CLOUDFLARE

What Happens in a TLS Handshake? | SSL Handshake

In a TLS/SSL handshake, clients and servers exchange SSL certificates, cipher suite requirements, and randomly generated data for creating session keys.

Share

TLS Handshake Learning Objectives

After reading this article you will be able to:

- Learn what a TLS handshake is
- Understand what a TLS handshake accomplishes
- Explain the steps in a TLS handshake
- Explore different types of TLS handshakes

Related Content

What is an SSL Certificate?

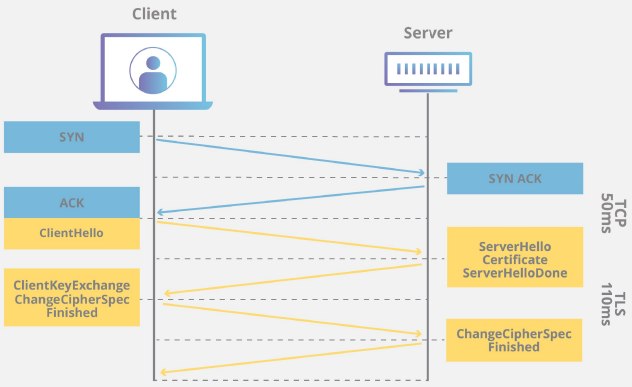
Keyless SSL

What is SSL?

What is Mixed Content?

What is HTTPS?

What is a TLS handshake?



TLS is an encryption protocol designed to secure Internet communications. A TLS handshake is the process that kicks off a communication session that uses TLS encryption. During a TLS handshake, the two communicating sides exchange messages to acknowledge each other, verify each other, establish the encryption algorithms they will use, and agree on session keys. TLS handshakes are a foundational part of how HTTPS works.

TLS vs. SSL handshakes

SSL, or Secure Sockets Layer, was the original encryption protocol developed for HTTP. SSL was replaced by TLS, or Transport Layer Security, some time ago. SSL handshakes are now called TLS handshakes, although the "SSL" name is still in wide use.

When does a TLS handshake occur?

A TLS handshake takes place whenever a user navigates to a website over HTTPS and the browser first begins to query the website's origin server. A TLS handshake also happens whenever any other communications use HTTPS, including API calls and DNS over HTTPS queries.

TLS handshakes occur after a TCP connection has been opened via a TCP handshake.

What happens during a TLS handshake?

During the course of a TLS handshake, the client and server together will do the following:

- Specify which version of TLS (TLS 1.0, 1.2, 1.3, etc.) they will use
- Decide on which cipher suites (see below) they will use
- Authenticate the identity of the server via the server's public key and the SSL certificate authority's digital signature
- Generate session keys in order to use symmetric encryption after the handshake is complete

What are the steps of a TLS handshake?

TLS handshakes are a series of datagrams, or messages, exchanged by a client and a server. A TLS handshake involves multiple steps, as the client and server exchange the information necessary for completing the handshake and making further conversation possible.

The exact steps within a TLS handshake will vary depending upon the kind of key exchange algorithm used and the cipher suites supported by both sides. The RSA key exchange algorithm is used most often. It goes as follows:

1. The 'client hello' message: The client initiates the handshake by sending a "hello" message to the server. The message will include which TLS version the client supports, the cipher suites supported, and a string of random bytes known as the "client random."
2. The 'server hello' message: In reply to the client hello message, the server sends a message containing the server's SSL certificate, the server's chosen cipher suite, and the "server random," another random string of bytes that's generated by the server.

3. Authentication: The client verifies the server's SSL certificate with the certificate authority that issued it. This confirms that the server is who it says it is, and that the client is interacting with the actual owner of the domain.
4. The premaster secret: The client sends one more random string of bytes, the "premaster secret." The premaster secret is encrypted with the public key and can only be decrypted with the private key by the server. (The client gets the public key from the server's SSL certificate.)
5. Private key used: The server decrypts the premaster secret.
6. Session keys created: Both client and server generate session keys from the client random, the server random, and the premaster secret. They should arrive at the same results.
7. Client is ready: The client sends a "finished" message that is encrypted with a session key.
8. Server is ready: The server sends a "finished" message encrypted with a session key.
9. Secure symmetric encryption achieved: The handshake is completed, and communication continues using the session keys.

All TLS handshakes make use of asymmetric encryption (the public and private key), but not all will use the private key in the process of generating session keys. For instance, an ephemeral Diffie-Hellman handshake proceeds as follows:

1. Client hello: The client sends a client hello message with the protocol version, the client random, and a list of cipher suites.
2. Server hello: The server replies with its SSL certificate, its selected cipher suite, and the server random. In contrast to the RSA handshake described above, in this message the server also includes the following (step 3):
3. Server's digital signature: The server uses its private key to encrypt the client random, the server random, and its DH parameter*. This encrypted data functions as the server's digital signature, establishing that the server has the private key that matches with the public key from the SSL certificate.
4. Digital signature confirmed: The client decrypts the server's digital signature with the public key, verifying that the server controls the private key and is who it says it is. Client DH parameter: The client sends its DH parameter to the server.
5. Client and server calculate the premaster secret: Instead of the client generating the premaster secret and sending it to the server, as in an RSA handshake, the client and server use the DH parameters they exchanged to calculate a matching premaster secret separately.
6. Session keys created: Now, the client and server calculate session keys from the premaster secret, client random, and server random, just like in an RSA handshake.
7. Client is ready:
8. Same as an RSA handshake.
9. Server is ready
10. Secure symmetric encryption achieved

*DH parameter: DH stands for Diffie-Hellman. The Diffie-Hellman algorithm uses exponential calculations to arrive at the same premaster secret. The server and client each provide a parameter for the calculation, and when combined they result in a different calculation on each side, with results that are equal.

To read more about the contrast between ephemeral Diffie-Hellman handshakes and other kinds of handshakes, and how they achieve forward secrecy, see What is Keyless SSL?

What is a cipher suite?

A cipher suite is a set of encryption algorithms for use in establishing a secure communications connection. (An encryption algorithm is a set of mathematical operations performed on data for making data appear random.) There are a number of cipher suites in wide use, and an essential part of the TLS handshake is agreeing upon which cipher suite will be used for that handshake.

To learn more about TLS/SSL, see How does SSL work? To test if a website uses TLS correctly, visit the Cloudflare Diagnostic Center.

Cryptographic Key
What is a Session Key?

SSL Glossary
What is Mixed Content?
SSL Handshake
What is an SSL Certificate?
SSL Certificate Types
Why Use TLS 1.3?
What is SNI?
What is Encrypted SNI?
What is Domain Spoofing?

Learning Center Navigation
Learning Center Home
DDoS Learning Center
CDN Learning Center
DNS Learning Center
Performance Learning Center
Security Learning Center
Serverless Learning Center
Bots Learning Center
Cloud Learning Center
Access Management Learning Center
Network Layer Learning Center
Privacy Learning Center
Video Streaming Learning Center