
Hackers breached Quora’s network and accessed sensitive personal data ~ 100 million users:

- cryptographically protected passwords
- full names
- email addresses
- data imported from linked networks
- non-public content and action (direct messages, answer requests, downvotes, and others)

“encrypted password (hashed using bcrypt with a salt that varies for each user)” (https://blog.quora.com/Quora-Security-Update )



Hackers breached Quora's network and accessed sensitive personal data ~ 100 million users:

- cryptographically protected passwords
- full names
- email addresses
- data imported from linked networks
- non-public content and action (direct messages, answer requests, downvotes, and others)

"encrypted password (hashed using bcrypt with a salt that varies for each user)" (https://blog.quora.com/Quora-Security-Update )

## It's all about the hash function

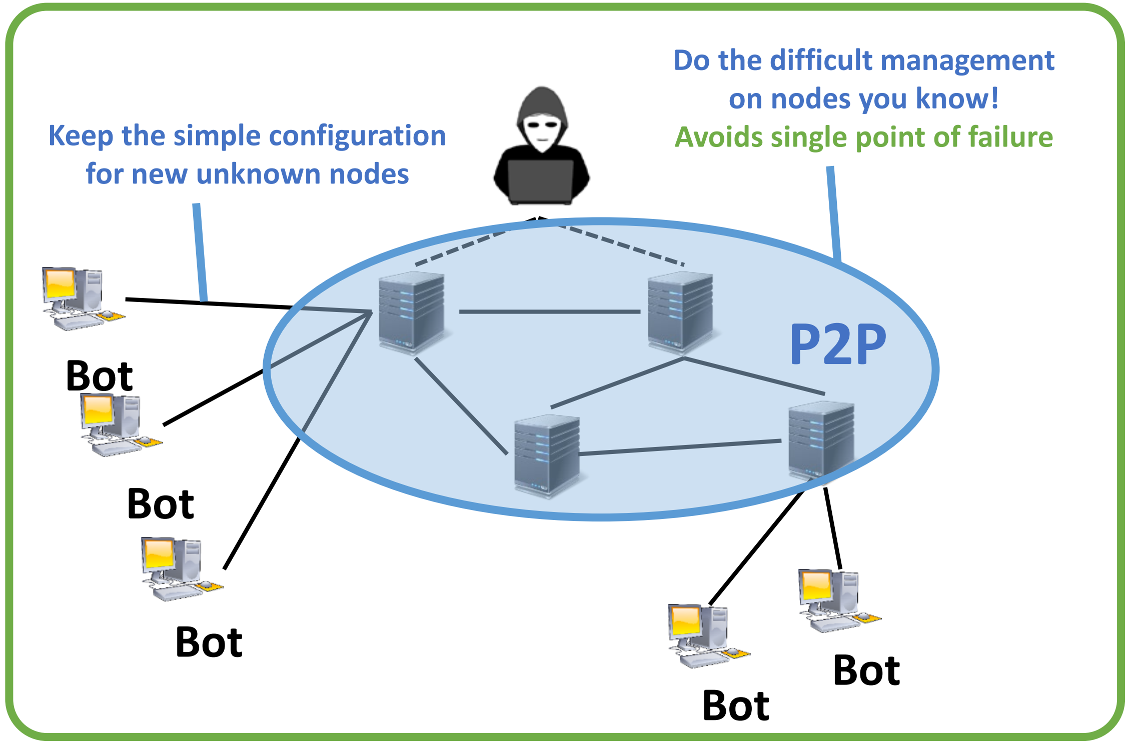
https://arstechnica.com/information-technology/2018/12/quora-says-hackers-stole-password-data-and-other-details-for-100-million-users/

# Last week – Botnets


Multiple (millions) compromised **hosts** under the control of a single entity

**"zombies" or "bots"**

uses




## Bot-net command & control (C&C)

System to keep track and send commands to bots

**Defenses**
- Attack C&C infrastructure
- Use honeypots to learn

# Last week – Desired properties

**Confidentiality, Integrity, Availability, Authentication, Authorization?**

| | |
|---|---|
| **Naming security:** The association between lower level names (eg. network addresses) and higher level names (e.g. Alice / Bob) must not be influenced by the adversary | **Integrity**<br>**Authentication**<br>**Availability (naming service)** |
| **Routing security:** The route over the network and the eventual delivery of messages must not be influenced by the adversary | **Integrity**<br>**Authentication**<br>**Availability**<br>**Authorization** |
| **Session security:** Messages within the same session, cannot be modified (keep ordering and no adding/removing messages) | **Integrity**<br>**Authentication** |
| **Content security:** The content of the messages must not be readable or influenced by adversaries | **Confidentiality**<br>**Integrity** |

# Last week - ARP spoofing

**ARP**: Mapping associations IP – MAC in the LAN network

```
*-------------------------*
| HTYPE (2 bytes)         |
| PTYPE (2 bytes)         |
| HLEN (1)     | PLEN (1) |
| OPERATION (2)           |
| Sender HA (HLEN)        |
| Sender PA (PLEN)        |
| Target HA (HLEN)        |
| Target PA (PLEN)        |
*-------------------------*
```

**No Integrity check, nor Authentication**

**Implications**

- **Impersonation**
- **Man in the middle**: provide two hosts (sender/receiver) with your MAC address
    - Monitor communication or tamper with it
- **Abuse resource allocation**
- **Denial of Service**: avoid that packets arrive to one host

**Defenses**
- Static associations for critical services
- Cross check (separation of privilege)

# Last week – DNS Spoofing

**DNS**: Mapping associations IP – domain in a WAN network

**DNS Spoofing**

**Cache poisoning**: corrupt the DNS resolver with fake pairs (IP,domain)

**DNS Hijacking**: corrupt the DNS responses (man in the middle) with fake pairs

**What can you achieve?**

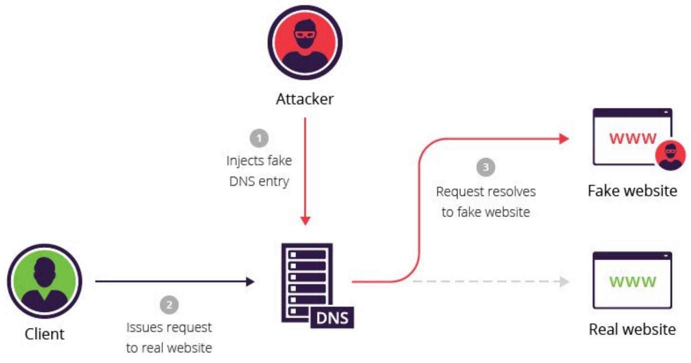
- **Denial of Service**: avoid that packets arrive to one host
- **Redirection**: reroute clients to malicious host
  - Malicious host attacks client (e.g., serving malware…)
  - Malicious host act as man in the middle (e.g., monitoring)

