Let's Know

What makes HTTPS a secured protocol ?

Story Time

Hi Rita !

I really like that girl in our neighbourhood

No, keep this a secret.. pls

Hi Ram

Should I tell her ?

Ram shared his feelings to Rita, do you think this thing will remain a secret forever ?

Here msgs are sent over Internet.

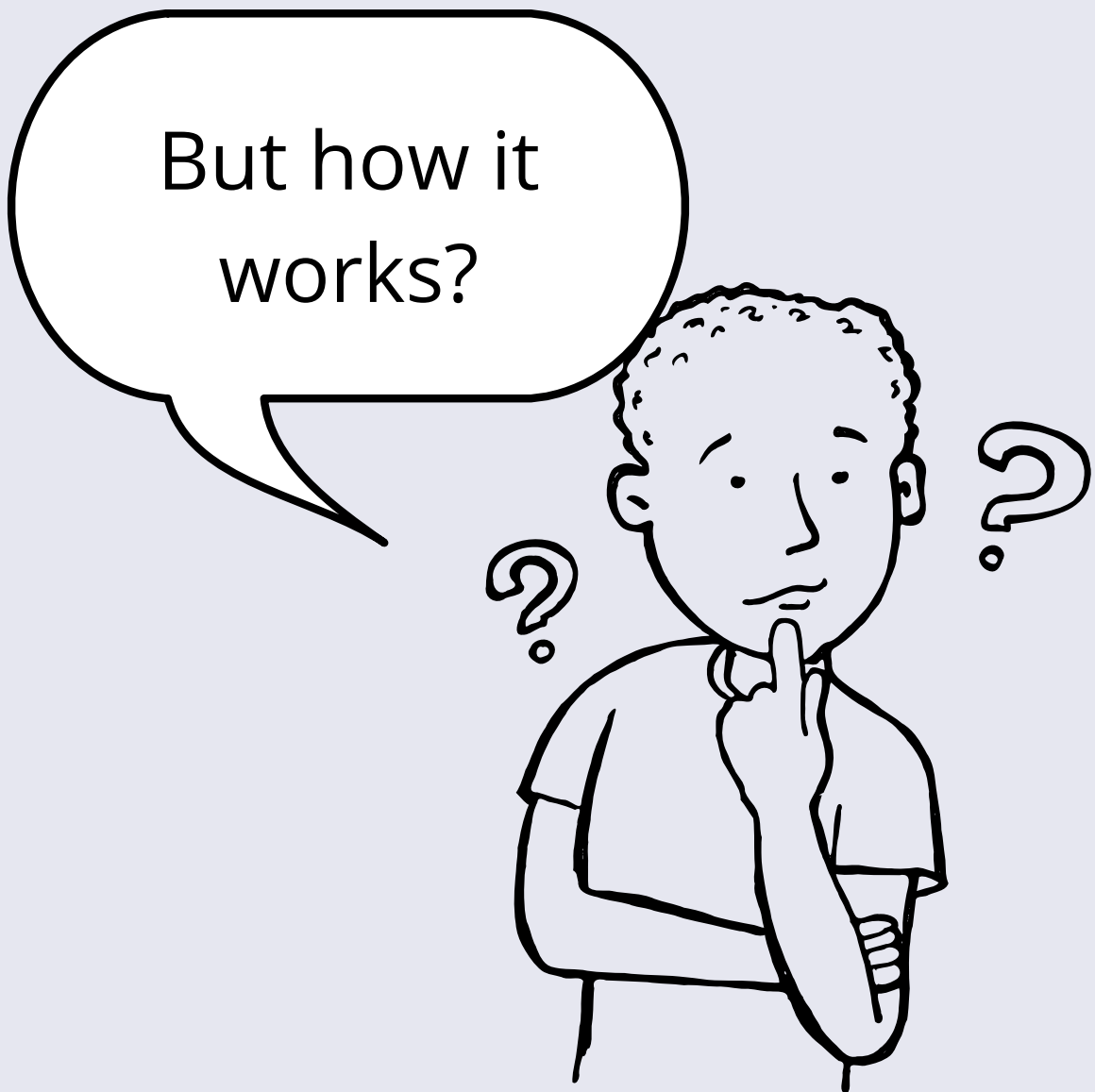
If they are not strongly encrypted, someone can sniff these msgs in b/w and Ram's

Secret is no more secret...

