Unit 4 **Electronic Mail Security, Web Security, Intrusion, Firewalls, Biometric security**

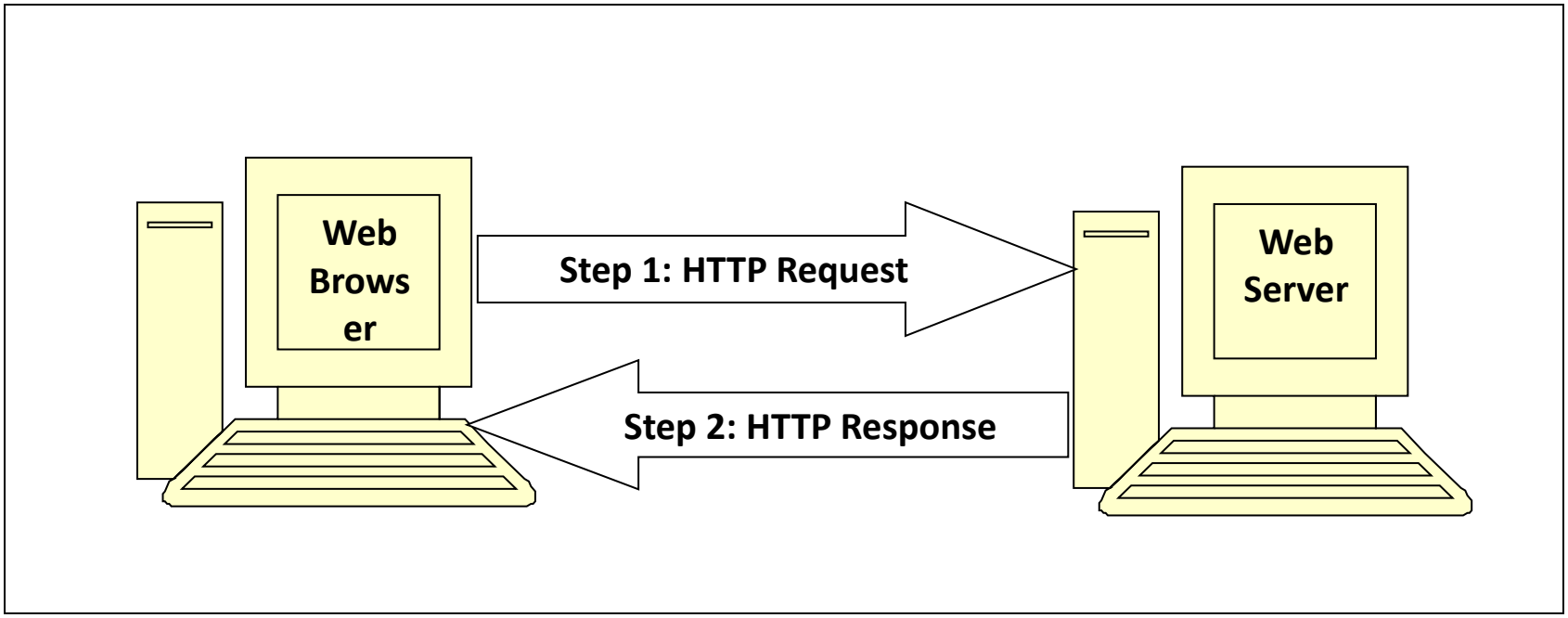
Electronic Mail Security: Pretty Good Privacy, S/MIME, DomainKeys Identified Mail.
IP Security: Overview, Architecture, Authentication Header, Encapsulating Security Payload, Combining Security Associations, Key Management
Web Security: Web Security Considerations, Secure Socket Layer and
Transport Layer Security, HTTPS standard , Secure Socket Shell
Intrusion: Intruders, Intrusion Techniques, Intrusion Detection,
Firewalls: Firewall Design Principles, Types of Firewalls
Security in Online transactions

IP Security

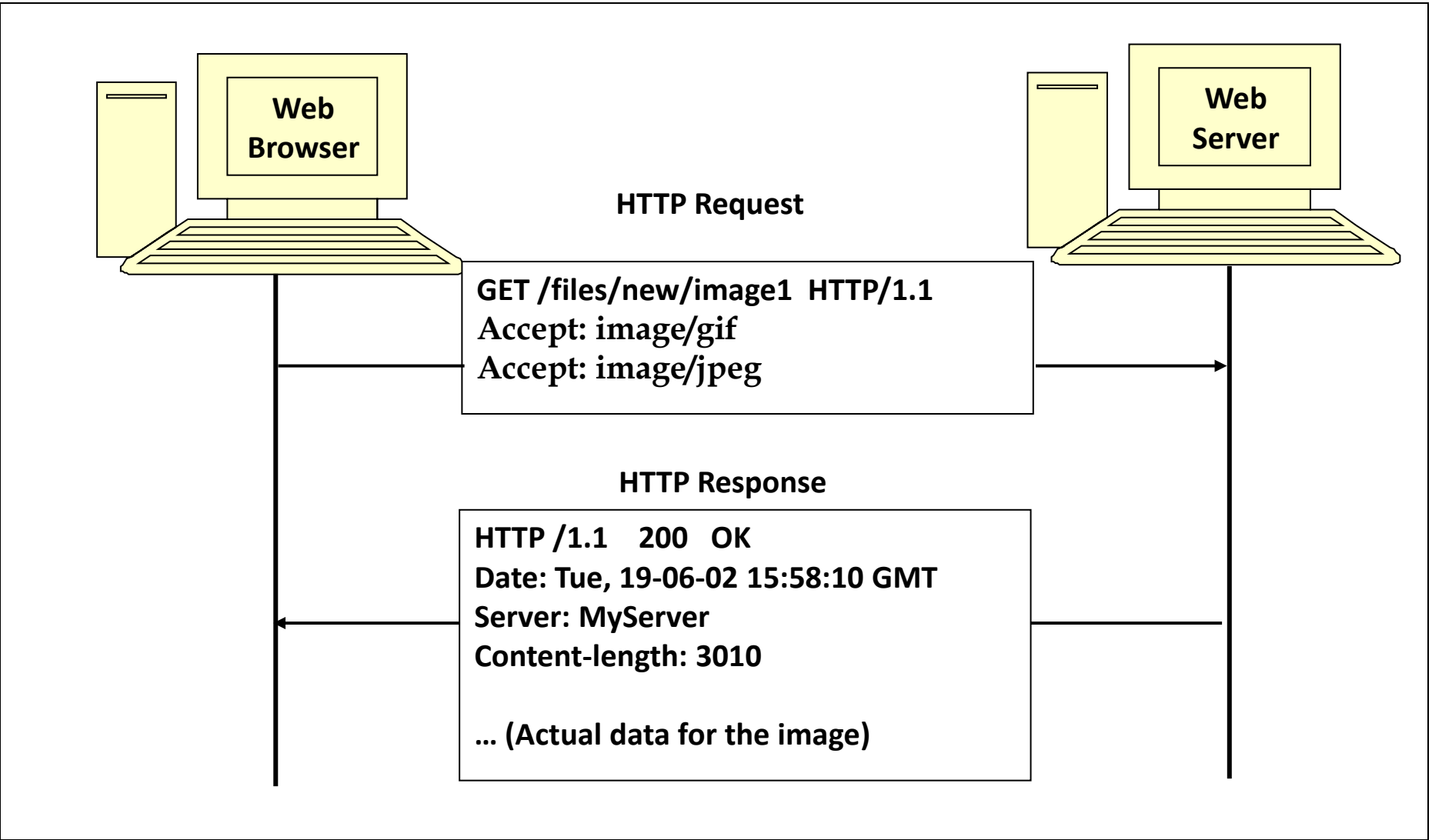
HTTP Protocol

- Hyper Text Transfer Protocol
- Used on the Internet
- Based on Request-Response Model

Static Web Page



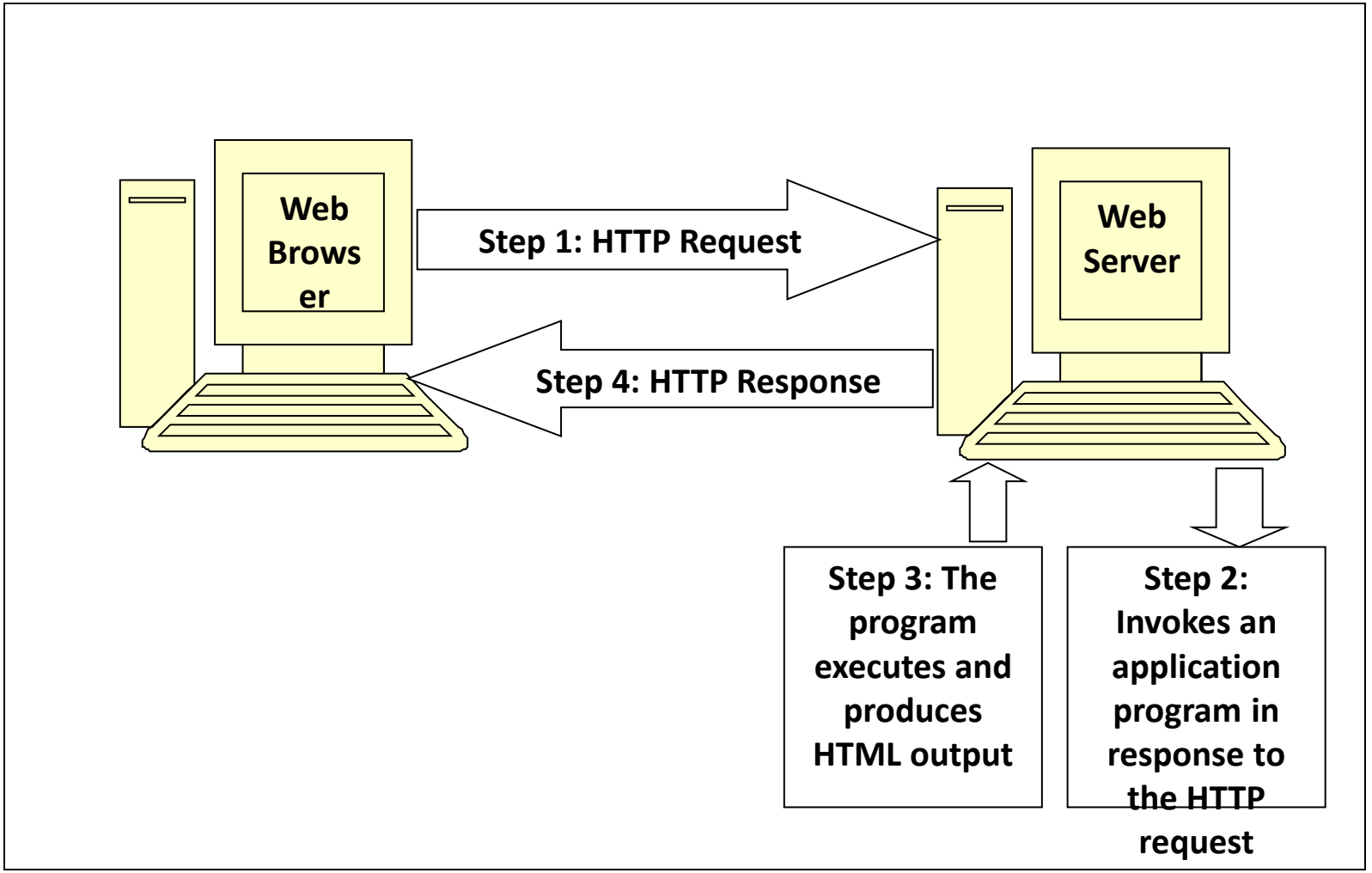
Sample HTTP Interaction



Dynamic Web Page

- Client sends HTTP Request
- Server executes a program
- Server sends back an HTTP Response

