

# **http:// vs https://**

"Securing Online Communication"

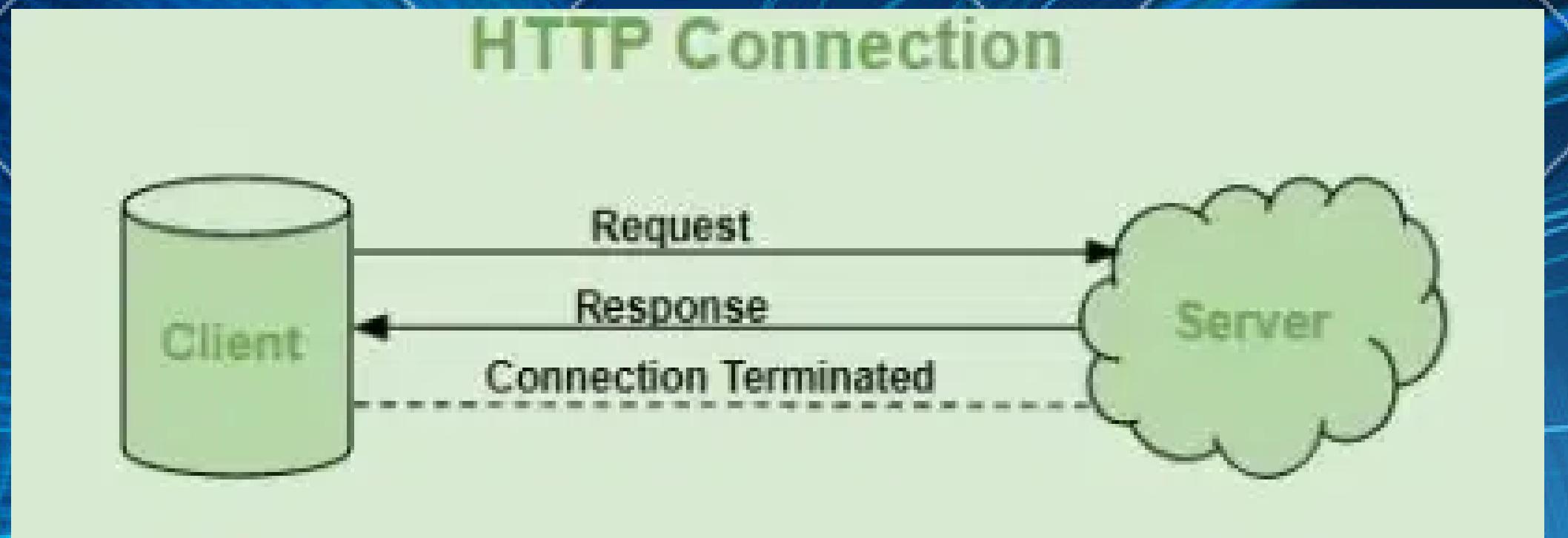


# HTTP Connection



HTTP provides a standard between a web browser and a web server to establish communication. It is a set of rules for transferring data from one computer to another. Data such as text, images, and other multimedia files are shared on the World Wide Web

# HTTP Connection



# HTTP Methods

- Methods are commands to server for request or command

Method	Description
GET	Request to read a Web page
HEAD	Request to read a Web page's header
PUT	Request to store a Web page
POST	Append to a named resource (e.g., a Web page)
DELETE	Remove the Web page
TRACE	Echo the incoming request
CONNECT	Reserved for future use
OPTIONS	Query certain options

# Method classified as:

- Safe Methods: Do not modify resources- retrieval only
- Idempotent Method: Can be called many times , same outcome

HTTP Method	Safe	Idempotent
GET	✓	✓
POST	✗	✗
PUT	✗	✓
DELETE	✗	✓
OPTIONS	✓	✓
HEAD	✓	✓

# Get, Put and Post

- **Get:** GET method means retrieve whatever information is identified by the Request-URI.

- Conditional Get : If-Modified-Since, If-Unmodified-Since, If-Match, If-None-Match
- Partial Get : Range header field.