Here comes TLS to rescue

Transport Layer Security Protocol

This protocol will make sure that Ram's msgs are secret and only Rita gets them no-one else



Few facts we should know-

We definitely need some encrypting algorithm.

We have 2 famous ones-

1. RSA

2. Diffie-Hellman Key Exchange

We will go with Diffie-Hellman, for now...

Diffie-Hellman Key Exchange

It is a symmetric cryptography algo. which uses a single key to encrypt and decrypt msgs

But how does Client and Server get a common key.

Client and Server both have their Public and Private Keys.

Deffie-Hellman combines these keys and generate a symmetric key, let's see how ....

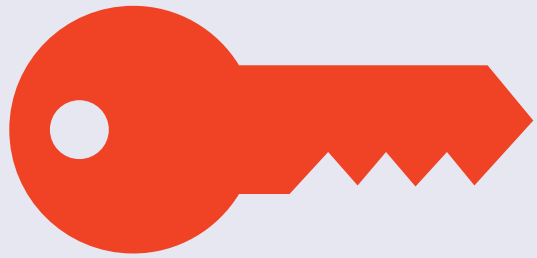
Diffie-Hellman Key Exchange

1 Private Key (Client)

3 Private Key (Server)

2 Public Key (Client)

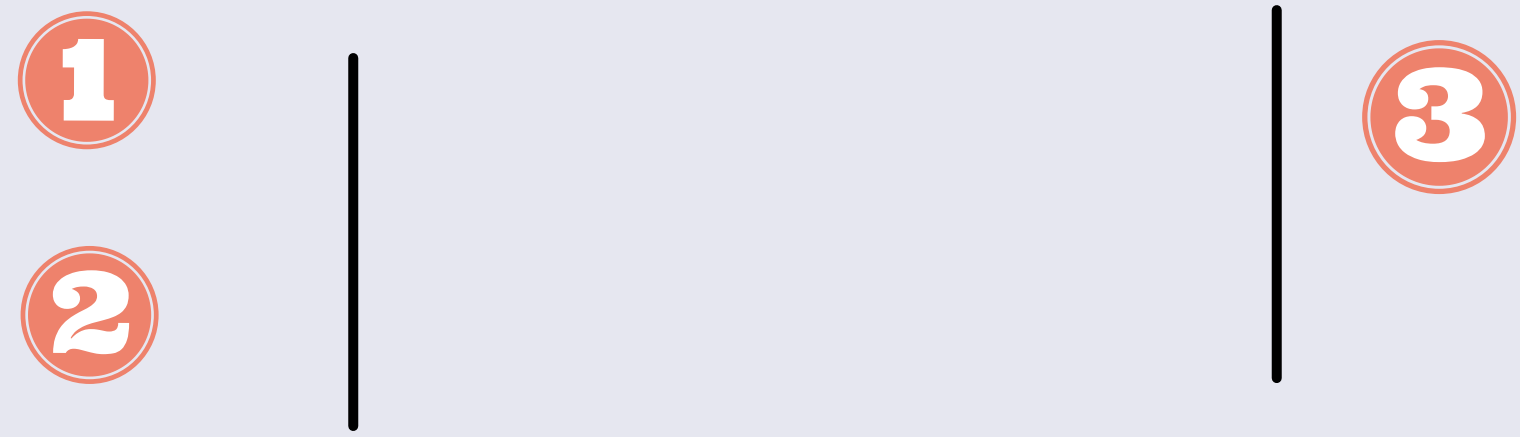
4 Public Key (Server)

