TLS/SSL handshake

Securing your Browser

106118056

MEIVENKATKUMAR LN

CSE -B II year

SSL/TLS Handshake Protocol

An Introduction to HTTP

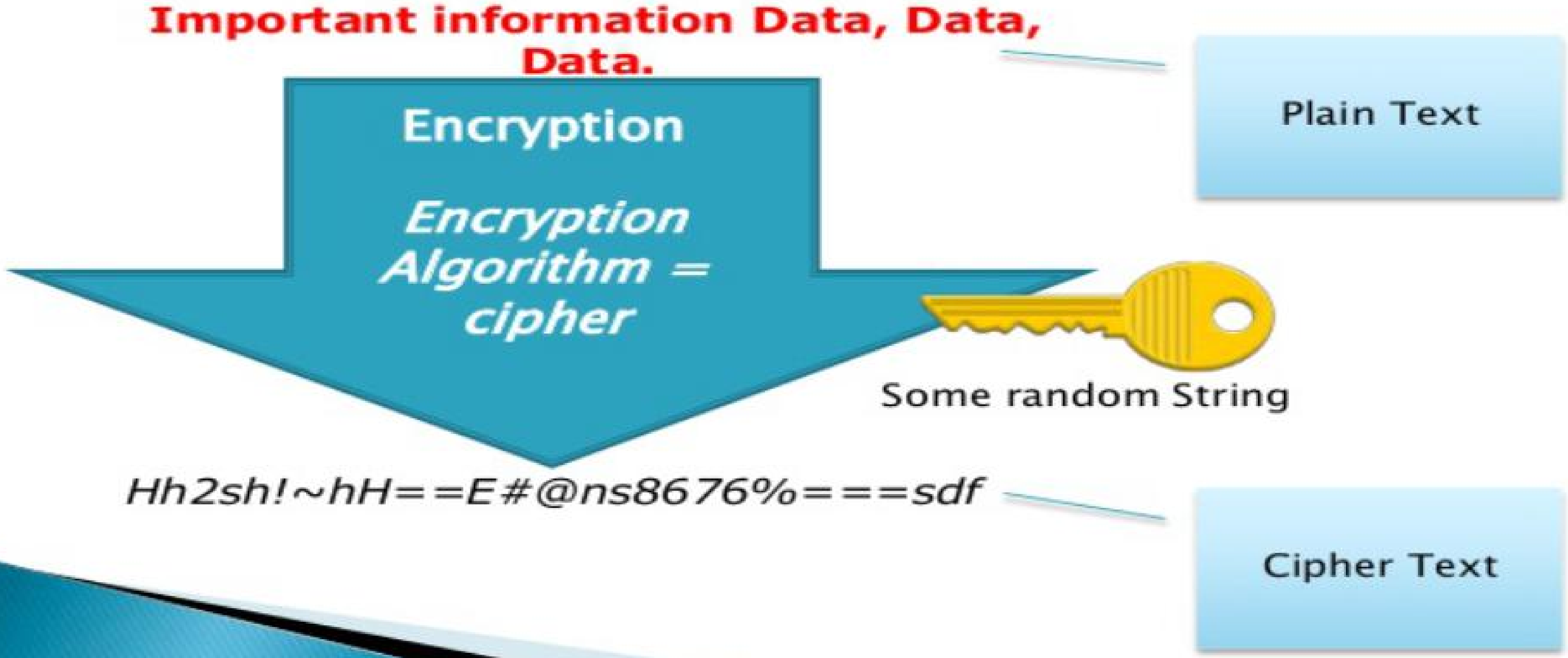
- Hyper Text Transfer Protocol
- One of the application layer protocols that make up the Internet
  - HTTP over TCP/IP
  - Like SMTP, POP, IMAP, NNTP, FTP, etc.
- The underlying language of the Web
- Three versions have been used, two are in common use and have been specified:
  - RFC 1945 HTTP 1.0 (1996)
  - RFC 2616 HTTP 1.1 (1999)

HTTPS

=

HTTP + SSL

Cryptography



Important information Data, Data, Data.