\*\*Response to GET request is cacheable

- **PUT:** PUT method requests that the enclosed entity be stored under the supplied Request-URI

- If a new resource is created server responded 201 (Created).
- Responses to PUT method are not cacheable

- **POST:** POST method requests that a web server accepts and stores/processes the data enclosed in the body of the request message.
- Function of POST method is determined by the server and is dependent on the Request-URI

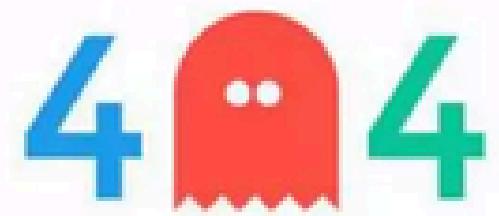
\*\* **GET** requests data from a specified resource **POST** submits data to be processed to a specified resource

\*\*PUT is like a file upload. A put to a URI affects exactly that URI. A **POST** to a URI could have any effect at all.

# HTTP Status Codes

- This class of status code indicates a provisional response

Code	Meaning	Examples
1xx	Information	100 = server agrees to handle client's request
2xx	Success	200 = request succeeded; 204 = no content present
3xx	Redirection	301 = page moved; 304 = cached page still valid
4xx	Client error	403 = forbidden page; 404 = page not found
5xx	Server error	500 = internal server error; 503 = try again later

