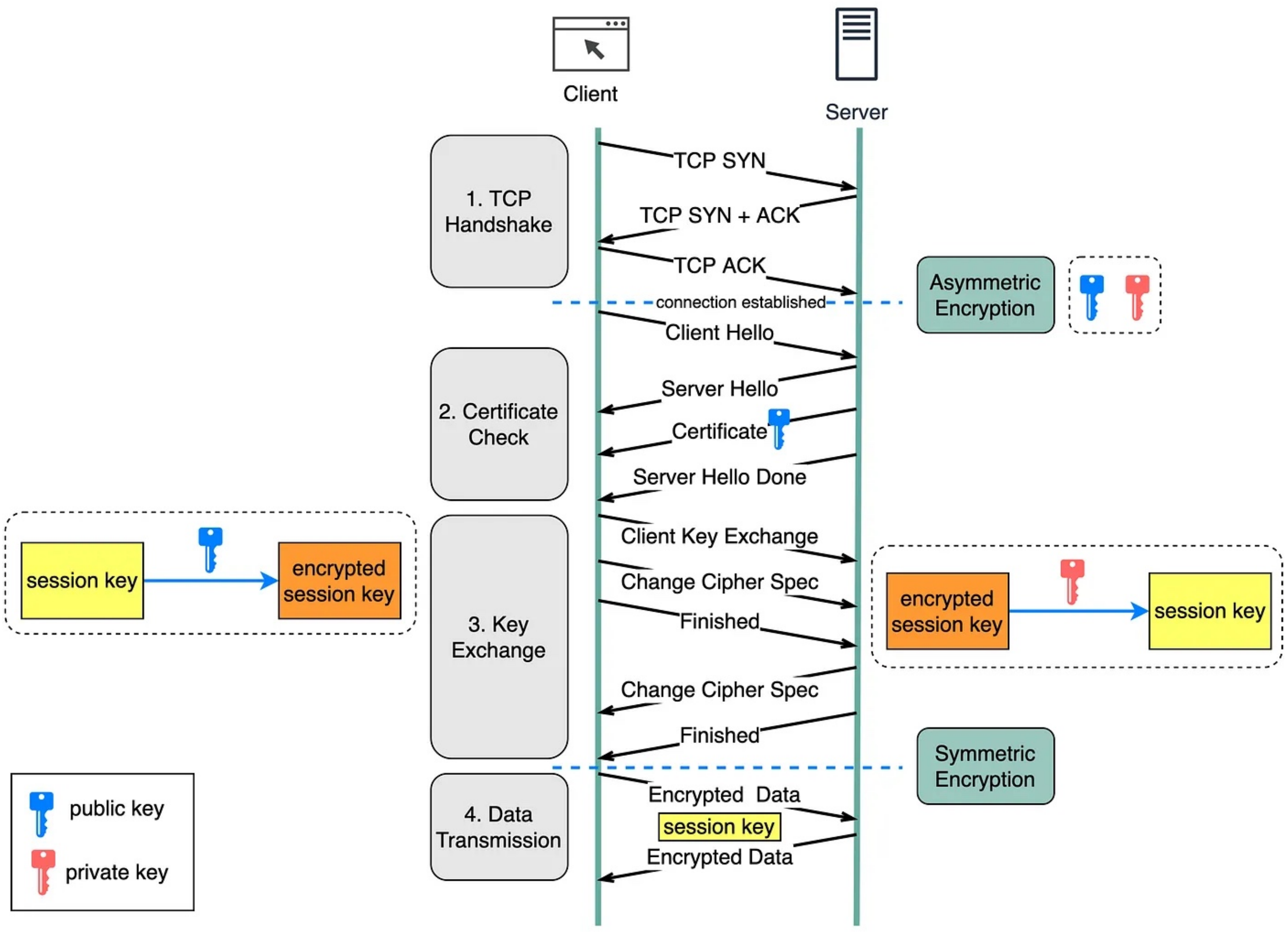
How does HTTPS Work?

blog.bytebytego.com



After the TCP Handshake :

1. Client Hello :- The client proposes cryptographic parameters and sends its random value.
Also sends list of encryption algorithms to choose from.

2. Server Hello :- The server selects cryptographic algorithm and send its random value.

3. Server Certificate :- The server sends its public key

to the client. along with the SSL certificate

4. Server Hello Done :- The server indicates the end of it's Hello phase.

5. Client key Exchange :- After certificate validation, the client generates a session key and encrypts it using server's public key and sends it to the server.