**Defenses**

- DNSSEC: signed responses. Authenticity and integrity without confidentiality
- DNS-over-HTTPS: integrity, authenticity and confidetiality
- DNSCrypt, DNSCurve

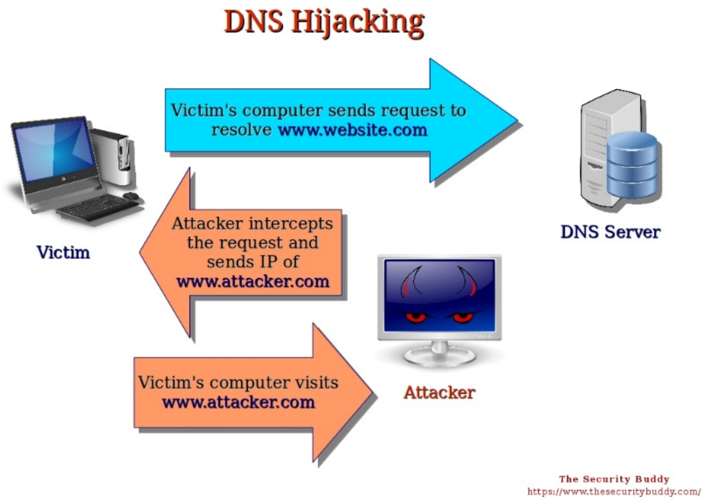
# Last week – DNS Spoofing

**Cache poisoning**: corrupt the DNS resolver with fake pairs (IP,domain)




https://www.incapsula.com/web-application-security/dns-spoofing.html

**DNS Hijacking**: corrupt the DNS responses (man in the middle) with fake pairs



# Last week – BGP Hijacking

**BGP**: Announcement of routes on the internet

**BGP Hijacking**: An adversary controls or compromises a router *somewhere* on the Internet, injects false low-cost routes to redirect portions of traffic to themselves.

**What can you achieve?**

- **Redirection**: surveillance, injection, modification, or censorship.

# Spoofing: lesson to be learned

## The network is hostile!

**Threat model**: assumes network “insiders” are trusted to provide authoritative information.

Also most Internet protocols are designed with **no integrity or confidentiality**.

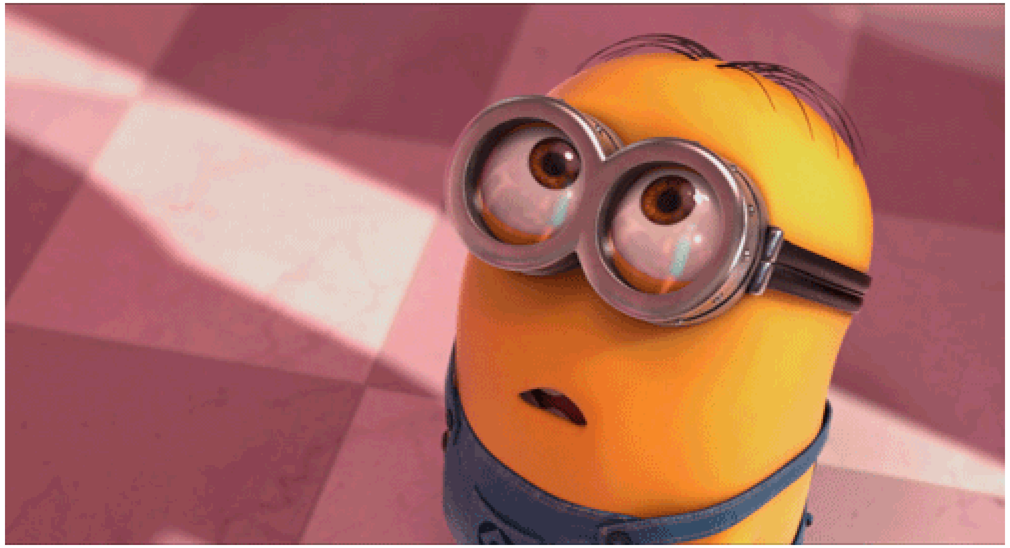
## The solution is intimately linked to cryptography

There is **no centralized authority** to act as either (a) originator of policy or (b) provide a trusted computing base

## But also… who has authority?

Not a cryptographic question! related to name resolution & security policy

# New content!!



# So what about IP?

Refresher

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version|  IHL  |Type of Service|          Total Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Identification        |Flags|      Fragment Offset    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Time to Live |    Protocol   |         Header Checksum       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Source Address                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination Address                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Packet Data                          |
                Example Internet Datagram Header
```

# So what about IP?

Refresher

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |Version|  IHL  |Type of Service|          Total Length         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |         Identification        |Flags|      Fragment Offset    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  Time to Live |    Protocol   |         Header Checksum       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                       Source Address                          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                    Destination Address                        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                    Options                    |    Padding    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                          Packet Data                          |
                    Example Internet Datagram Header
```

**No Integrity check, nor Authentication → Spoofing is possible**

# So what about IP?
# IPSec - Internet Protocol Security

- Cryptographic security properties at the IP level
  - Key exchange based on public key cryptography or shared symmetric keys
  - **Authentication Header (AH)**: authentication & integrity (HMAC), protection from replay attacks (sequence number)
  - **Encapsulating Security Payload (ESP)**: confidentiality

# So what about IP?
# IPSec - Internet Protocol Security

- Cryptographic security properties at the IP level
  - Key exchange based on public key cryptography or shared symmetric keys
  - **Authentication Header (AH)**: authentication & integrity (HMAC), protection from replay attacks (sequence number)
  - **Encapsulating Security Payload (ESP)**: confidentiality

- Two modes:
  - **Transport**:

    protects IP packet payload using AH/ESP

    sent with the **original IP headers**
  - **Tunnel**:

    protects the whole packet (Headers + Payload) is protected and placed inside another packet

# So what about IP?
# IPSec - Internet Protocol Security

- Cryptographic security properties at the IP level
  - Key exchange based on public key cryptography or shared symmetric keys
  - **Authentication Header (AH)**: authentication & integrity (HMAC), protection from replay attacks (sequence number)
  - **Encapsulating Security Payload (ESP)**: confidentiality

