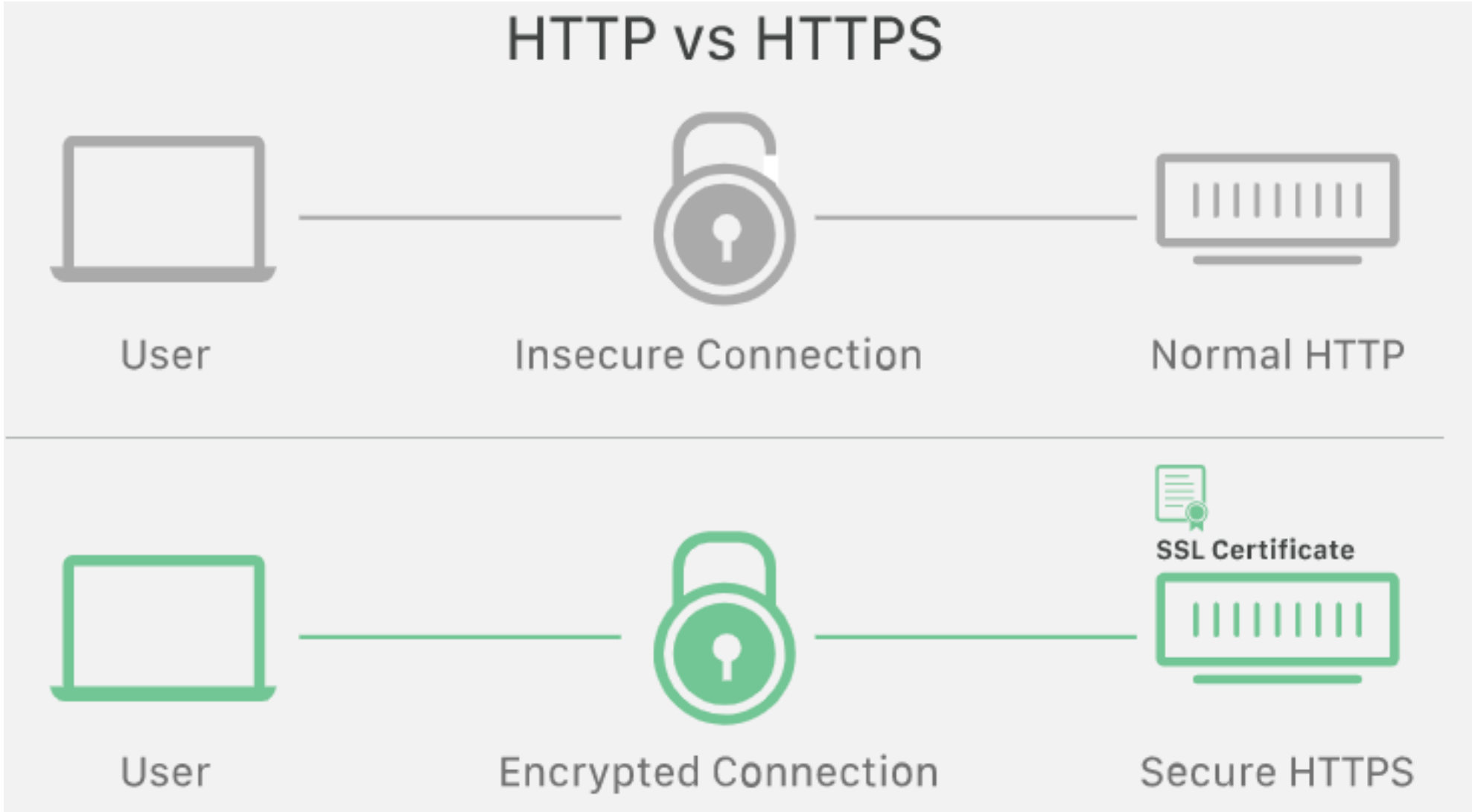
