HTTP (HyperText Transfer Protocol)

Introduction

- บริการเอกสารบนเว็บไซต์
- TCP protocol port 80
- Client-Server Model
  - HTTP Request: client (web browser - Chrome, Brave, Firefox, Opera, Safari)
  - HTTP Response: server (web server - Apache, Nginx, IIS)
- Stateless Protocol - Request/Response แต่ละครั้งไม่เกี่ยวข้องกัน, server ไม่มีการเก็บประวัติการทำงาน
  - ใช้ cookies, session IDs, or URL rewriting ในการเก็บประวัติ เช่น login , shopping cart

เก็บฝั่ง client ↙ ↳ เก็บฝั่ง server

World Wide Web (WWW)

- 1989 - Tim Berners-Lee
  - CERN - European Organization for Nuclear research, นักวิจัยใน Europe แชร์ research
- 1990 - Commercial Web
- Web page - documents
  - distributed - allows the growth of the Web
  - linked - allows one web page to refer to another web page
- Hypertext - refer to another web page stored in another server
- Hypermedia - a web page can be a text document, an image, an audio file, or a video file
- Uniform Resource Locator (URL)

```
protocol://host/path/filename       Used most of the time
protocol://host:port/path/filename  Used when port number is needed
```

↳ หากทำงานบน port อื่น

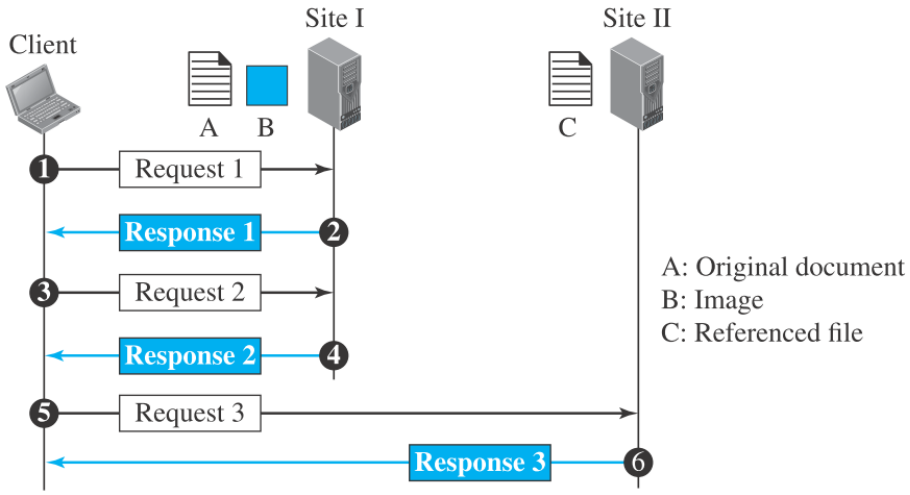
  - default filename - กำหนดโดย web server

```
http://nt-lab.cs.psu.ac.th          http://nt-lab.cs.psu.ac.th/index.html
```

- Web Documents
  - Static documents - fixed-content documents : nithi
  - Dynamic Documents - created by a web server
  - Active Documents - a script run at the client site : lms

How HTTP work

- 1,.2 เปิดเอกสาร A อยู่ที่ Site I
  - 3, 4 อ้างถึง รูปภาพ B อยู่ที่ Site I ส่งแล้วยับเลย ไม่มีการจำ session
  - 5, 6 อ้างถึง เอกสาร C อยู่ที่ Site II