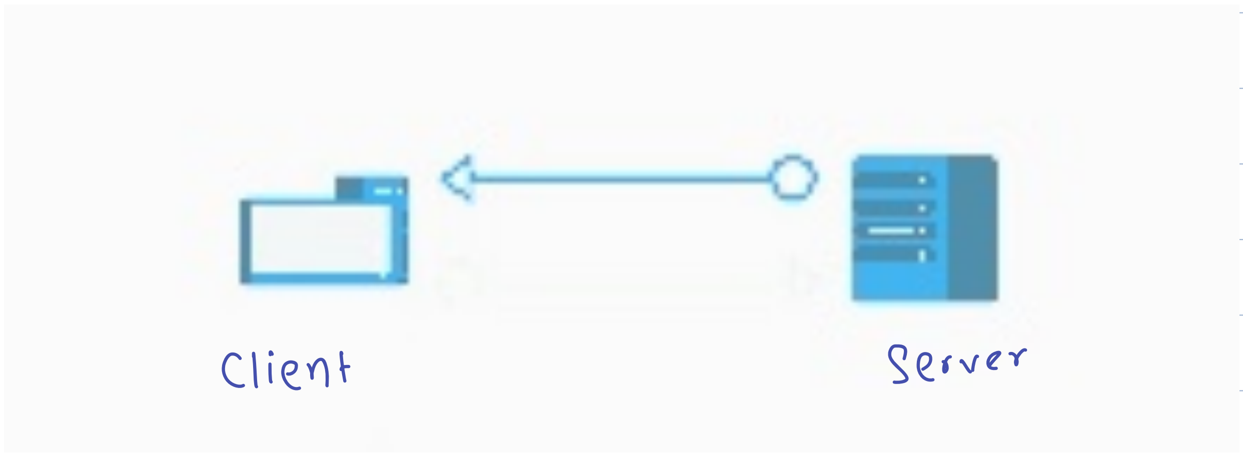
6. Change Cipher Spec :- The client switches to encrypted communication

7. Client Finished :- The Client sends a verification message encrypted with the session key.
Encrypted hash of all the TLS messages.

8 Change Cipher spec :- The server switches to encrypted communication

9. Server Finished :- The server sends a verification message encrypted with the session key.
Encrypted hash of all the TLS messages.

Example of TLS :-



1. Client sends Client Hello message to the server

Client Hello {
Version: TLS 1.2 (0x0303) ⇒ TLS version
ClientRandom: 5AC3A452CFF . . . . . . ⇒ Random no.
SessionID : 00 (If no prev session exist) for any prev session or 00 for new session
Cipher Suites : C02F, 009C
}
↳ gives options to the server to select a cipher suit. Here C02F and 009C are cipher suite options.

Cipher Suites :-
- C02F ⇒ (TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256)
- 009C ⇒ TLS_RSA_WITH_AES128_GCM_SHA256)

2. Server sends Server Hello to Client

Server Hello {
Version : TLS 1.2 (0x0303)
ServerRandom : 123FDE . . . .

SessionID : AB24 The server assigns a session ID
Cipher Suite : 009C select the suite from given options
}

3. Server Certificate (from server to Client)

Certificate {
Certificate Type :: X.509 X.509 certificate
Public key: B4CD 34125 . . . . server's public key which client will use to encrypt the session key.
}

4. Server Hello Done (from server to Client)

ServerHello Done {
(null)
}

5. Client key Exchange (Client to server) ::

Client key Exchange {
Encrypted PreMaster Secret : 9B34 A7 . . . --
}

Client generates a pre-master secret (session key) and encrypts it using Public key of the server. and sends the Encrypted PreMaster Secret to the server.

On the server side, it decrypts the Encrypted PreMaster secret and gets the Session key.

6. Change Cipher Spec (Client to Server)

```
Change Cipher Spec {
        Value : 01
  }
```

The value 01 indicates that client is transitioning to encrypted messages from now on.

7. Client Finished :: (Client to server)

It contains hash of all previous handshake messages to verify handshake was successful.

```
Finished {
        Encrypted Handshake Hash :: FE 23AB . .  . .
    }
```

Encrypted (Hash $\boxed{\text{all msgs}}$) $\rightarrow$

8. Change Cipher Spec and Server finished are same as the above two but from server to client.