1 + 2 + 3 = 

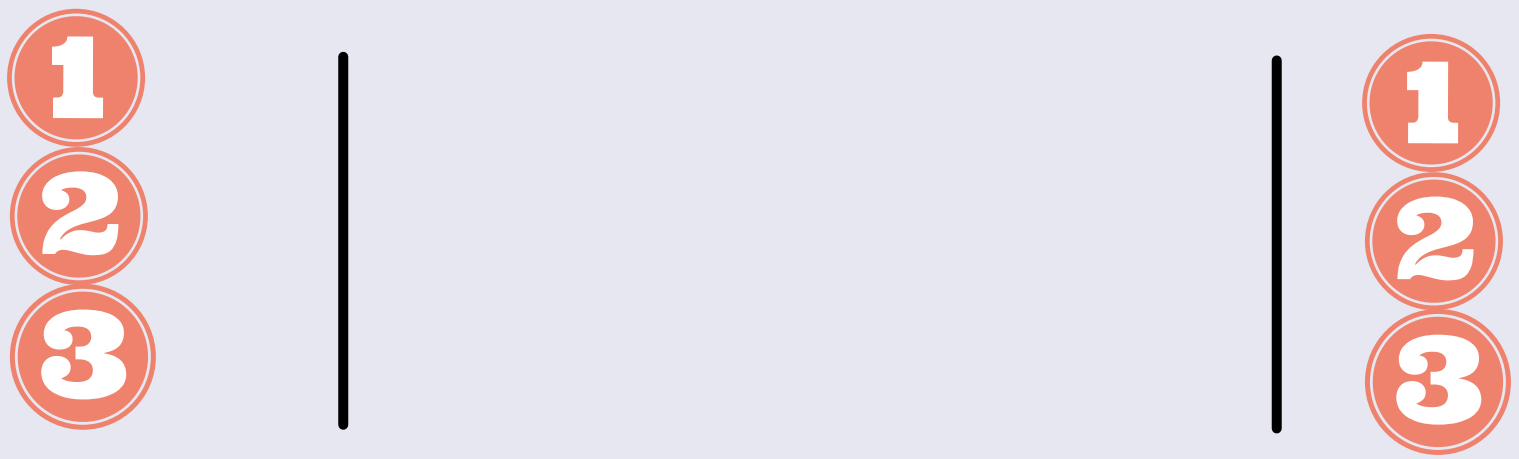
Combining the three keys will yield us our Symmetric Key

Remember the no.s with their corresponding keys

Before TLS Handshake

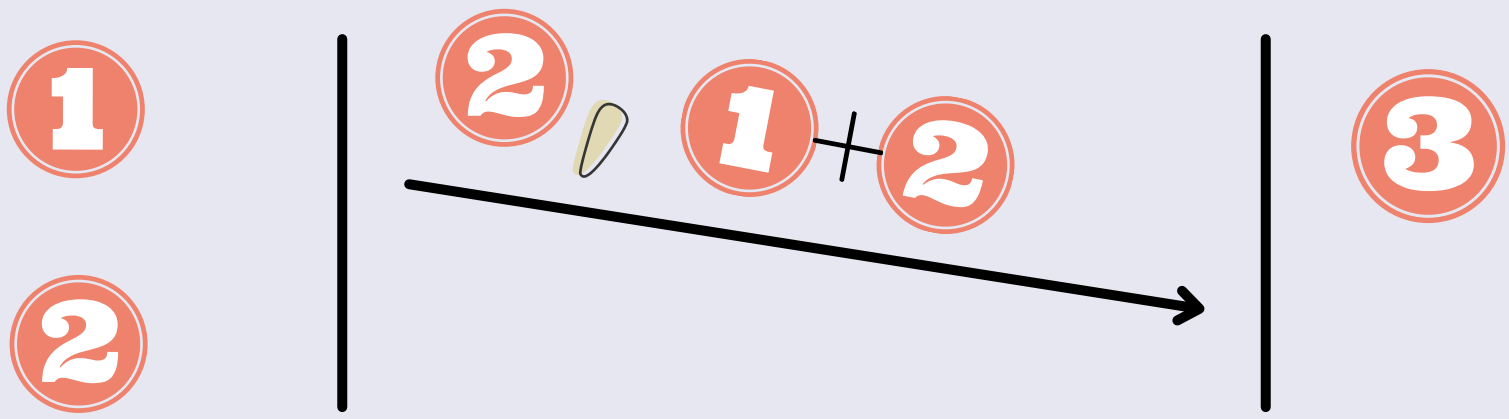


What we want After TLS Handshake



Let's see how key no. 1, 2 and 3 will be sent from client to server and vice - versa