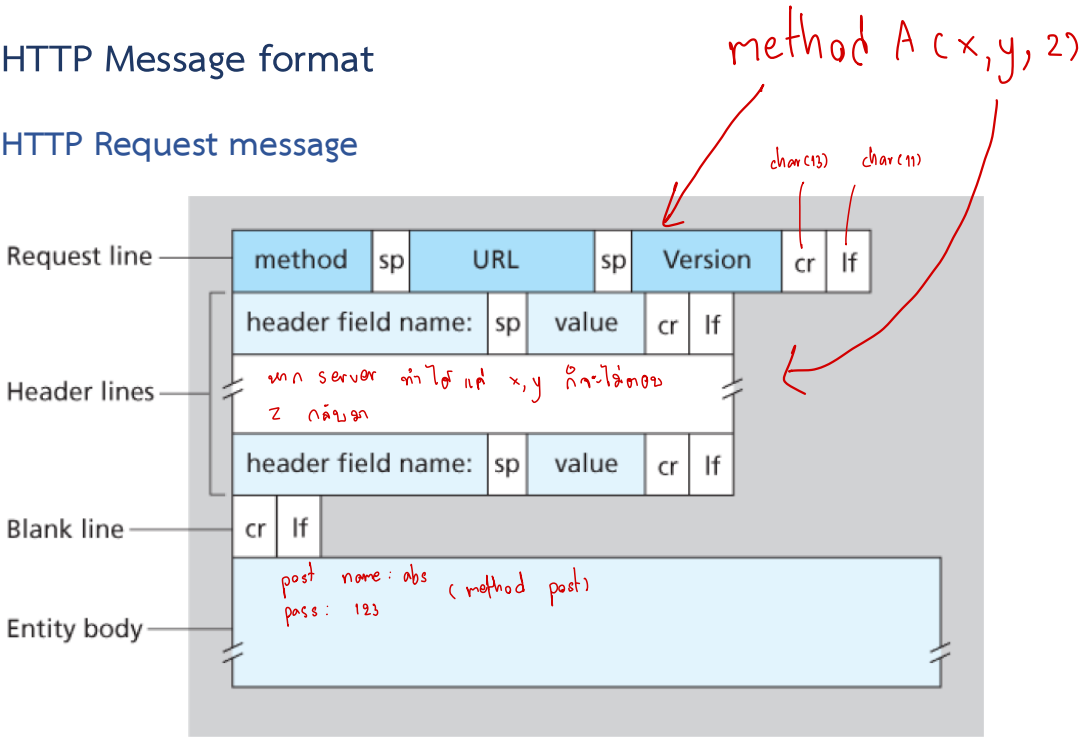


HTTP vs. HTTPS

- HTTP - Data is transmitted in plain text. Vulnerable to eavesdropping and tampering.
- HTTPS (HTTP Secure) - HTTP communication over Transport Layer Security (TLS) or its predecessor, **Secure Sockets Layer (SSL)**. Encrypts the communication, providing data integrity and authentication. Essential for sensitive data (passwords, credit card numbers).

HTTP Message format

HTTP Request message



- GET method

```
GET /index.html HTTP/1.1
Host: www.example.com
User-Agent: Mozilla/5.0
Accept: text/html
Connection: keep-alive
<blank line>
```

- POST method

```
POST /submit-form HTTP/1.1
Host: www.example.com
Content-Type: application/x-www-form-urlencoded
Content-Length: 29
<blank line>
name=John+Doe&age=30
```

- common HTTP methods

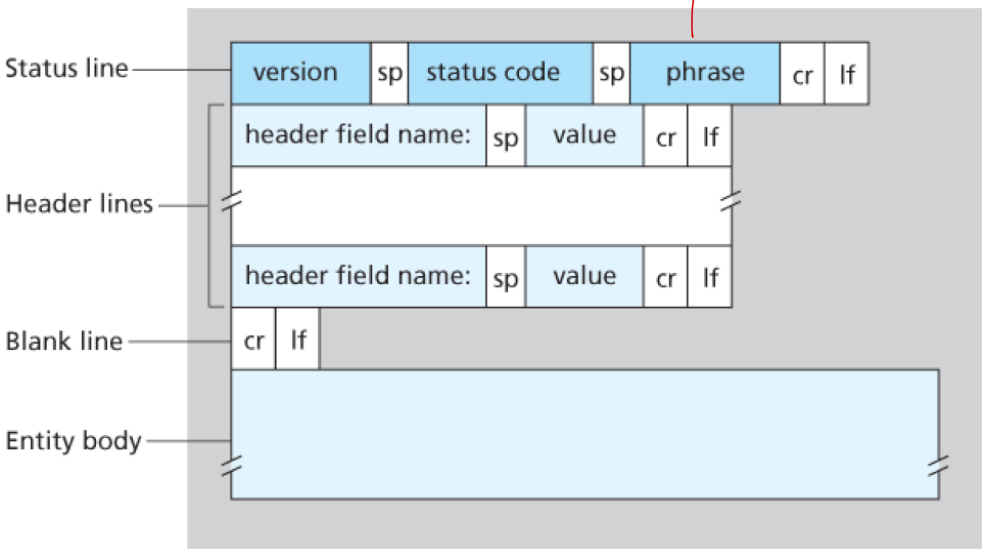
Method	Description
GET	Retrieves data from the server
POST	Submits data to the server
PUT X	Updates data on the server
DELETE X	Deletes data from the server
HEAD ใช้เฉพาะ หน้าที่ทำงานอย่างไร	Like GET but returns only headers
OPTIONS	Describes communication options
PATCH	Partially updates data