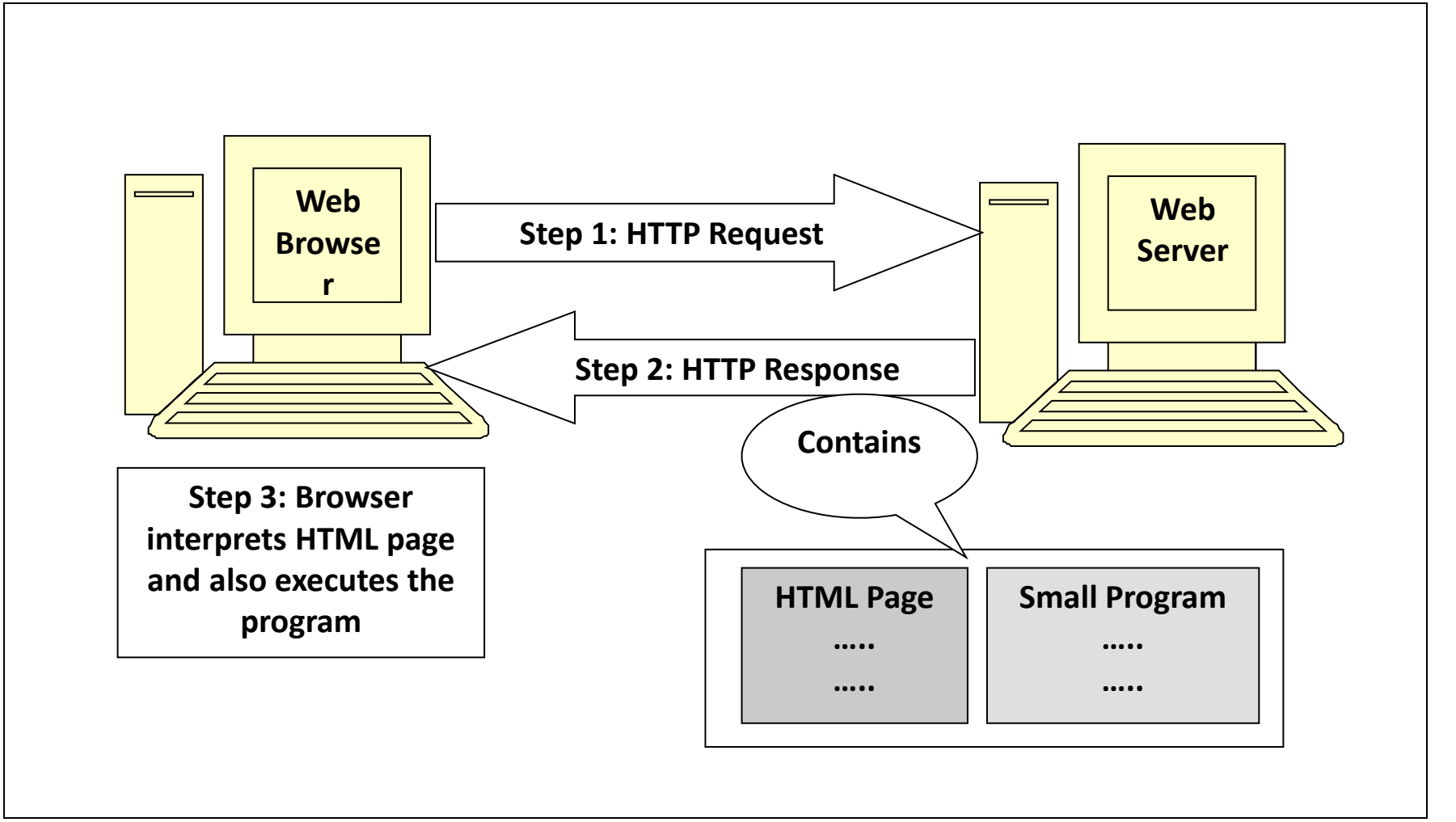
Dynamic Web Page



Active Web Page

- Client sends HTTP Request
- Server sends back HTML Page and a Client-side Program
- Examples: Applet, ActiveX Control

Active Web Page



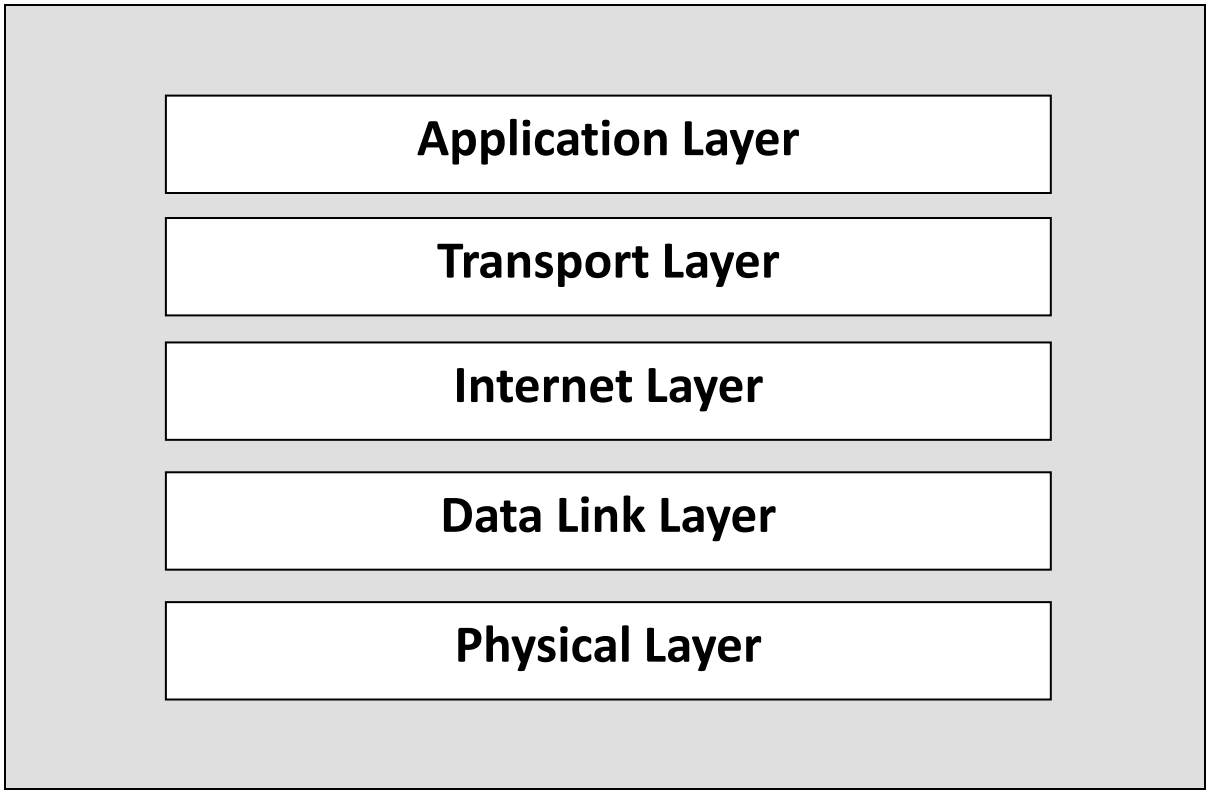
TCP/IP

- Transmission Control Protocol/Internet Protocol
- Convention for communication on the Internet
- Consists of five layers of software

TCP/IP Layers

Layer Number	Layer Name
5 (Highest)	Application
4	Transport
3	Internet
2	Data link
1 (Lowest)	Physical