Plain Text

Encryption

Encryption Algorithm = cipher

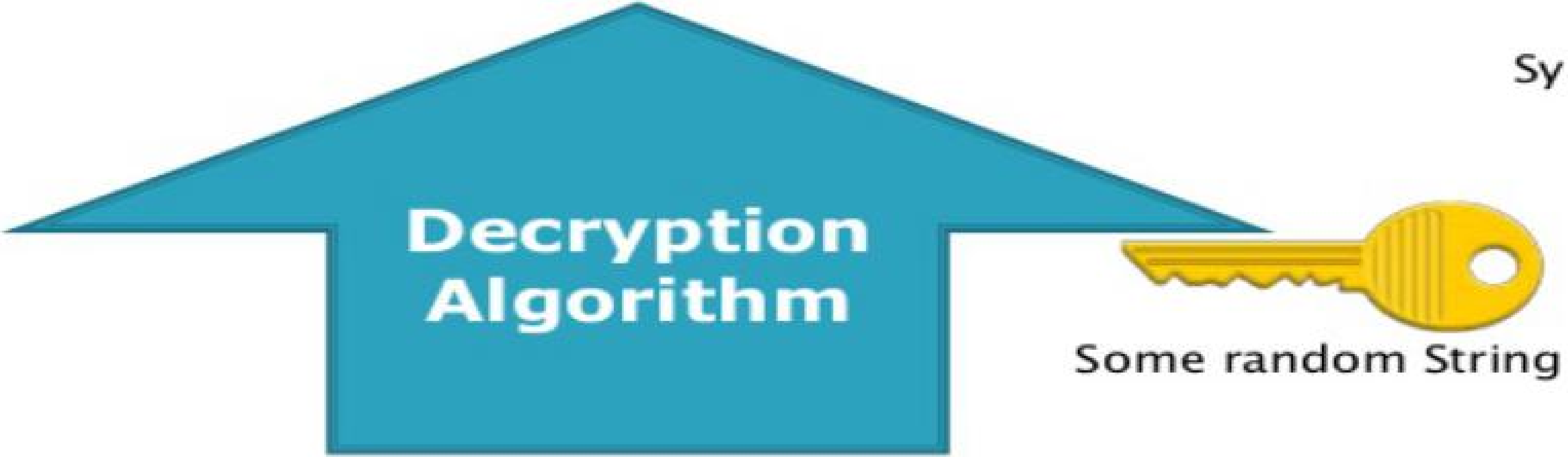
Some random String

Hh2sh!~hH==E#@ns8676%===sdf

Cipher Text

Cryptography cont.

Important information Data, Data, Data.



