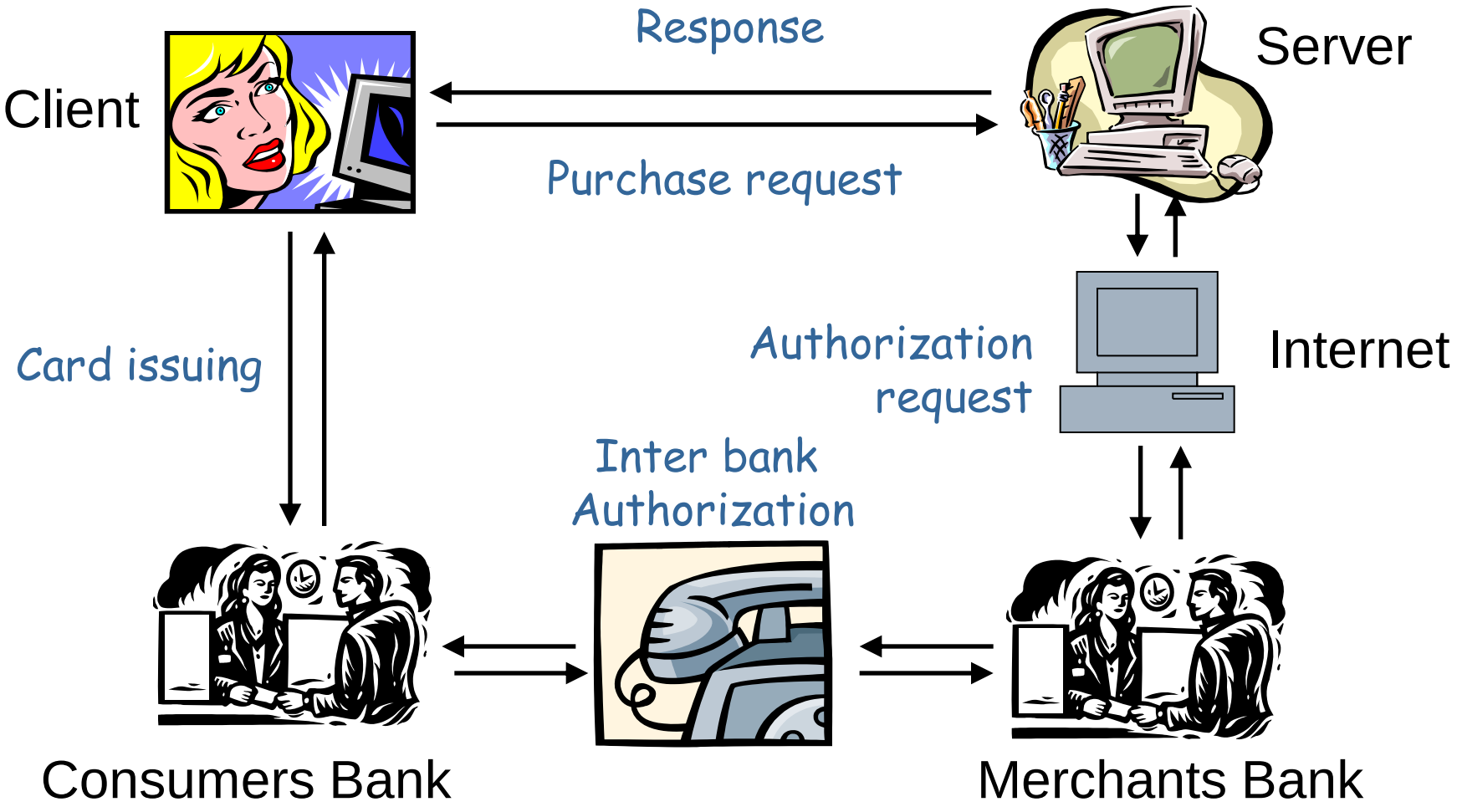
VERY IMPORTANT. USAGE INCREASING

OBSOLETE

VERY SLOW ACCEPTANCE; DEAD

RAPID EXPANSION

Electronic payment systems
Credit card + SSL: Participants

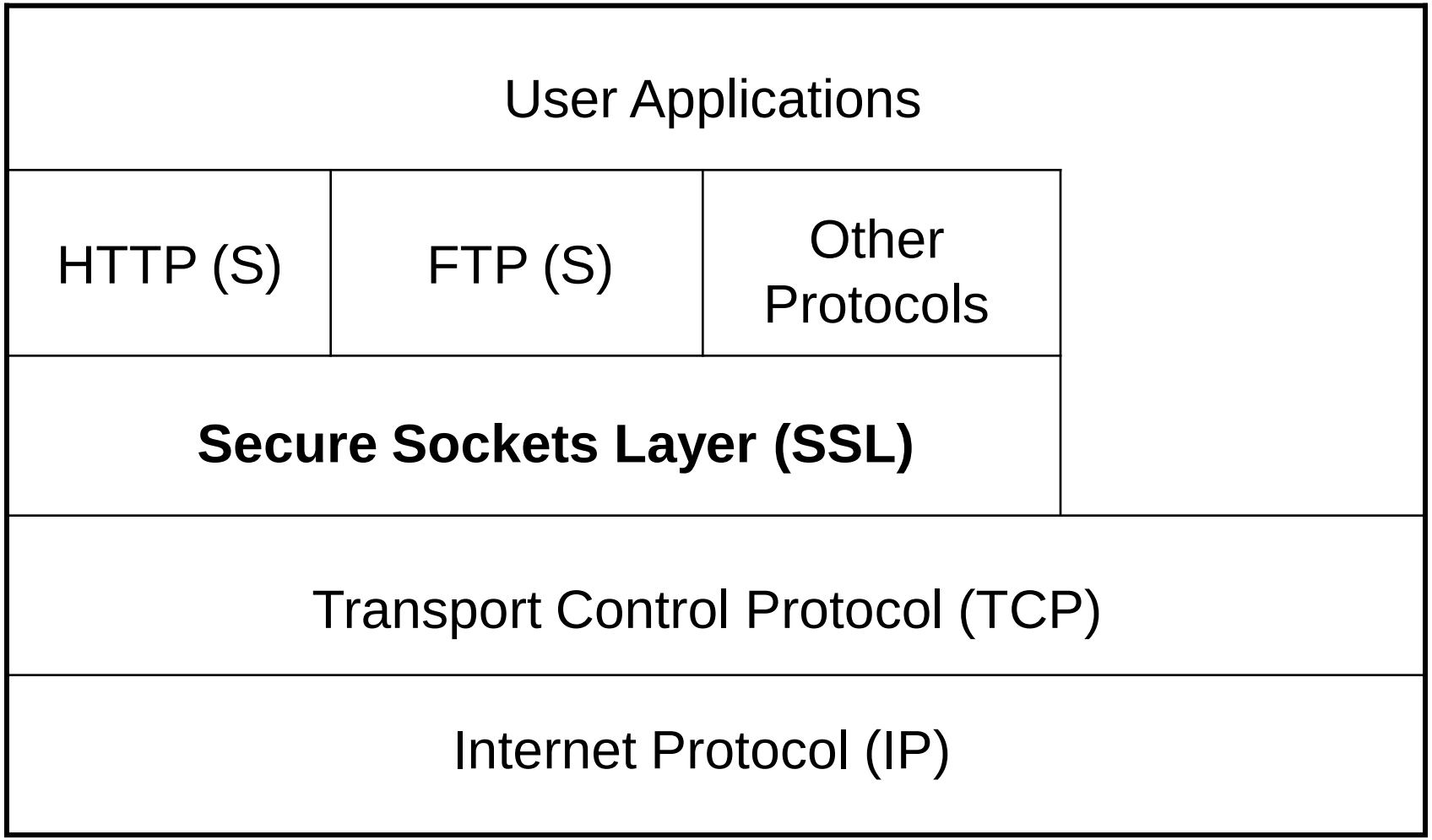


Secure Sockets Layer (SSL)

= Security protocol:

- Created by Netscape® (1994)
- Works between the application level and the transport protocol (usually TCP/IP).
  - Needs a reliable end-to-end transport service
- Adds security features to information stream (confidentiality, integrity…)
- Provides the service to protocols at application level: HTTP (https), FTP, Telnet, POP 3, SMTP, ...

Secure Sockets Layer (SSL)



Secure Sockets Layer (SSL)

Characteristics:

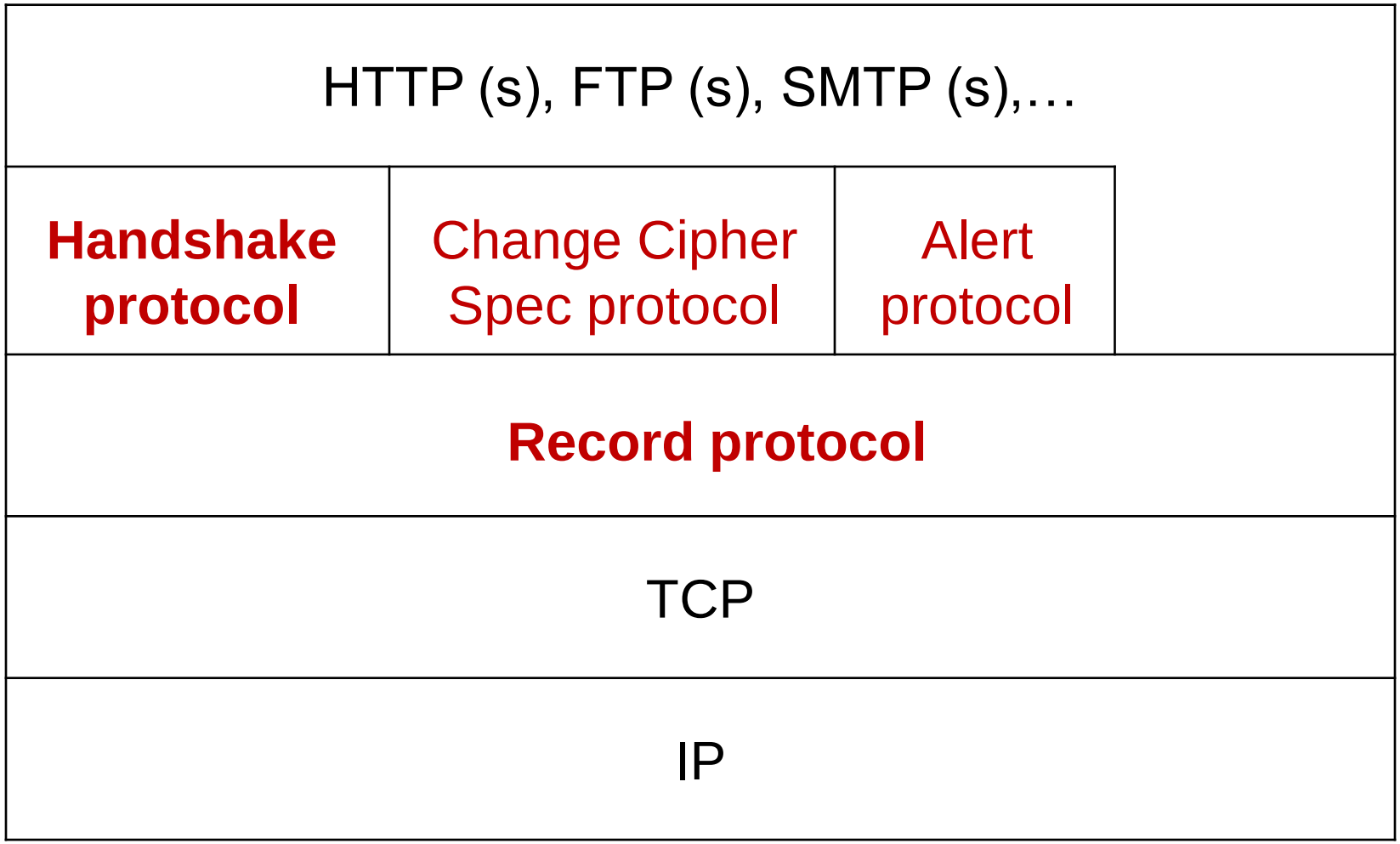
- Establishes a secure channel on the transport level between two parties
  - NOT a payment protocol -- can be used for any secure communications, like credit card numbers
- Supports compression (optional)
- Provides various security services

Secure Sockets Layer (SSL)

Security services:

- Peers authentication: X.509 v3 certificates
  - Server
  - Client (optional)
- Integrity: MACs
- Confidentiality: symmetric encryption with a session key

Secure Sockets Layer (SSL)



Secure Sockets Layer (SSL)

2 Sub-layers:

- Higher layer:
  - **Handshake protocol**
    - Authenticates peers' identity
    - Negotiates cipher suite
    - Establishes secret information
  - Change Cipher Spec protocol
  - Alert protocol
- Lower layer:
  - **Record protocol**
    - Packs/unpacks records, compression (optional)
    - MAC calculation/verification and encryption/decryption

SSL: Handshake Protocol

- Most complex protocol within SSL
- Permits mutual authentication of server and client (optionally)
- Negotiates the cryptographic algorithms:
  - Key exchange
  - Encryption and MAC
- Negotiates the cryptographic keys
  - A master secret from which keys for encryption and MAC are derived
- Takes place before the transmission of application data