- Two modes:
  - **Transport**:

    protects IP packet payload using AH/ESP

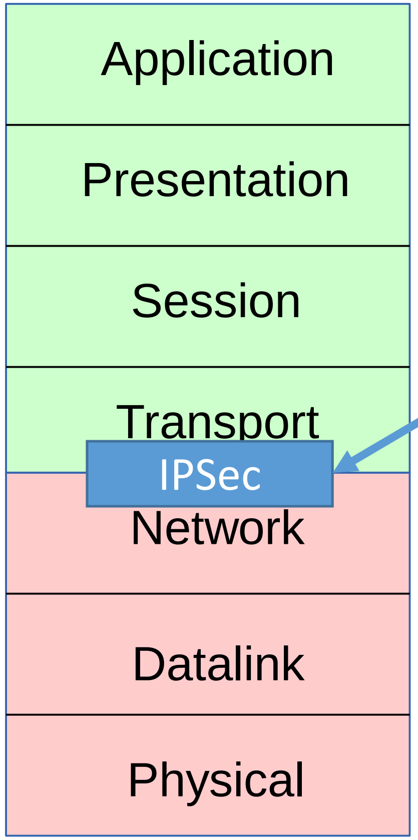
    sent with the **original IP headers**
  - **Tunnel**:

    protects the whole packet (Headers + Payload) is protected and placed inside another packet

**Weak deployment**

**..but mandatory in IPv6**

# Where does IPSec happen?

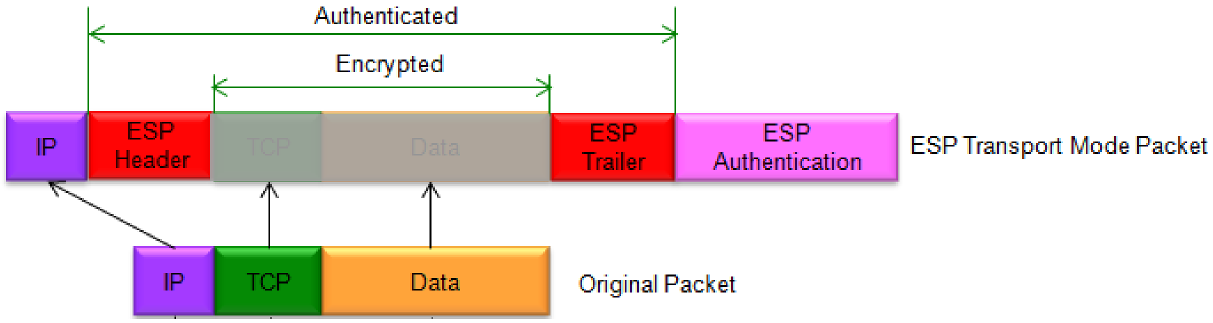



Open Systems
Initiative
(OSI) Model '94

IPSec in **TRANSPORT MODE**, **<u>encrypts payload</u>** but keeps the headers.

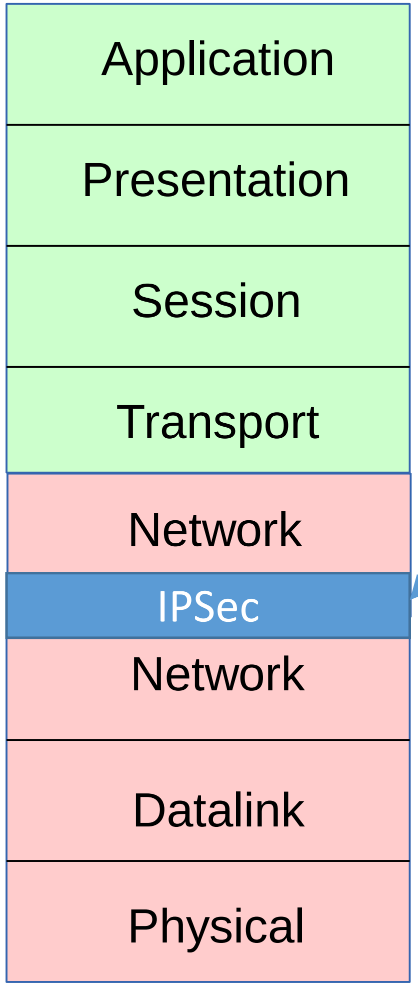
Transmission Control Protocol (TCP), UDP

Internet Protocol (IP)




http://www.sharetechnote.com/html/IP_Network_IPSec_ESP.html

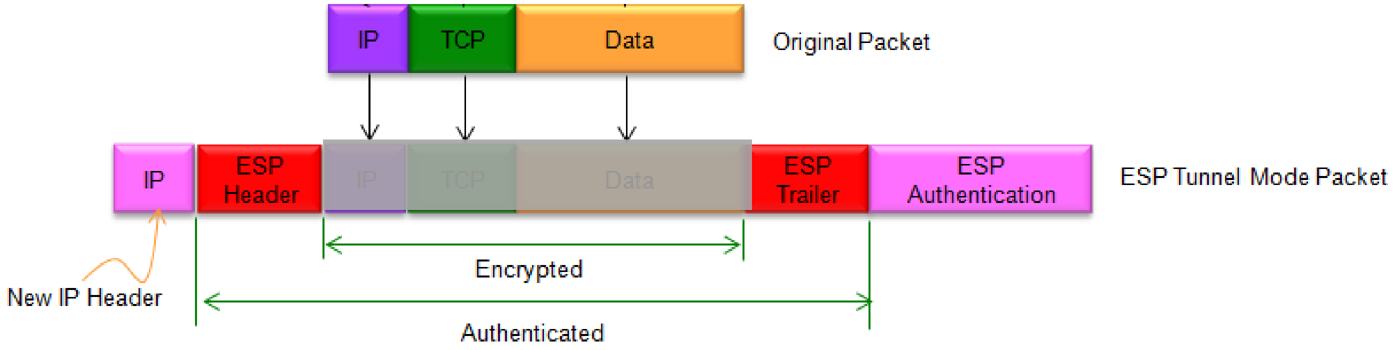
# Where does IPSec happen?




IPSec in **TUNNEL MODE**, **encrypts payload** and **the headers**.