TCP/IP Layers - Pictorially

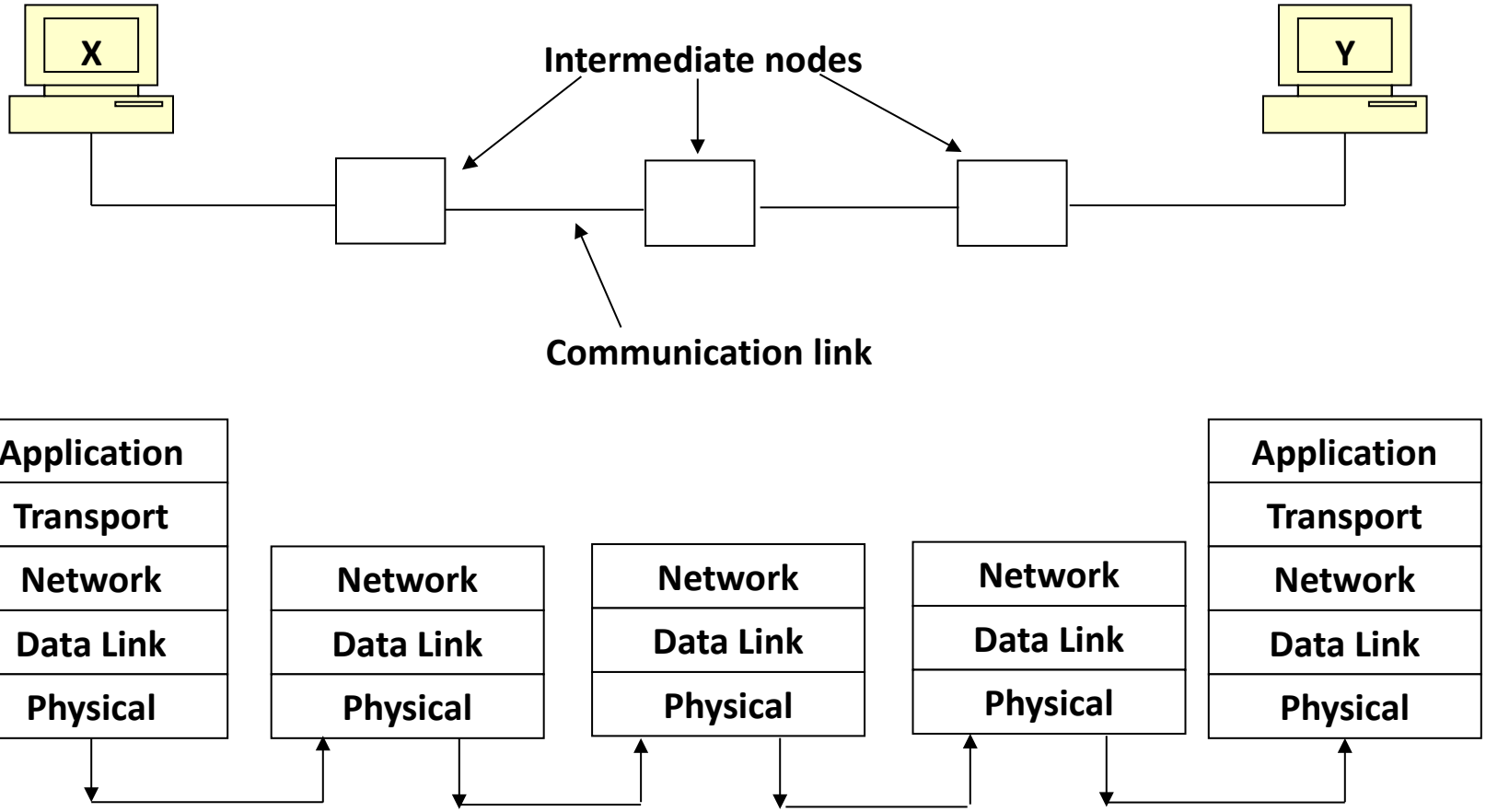


TCP/IP Concept

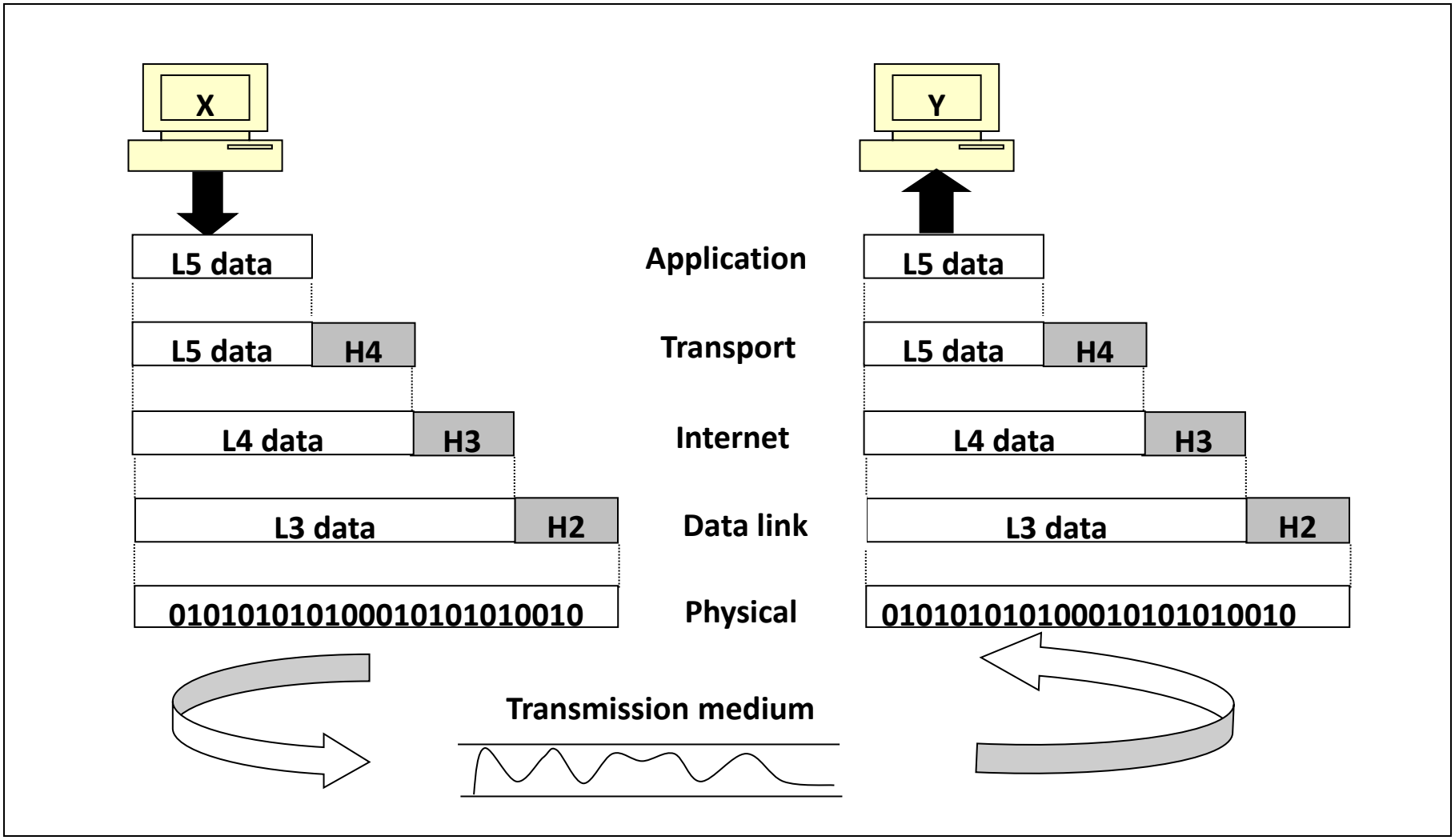
- All layers except physical layer communicate with adjacent layers on the same computer

- Physical layer is the only layer where actual transmission between two computers happens