SSL Handshake Protocol - Overview

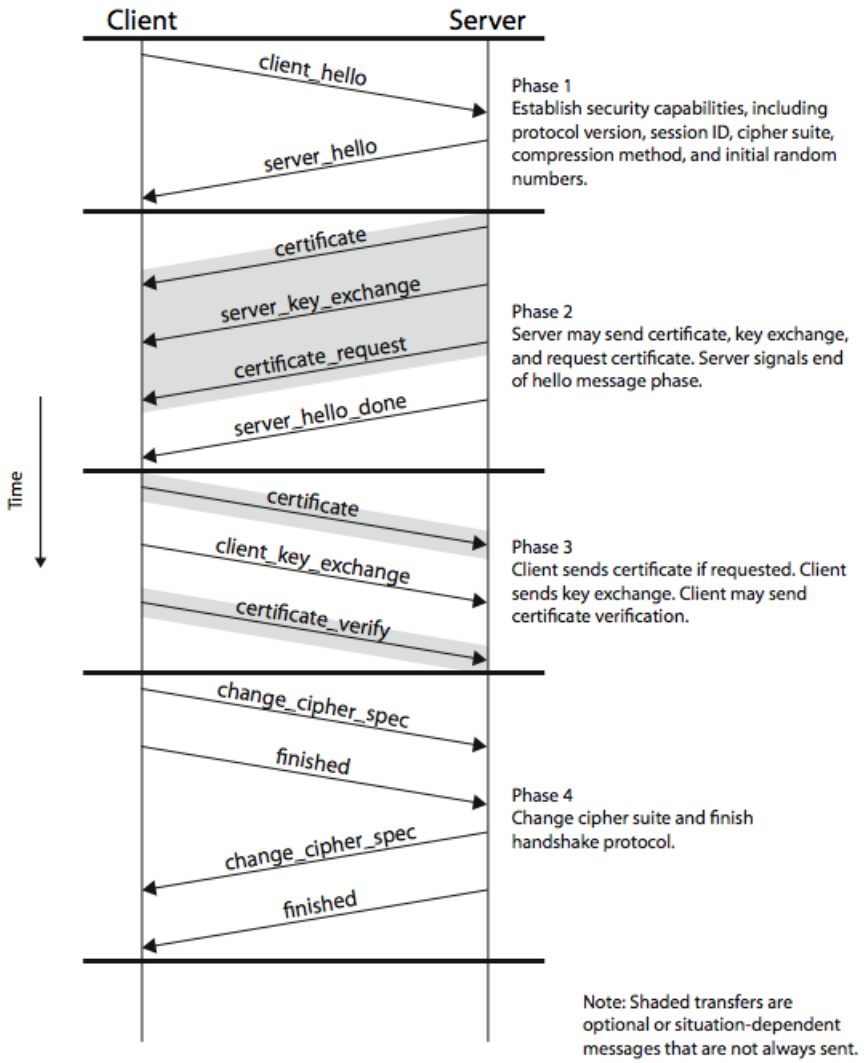
Phase 1: Establish security capabilities

Phase 2: Server authentication and key exchange

Phase 3: Client authentication and key exchange

Phase 4: Finish

Source: W. Stallings and L. Brown. Slides of Chapter 17. Cryptography and Network Security. 4th edition.



SSL Handshake Protocol: Key Exchange algorithm

- **RSA**
  - Certified server's public key
  - Client sends premaster secret encrypted (enveloped)
- Fixed Diffie-Hellman (DH)
  - Certified server's DH public parameters
  - Client sends its DH public parameters in certificate or key exchange message

SSL Handshake Protocol: Key Exchange algorithm

- **Ephemeral Diffie-Hellman (EDH)**
  - Peers exchange their DH public parameters signed with their RSA or DSS private keys
  - Corresponding certified RSA or DSS public keys
- Anonymous Diffie-Hellman (ADH)
  - Non authenticated DH public parameters
  - Vulnerable to Man-in-the-middle attacks

Diffie-Hellmann key exchange

New Directions in Cryptography, Whitfield Diffie, Martin E. Hellman. *IEEE Transactions in Information Theory*, vol. IT-22, pp 664-654. Noviembre de 1976

Diffie-Hellmann key exchange

A and B negotiate the following 2 public values:

- p: a very big prime number (> 512 bits)
- g: a primitive root (generator)

A chooses a very large random integer x and sends to B:

$$X = g^x \bmod p$$

B chooses a very large random integer y and sends to A:

$$Y = g^y \bmod p$$

A and B calculate respectively:

$K_A = Y^x = g^{yx} \bmod p$ and $K_B = X^y = g^{yx} \bmod p$

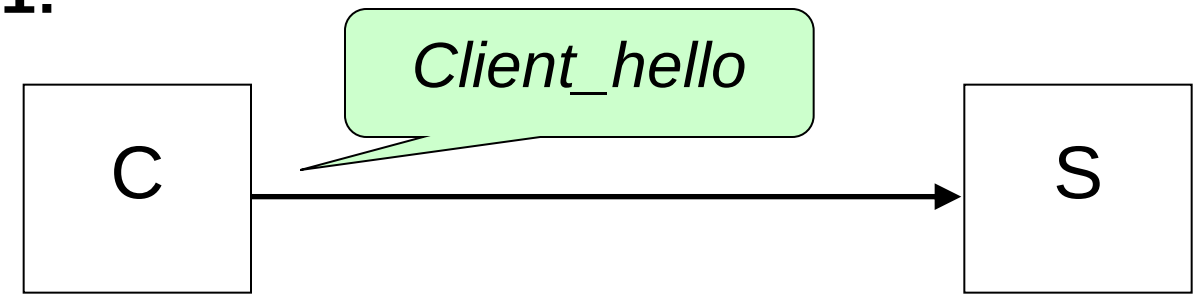
Key:
$K = Y^x = X^y$

SSL Handshake Protocol: Master Secret Computation

- RSA:
  - Server decrypts premaster secret using its private key
- Diffie Hellman:
  - Both client and server exchange Diffie-Hellman public keys
  - Both perform calculation of premaster secret
- Both client and server derive master secret from premaster secret and the exchanged random values

SSL Handshake Protocol - Details

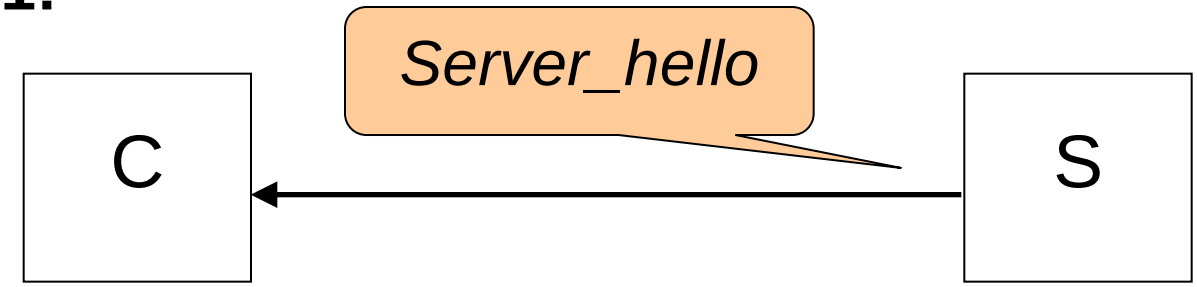
Phase 1:



- Client:
  - Creates random number including a time stamp
  - Sends supported Cipher Suites consisting in:
    - Key exchange method
    - Cipher algorithms for data transfer
    - Message digest for creating MAC
  - Sends type of compression (if used)

SSL Handshake Protocol - Details

Phase 1:



- Server:
  - Creates random number
  - Sends selected cipher suite
  - Sends compression method (if used)

SSL Handshake Protocol - Details

Phase 2:

- Server:
  - Sends certificate (except ADH)
  - Sends public key parameters (except DH and RSA)
  - Requests client certificate (optional)
  - Server Hello Done
- Client:
  - Server's certificate validation

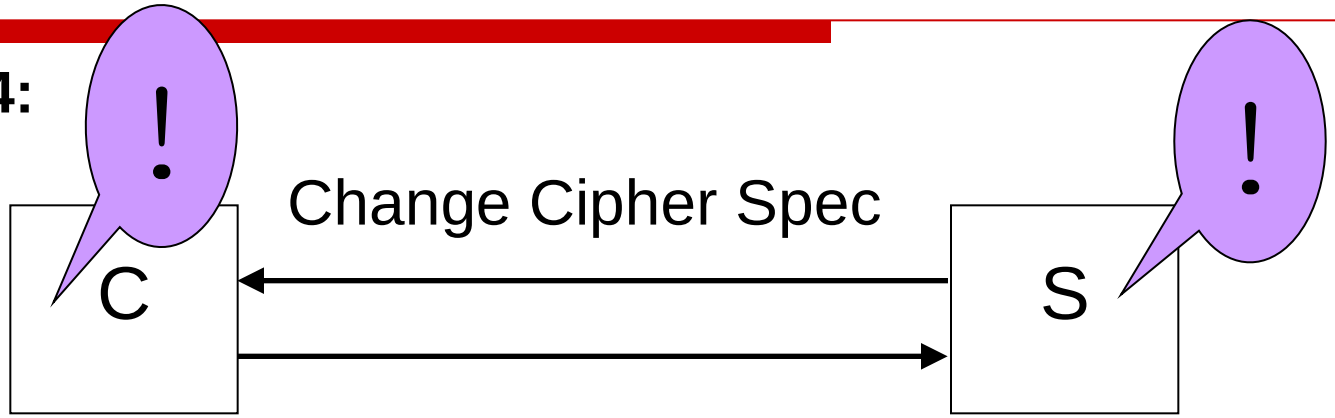
SSL Handshake Protocol - Details

Phase 3:

- Client:
  - Sends certificate (if requested)
  - Creates pre-master secret for key exchange (for RSA).
  - Sends encrypted pre-master secret (for RSA) or public parameters (for EDH and ADH) in key exchange msg
- Server:
  - Client's certificate validation

SSL Handshake Protocol - Details

Phase 4:

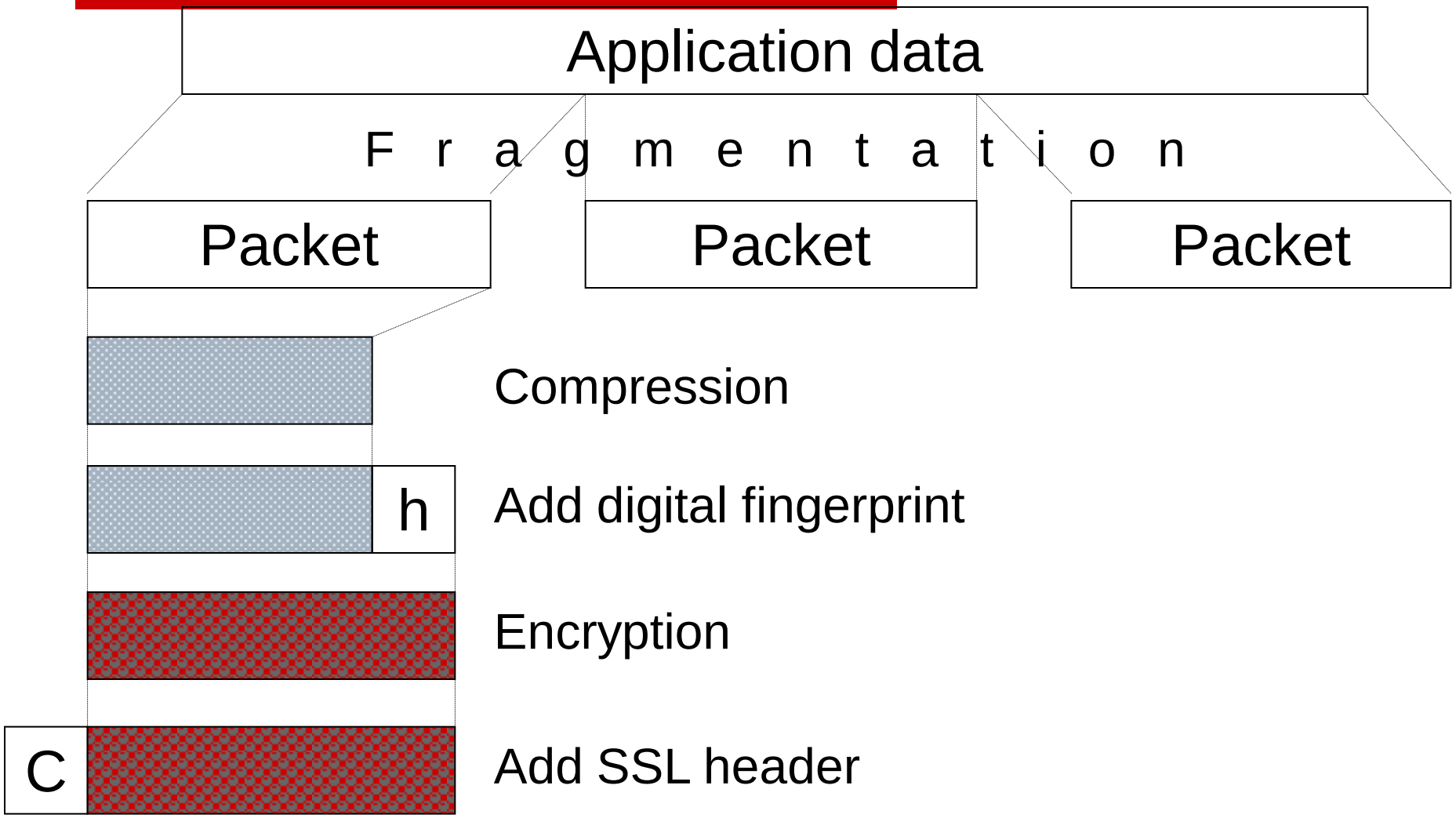


- Client:
  - If DH, EDH, ADH: computes premaster secret
  - Pre-master secret → master secret
  - Sends Change Cipher Spec
- Server:
  - If RSA: decrypts pre-master secret
  - If DH, EDH, ADH: computes premaster secret
  - Pre-master secret → master secret
  - Confirms Change Cipher Spec

SSL: Change Cipher Spec Protocol

- Indicates a change in the used ciphers (algorithm, keys etc.).
- Single message encrypted with current cipher spec sent by client and server.
- Phase 4 in Handshake protocol.

