Transport Layer Security, or TLS, is a widely adopted security protocol designed to facilitate privacy and data security for communications over the Internet. A primary use case of TLS is encrypting the communication between web applications and servers, such as web browsers loading a website. TLS can also be used to encrypt other communications such as email, messaging, and voice over IP (VOIP). In this article we will focus on the role of TLS in [web application security](https://www.cloudflare.com/learning/security/what-is-web-application-security/).

listen 443 ssl; SSL\_certificate Full\_path\_of\_publickey SSL\_certificate\_Key Full\_path\_of\_private\_key ssl\_protocols TLSv1.3 // Enabling TLS protocal.

For enabling HTTPv2 protocal. listen 443 ssl https2;

## What’s the difference between TLS and SSL?

TLS evolved from a previous encryption protocol called Secure Socket Layer ([SSL](https://www.cloudflare.com/learning/ssl/what-is-ssl/)), which was developed by Netscape. TLS version 1.0 actually began development as SSL version 3.1, but the name of the protocol was changed before publication in order to indicate that it was no longer associated with Netscape. Because of this history, the terms TLS and SSL are sometimes used interchangeably.

## What’s the difference between TLS and HTTPS?

[HTTPS](https://www.cloudflare.com/learning/ssl/what-is-https/) is an implementation of TLS encryption on top of the [HTTP](https://www.cloudflare.com/learning/ddos/glossary/hypertext-transfer-protocol-http/) protocol, which is used by all websites as well as some other web services. Any website that uses HTTPS is therefore employing TLS encryption.

## Why should you use TLS?

TLS encryption can help protect web applications from attacks such as [data breaches](https://www.cloudflare.com/learning/security/what-is-a-data-breach), and [DDoS attacks](https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/). Additionally, TLS-protected HTTPS is quickly becoming a standard practice for websites. For example, the Google Chrome browser is cracking down on non-HTTPS sites, and everyday Internet users are starting to become more wary of websites that don’t feature the HTTPS padlock icon.

## How does TLS work?

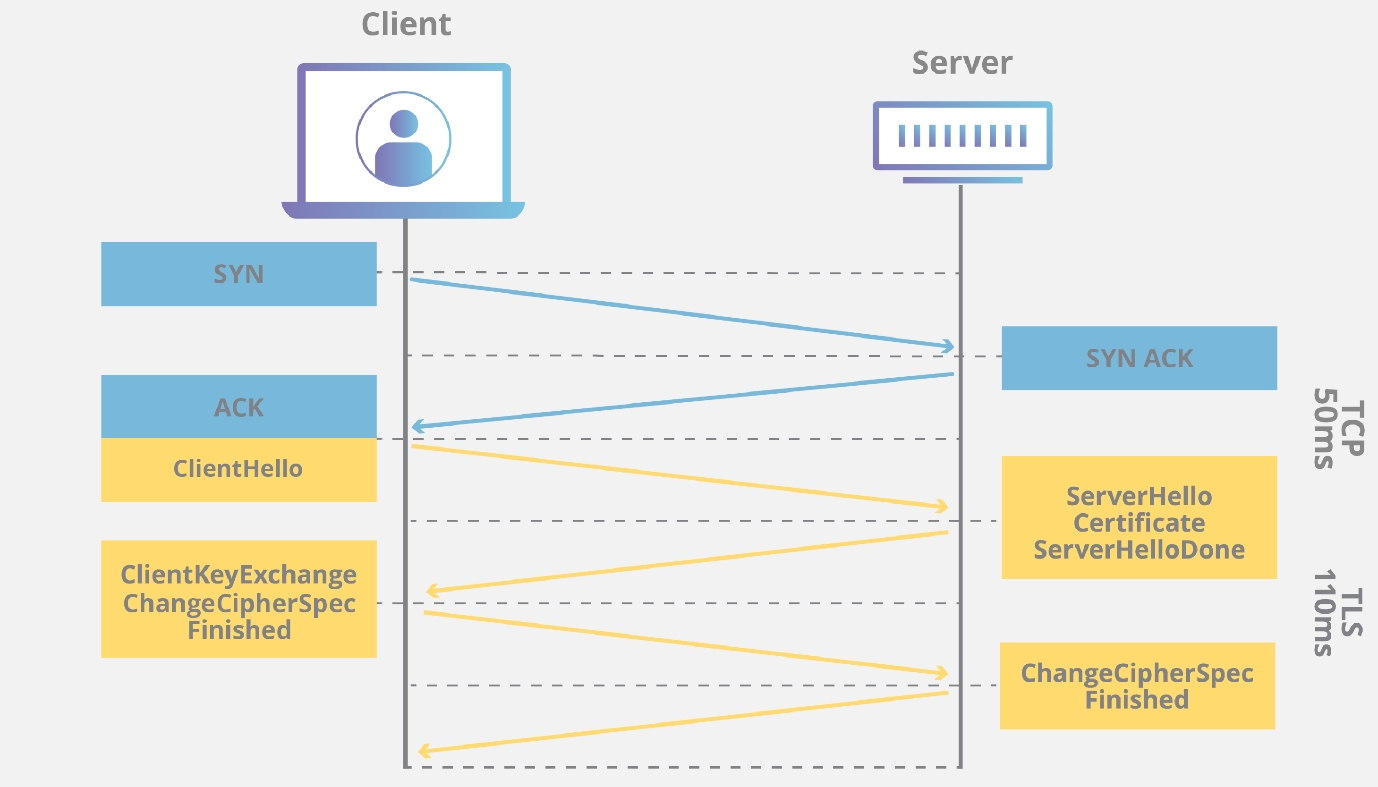
TLS can be used on top of a transport-layer security protocol like [TCP](https://www.cloudflare.com/learning/ddos/glossary/tcp-ip/). There are three main components to TLS: Encryption, Authentication, and Integrity.