Decryption Algorithm

Some random String

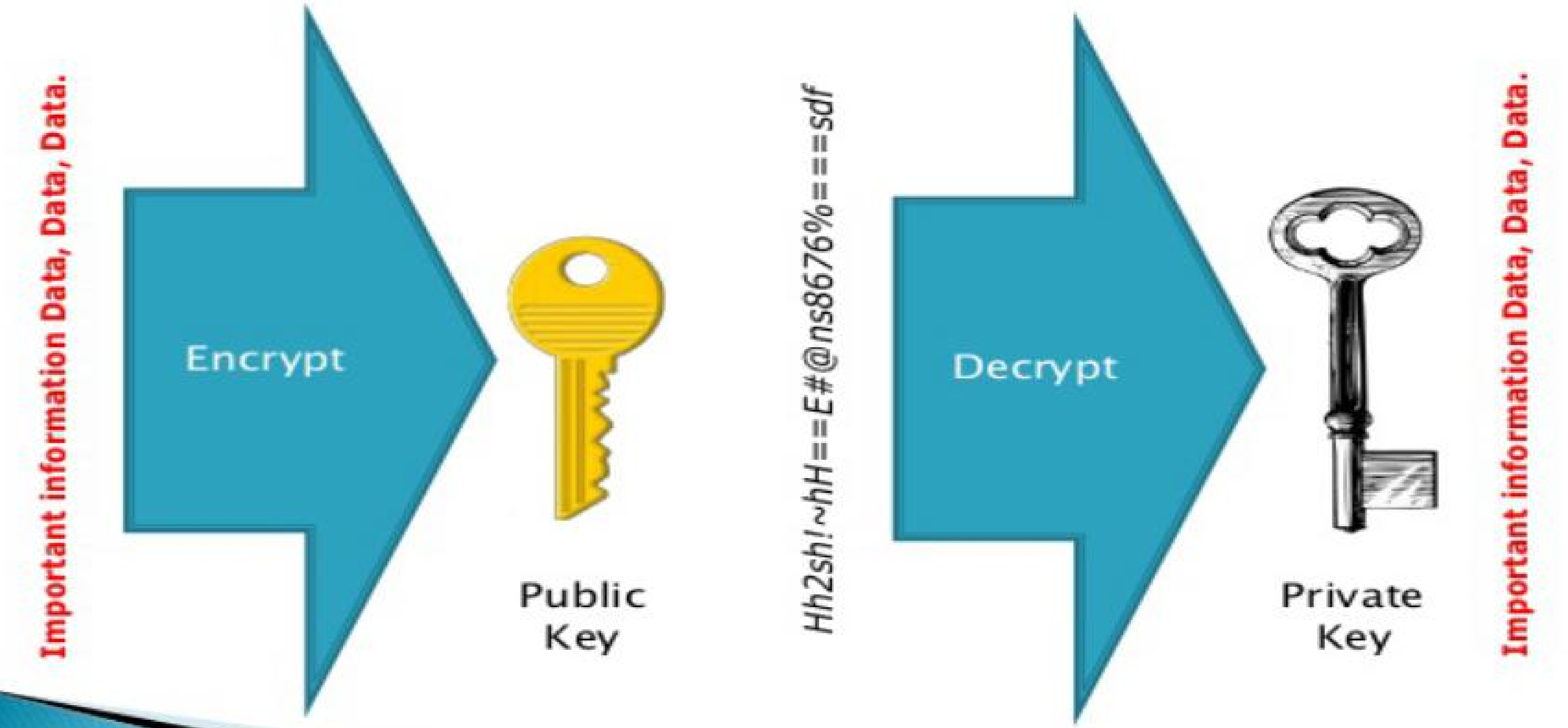
Hh2sh!~hH==E#@ns8676%===sdf

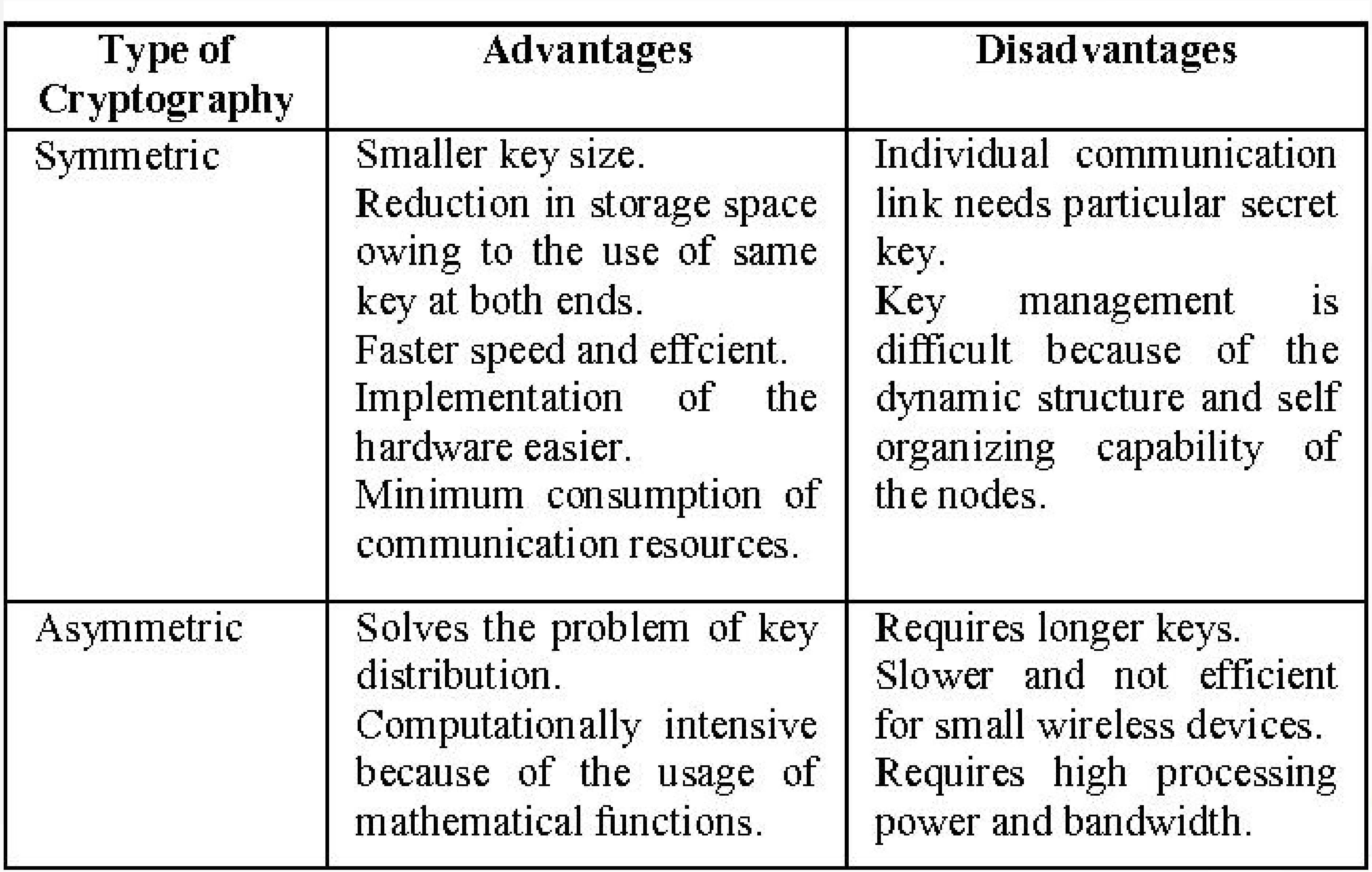
Symmetric Key

byte 00	byte 01	byte 02	byte 03	byte 04	byte 05	byte 06	byte 07	byte 08	byte 09	byte 10	byte 11	byte 12	byte 13	byte 14	byte 15

byte 00	byte 04	byte 08	byte 12
byte 01	byte 05	byte 09	byte 13
byte 02	byte 06	byte 10	byte 14
byte 03	byte 07	byte 11	byte 15

Asymmetric (public-key) encryption


Important information Data, Data, Data.
Encrypt
Public Key
Hh2sh!~hH==E#@ns8676%===sdf
Decrypt
Private Key
Important information Data, Data, Data.



Type of Cryptography	Advantages	Disadvantages
Symmetric	Smaller key size. Reduction in storage space owing to the use of same key at both ends. Faster speed and effcient. Implementation of the hardware easier. Minimum consumption of communication resources.	Individual communication link needs particular secret key. Key management is difficult because of the dynamic structure and self organizing capability of the nodes.
Asymmetric	Solves the problem of key distribution. Computationally intensive because of the usage of mathematical functions.	Requires longer keys. Slower and not efficient for small wireless devices. Requires high processing power and bandwidth.