- common HTTP request headers

Request Header	Description
Host	ที่อยู่ของ server
User-Agent	รุ่นของ browser
Accept	รองรับเอกสาร
Accept-Language	รองรับภาษา
Accept-Encoding	รองรับการบีบอัดข้อมูล zip ทำงานได้ดีกับ text
Connection	เชื่อมต่อแบบ persistent

HTTP Response message

code → eng



- Status code 200

```
HTTP/1.1 200 OK
Date: Wed, 02 Jul 2025 04:00:00 GMT
Server: Apache/2.4.41 (Ubuntu)
Last-Modified: Mon, 30 Jun 2025 10:00:00 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 123
<blank line>
<!DOCTYPE html>
<html> <head> <title>Example Page</title> </head>
<body> <h1>Welcome!</h1> </body>
</html>
```

วันที่ตอนเรียกเข้าไฟล์

วันที่ของเอกสารตอนบันทึก

html css java script

- common HTTP Status codes

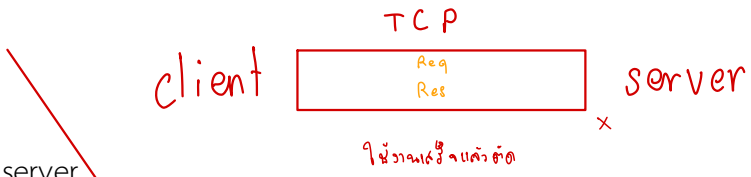
Status Code	Meaning
200	OK
301	Moved Permanently เอกสารโดนย้าย
400	Bad Request คำสั่งผิด
401	Unauthorized ไม่มีสิทธิ์เข้าถึง
403	Forbidden ไฟล์โดนซ่อน
404	Not Found หาไฟล์ไม่เจอ
500	Internal Server Error

304 not modify ไม่มีการเปลี่ยนแปลง ข้อมูลดึงมาจาก cache

- common HTTP Response headers

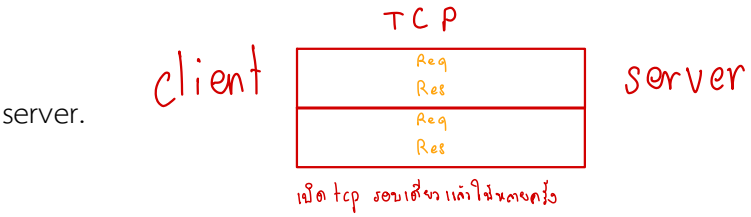
Response Header	Description
Date	วันที่เปิดเว็บไซต์
Server	รุ่นของ server
Upgrade	ให้เปลี่ยน protocol ที่ปลอดภัย
Connection	เชื่อมต่อแบบ persistent
Last-Modified	วันที่แก้ไขเอกสารล่าสุด
Accept-Ranges	สามารถแยก download เอกสารเป็น bytes
Cache-Control	เวลาซึ่งสามารถเก็บเอกสารไว้ใน cache (วินาที)
Expires	วันที่เอกสารหมดอายุ (วันที่อ่าน + cache)
Vary	สามารถใช้คำสั่งเหล่านี้ได้ คำสั่ง ที่ server ยอมรับ
Content-Encoding	การบีบอัดข้อมูล
Content-Length	ขนาดเอกสาร
Keep-Alive	เปิดการเชื่อต่อ persistent (นาที, จำนวนครั้งที่ใช้ได้)
Content-Type	ประเภทของเอกสาร

Persistent Connection (Keep-Alive)



- HTTP/1.0

1. The client opens a TCP connection to the server.
2. The client sends an HTTP request.
3. The server sends an HTTP response.
4. The connection is immediately closed.



- HTTP/1.1 (Keep-Alive)

1. The client opens a TCP connection to the server.
2. The client sends an HTTP request.
3. The server sends an HTTP response.
4. The TCP connection remains open.
5. The client can then send subsequent HTTP requests over the same open connection.
6. The connection is only closed after a certain period of inactivity (a configurable timeout) or if either the client or server explicitly signals to close it.

