URL

URL stands for Uniform Resource Locator, and it is a unique address that identifies a resource on the internet. It specifies the protocol used to access the resource, the domain or IP address where the resource is located, and the specific path to the resource. URLs are used to locate and access websites, web pages, files, images, and other types of resources on the internet. They typically begin with a protocol identifier, such as "http://" or "https://", followed by the domain name or IP address, and finally the path to the specific resource.

Components

1. *Scheme:* This indicates the protocol used to access the resource. Examples include *http, https, ftp, file*, etc.
2. *Hostname or IP address:* This identifies the server where the resource is located. It can be a domain name like *google.com* or an IP address like *216.58.194.174.*
3. *Port number:* This is an optional parameter that specifies the port to use when connecting to the server. If not specified, the default port for the protocol is used (80 for *http*, 443 for *https*, etc.).
4. *Path*: This specifies the location of the resource on the server's file system. It can be a directory path or the name of a file.
5. *Query string*: This is an optional parameter that can be used to pass data to the server. It consists of one or more key-value pairs separated by *&* symbols, and is appended to the end of the URL after a *?* character. For example: *http://example.com/search?q=term.*
6. *Fragment identifier*: This is an optional parameter that identifies a specific part of the resource. It is indicated by a # character followed by the identifier, and is used to link to a specific section of a web page. For example: *https://example.com/page#section2*

HTTP And HTTPS

HTTP (Hypertext Transfer Protocol) is a protocol used to transfer data over the internet. It defines how data is formatted and transmitted between web servers and clients.

HTTPS (Hypertext Transfer Protocol Secure) is a more secure version of HTTP. It uses encryption to protect the communication between web servers and clients, making it more difficult for hackers to intercept and steal data.

Difference :

HTTP	HTTPS
Stands for Hypertext Transfer Protocol	Stands for Hypertext Transfer Protocol Secure
Sends data in plain text	Sends data encrypted with SSL/TLS
Uses port 80 by default	Uses port 443 by default
No security measures	Provides security measures through SSL/TLS encryption
Vulnerable to attacks such as MITM	Resistant to attacks such as MITM
Faster than HTTPS due to no encryption overhead	Slower than HTTP due to SSL/TLS encryption overhead
Used for websites where security is not a concern	Used for websites that require secure data transmission
No SSL/TLS certificate required	SSL/TLS certificate required to enable encryption

Client-Server Model Encryption :

In the client-server model, a client sends a request to a server, which processes the request and returns a response. Encryption is used to secure the communication between the client and the server. In short:

- *Encryption* is the process of converting information into a secret code to prevent unauthorized access.
- The client-server model is a communication model in which a client requests services from a server.
- *HTTPS (Hypertext Transfer Protocol Secure)* is a protocol that encrypts data sent between the client and the server using SSL/TLS.
- *SSL (Secure Sockets Layer) and TLS (Transport Layer Security)* are protocols that provide encryption and authentication to ensure secure communication.
- *HTTP (Hypertext Transfer Protocol)* is a protocol that transfers data between the client and the server, but it does not provide encryption.

SSL And TLS Certificates:

SSL and TLS are protocols for encrypting internet traffic between a web server and a client. *SSL* stands for *Secure Sockets Layer*, and *TLS* stands for *Transport Layer Security*. Both protocols ensure that data is transmitted securely, preventing unauthorized access or interception.

SSL and TLS use certificates to authenticate the identity of a web server. These certificates are issued by trusted third-party organizations called *Certificate Authorities (CAs)*. When a client connects to a web server using SSL or TLS, the server presents its SSL/TLS certificate to the client. The client then verifies the certificate's authenticity by checking it against the trusted CAs. If the certificate is valid, a secure connection is established between the client and the server.

Public Key And Symmetric Key :

- *Public key encryption* uses two keys: a public key and a private key. The public key is used to encrypt data, and the private key is used to decrypt it. The public key is freely available to anyone who wants to send you encrypted data, but only the person who has the private key can decrypt the data.
- *Symmetric key* encryption, on the other hand, uses only one key for both encryption and decryption. The same key is used to both encrypt and decrypt the data. This makes symmetric key encryption much faster than public key encryption.