The SSL Handshake

By Naveen PS

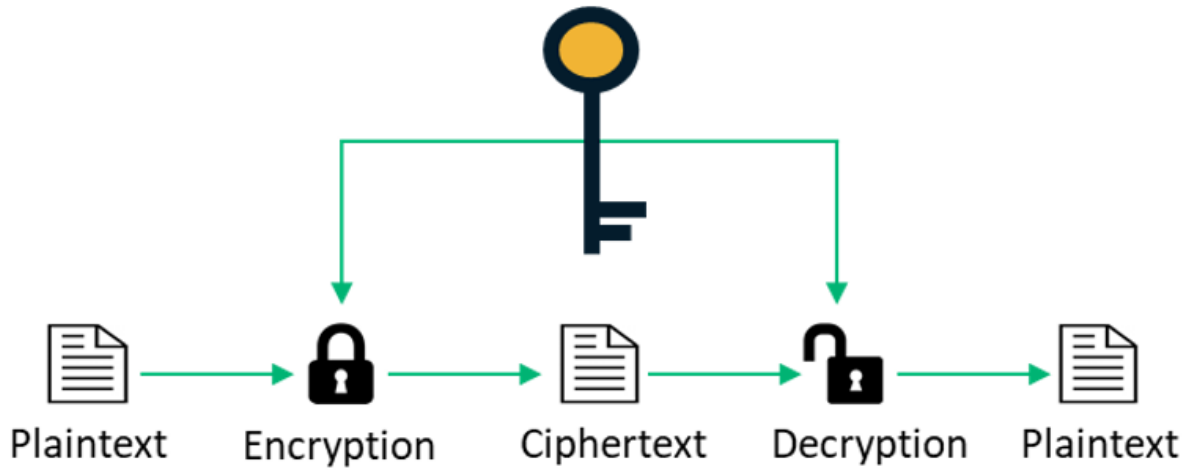
MAKERDEMY
LEARN EXPLORE MAKE

The SSL Handshake

Encryption is the process of converting human-readable data into unintelligible ciphertext. This scrambling of data is the result of an algorithmic operation that uses a cryptographic key. Simply put, encryption takes your data and makes it random enough so that anybody who steals it can't read it unless they have the key to turn it back into a legible form.

The use of encryption is necessary if we want privacy and for protecting our data at rest, in use, or in motion. There are two main types – symmetric encryption vs asymmetric encryption – which we will compare in this article.

Symmetric Encryption



Graphic of how symmetric encryption works

In the case of symmetric encryption, the same key is used for both encrypting and decrypting messages. Because the entire mechanism is dependent on keeping the key a shared secret – meaning that it needs to be shared with the recipient in a secure way so that only they can use it to decrypt the message – it does not scale well.

Symmetric encryption algorithms can use either block ciphers or stream ciphers. With block ciphers, a number of bits (in chunks) is encrypted as a single unit. For instance, AES uses a block size of 128 bits with options for three different key lengths – 128, 192, or 256 bits.

Symmetric encryption suffers from key exhaustion issues and, without proper maintenance of a key hierarchy or effective key rotation, it's possible that every usage can leak information that can be potentially leveraged by an attacker to reconstruct the secret key. Although there are key management issues with symmetric encryption, its faster and functions without a lot of overheads on network or CPU resources. Therefore, it's often used in combination with asymmetric encryption, which we'll look into in the following section.

Key Takeaways of Symmetric Encryption

- There's a single shared key that's used for encryption and decryption.

Differentiator	Symmetric Key Encryption	Asymmetric Key Encryption
1. Symmetric Key vs Asymmetric key	Only one key (symmetric key) is used, and the same key is used to encrypt and decrypt the message.	Two different cryptographic keys (asymmetric keys), called the public and the private keys, are used for encryption and decryption.
2. Complexity and Speed of Execution	It's a simple technique, and because of this, the encryption process can be carried out quickly.	It's a much more complicated process than symmetric key encryption, and the process is slower.
3. Length of Keys	The length of the keys used is typically 128 or 256 bits, based on the security requirement.	The length of the keys is much larger, e.g., the recommended RSA key size is 2048 bits or higher.
4. Usage	It's mostly used when large chunks of data need to be transferred.	It's used in smaller transactions, primarily to authenticate and establish a secure communication channel prior to the actual data transfer.