TCP/IP Communication



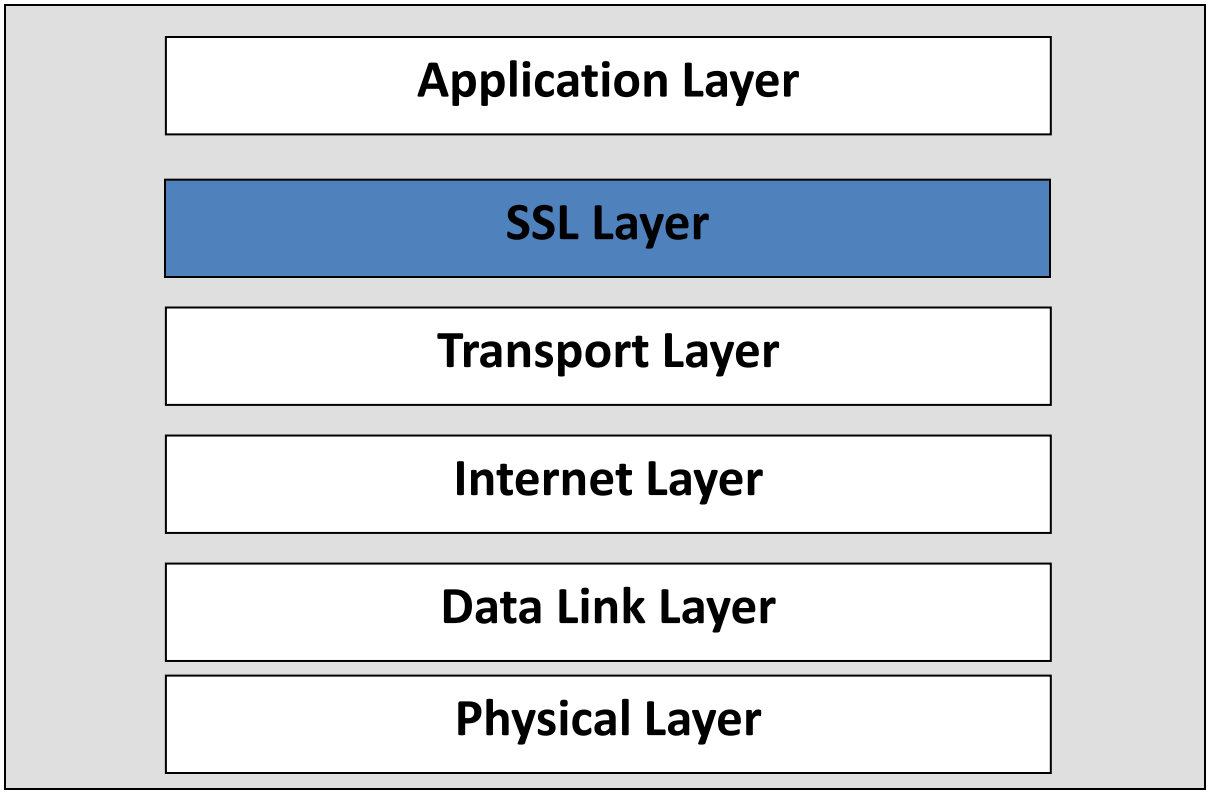
Data Exchange using TCP/IP Layers

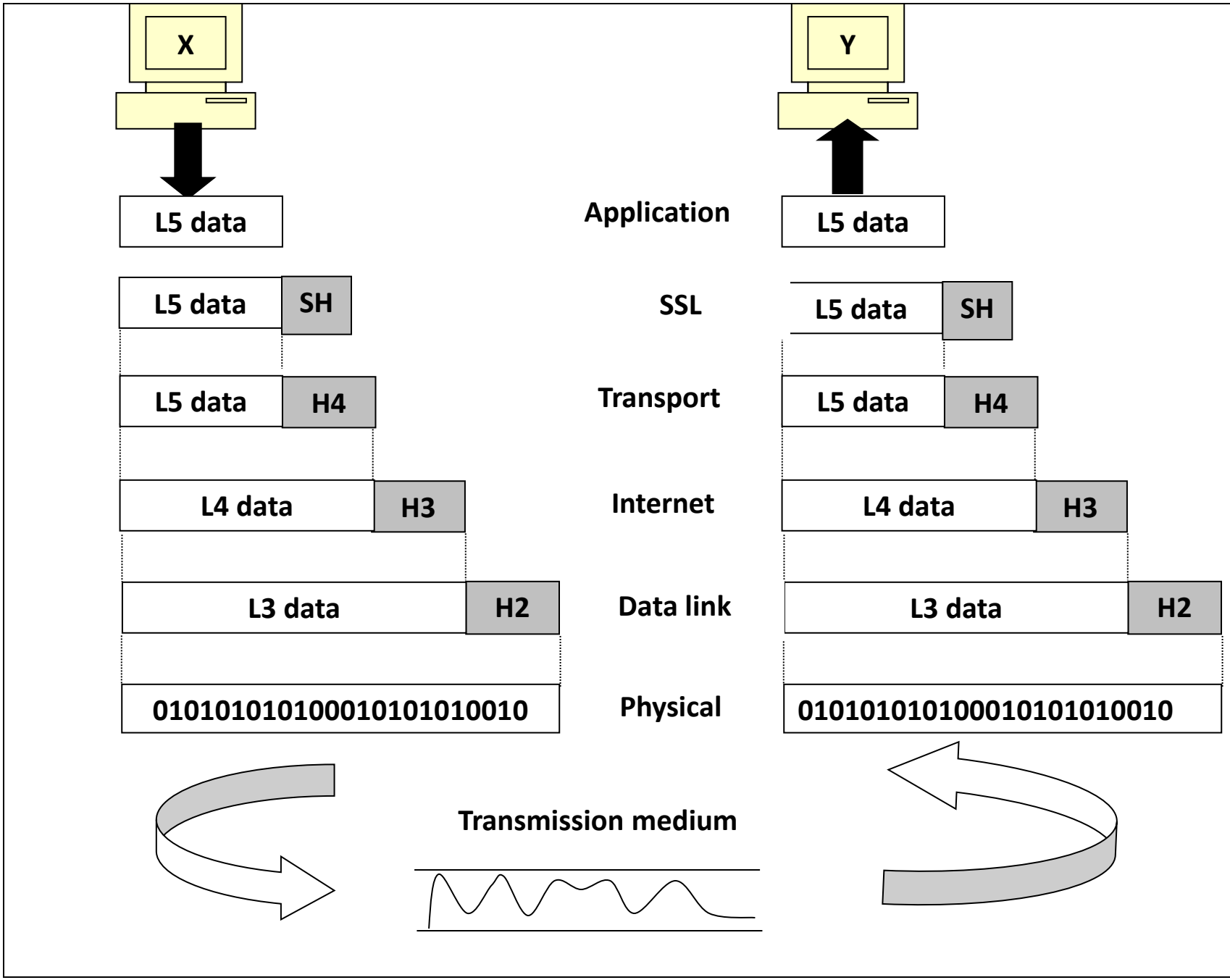


Secure Socket Layer

- Internet Protocol for secure exchange of information between a Web browser and a Web Server
- Provides authentication and confidentiality
- Provides a secure pipe between the Web browser and the Web Server
- Comes in 3 versions : 2,3 and 3.1

Position of SSL in the TCP/IP

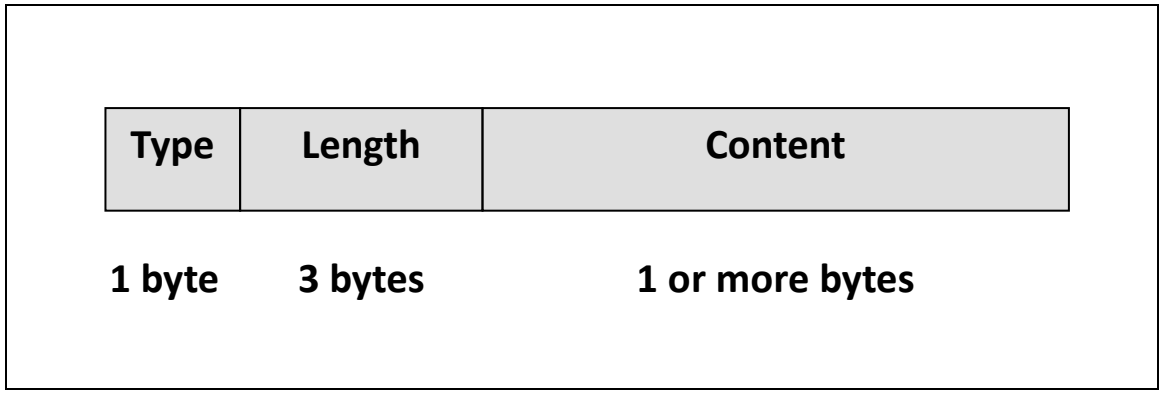



X
Y
L5 data
Application
L5 data
L5 data
SH
SSL
L5 data
SH
L5 data
H4
Transport
L5 data
H4
L4 data
H3
Internet
L4 data
H3
L3 data
H2
Data link
L3 data
H2
0101010101000101010100 10
Physical
0101010101000101010100 10
Transmission medium

SSL Working/SSL Sub-Protocols

- Handshake Protocol
- Record Protocol
- Alert Protocol

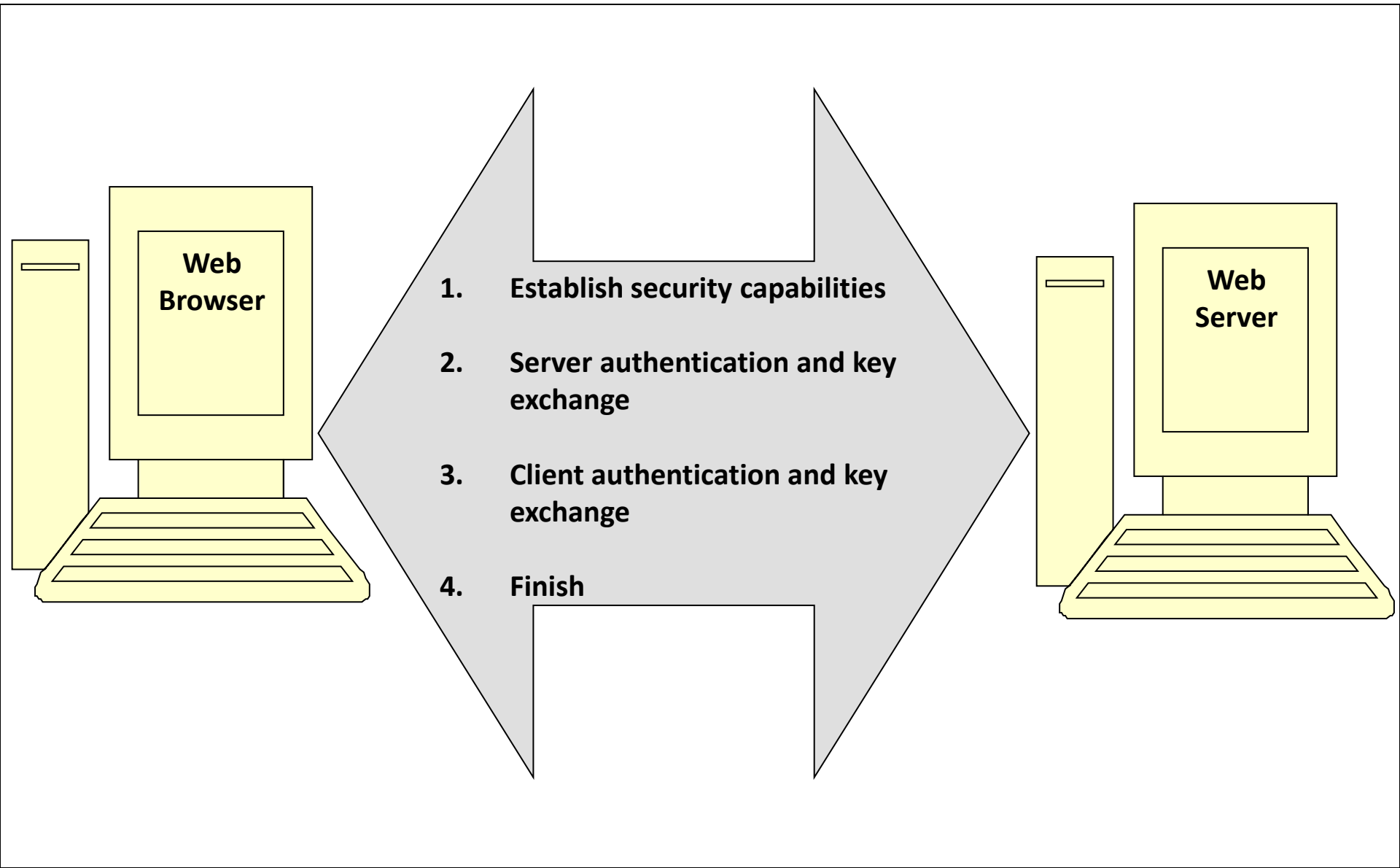
SSL Handshake Message Format



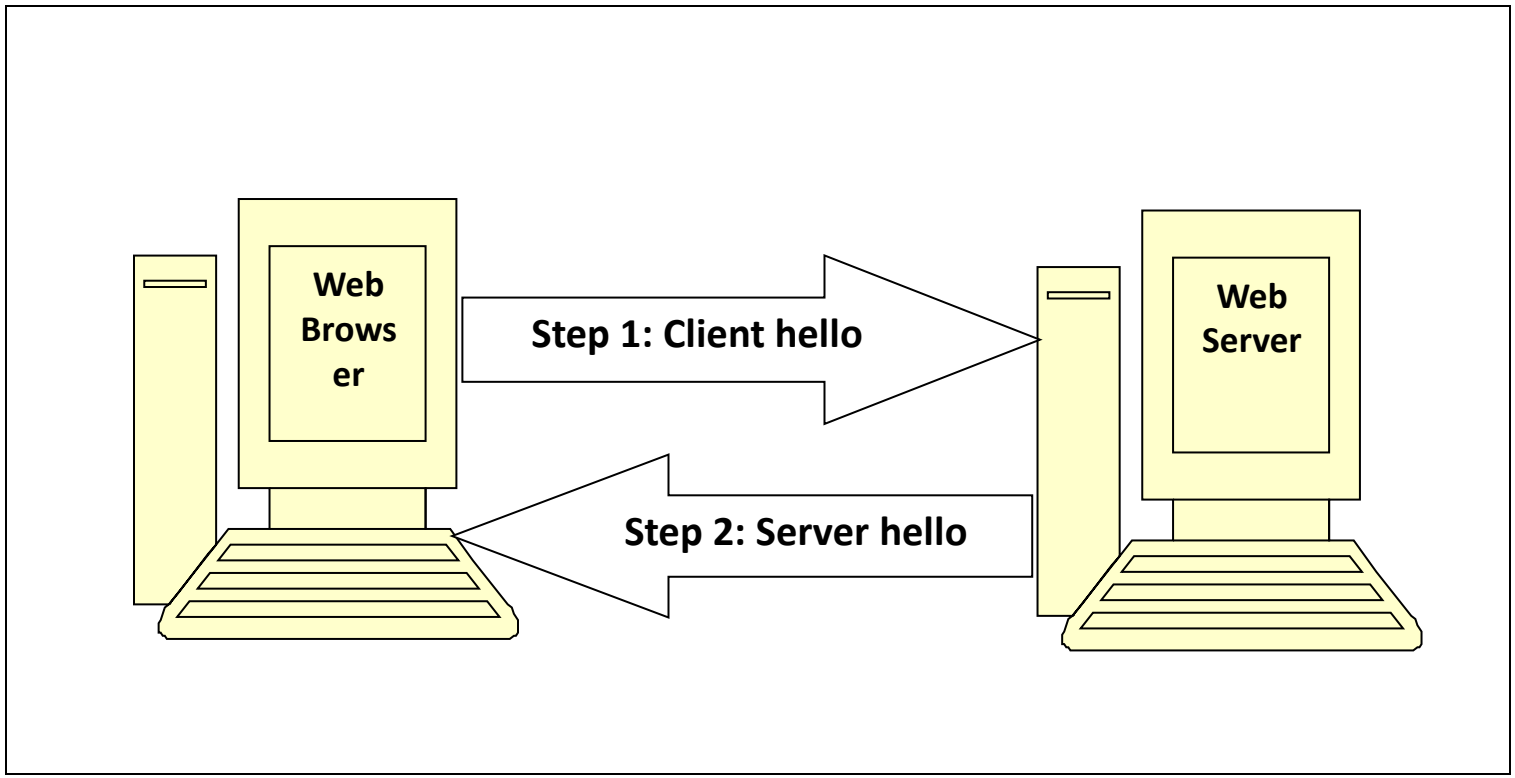
SSL Handshake Messages

Message Type	Parameters
Hello request	None
Client hello	Version, Random number, Session id, Cipher suite, Compression method
Server hello	Version, Random number, Session id, Cipher suite, Compression method
Certificate	Chain of X.509V3 certificates
Server key exchange	Parameters, signature
Certificate request	Type, authorities
Server hello done	None
Certificate verify	Signature
Client key exchange	Parameters, signature
Finished	Hash value