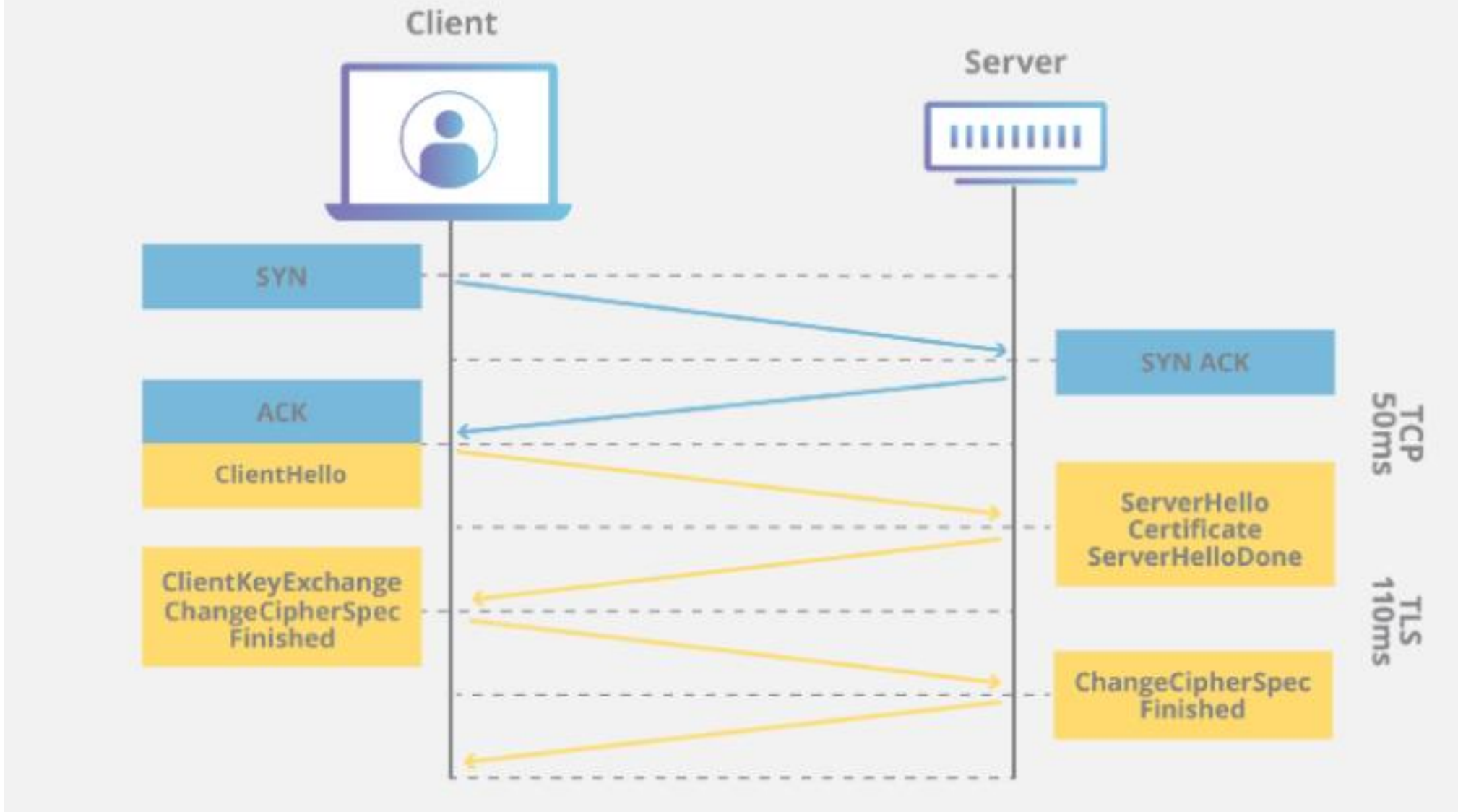
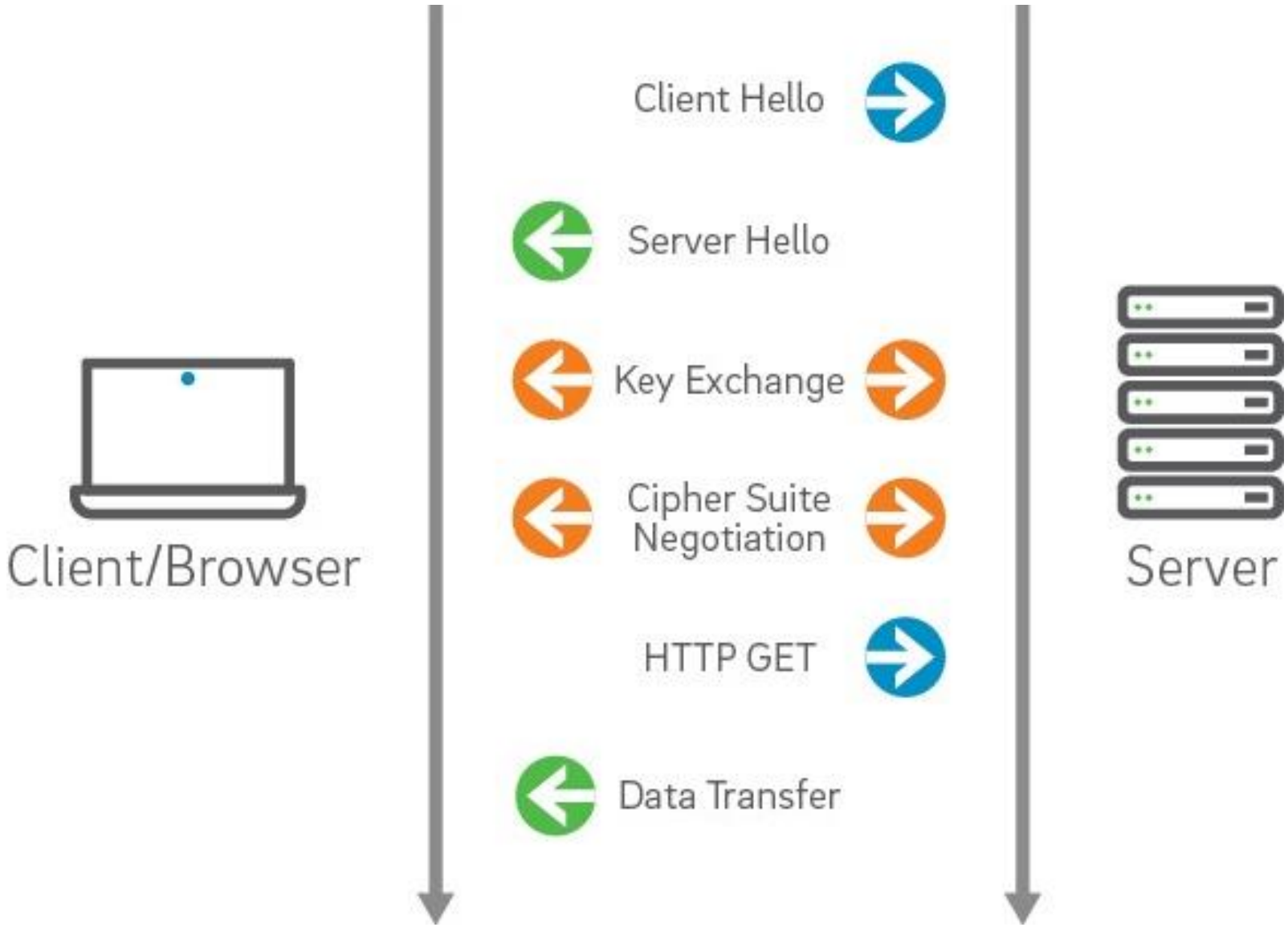