SSL: Record protocol



SSL: Record protocol

1. Fragmentation
   - Division of messages > 2^{14} bytes in smaller blocks
   - Or combining multiple higher level protocol data messages into single units
2. Compression
   - With negotiated algorithm (optional)
3. Message authentication code
   - Verification of data coming from TCP level
   - Authentication of messages from higher levels
   - With negotiated algorithm and keys
   - Concatenate message with secret number and sequence number and calculate its hash (TLS uses HMAC algorithm instead)

SSL: Record protocol

4. Encryption
   - Information to be encrypted: application data + MAC
   - With negotiated algorithm and exchanged keys
   - Stream ciphers:
     - RC-4 (40 y 128 bits)
   - Block ciphers:
     - IDEA (128 bits)
     - RC-2 (40 bits)
     - DES (40 y 56 bits)
     - Triple DES (168 bits)
     - Fortezza
5. Header
   - Identification of protocol
   - Byte length

SSL: Alert Protocol

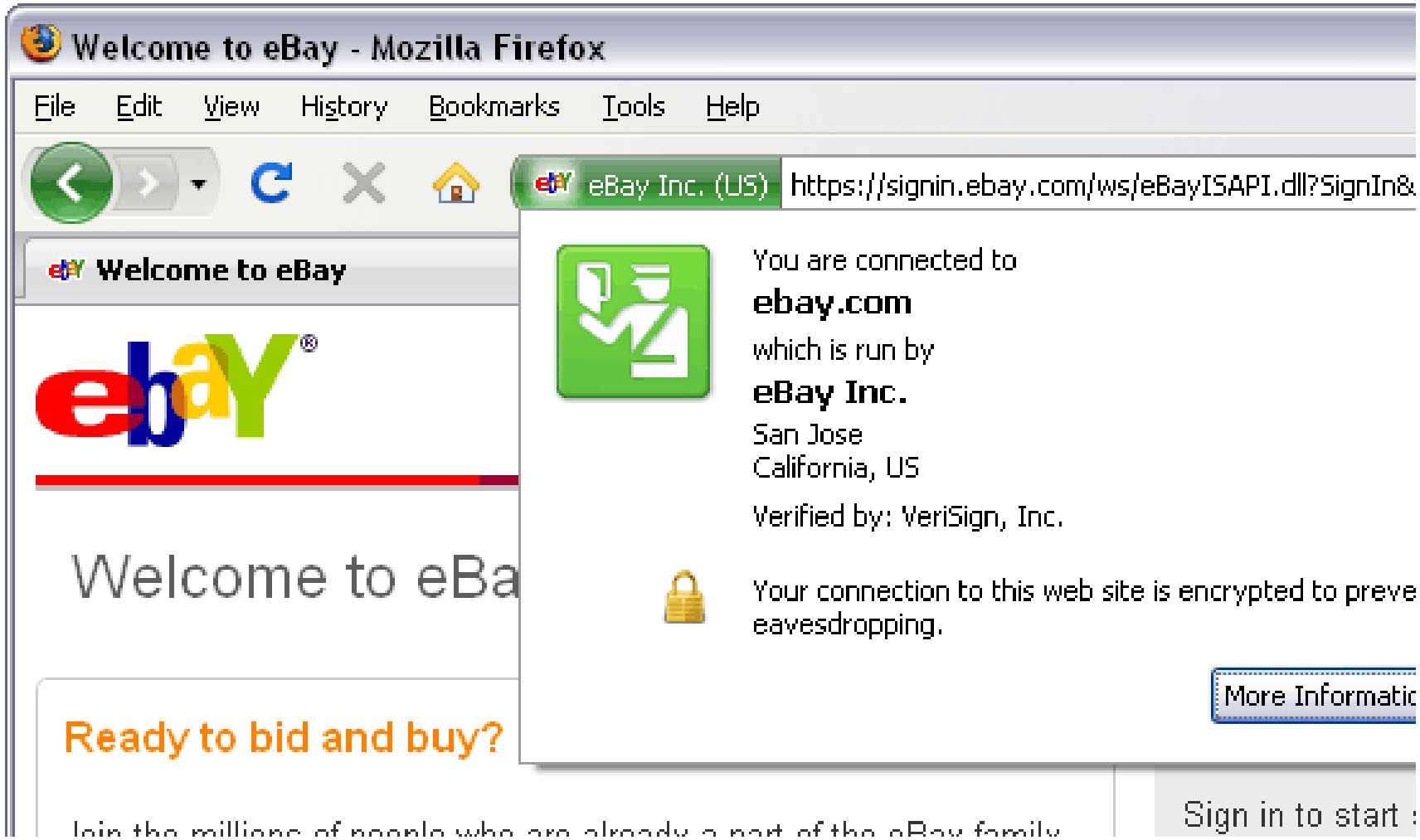
- Information about certain events
  - Description
  - Severity
- Events
  - Error conditions (bad MAC)
  - Certificate expired
  - Illegal parameter
  - Planned connection termination