SSL/TLS Handshake Protocol

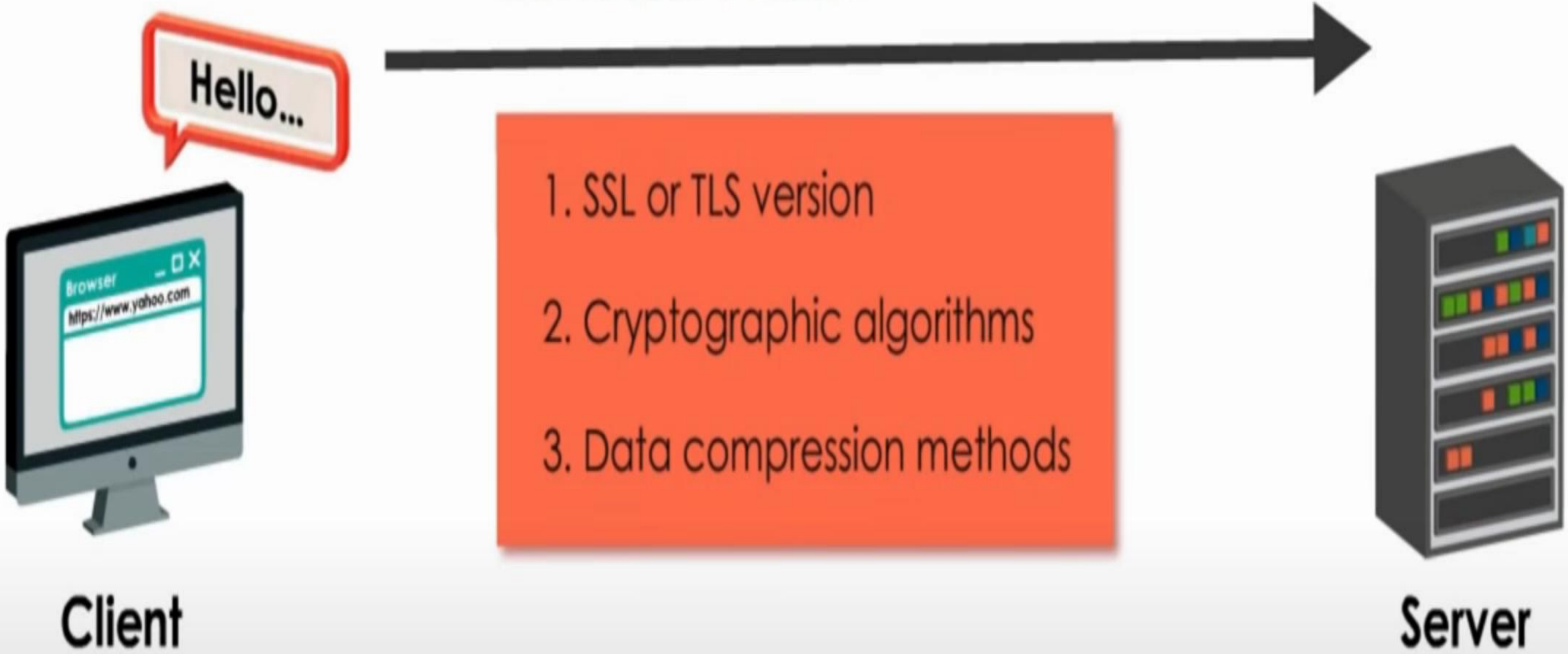
SSL Session

- Uses asymmetric encryption to privately share the session key
  - Asymmetric has a lot of overhead

- Uses symmetric encryption to encrypt data
  - Symmetric encryption is quicker and uses less resource