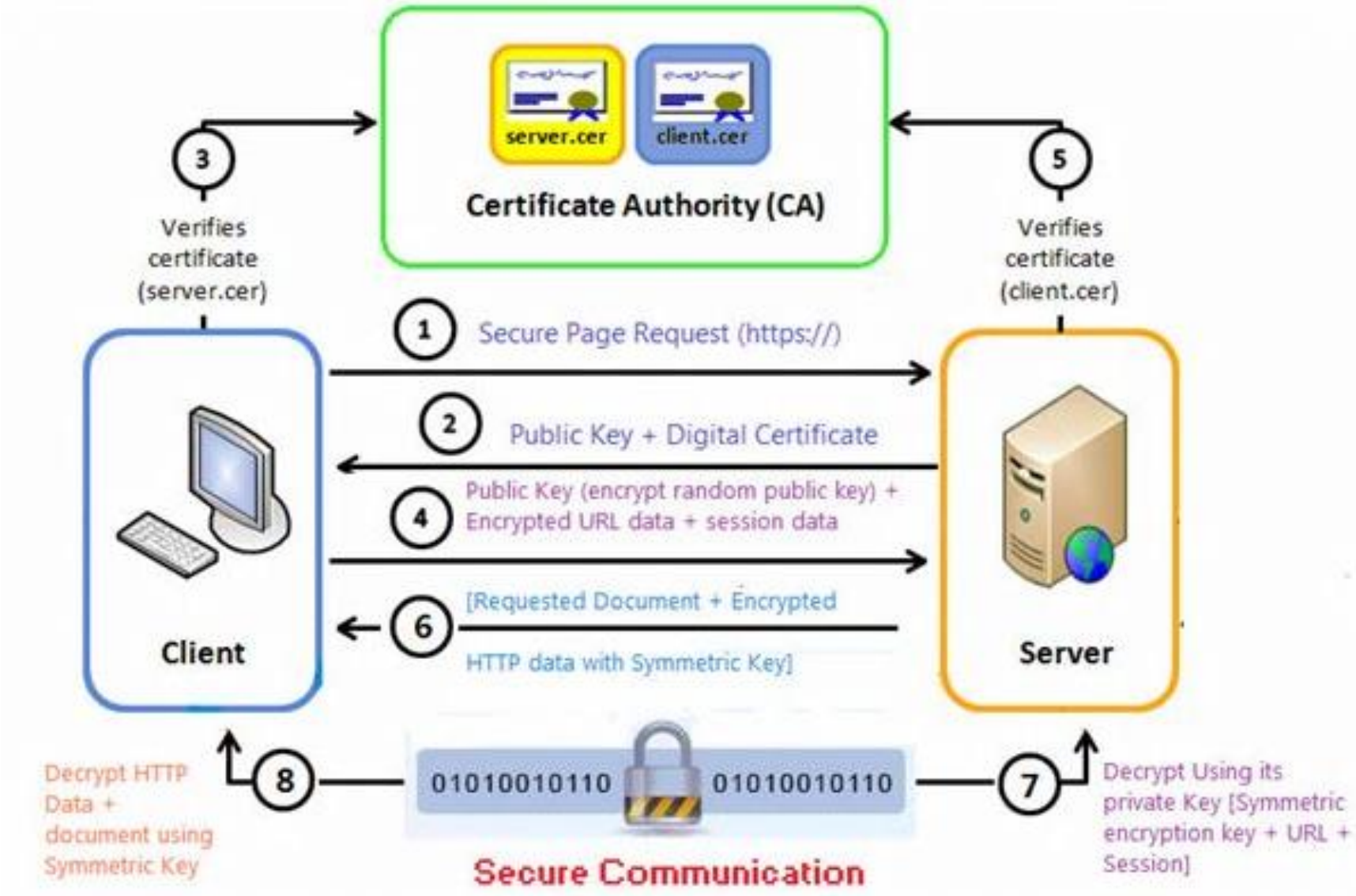