* **Encryption:** hides the data being transferred from third parties.
* **Authentication:** ensures that the parties exchanging information are who they claim to be.
* **Integrity:** verifies that the data has not been forged or tampered with.

A TLS connection is initiated using a sequence known as the TLS handshake. The TLS handshake establishes a cypher suite for each communication session. The cypher suite is a set of algorithms that specifies details such as which shared encryption keys, or session keys, will be used for that particular session. TLS is able to set the matching session keys over an unencrypted channel thanks to a technology known as public key cryptography.

The handshake also handles authentication, which usually consists of the server proving its identity to the client. This is done using public keys. Public keys are encryption keys that use one-way encryption, meaning that anyone can unscramble data encrypted with the private key to ensure its authenticity, but only the original sender can encrypt data with the private key.

Once data is encrypted and authenticated, it is then signed with a message authentication code (MAC). The recipient can then verify the MAC to ensure the integrity of the data. This is kind of like the tamper-proof foil found on a bottle of aspirin; the consumer knows no one has tampered with their medicine because the foil is intact when they purchase it.



## How does TLS affect web application performance?

Because of the complex process involved in setting up a TLS connection, some load time and computational power must be expended. The client and server must communicate back and forth several times before any data is transmitted, and that eats up precious milliseconds of load times for web applications, as well as some memory for both the client and the server.

Thankfully there are technologies in place that help to mitigate the lag created by the TLS handshake. One is TLS False Start, which lets the server and client start transmitting data before the TLS handshake is complete. Another technology to speed up TLS is TLS Session Resumption, which allows clients and servers that have previously communicated to use an abbreviated handshake.

These improvements have helped to make TLS a very fast protocol that shouldn’t noticeably affect [load times](https://www.cloudflare.com/learning/performance/why-site-speed-matters/). As for the computational costs associated with TLS, they are mostly negligible by today’s standards. For example, when Google moved their entire Gmail platform to HTTPS in 2010, there was no need for them to enable any additional hardware. The extra load on their servers as a result of TLS encryption was less than 1%.