SSL/TLS Handshake Protocol

ClientHello



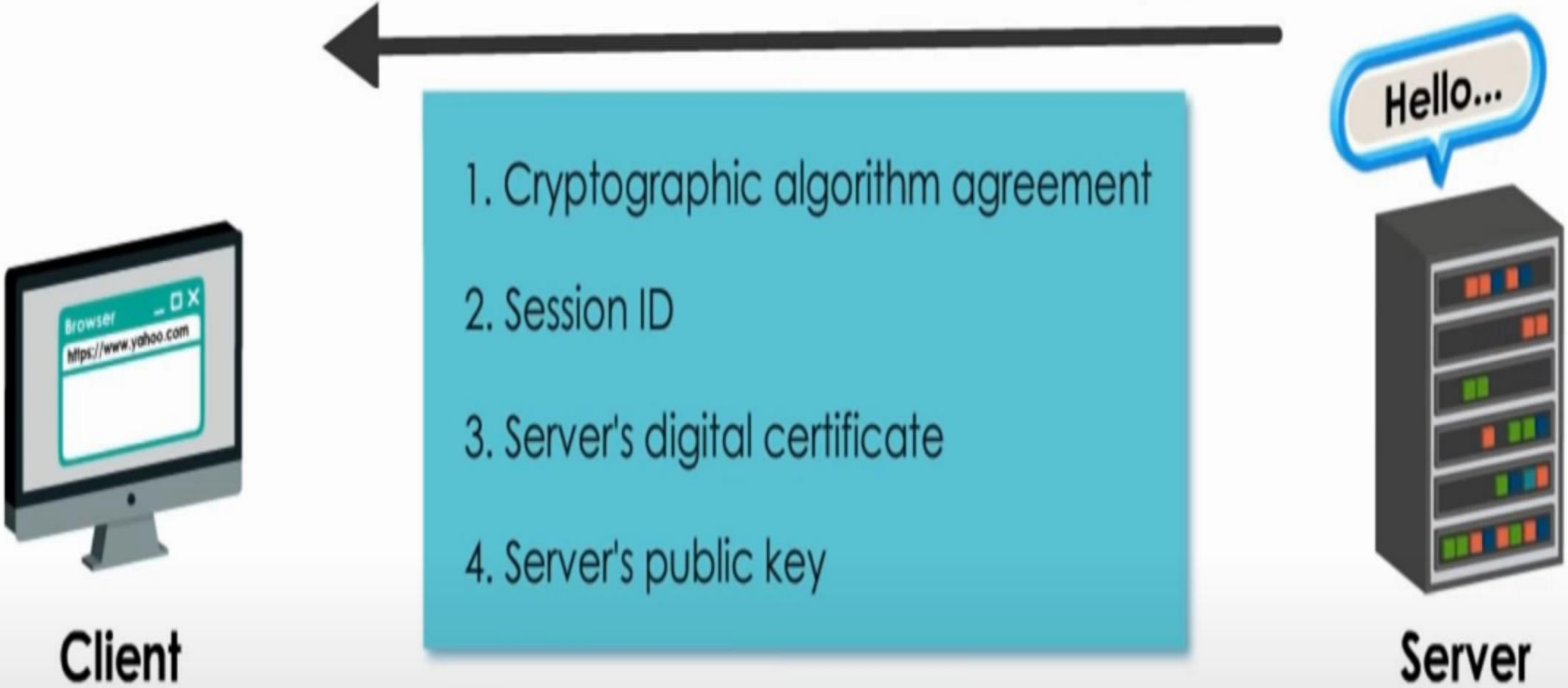
1. SSL or TLS version

2. Cryptographic algorithms

3. Data compression methods

```
*** ClientHello, TLSv1.2
RandomCookie: *** ClientHello, TLSv1.2
RandomCookie: GMT: -1892413556 bytes = { GMT: -351008774 bytes = { 169, 131, 204, 213,
154, 96, 7, 136, 43, 142, 232, 138, 148, 171, 52, 226, 155, 202, 145, 57, 210, 132, 227, 182,
67, 222, 161, 28, 20 }
Session ID: 239, 10, 92, 143, 185, {}
93, Cipher Suites: [Unknown 0x8a:0x8a, TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384, Unknown 0xcc:0xa9, Unknown 0xcc:0xa8,
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA, TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,
TLS_RSA_WITH_AES_128_GCM_SHA256, TLS_RSA_WITH_AES_256_GCM_SHA384,
TLS_RSA_WITH_AES_128_CBC_SHA, TLS_RSA_WITH_AES_256_CBC_SHA,
SSL_RSA_WITH_3DES_EDE_CBC_SHA]
.....................................................
```

ServerHello

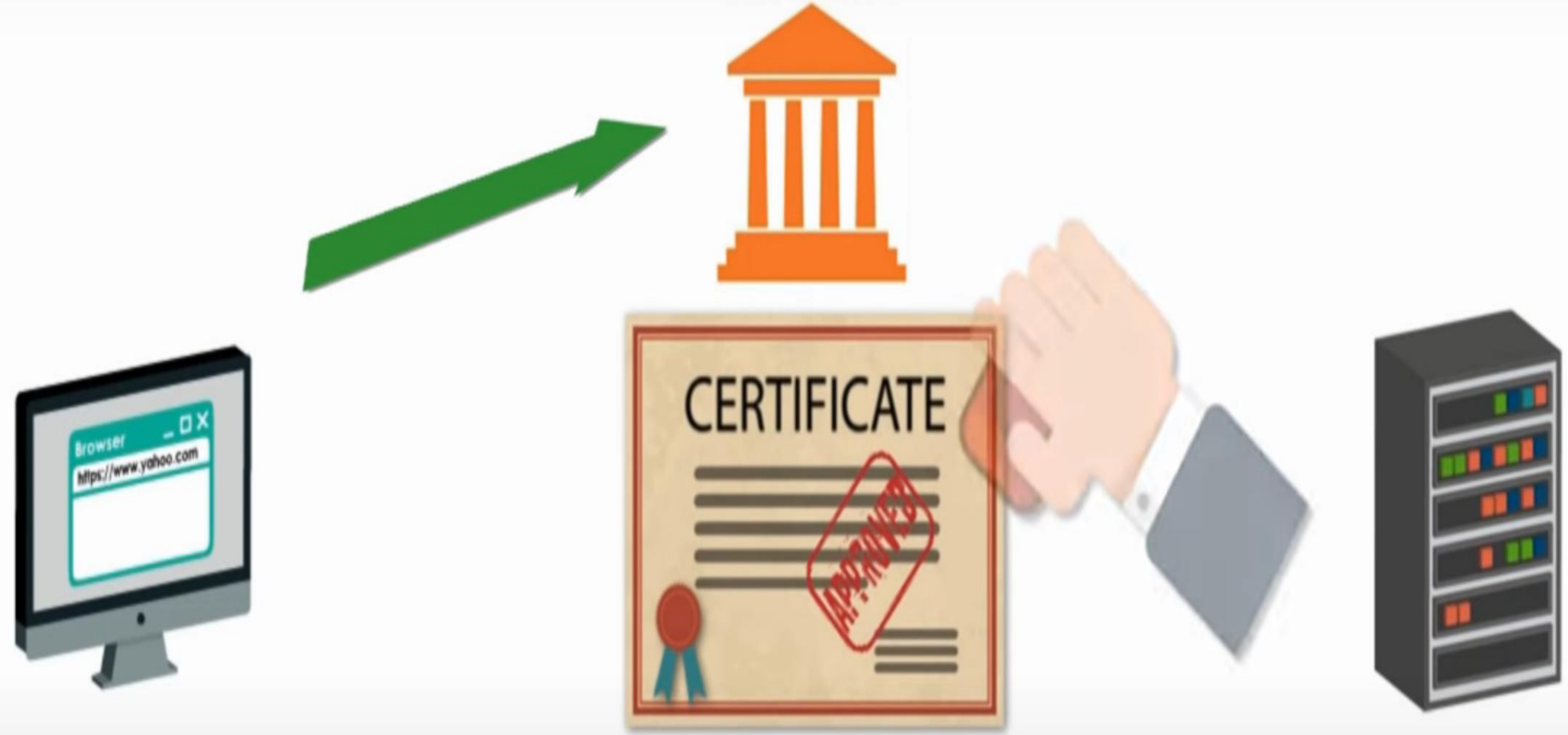


1. Cryptographic algorithm agreement
2. Session ID
3. Server's digital certificate
4. Server's public key

```
*** ServerHello, TLSv1.2
RandomCookie: GMT: 1518335451 bytes = { 19, 150, 56, 42, 168, 202, 151, 43,
174, 226, 187, 53, 135, 67, 244, 170, 59, 176, 105, 150, 50, 112, 167, 83, 192, 48,
171, 64 }
Session ID: {91, 128, 246, 219, 26, 93, 46, 172, 85, 212, 221, 79, 20, 186, 108, 134,
200, 239, 150, 102, 172, 24, 125, 171, 137, 53, 5, 130, 53, 228, 2, 195}
Cipher Suite: TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
Compression Method: 0
Extension renegotiation_info, renegotiated_connection: <empty>
***
```

It also contains a Digital Certificate.