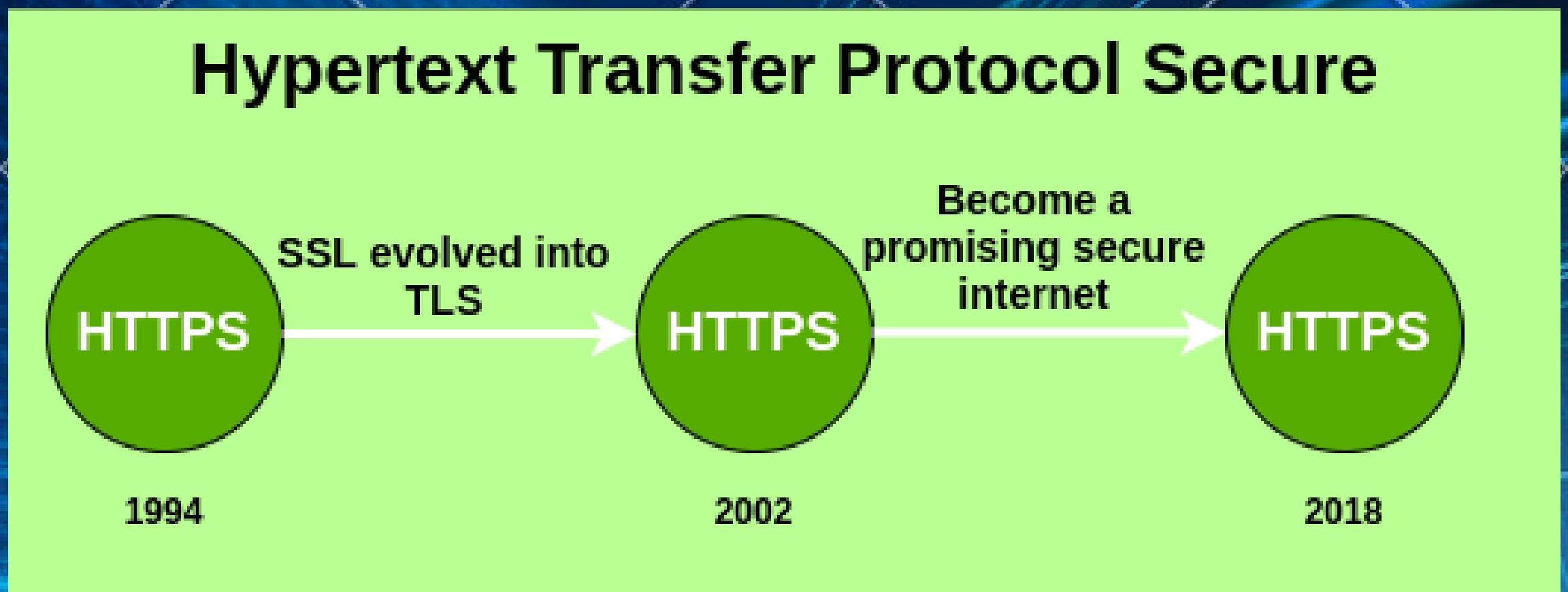


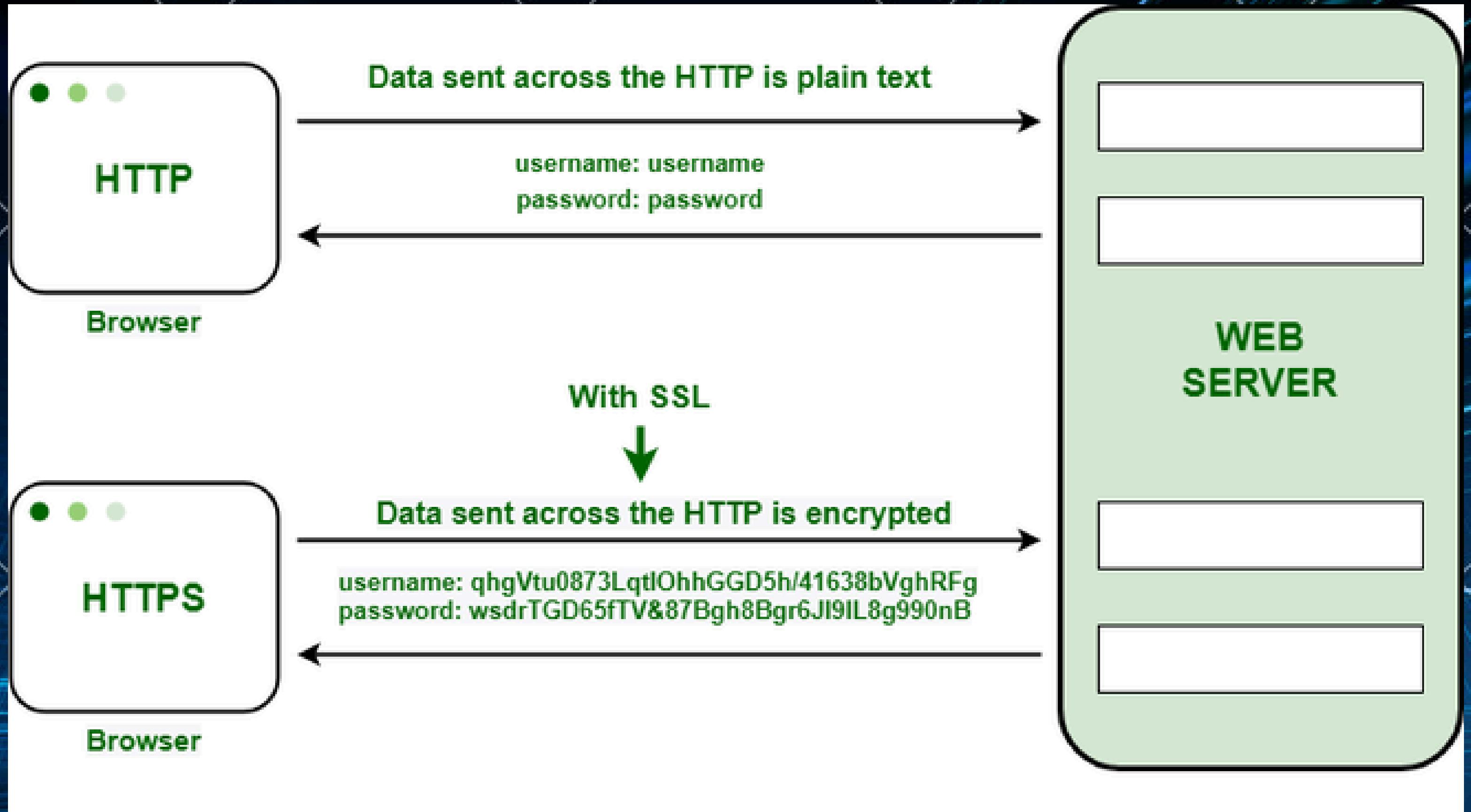
# HTTPS Connection

HTTP Secure (HTTPS), is a combination of the Hypertext Transfer Protocol with the SSL/TLS convention to supply encrypted communication and secure distinguishing proof of an arranged web server.