Transmission Control Protocol (TCP), UDP

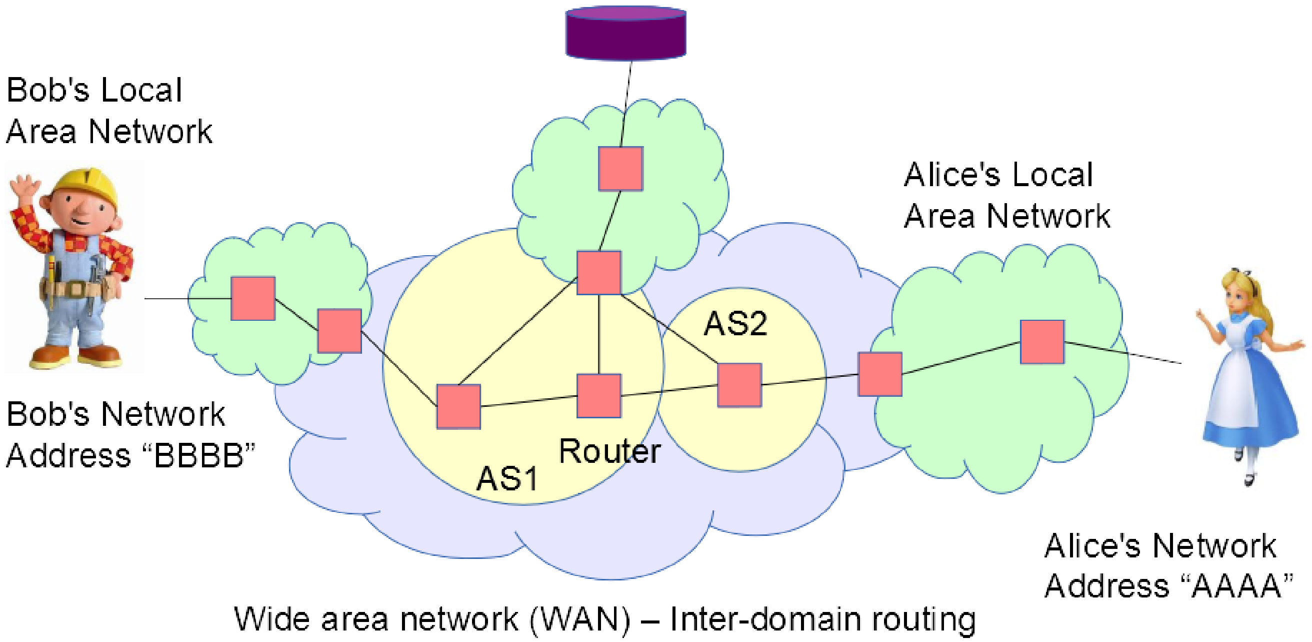
Internet Protocol (IP)



