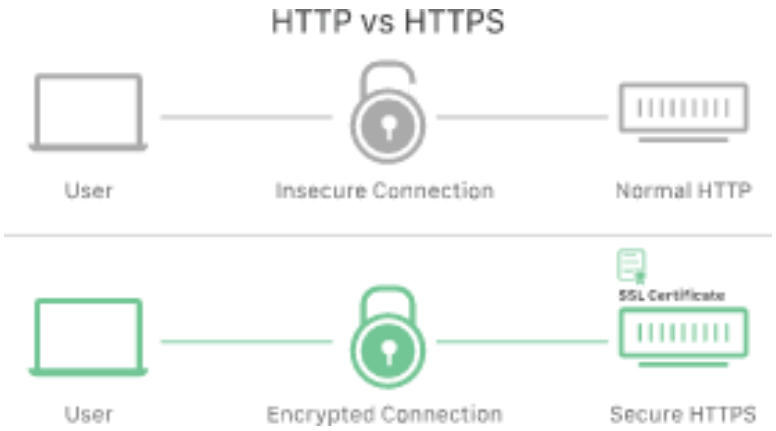
What is SSL?

SSL, or Secure Sockets Layer, is an encryption-based Internet security protocol. It was first developed by Netscape in 1995 for the purpose of ensuring privacy, authentication, and data integrity in Internet communications. SSL is the predecessor to the modern TLS encryption used today.

A website that implements SSL/TLS has "HTTPS" in its URL instead of "HTTP."



How does SSL/TLS work?

- In order to provide a high degree of privacy, SSL encrypts data that is transmitted across the web. This means that anyone who tries to intercept this data will only see a garbled mix of characters that is nearly impossible to decrypt.
- SSL initiates an authentication process called a handshake between two communicating devices to ensure that both devices are really who they claim to be.
- SSL also digitally signs data in order to provide data integrity, verifying that the data is not tampered with before reaching its intended recipient.

There have been several iterations of SSL, each more secure than the last. In 1999 SSL was updated to become TLS.

Why is SSL/TLS important?

Originally, data on the Web was transmitted in plaintext that anyone could read if they intercepted the message. For example, if a consumer visited a shopping website, placed an order, and entered their credit card number on the website, that credit card number would travel across the Internet unconcealed.

SSL was created to correct this problem and protect user privacy. By encrypting any data that goes between a user and a web server, SSL ensures that anyone who intercepts the data can only see a scrambled mess of characters. The consumer's credit card number is now safe, only visible to the shopping website where they entered it.

SSL also stops certain kinds of cyber attacks: It authenticates web servers, which is important because attackers will often try to set up fake websites to trick users and steal data. It also prevents attackers from tampering with data in transit, like a tamper-proof seal on a medicine container.

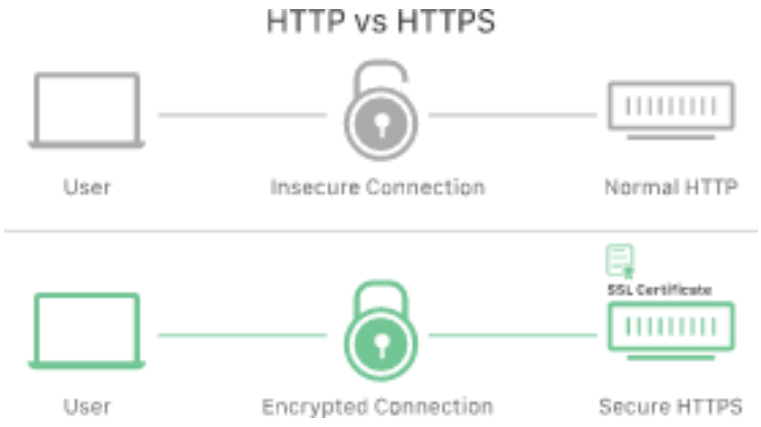
]Are SSL and TLS the same thing?

SSL is the direct predecessor of another protocol called TLS (Transport Layer Security). In 1999 the Internet Engineering Task Force (IETF) proposed an update to SSL. Since this update was being developed by the IETF and Netscape was no longer involved, the name was changed to TLS. The differences between the final version of SSL (3.0) and the first version of TLS are not drastic; the name change was applied to signify the change in ownership.

Since they are so closely related, the two terms are often used interchangeably and confused. Some people still use SSL to refer to TLS, others use the term "SSL/TLS encryption" because SSL still has so much name recognition.

HTTP vs. HTTPS: What are the differences?

HTTPS is HTTP with encryption and verification. The only difference between the two protocols is that HTTPS uses TLS (SSL) to encrypt normal HTTP requests and responses, and to digitally sign those requests and responses. As a result, HTTPS is far more secure than HTTP. A website that uses HTTP has http:// in its URL, while a website that uses HTTPS has https://.



What is HTTP?

HTTP stands for Hypertext Transfer Protocol, and it is a protocol – or a prescribed order and syntax for presenting information – used for transferring data over a network. Most information that is sent over the Internet, including website content and API calls, uses the HTTP protocol. There are two main kinds of HTTP messages: requests and responses.

What is HTTPS?

The S in HTTPS stands for "secure." HTTPS uses TLS (or SSL) to encrypt HTTP requests and responses, instead of the text, an attacker would see a bunch of seemingly random characters.

In HTTPS, how does TLS/SSL encrypt HTTP requests and responses?

TLS uses a technology called public key cryptography: there are two keys, a public key and a private key, and the public key is shared with client devices via the server's SSL certificate. When a client opens a connection with a server, the two devices use the public and private key to agree on new keys, called session keys, to encrypt further communications between them.