HTTP version

- **HTTP/0.9**: The initial version, very simple.
- **HTTP/1.0**: Introduced headers, different methods, and status codes.
- **HTTP/1.1**: The most widely used version for a long time. Introduced persistent connections, pipelining, and host headers.
- **HTTP/2**: Designed to improve performance over HTTP/1.1. Introduced multiplexing (sending multiple requests/responses over a single connection), header compression, and server push.
- **HTTP/3**: The latest major version, using QUIC (Quick UDP Internet Connections) instead of TCP as its underlying transport protocol. Aims to further reduce latency and improve performance, especially in unreliable network conditions.

Example Message

- **Firefox** - Request message

```
Hypertext Transfer Protocol
    GET / HTTP/1.1\r\n
    Host: nt-lab.cs.psu.ac.th\r\n
    User-Agent: Firefox/140.0\r\n
    Accept: text/html,application/xhtml+xml,application/xml\r\n
    Accept-Language: en-US,en\r\n
    Accept-Encoding: gzip, deflate\r\n
    Connection: keep-alive\r\n
    \r\n
```

- **Apache** - Response message

```
Hypertext Transfer Protocol
    HTTP/1.1 200 OK\r\n
    Date: Wed, 02 Jul 2025 03:38:05 GMT\r\n
    Server: Apache\r\n
    Upgrade: h2,h2c\r\n
    Connection: Upgrade, Keep-Alive\r\n
    Last-Modified: Mon, 30 Jun 2025 04:23:22 GMT\r\n
    Accept-Ranges: bytes\r\n
    Cache-Control: max-age=3600, public\r\n
    Expires: Wed, 02 Jul 2025 04:38:05 GMT\r\n
    Vary: Accept-Encoding\r\n
    Content-Encoding: gzip\r\n
    Content-Length: 172\r\n
    Keep-Alive: timeout=5, max=100\r\n
    Content-Type: text/html; charset=UTF-8\r\n
    \r\n
    <!DOCTYPE html>\n
    <html lang="en">\n
    ...
```

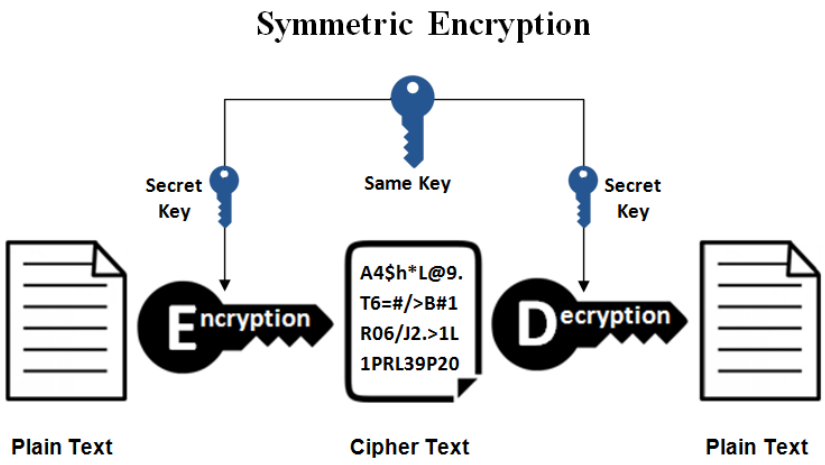
HTTPS (HyperText Transfer Protocol Secure)