SSL Handshake Process



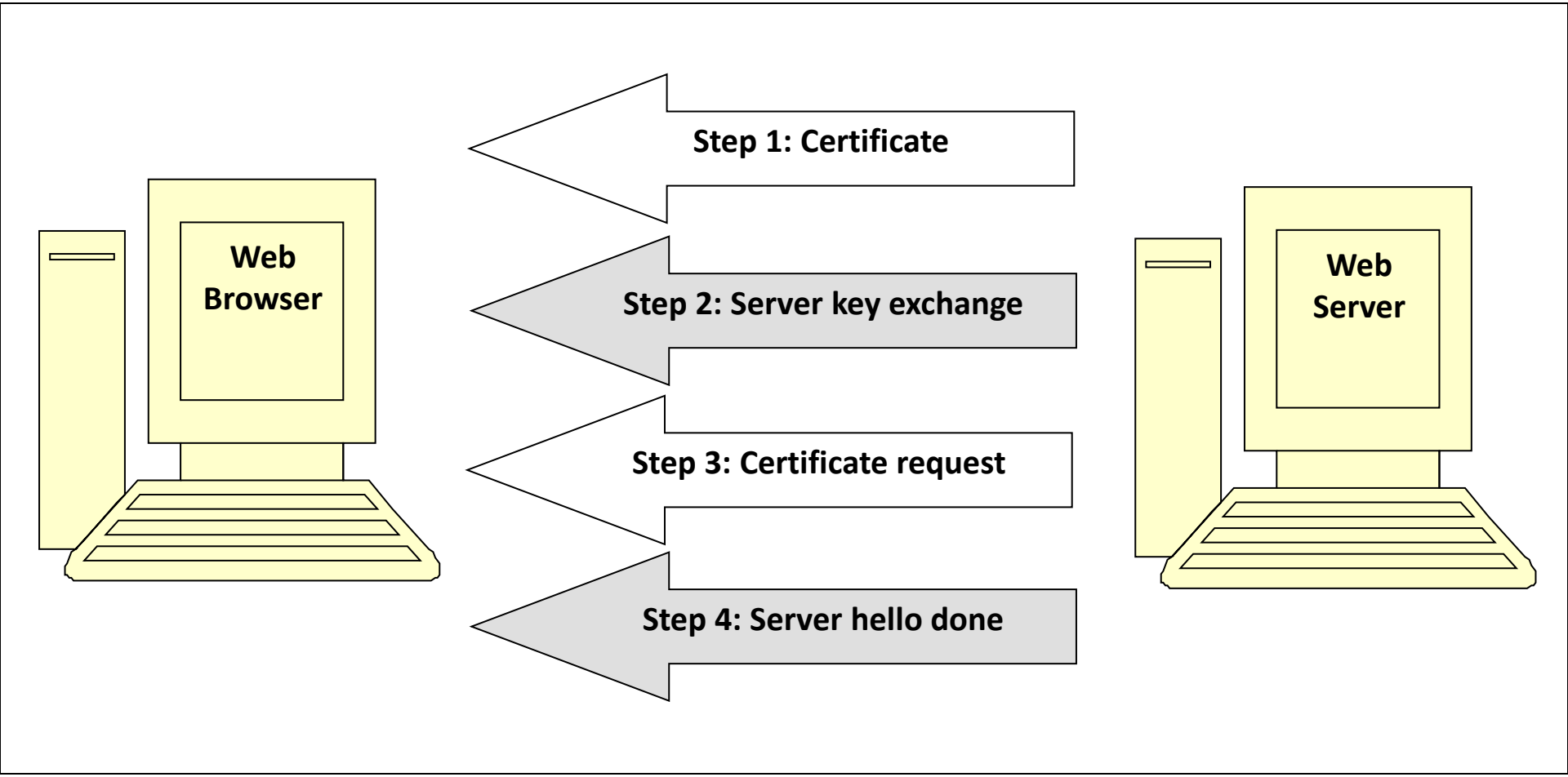
SSL Handshake – Phase 1



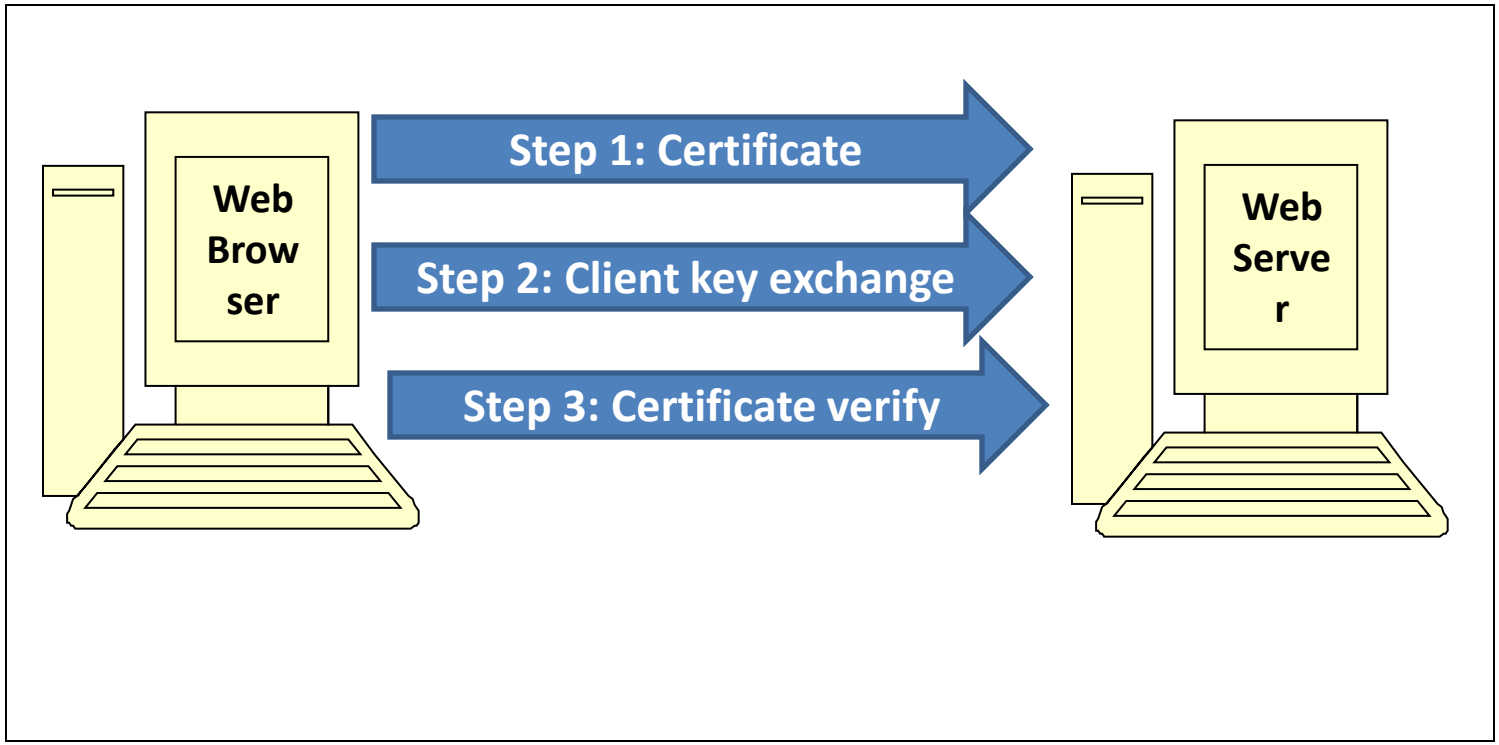
- Hello Client
  - Version
    - Highest version of SSL supported
  - Random
    - 32-bit time identifies the current system date and time on client
    - 28 byte random number generated by the random number generator software built inside the client computer
  - Session id
    - Variable length session identifier
    - If non-zero there is already a session between client and the server
    - Zero value indicates client wants to create a new connection with the server
  - Cipher Suite – list of cryptographic algorithms
  - Compression method - list of compression algorithms

- Server Hello
  - Version
    - lower version among the one client suggested of SSL supported and highest supported by server
  - Random
    - 32-bit time identifies the current system date and time on client
    - 28 byte random number generated by the random number generator software built inside the client computer
    - Independent of the client
  - Session id
    - If non-zero the server uses the same
    - Else creates a new session id
  - Cipher Suite – single which the server selects form the list of cryptographic algorithms
  - Compression method - single which the server selects form the list of compression algorithms