Extended Validation SSL (EV-SSL)

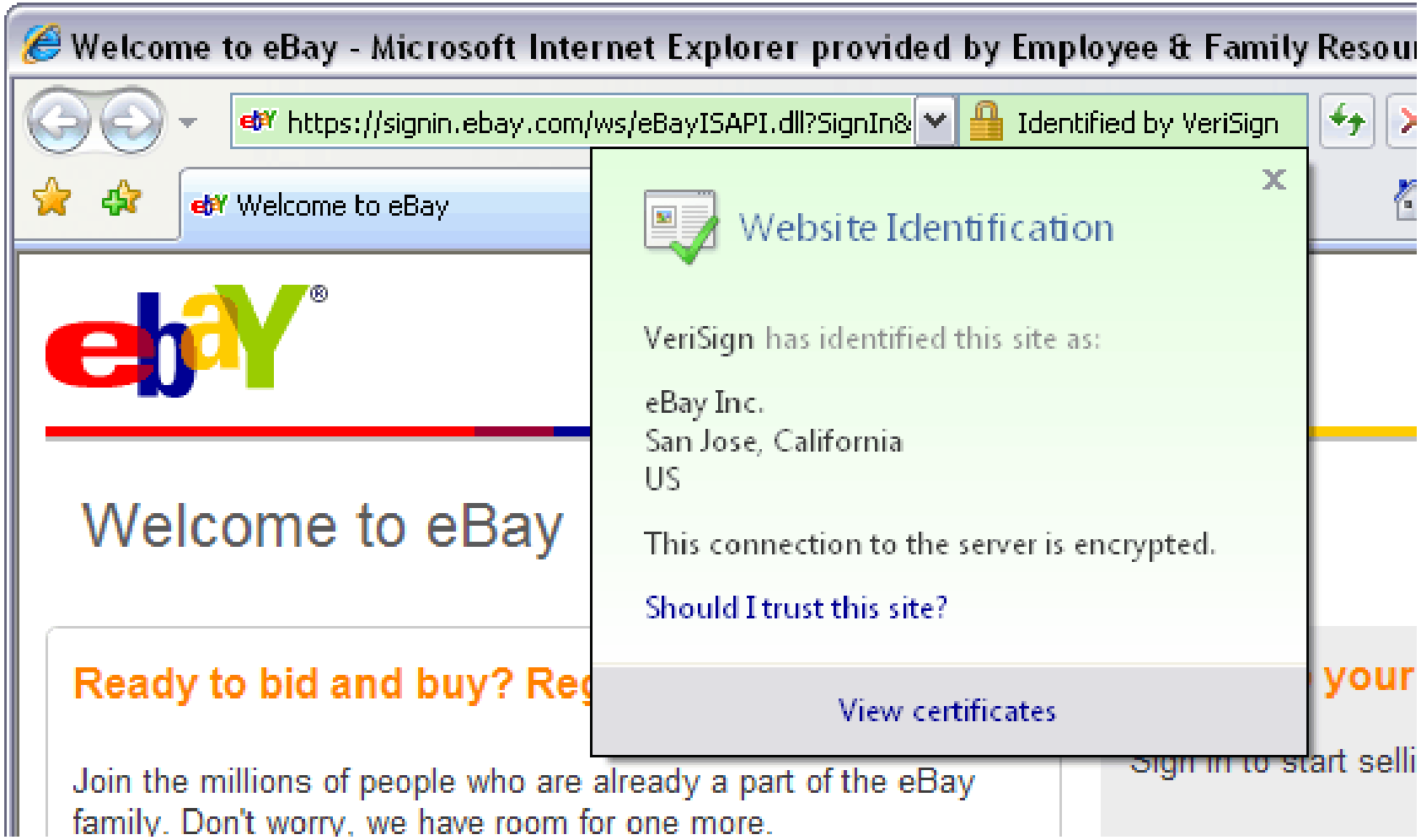
- New type of X.509 SSL certificates
  - Identification using the certificate policy extension
- Requires companies to go through a more thorough and complete company validation process in order to establish the legal identity that controls a web site

  http://cabforum.org/EV_Certificate_Guidelines_V11.pdf
- Browsers with EV support display more information for EV certificates than for previous SSL certificates.

Extended Validation SSL (EV-SSL)



Extended Validation SSL (EV-SSL)



Extended Validation SSL (EV-SSL)

- https://www.phish-no-phish.com/