All HTTP requests and responses are then encrypted with these session keys, so that anyone who intercepts communications can only see a random string of characters, not the plaintext.

What is Time Stamping?

Time stamping is an increasingly valuable complement to digital signing practices, enabling organizations to record when a digital item—such as a message, document, transaction or piece of software—was signed. For some applications, the timing of a digital signature is critical, as in the case of stock trades, lottery ticket issuance and some legal proceedings. Even when time is not intrinsic to the application, time

stamping is helpful for record keeping and audit processes, because it provides a mechanism to prove whether the digital certificate was valid at the time it was used. The growing importance of digital signing solutions has created a corresponding demand for time stamping, so many software programs, such as Microsoft Office, support time stamping capabilities.

The **Time-Stamp Protocol**, or **TSP** is a cryptographic protocol for certifying timestamps using X.509 certificates and public key infrastructure. The timestamp is the signer's assertion that a piece of electronic data existed at or before a particular time. The protocol is defined in RFC 3161. One application of the protocol is to show that a digital signature was issued before a point in time, for example before the corresponding certificate was revoked.

The TSP protocol is an example of trusted timestamping. It has been extended to create the ANSI ASC X9.95 Standard. In the protocol a Time Stamp Authority (TSA) is a trusted third party that can provide a timestamp to be associated with a hashed version of some data. It is a request-response protocol, where the request contains a hash of the data to be signed. This is sent to the TSA and the response contains a Time Stamp Token (TST) which itself includes the hash of the data, a unique serial number, a timestamp and a digital signature. The signature is generated using the private key of the TSA. The protocol can operate over a number of different transports, including email, TCP sockets or HTTP. When presented with a TST, someone may verify that the data existed at the timestamp in the TST by verifying the signature using the public key of the TSA and that the hash of the data matches that included in the TST.

Secure Electronic Transaction (SET)

Secure electronic transaction (SET) was an early communications protocol used by e-commerce websites to secure electronic debit and credit card payments. Secure electronic transaction was used to facilitate the secure transmission of consumer card information via electronic portals on the internet. Secure electronic transaction protocols were responsible for blocking out the personal details of card information, thus preventing merchants, hackers, and electronic thieves from accessing consumer information.

Secure electronic transaction protocols were supported by most of the major providers of electronic transactions, such as Visa and MasterCard. These protocols allowed merchants to verify their customers' card information without actually seeing it, thus protecting the customer. The information on the cards was transferred directly to the credit card company for verification.

The process of secure electronic transactions used digital certificates that were assigned to provide electronic access to funds, whether it was a credit line or bank account. Every time a purchase was made electronically, an encrypted digital certificate was generated for participants in the transaction–the customer, merchant, and financial institution–along with matching digital keys that allowed them to confirm the certificates of the other party and verify the transaction. The algorithms used would ensure that only a party with the corresponding digital key would be able to confirm the transaction. As a result, a consumer’s credit card or bank account information could be used to complete the transaction without revealing any of their personal details, such as their account numbers. Secure electronic transactions were meant to be a form of security against account theft, hacking, and other criminal actions.

SSL versus SET
https://www.geeksforgeeks.org/difference-between-secure-socket-layer-ssl-and-secure-electronic-transaction-set/

Electronic Money

Electronic money refers to the currency electronically stored on electronic systems and digital databases, as opposed to physical paper and coin money, and is used to make it easier for users to transact electronically. The value of the electronic currency is backed by fiat currency.

- Electronic money refers to the currency electronically stored on electronic systems and digital databases used to make it easier to transact electronically. It is popularly referred to by many names, including digital cash, digital currency, e-money, and so on.
- Fiat money, simply put, is a legal tender, whose value as a currency is established by an issuing government and consequently, is also regulated by it.
- Electronic money can be classified into two broad categories: hard and soft.

Email Security

Email security is the process of preventing email-based cyber attacks and unwanted communications. It spans protecting inboxes from takeover, protecting domains from spoofing, stopping phishing attacks, preventing fraud, blocking malware delivery, filtering spam, and using encryption to protect the contents of emails from unauthorized persons.

Security and privacy were not built into email when it was first invented, and despite email's importance as a communication method, these are still not built into email by default. As a result, email is a major attack vector for organizations large and small, and for individual people as well.

What kinds of attacks occur via email?

Some of the common types of email attacks include:

- Fraud: Email-based fraud attacks can take a variety of forms, from the classic advance-fee scams directed at everyday people to business email compromise (BEC) messages that aim to trick large enterprise accounting departments into transferring money to illegitimate accounts. Often the attacker will use domain spoofing to make the request for funds look like it comes from a legitimate source.
- Phishing: A phishing attack tries to get the victim to give the attacker sensitive information. Email phishing attacks may direct users to a fake webpage that collects credentials, or simply pressure the user to send the information to an email address secretly controlled by the attacker. Domain spoofing is also common in attacks like these.
- Malware: Types of malware delivered over email include spyware, scareware, adware, and ransomware, among others. Attackers can deliver malware via email in several different ways. One of the most common is including an email attachment that contains malicious code.
- Account takeover: Attackers take over email inboxes from legitimate users for a variety of purposes, such as monitoring their messages, stealing information, or using legitimate email addresses to forward malware attacks and spam to their contacts.
- Email interception: Attackers can intercept emails in order to steal the information they contain, or to carry out on-path attacks in which they impersonate both sides of a conversation to each other. The most common method for doing this is monitoring network data packets on wireless local area networks (LANs), as intercepting an email as it transits the Internet is extremely difficult.