Certificate Authority(CA)

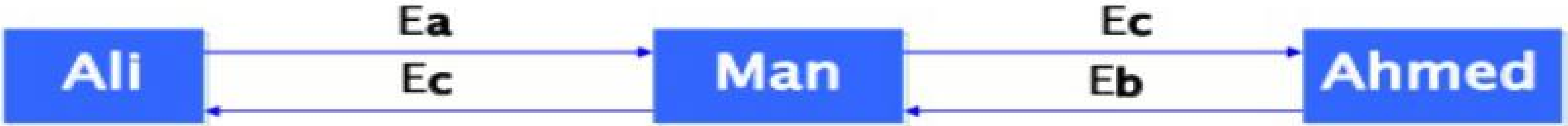


Client

Server

Man–in–the–Middle (MITM) Attack Concept

- There were away to get around the encryption instead o0f trying to break it



E{a,b,c} = Ali's, Ahmed's, and Man's public keys, respectively

- Ali wants to send secure messages to Ahmed.
- Man intercepts Ali’s messages.
- Man talks to Ali and pretends to be Ahmed.
- Man talks to Ahmed and pretends to be Ali.

MITM Attack Concept

- Ali uses the *public key* she thinks she received from Ahmed (Man’s)

- Ahmed uses the key he thinks is Ali’s (also Man’s)

- As a result, Man not only gains *access* to secure information but also can *modify* it (e.g. *transfer money to a different account* etc.)

MITM and Certificates

- Digital Certificates designed to solve the problem but do they always help ?

- The MITM would have to create his own certificate with a private/public key.

- He still sit between client and server, acting as server to the client and client to the server, listening in on everything sent between the two.

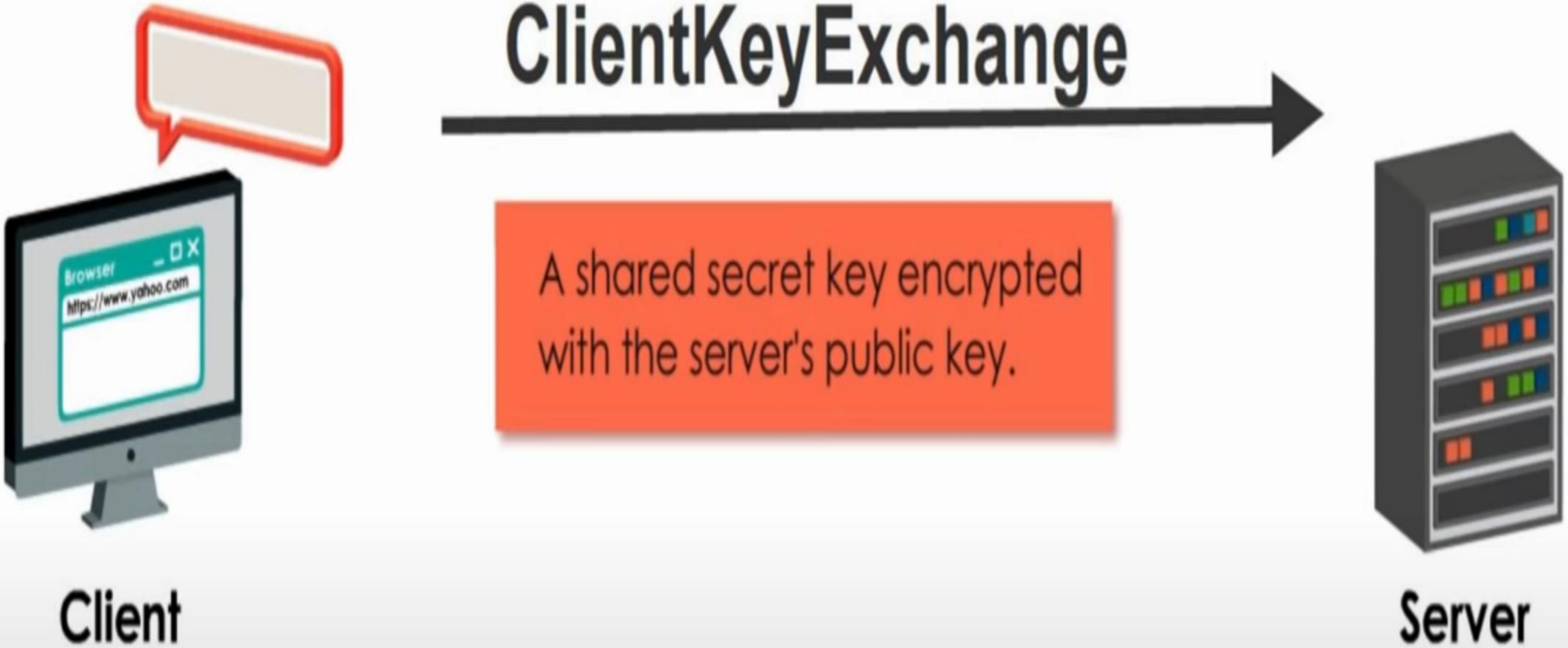
The solution “chain of trust”