Introduction

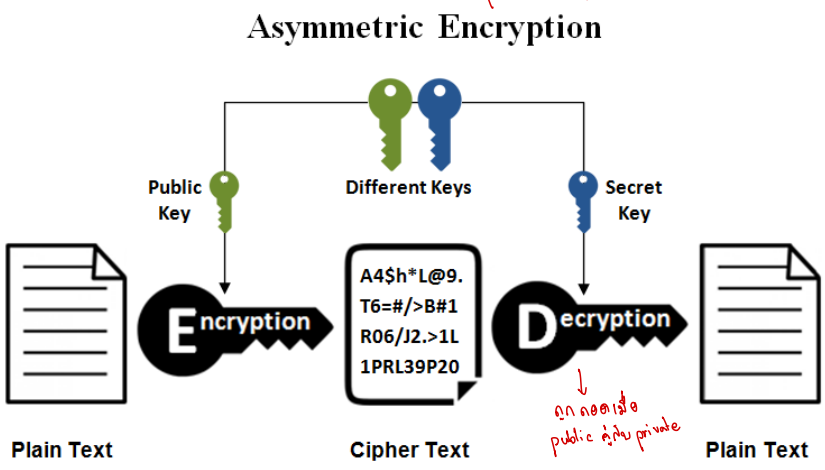
- TCP protocol port 443
- Transport Layer Security (TLS): เข้ารหัสข้อมูลระหว่างส่ง, Secure Sockets Layer (SSL)
- HTTPS: HTTP + TCP + TLS
  - Data Privacy: ป้องกันไม่ให้ผู้อื่นอ่านข้อมูล
  - Data Integrity: รับประกันว่าข้อมูลไม่ถูกเปลี่ยนแปลงระหว่างส่ง
  - Authentication: ยืนยันตัวตนของเว็บไซต์โดยใช้ใบรับรองดิจิทัล Certificate Authorities (CA)

Data Encryption

- Symmetric Encryption client กับ server คุยกันทำไง key ไหน



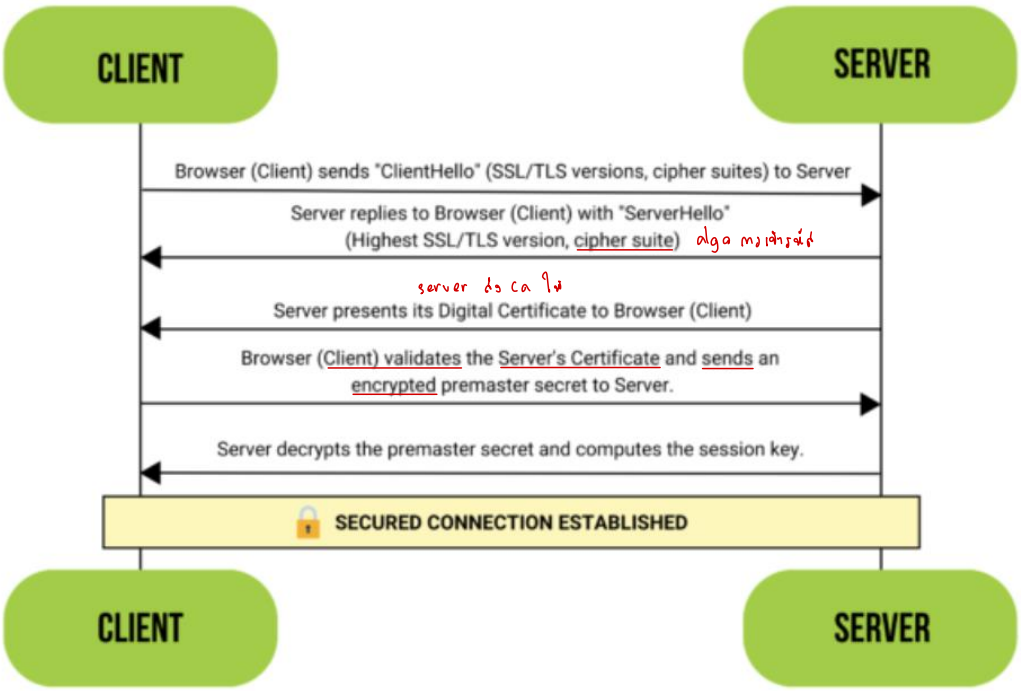
- Asymmetric Encryption server สร้าง key มาคู่นึง โดยเก็บ private key ไว้



Certificate Authorities (CA)

- ใบรับรองดิจิทัล DigiCert, GlobalSign, Let's Encrypt, etc
    - Domain name
    - Public key
    - CA signature
    - Expiry date

TLS Handshake



1. **ClientHello**: SSL/TLS versions, cipher suites
2. **ServerHello**: highest SSL/TLS version, cipher suite
3. **Server's Credentials**: server present "server's certificate" verified by a Certificate Authority (CA)
4. **Client's Verification and Key Generation**: client validate "server's certificate", uses server's public key to encrypt a "premaster secret" a unique session key
5. **Establishing a Secure Connection**: server decrypt "premaster secret" with private key, server and client compute the session key, use session key for symmetric encryption of all communication