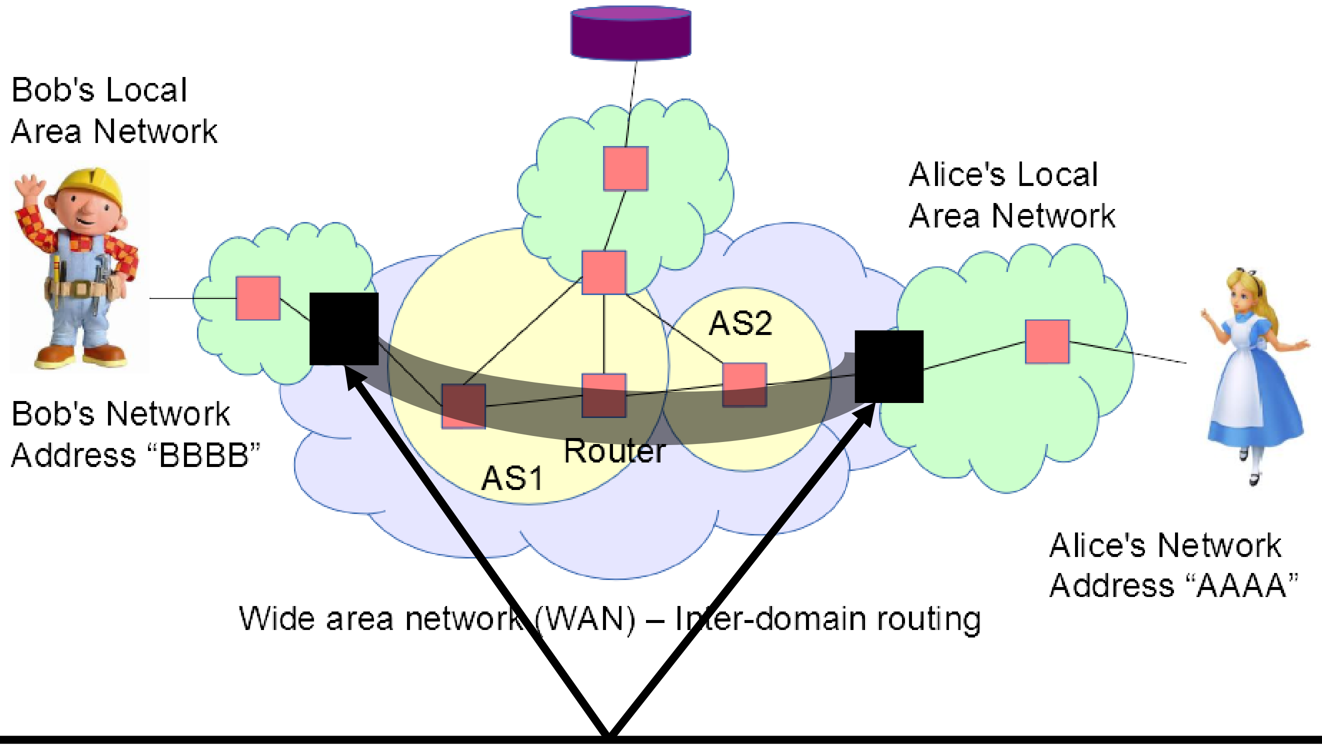

Open Systems Initiative (OSI) Model '94

http://www.sharetechnote.com/html/IP_Network_IPSec_ESP.html

# Virtual Private Network

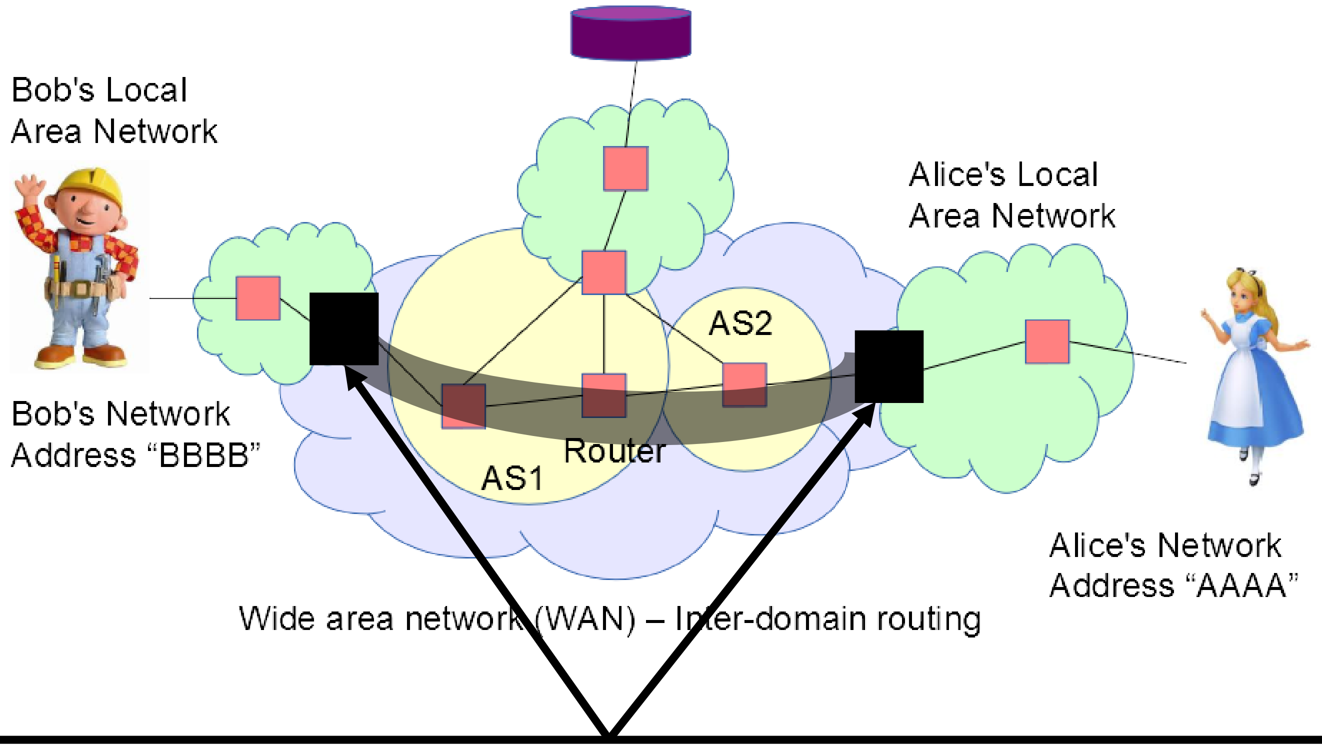


# Virtual Private Network




- Builds on IPSec in tunnel mode
  - Looks like one single network (Bob routes to Alice as if it was a LAN)
  - Inside VPN "tunnel" fully protected packets: confidentiality, authentication, integrity, reply

# Virtual Private Network



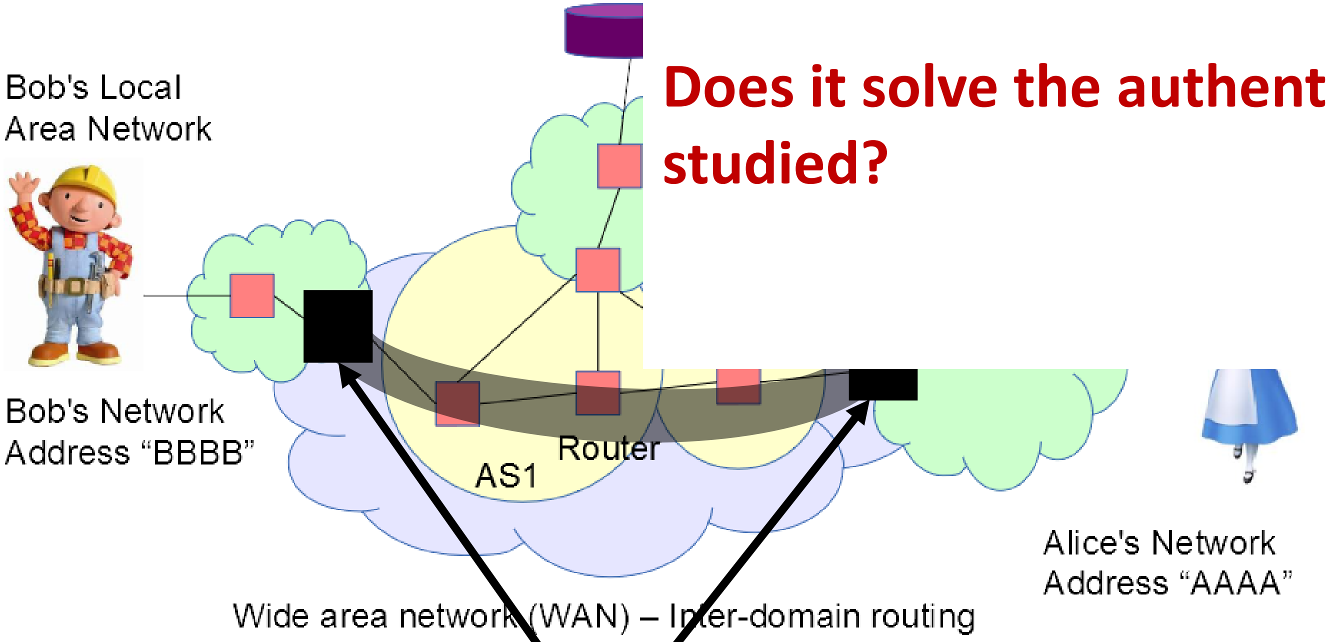

**VPN Uses**

- Join two networks from a company – e.g., two campuses from a university
- Accessing an intranet – e.g., accessing EPFL internal services from home
- Accessing the internet from a location – e.g., accessing publisher web from EPFL

# Virtual Private Network

**Does it protect against Denial of Service?**

**Does it solve the authentication problem we have studied?**




- IPSec in tunnel mode. The VPN
  - Looks like one single network
  - Routing internally
  - Inside VPN "tunnel" fully protected packets: confidentiality, authentication, integrity, reply