SSL Handshake – Phase 2
Server Authentication and key Exchange

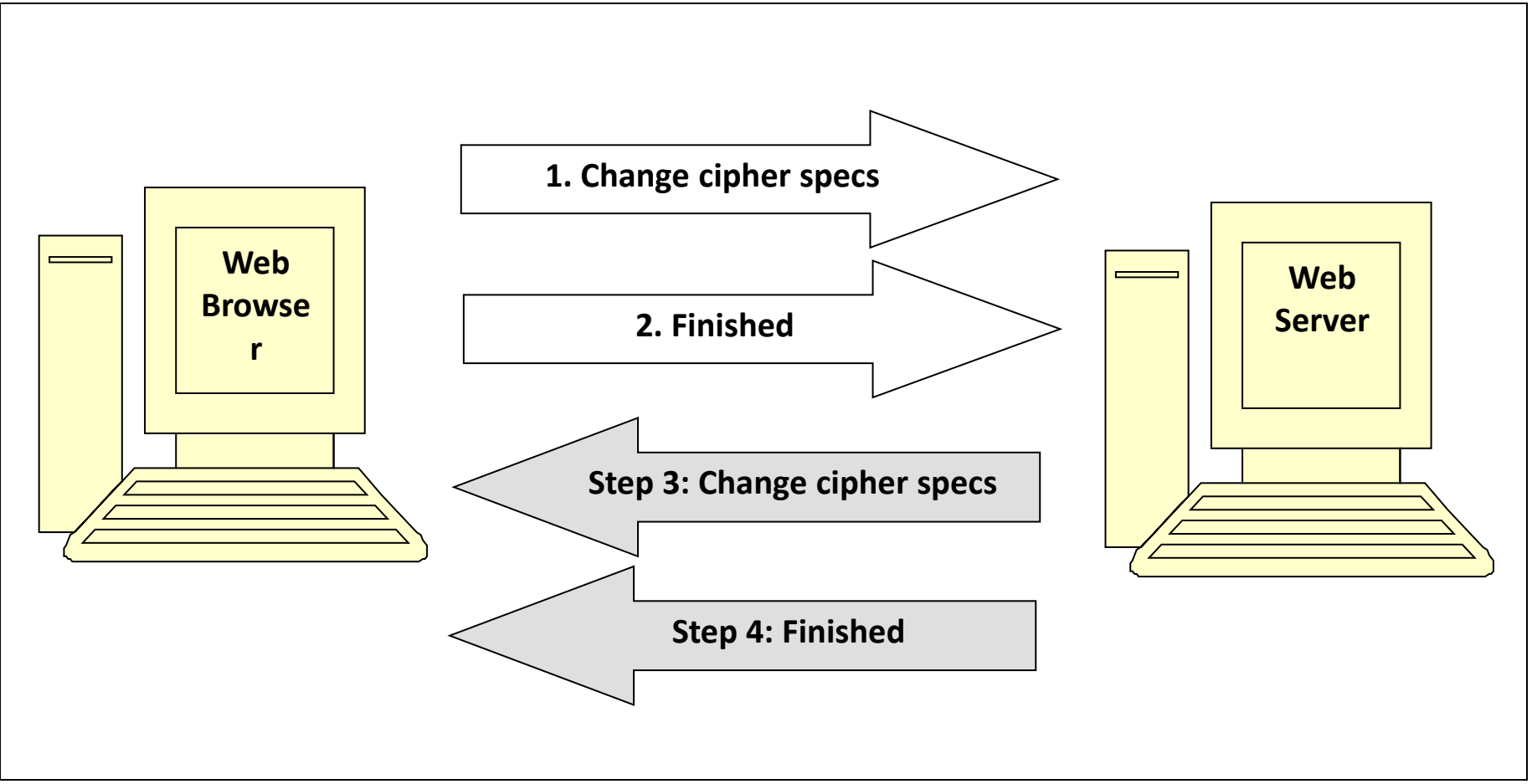


SSL Handshake – Phase 3- Client Authentication and Key Exchange



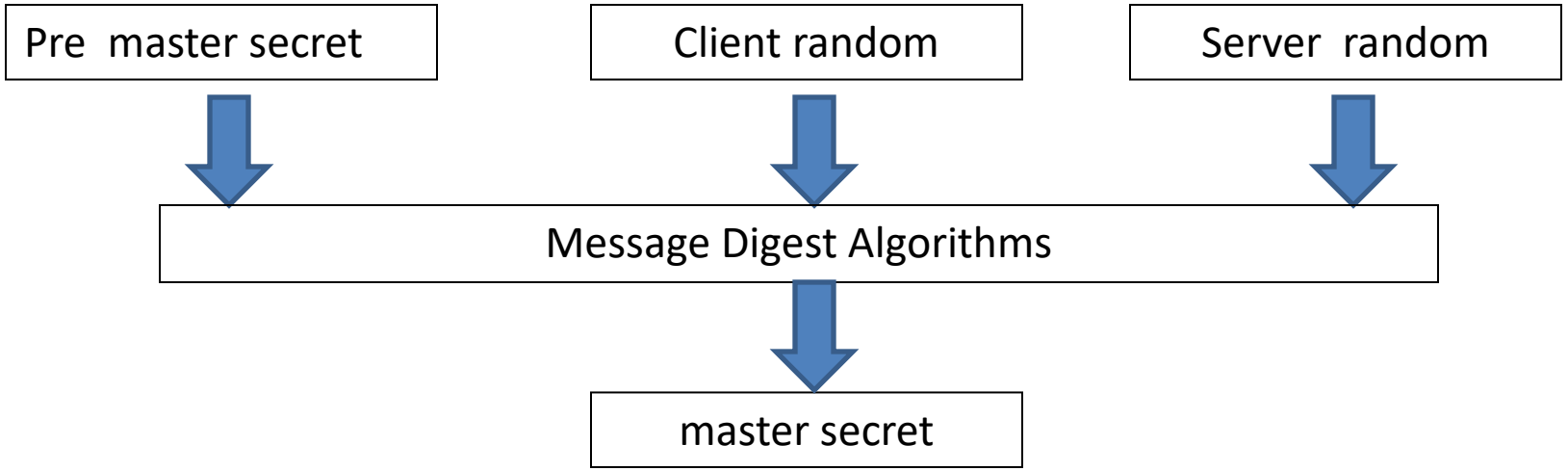
- Certificate – optional
  - Only if server requests for the clients digital certificate
  - If client does not have one client sends a *no certificate* message
- Server key exchange
  - Allows client to send info to the server
  - Related to the symmetric key to be used in the session
  - Client creates a 48-byte pre-master secret and encrypts it with the server's public key and sends this to server
- Certificate Verify
  - If server demanded client authentication
  - Client combines the premaster key with the random numbers exchanged by the client and the server earlier hashes and signs with its private key

SSL Handshake – Phase 4 - Finish

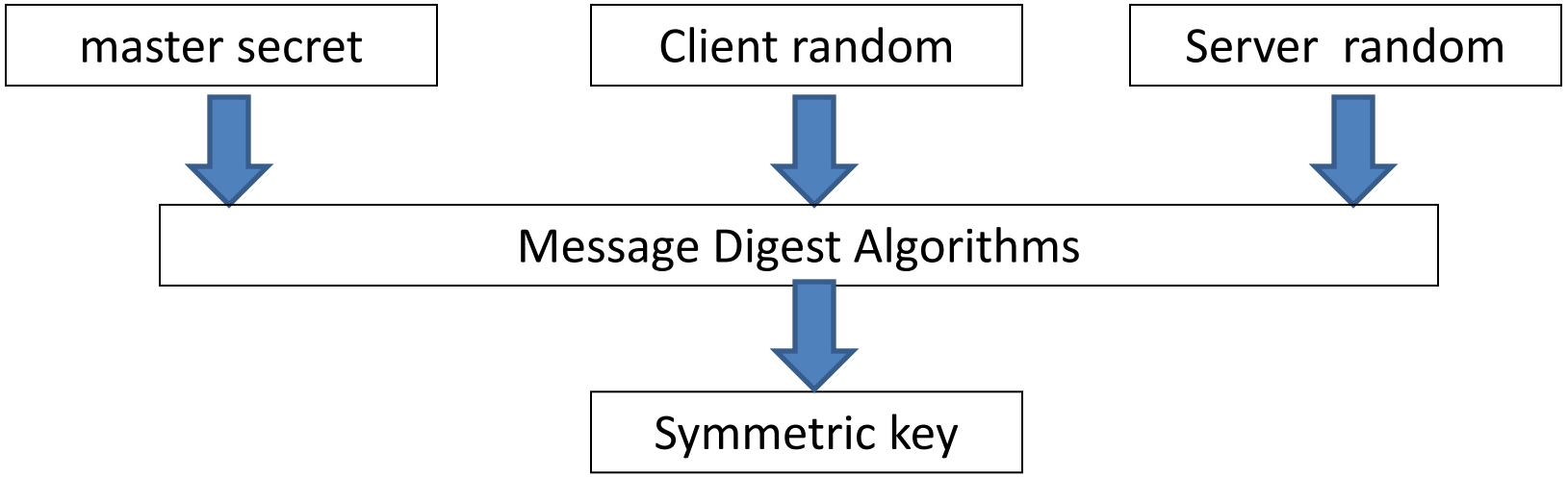


Master Key

- Client and the server create a master secret
  - based on the pre-master key
  - This is a 48-byte quantity
  - Used to generate keys and secrets for encryption and MAC computations
  - Calculated and Computed MD of pre-master secret, client random and server random