Step-1

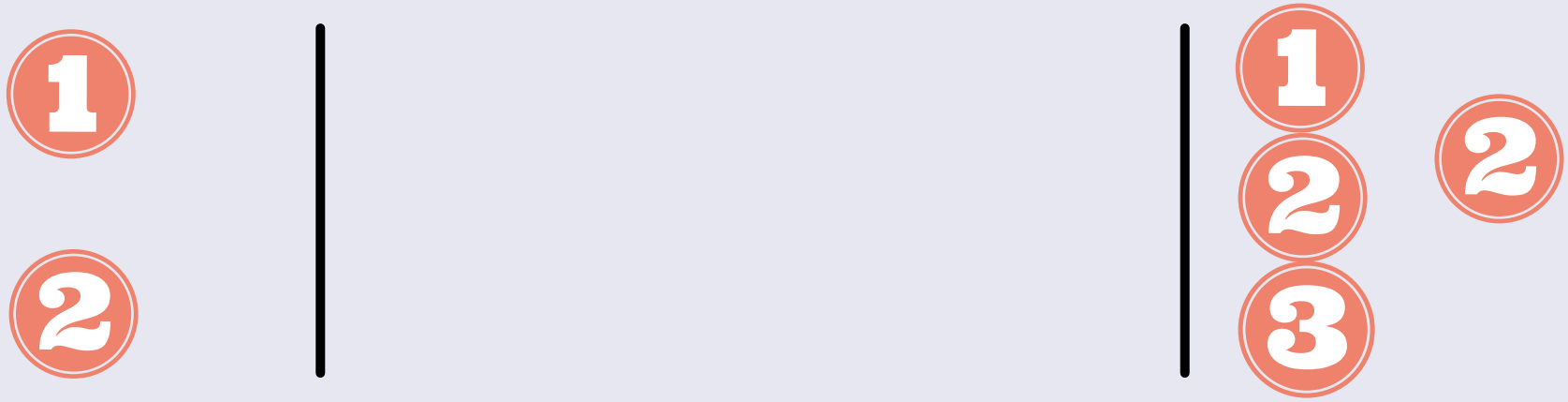


Client sends it's Public Key(2) and a combination of it's Public + Private Key(1+2)

Sending public key over network is fine, but incase of private key he combined that with Public, so it becomes hard to separate out Private Key

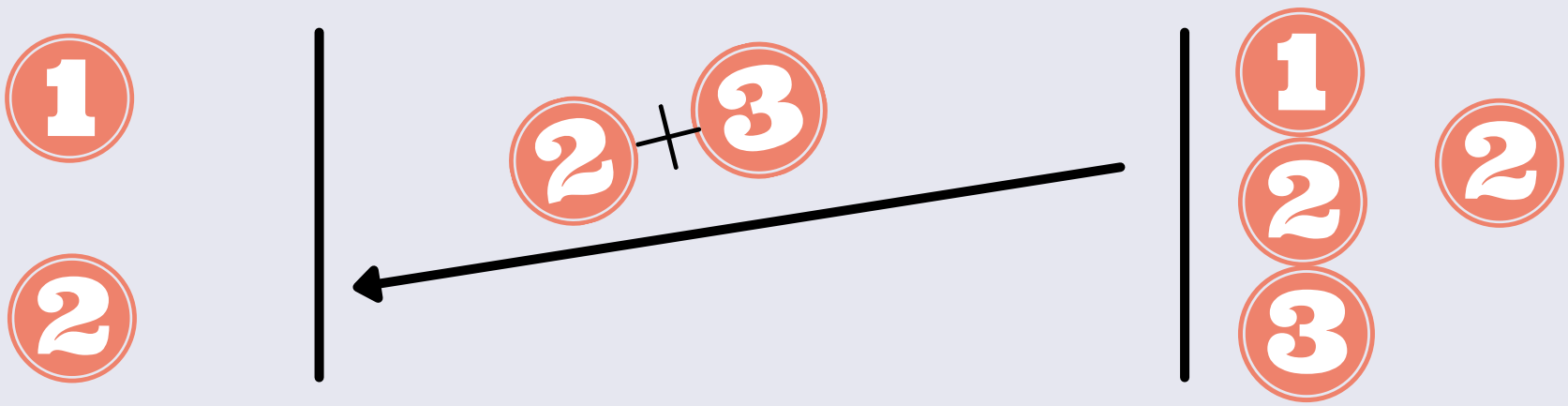
Also, a msg encrypted with a Public key can only be decrypted by the individual Private Key, so sending 1+2, is somewhat safe.