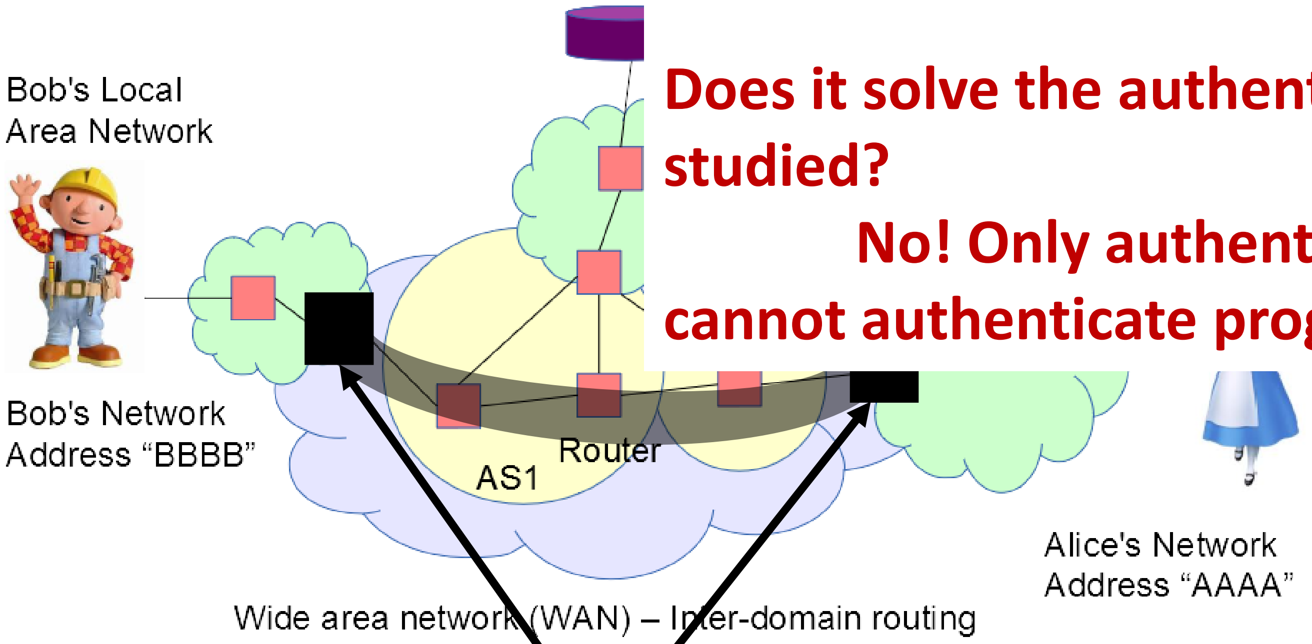
# Virtual Private Network

**Does it protect against Denial of Service?**

**No! Your IP still exists**

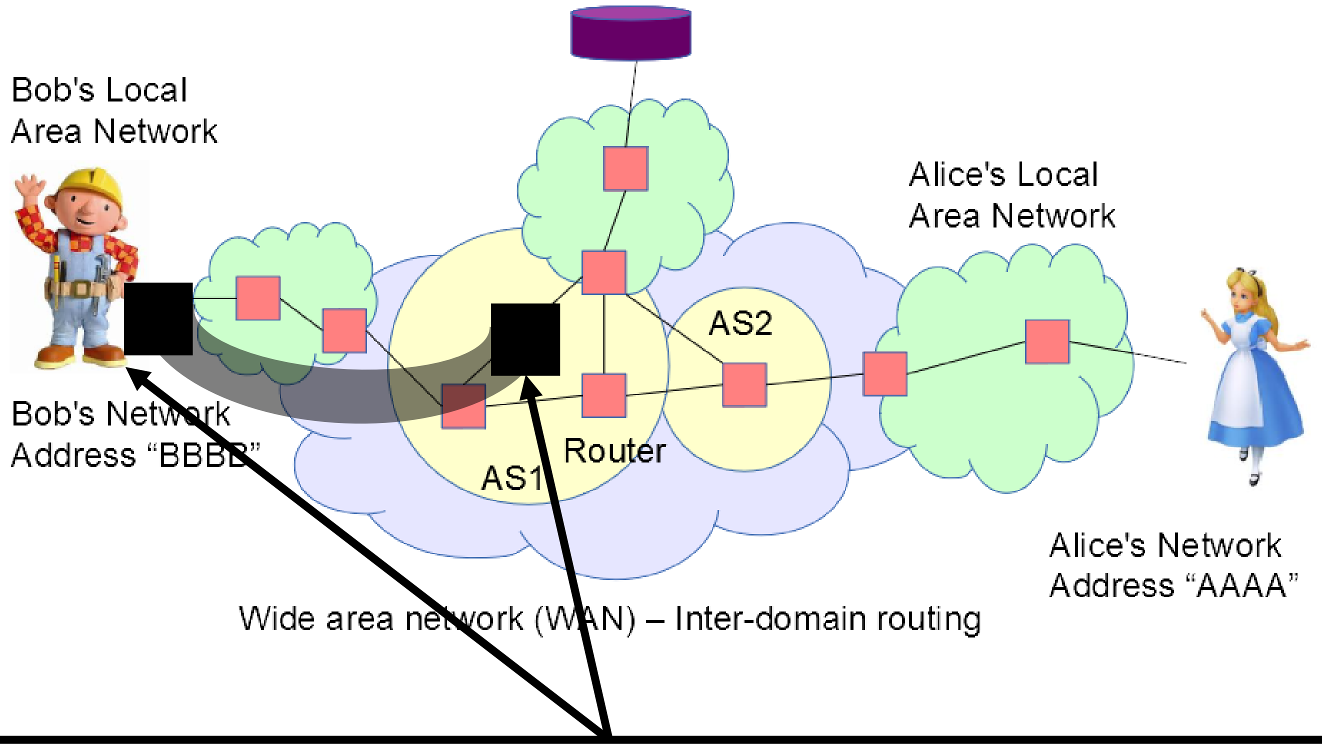
**Does it solve the authentication problem we have studied?**

**No! Only authentication at network level. It cannot authenticate programs or applications**




- IPSec in tunnel mode. The VPN
  - Looks like one single network
  - Routing internally
  - Inside VPN "tunnel" fully protected packets: confidentiality, authentication, integrity, reply

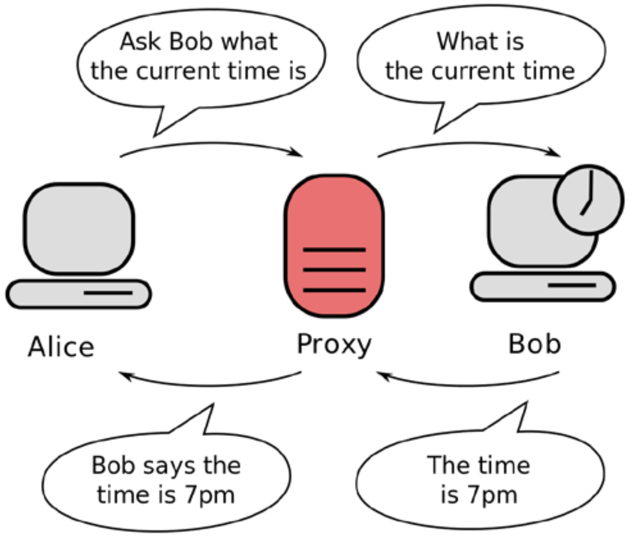
# Virtual Private Network - other common configuration