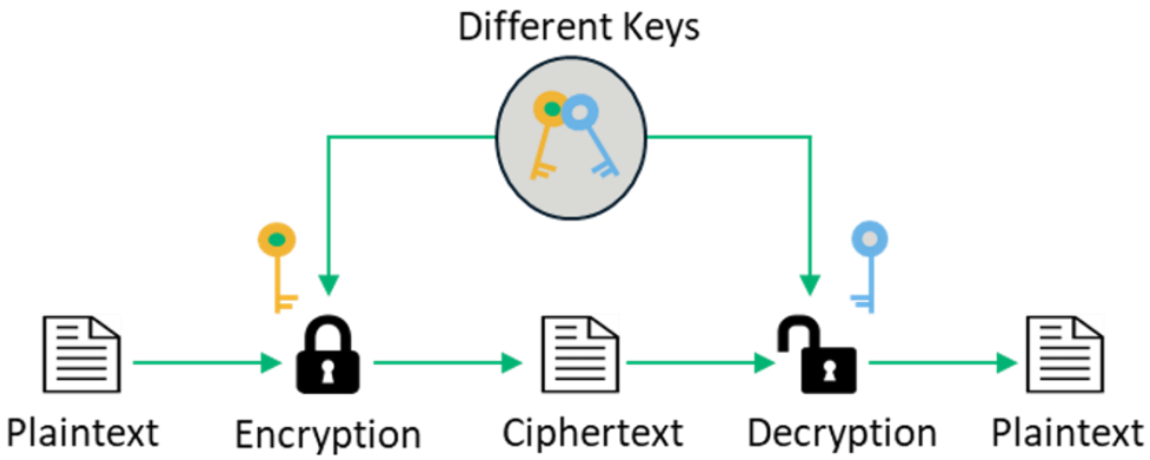
5. Security	The secret key is shared. Consequently, the risk of compromise is higher.	The private key is not shared, and the overall process is more secure as compared to symmetric encryption.
Examples of Algorithms	Examples include RC4, AES, DES, 3DES, etc.	Examples include RSA, Diffie-Hellman, ECC, etc.

- It doesn't scale very well because the secret key must not be lost or shared with unauthorized parties, or else they can read the message.

Asymmetric Encryption

Asymmetric encryption uses a pair of related keys – a public and a private key. The public key, which is accessible to everyone, is what's used to encrypt a plaintext message before sending it. To decrypt and read this message, you need to hold the private key. The public and the private keys are mathematically related, but the private key cannot be derived from it.

In asymmetric encryption (also known as public-key cryptography or public key encryption), the private key is only shared with the key's initiator since its security needs to be maintained.



Symmetric vs asymmetric encryption:

Because asymmetric encryption is a more complicated process than its symmetric counterpart, the time required is greater. However, this type of encryption offers a higher level of security as compared to symmetric encryption since the private key is not meant to be shared and is kept a secret. It is also a considerably more scalable technique.

Key Takeaways of Asymmetric Encryption

- It involves the use of two mathematically related keys. The public key (the one that's known to everybody) and the private key (which is only known by you) are required for encrypting and decrypting the message. The private key cannot be derived from the public key.
- The public key is used by others to encrypt the messages they send to you, but to decrypt and read these messages, one needs access to the private key.

What Is the Difference Between Symmetric and Asymmetric Encryption?