# HTTPS Connection





# Cookies

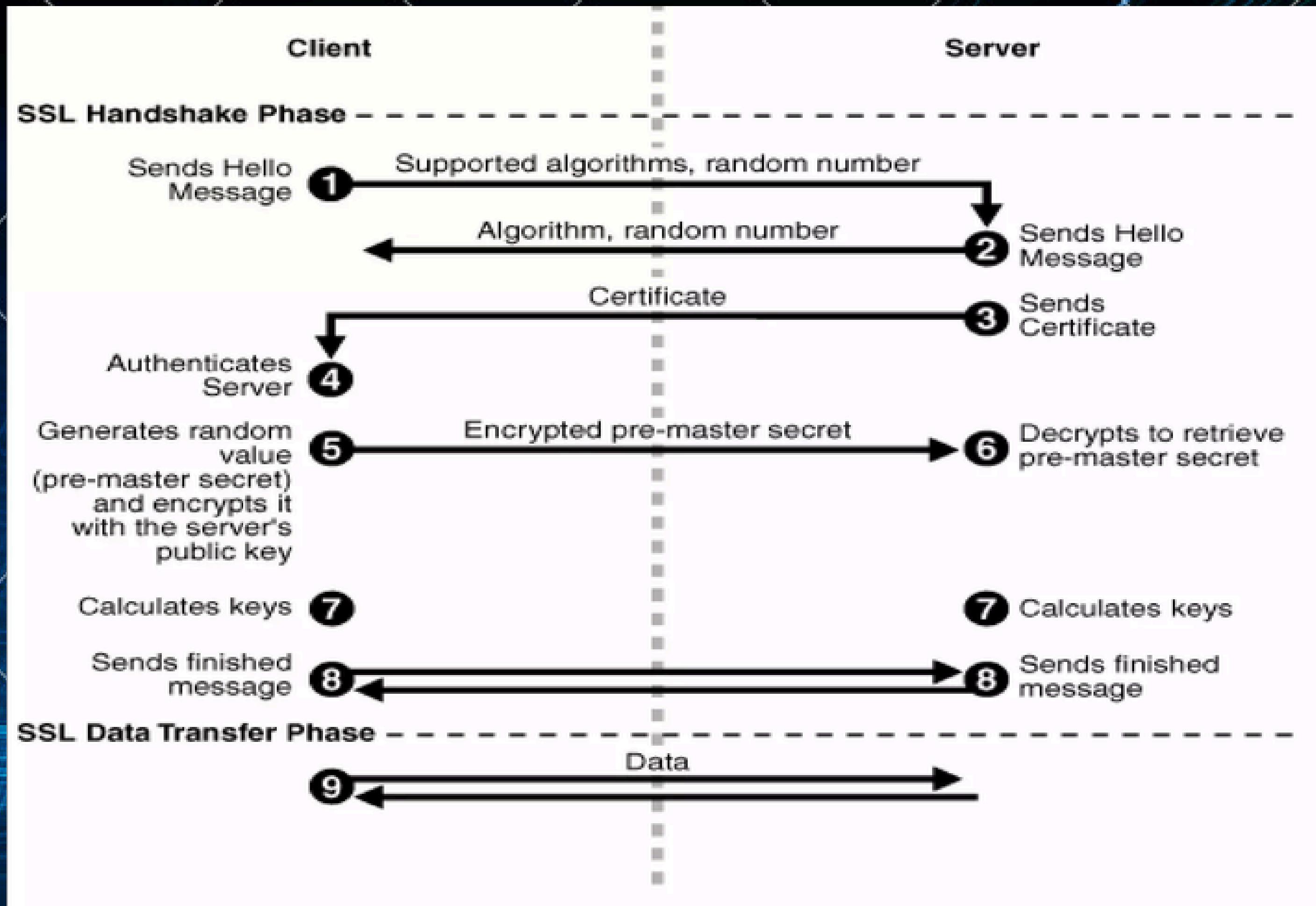
- Sessions are used for maintaining user specific state and authenticated user identities, among many interactions
- privacy and security implications



# SSL/TSL

SSL originally developed by Netscape

- TCP & SSL: provides a reliable & **secure** end-to-end service.
- Uses Public private key to encrypt
- Asymmetric then symmetric
- Key exchange(Deffie-hellman), Cipher (AES), Hash(MD5), Version, random number
- Ensures confidentiality, Message intergrity and key Authentication.



# Main Differences

	HTTP	HTTPS
<b>URL</b>	<code>http://</code>	<code>https://</code>
<b>Security</b>	Unsecure	Enhanced security
<b>Port</b>	PORT 80	PORT 443
<b>OSI Layer</b>	Application Layer	Transport Layer
<b>TLS Certificates</b>	No	Yes
<b>Domain Validation</b>	Not required	Domain validation (+ legal validation)
<b>Encryption</b>	No	Yes

# HTTP

HTTP Works at the Application Layer.

HTTP speed is faster than HTTPS.

HTTP does not use encryption, which results in low security in comparison to HTTPS.

# HTTPS

HTTPS works at Transport Layer.

HTTPS speed is slower than HTTP.

HTTPS uses Encryption which results in better security than HTTP.



# thank you