- To verify the authenticity and identity of the certificates themselves.
- linked back to a trustworthy source of certificates.
- Web browsers and operating systems will only trust certificates that directly or indirectly link back to one of a handful of CAs, the "root CAs.‟
- Any certificate that doesn't link back to a root CA such as a self-signed certificate will generate a big scary warning in the browser.
  - **How to create a self-signed SSL Certificate ...**
  - http://www.akadia.com/services/ssh_test_certificate.html

Digital Certificate Contents

Version
Certificate Serial Number
Signature Algorithm Identifier
Issuer Name
Validity (Not Before / Not After)
Subject Name
Subject Public Key Information
Issuer Unique Identifier
Subject Unique Identifier
Extensions
Certification Authority's Digital Signature

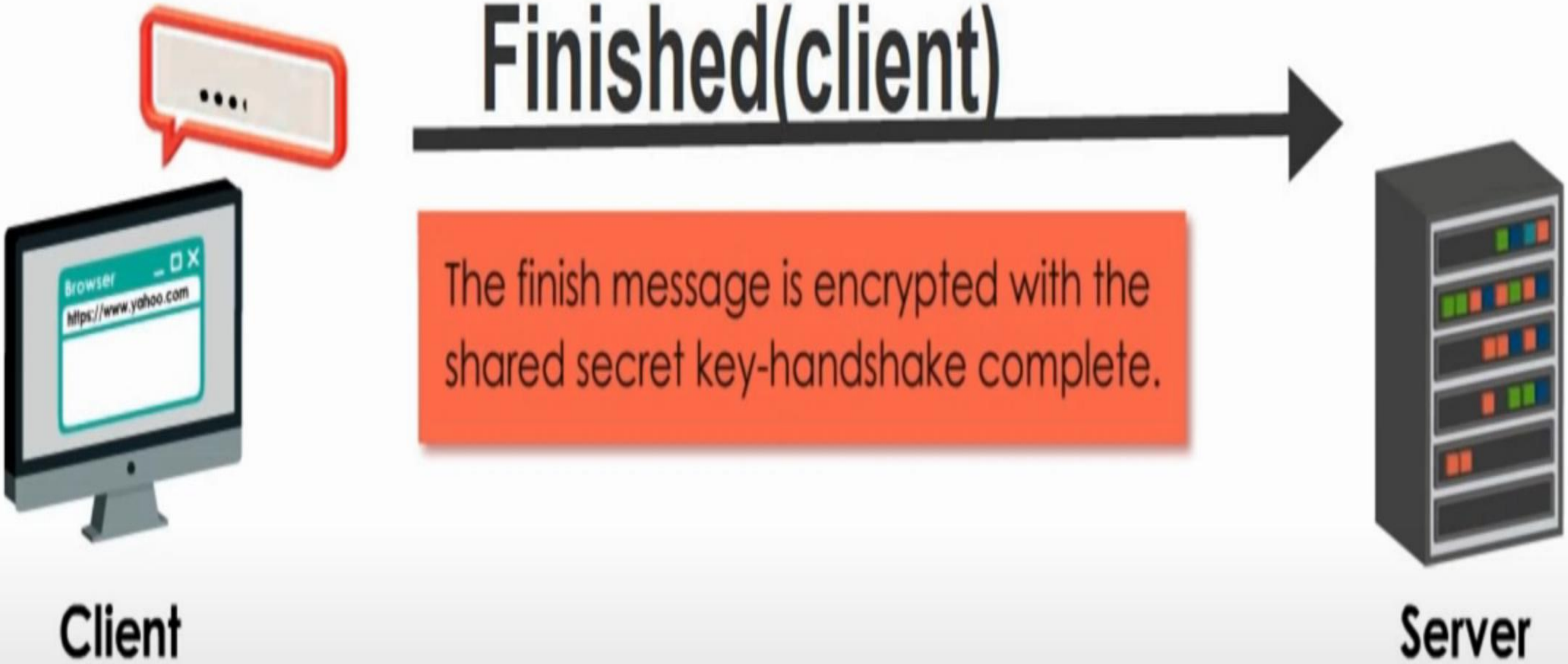
ClientKeyExchange



Client

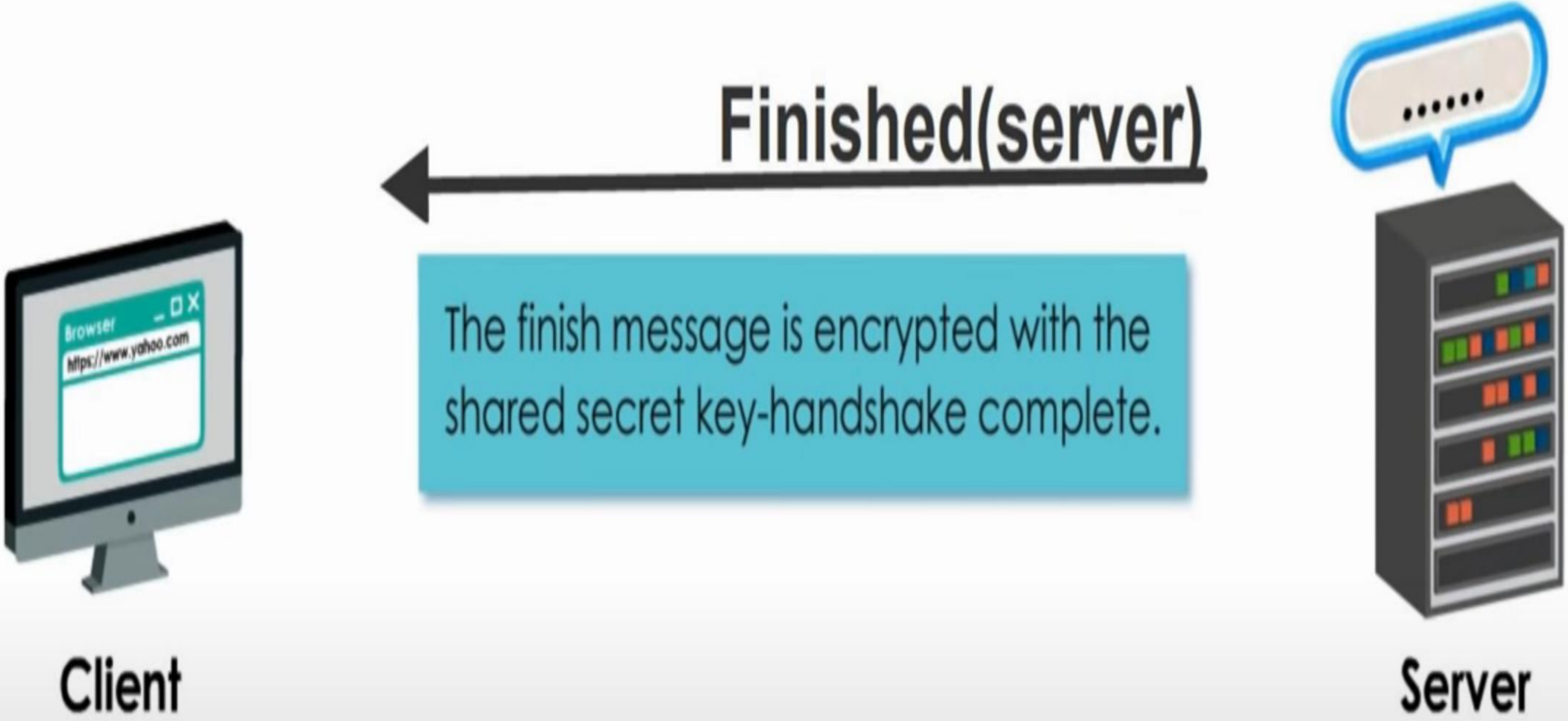
Server

A shared secret key encrypted with the server's public key.



Finished(client)

The finish message is encrypted with the shared secret key-handshake complete.

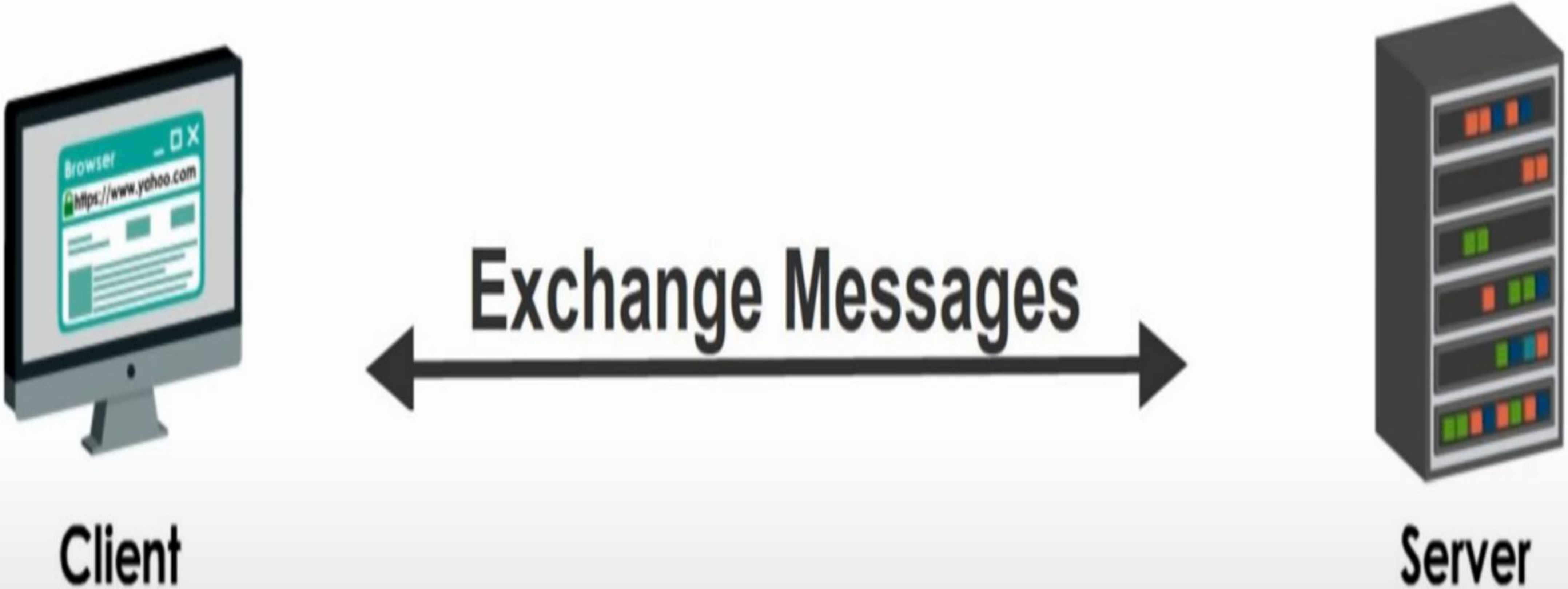


Finished(server)

The finish message is encrypted with the shared secret key-handshake complete.

Client

Server



Exchange Messages

Client

Server

Thank You