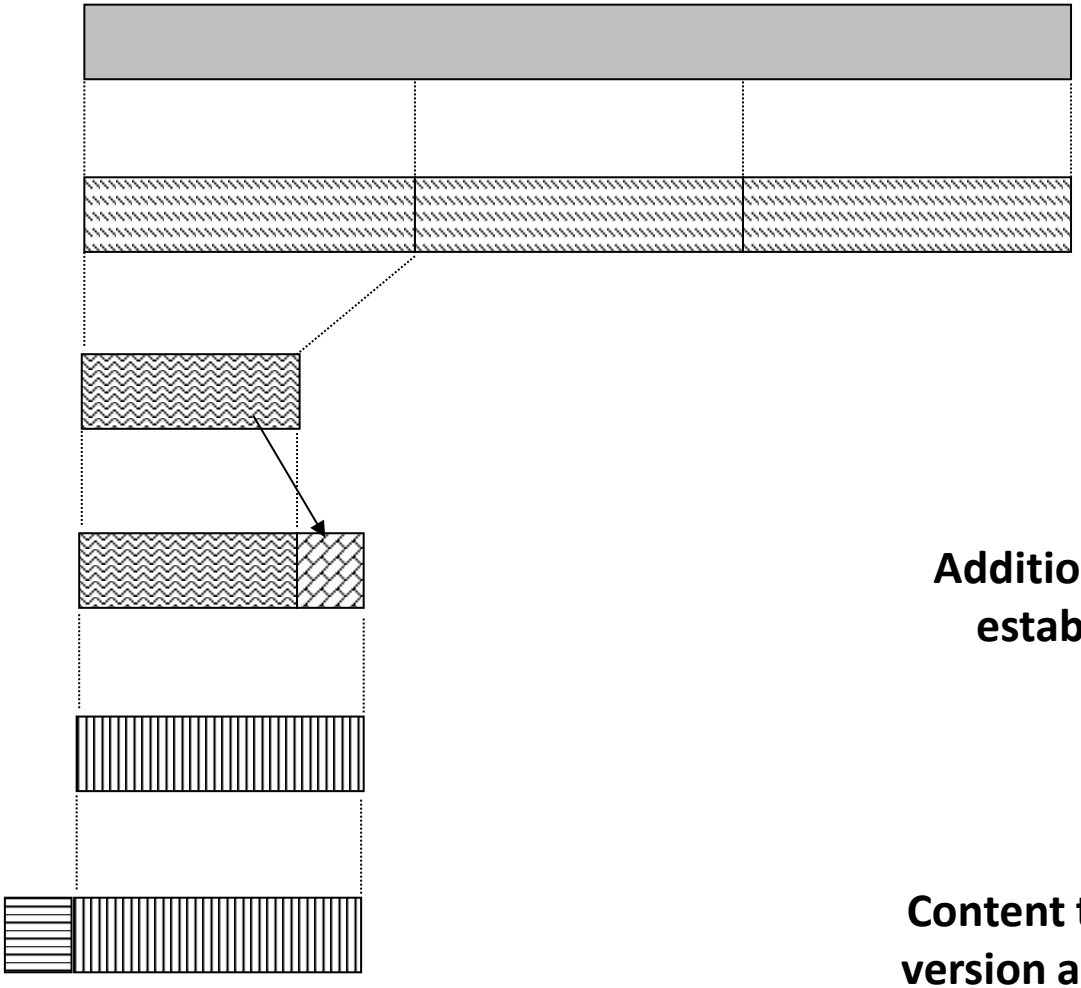
Symmetric Key



Record Protocol

- After a successful handshake
- Authenticated each other and decided on what algorithms to use
- Two services are provided
  - Confidentiality – using secret key
  - Integrity – MAC is defined that is used for assuring the message integrity

SSL Record Protocol



Application data

Fragmentation <= 16kb

Compression (lossless)

Addition of MAC(using secret key established) similar to HMAC

Encryption

Append header
Content type, major version, minor version and compressed length(3.1)
Major – 3 minor -1

Permitted Encryption Algorithms

- Stream Cipher
  - RC4
- Block Cipher
  - AES
  - IDEARC2
  - DES
  - DES-3

Alert Protocol

- Client or server detects an error
- Detecting party sends a alert message
- If error is fatal both close the SSL connection
- Destroy all the keys associated with the connection
- Message consists of two bytes
- Errors are fatal and nonfatal

Fatal/Non - Fatal Errors

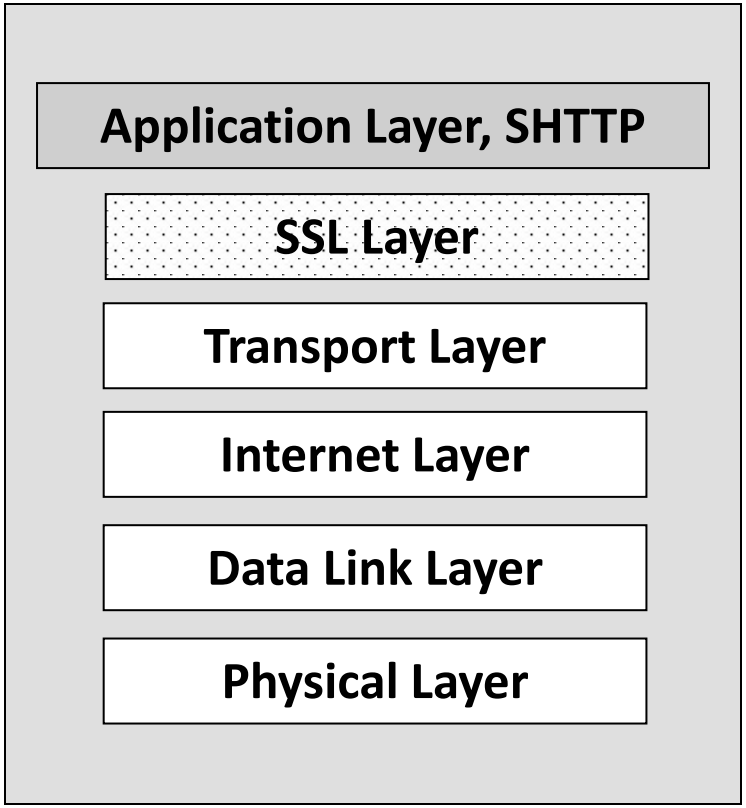
Fatal Errors	Non Fatal Errors
Unexpected message	No certificate
Bad record	Bad Certificate
Decompression Failure	Unsupported Certificate
Handshake failure	Certificate Revoked/Expired/Unknown
Illegal Parameters	Close Notify

Closing and resuming SSL connections

- Before ending both client and server must inform each other that their side is ending
- Each sends CLOSE NOTIFY – graceful closure
- Such connection can be resumed later
- If closed with out a CLOSE NOTIFY the SSL connection cannot be resumed

Secure Hypertext Transfer Protocol(SHTTP)

- Supports authentication and encryption of HTTP traffic
- Works at individual message level
- Encrypts and signs individual messages
- SSl does not differentiate between messages
- Aims at making the connection secure regardless of the messages



Transport Layer Security (TLS)

- Goal is to come out with an Internet Standard version of SSL
- Netscape wanted to standardize SSL

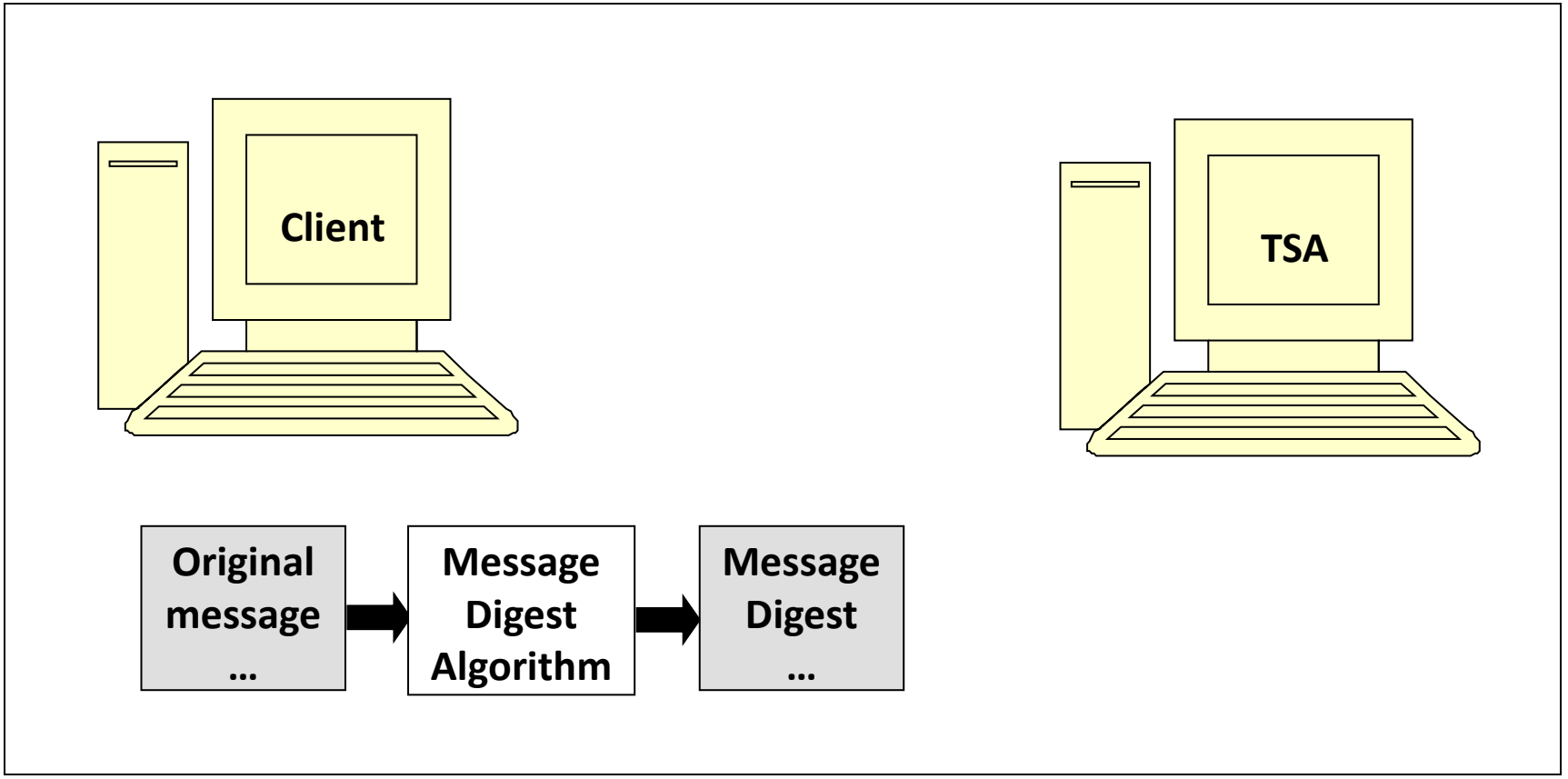
Differences between SSL and TLS

Property	SSL	TLS
Version	3.0	1.0
Cipher Suite	Support Fortezza	No support
Cryptography Secret	Computed	Uses a pseudorandom function to create a master secret
Alert Protocol	Explained	No Certificate deleted some more addedeg. Un known CA protocol version
Handshake protocol		Details are changed
Record Protocol	Uses MAC	Uses HMAC