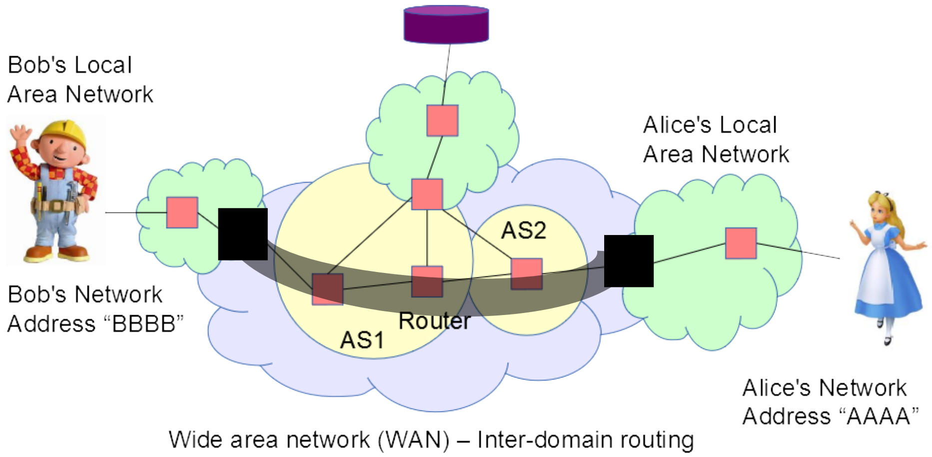



**VPN out of Bob's LAN** (VPN as a Service)

AS1 can see the connection from the VPN to the server (if that connection is in the clear, it can spy)

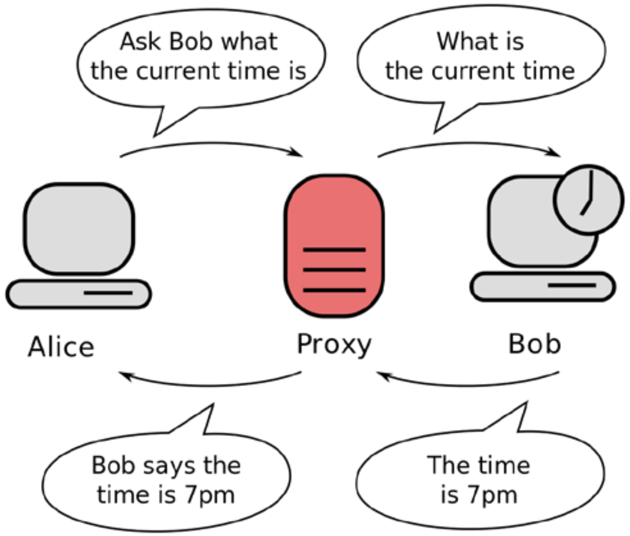
# Is a VPN the same as a proxy?

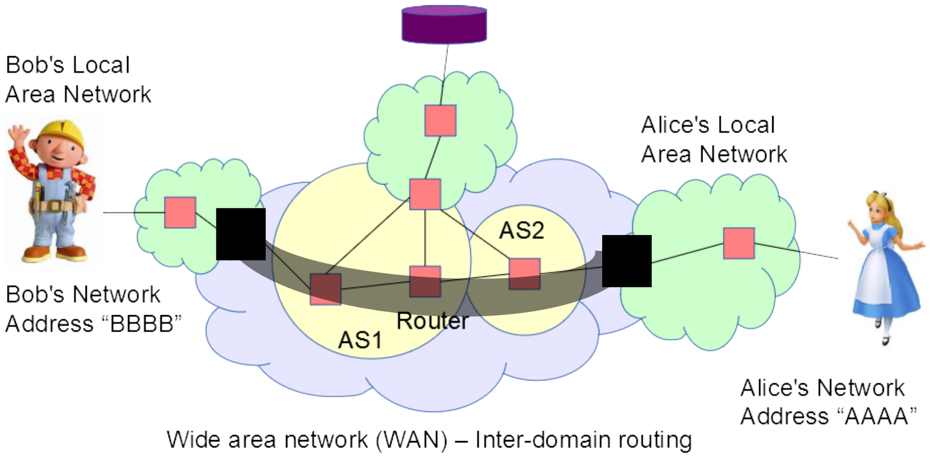


# Is a VPN the same as a proxy?

**No!** They both hide the IP from the receiver but they offer very different properties!




A proxy does not guarantee encryption just change of IP address (some proxys also cache data)

VPN not only changes IP but also encrypts data between the borders of the VPN

# IP limitations

Refresher

- **No reliability**: messages can get dropped, there is no mechanism to ensure a message was received
- **No congestion/flow control**: no mechanism to avoid congestion either in the network or the end hosts
- **No sessions**: no way to associate messages together (and in both directions) into one logical “session”
- **No multiplexing**: no way to associate messages to a network address to specific applications / users on host.

The Transmission Control Protocol (TCP)

- Protocol run “inside/above” the IP protocol
- Addresses the issues above

# TCP header

Refresher

```
 0                   1                   2                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |          Source Port          |       Destination Port        |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                        Sequence Number                        |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                    Acknowledgment Number                      |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |  Data |           |U|A|P|R|S|F|                               |
 | Offset| Reserved  |R|C|S|S|Y|I|            Window             |
 |       |           |G|K|H|T|N|N|                               |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |           Checksum            |         Urgent Pointer        |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                    Options                    |    Padding    |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                             data                              |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

                        TCP Header Format
```