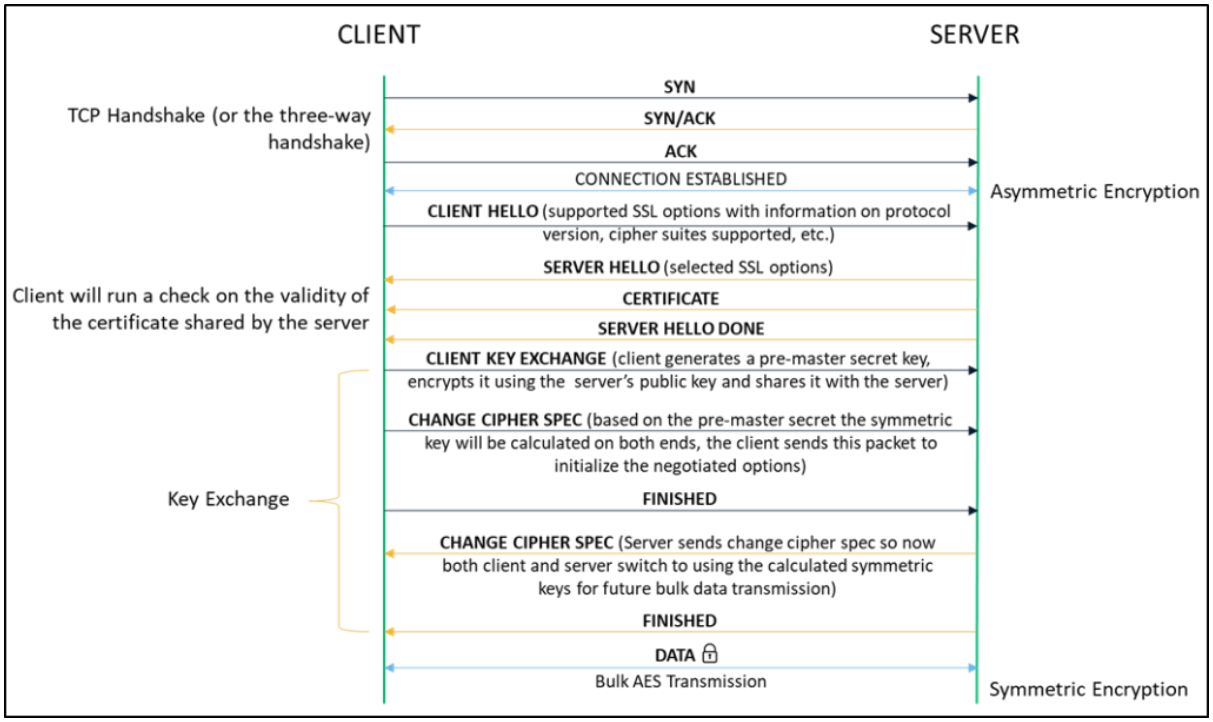
So, when we talk about symmetric vs asymmetric encryption, which is more secure? Asymmetric encryption is the more secure one, while symmetric encryption is faster. They're both very effective in different ways and, depending on the task at hand, either or both may be deployed alone or together.

Hopefully, you now have an understanding of the difference between symmetric encryption vs asymmetric encryption. The table below provides a more in-depth comparison between symmetric vs asymmetric encryption:

Symmetric vs Asymmetric Encryption in the Context of the SSL/TLS Handshake

When we surf the net using the insecure HTTP protocol, data travels in an unencrypted format that can easily be intercepted and stolen by anyone listening in on the network. SSL/TLS certificates are used to encrypt the communication channel between the client (web browsers like Chrome, Firefox, etc.) and the server you're attempting to connect with so you can browse securely over HTTPS. While there are a number of steps involved in the handshake, the entire encryption process (that begins using asymmetric encryption and later switches to symmetric encryption for bulk transmission) takes only a few milliseconds.

Every time we connect to a website over HTTPS, an encrypted communication channel is established between our client browser and the server hosting the site. Let's get a brief overview of where encryption comes into play when setting up a secured connection:


CLIENT
SERVER
TCP Handshake (or the three-way handshake)
SYN
SYN/ACK
ACK
CONNECTION ESTABLISHED
Asymmetric Encryption
CLIENT HELLO (supported SSL options with information on protocol version, cipher suites supported, etc.)
SERVER HELLO (selected SSL options)
Client will run a check on the validity of the certificate shared by the server
CERTIFICATE
SERVER HELLO DONE
CLIENT KEY EXCHANGE (client generates a pre-master secret key, encrypts it using the server's public key and shares it with the server)
CHANGE CIPHER SPEC (based on the pre-master secret the symmetric key will be calculated on both ends, the client sends this packet to initialize the negotiated options)
Key Exchange
FINISHED
CHANGE CIPHER SPEC (Server sends change cipher spec so now both client and server switch to using the calculated symmetric keys for future bulk data transmission)
FINISHED
DATA
Bulk AES Transmission
Symmetric Encryption

THE END