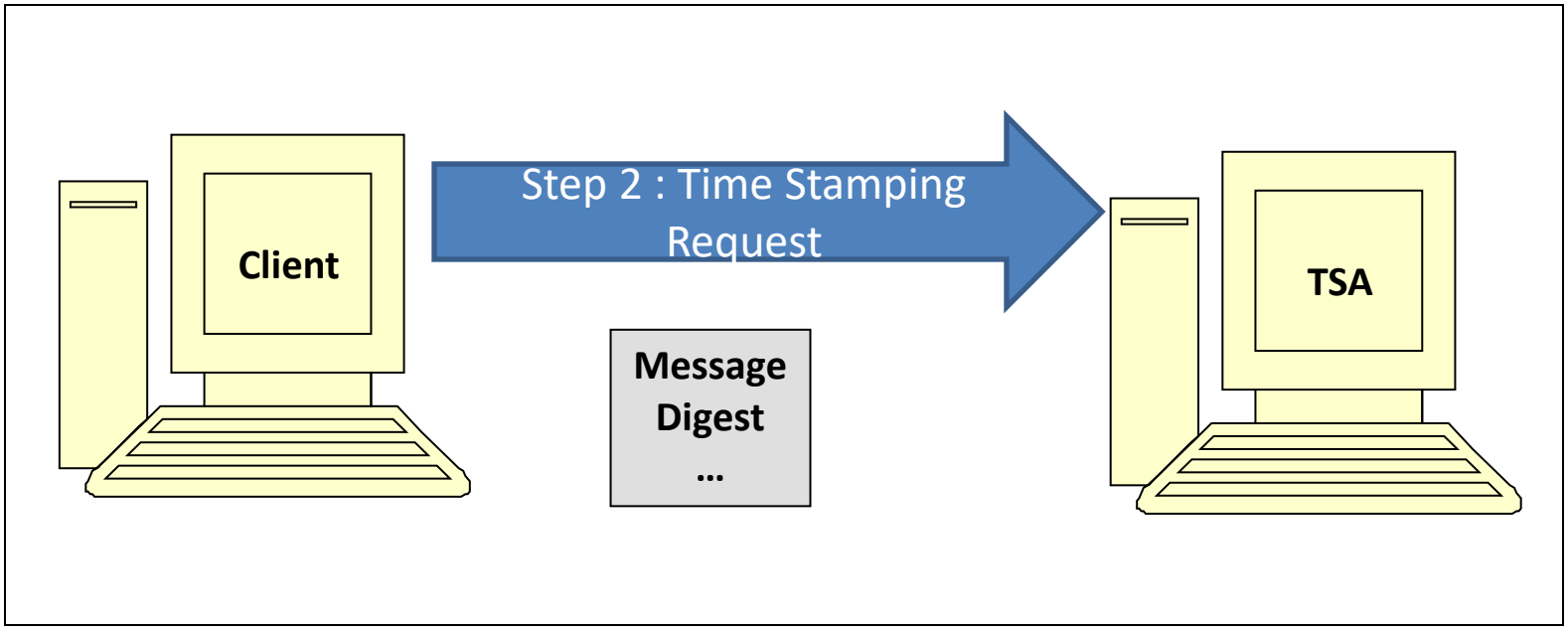
Time Stamping Protocol (TSP)

- Digital version of a notary service
- Prove that a document existed at a specific date and time
- Time Stamping Authority (TSA) is used

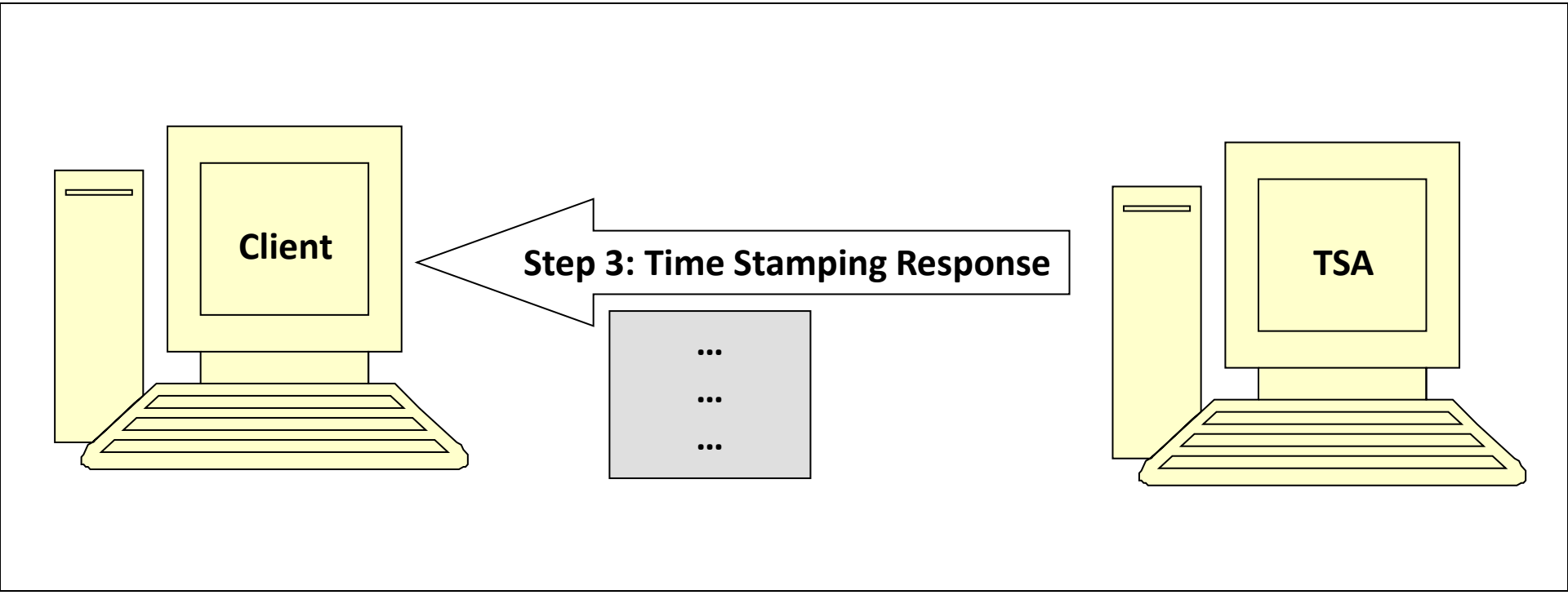
Time Stamping Protocol – Step 1
Message Digest Calculation



Time Stamping Protocol – Step 2 – Time Stamping Request



Time Stamping Protocol – Step 3 - Time Stamping Response

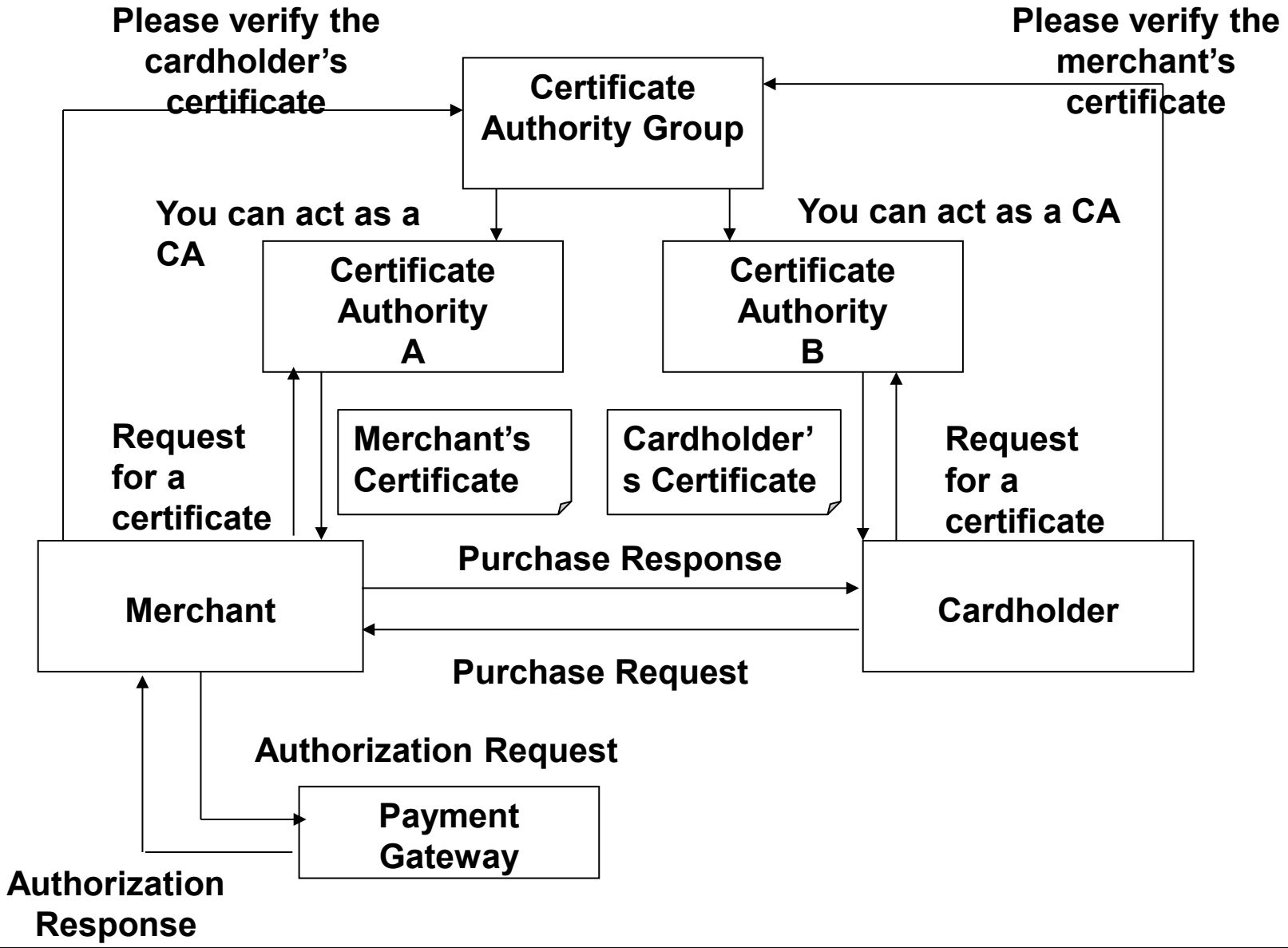


Secure Electronic Transaction (SET)

- Open encryption and security specifications
- Used for securing credit card payments on the Internet
- Merchant does not get to know the credit card details of the cardholder
- Requires software set up on the client as well as server

- SET is not a payment system
- Set of security protocol and formats
- Enable the users to employ the existing credit card payment structure on the internet
- SET services
  - Provides a secure communication channel among all the parties involved in a e-commerce transaction
  - Provides authentication b the use of digital certificates
  - Ensures confidentiality , because the information is only available to the parties involved in the transaction and only when necessary

SET Model


Please verify the cardholder's certificate
Certificate Authority Group
Please verify the merchant's certificate
You can act as a CA
Certificate Authority A
You can act as a CA
Certificate Authority B
Request for a certificate
Merchant's Certificate
Cardholder's Certificate
Request for a certificate
Purchase Response
Merchant
Cardholder
Purchase Request
Authorization Request
Payment Gateway
Authorization Response

SSL versus SET

Issue	SSL	SET
Main aim	Exchange of data in an encrypted form	E-commerce related payment mechanism
Certification	Two parties exchange certificates	All the involved parties must be certified by a trusted third party
Authentication	Mechanisms in place, but not very strong	Strong mechanisms for authenticating all the parties involved
Risk of merchant fraud	Possible, since customer gives financial data to merchant	Unlikely, since customer gives financial data to payment gateway
Risk of customer fraud	Possible, no mechanisms exist if a customer refuses to pay later	Customer has to digitally sign payment instructions
Action in case of customer fraud	Merchant is liable	Payment gateway is liable
Practical usage	High	Low at the moment, expected to grow