HTTP vs HTTPS
User
Insecure Connection
Normal HTTP
SSL Certificate
User
Encrypted Connection
Secure HTTPS

What is a TLS handshake?




Client Hello
Server Hello
Key Exchange
Cipher Suite Negotiation
HTTP GET
Data Transfer
Client/Browser
Server


server.cer
client.cer
Certificate Authority (CA)
3
Verifies certificate (server.cer)
5
Verifies certificate (client.cer)
1
Secure Page Request (https://)
2
Public Key + Digital Certificate
4
Public Key (encrypt random public key) + Encrypted URL data + session data
6
[Requested Document + Encrypted HTTP data with Symmetric Key]
Client
Server
8
Decrypt HTTP Data + document using Symmetric Key
01010010110
01010010110
7
Decrypt Using its private Key [Symmetric encryption key + URL + Session]
Secure Communication

1.브라우저는 서버에 연결하려는 요청을 보내고 보안 페이지(일반적으로 문서)를 요청합니다.

2.웹 서버는 서명 인증서와 함께 공개 키를 클라이언트로 다시 보냅니다.

3.브라우저는 인증서가 신뢰하는 CA에서 발급했는지 여부를 확인합니다. 클라이언트는 인증서에 있는 정보와 웹사이트에서 받은 정보를 비교하여 모든 내용을 확인합니다. 그렇다면 브라우저는 녹색 자물쇠를 표시하여 서버 인증서의 순도를 표시하고 클라이언트는 계속 진행합니다.

4.브라우저는 무작위 대칭 암호화 키를 생성한 다음 서버의 공개 키로 암호화합니다. 마지막으로 암호화된 URL 및 기타 암호화된 HTTP 데이터와 함께 서버로 보냅니다.

5.웹 서버는 수신 패킷을 자신의 개인 키를 사용하여 복호화하고 대칭 키를 사용하여 클라이언트 측에서 무작위로 생성된 URL 및 HTTP 데이터를 복호화합니다.

6.그런 다음 대칭 키로 암호화된 다른 데이터와 함께 클라이언트에서 요청한 문서가 브라우저로 다시 전송됩니다.

7.마지막으로 브라우저는 대칭 키를 사용하여 패킷을 해독하고 보안 핸드셰이킹이 설정됩니다.