After step-1

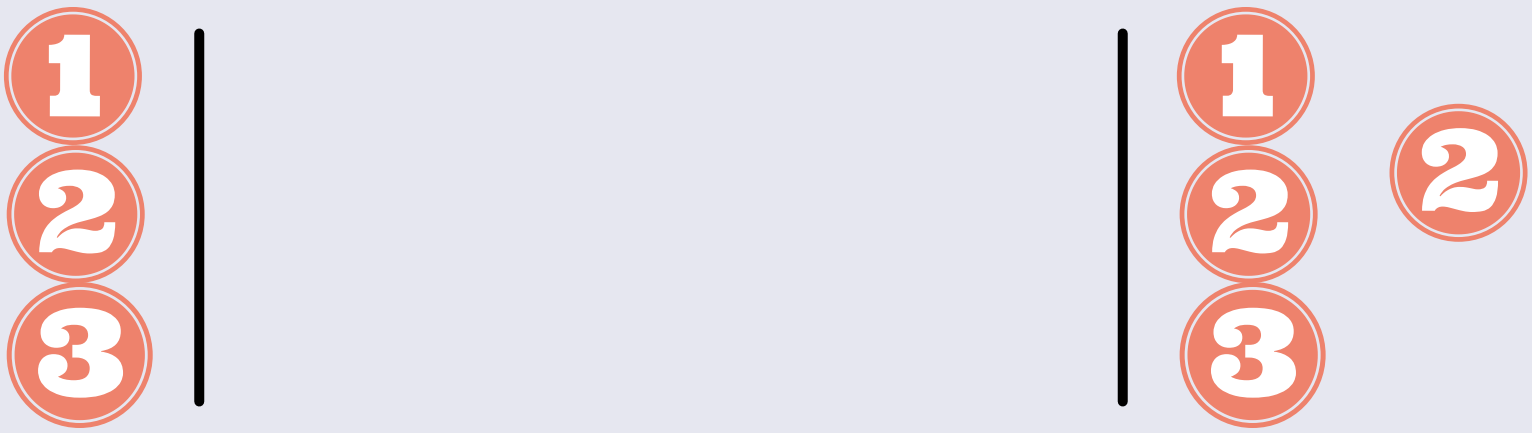


Now server has all 3 Keys, so he can form the Symmetric key, but client needs key no. 3

Step - 2



After step-2



Client

Server

In step 2, server sends it's Private Key(3) encrypted by Public key of Client(2), now Client will decrypt the msg from server and extract out Key 3

Now Client and Server both have Key no. 1, 2 and 3

By combining them they get the Symmetric Key

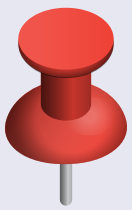
Now this key can be used to encrypt and decrypt the msgs, so the communication is safe now.

TLS Handshake Done

Now we got a safe way to communicate over internet

After TLS Handshake, now the requests and respose can be made securely between Client and Server.

TLS is the main reason why HTTPS is secured.



These Keys are actually big no.s, I have tried to keep things simple here...

There are few other things in between but for introduction it is enough.