Multiplexing

Reliability
Congestion control

Flow Control

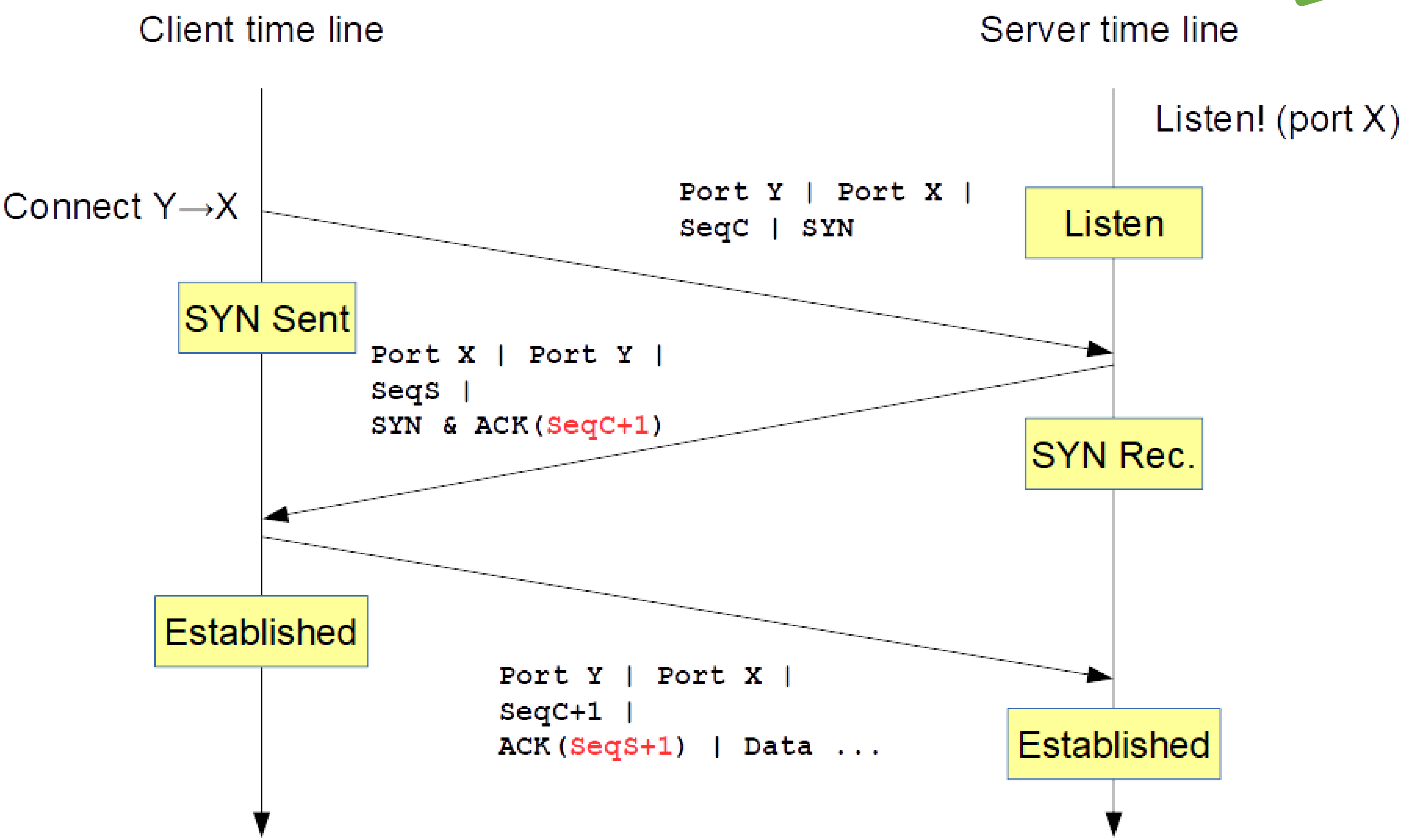
Well known Ports:

20-21 – FTP
22 – SSH
25 – SMTP
53 – DNS
80 – HTTP
110 – POP3
143 – IMAP
443 – HTTPS

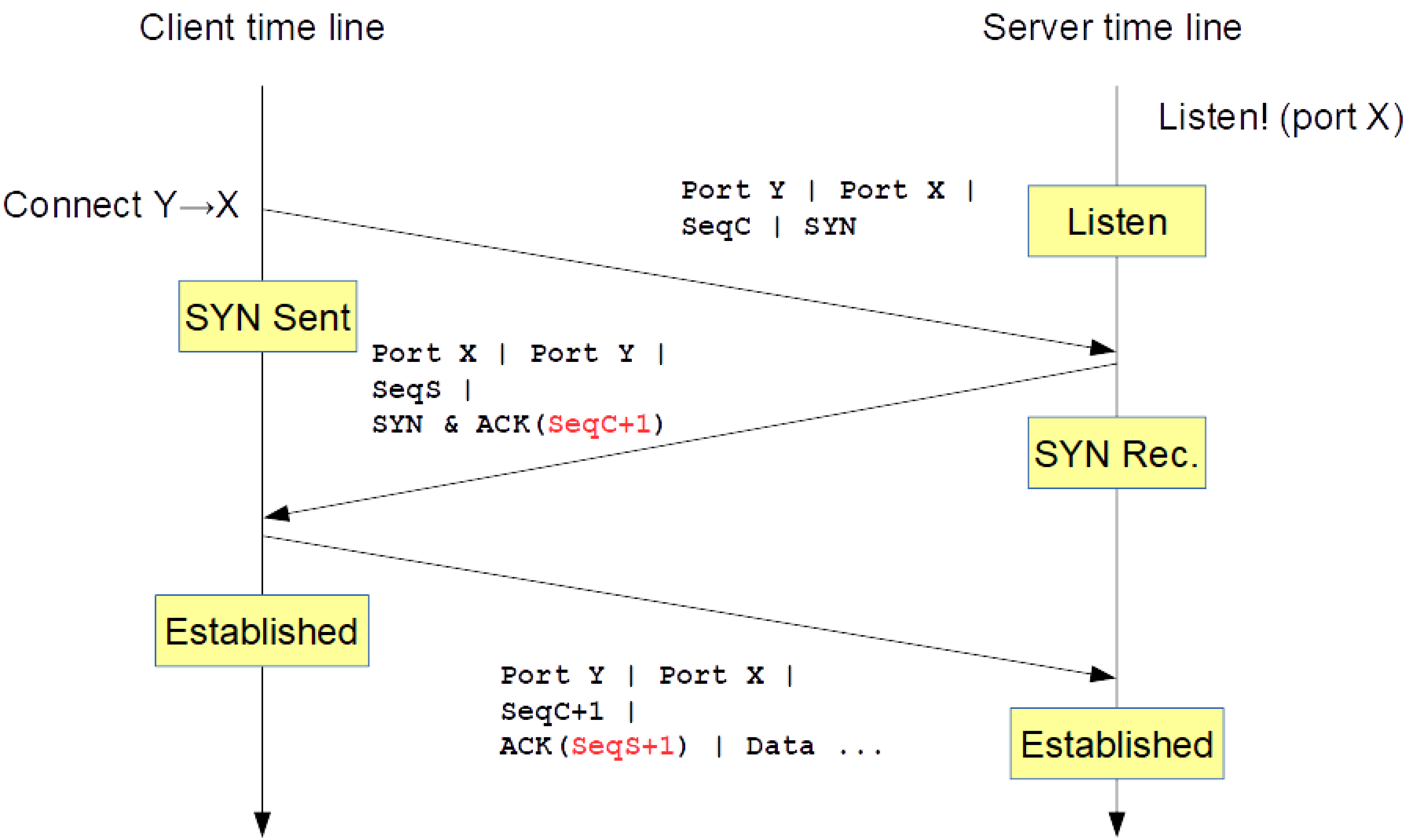
RFC793 (1981) http://www.ietf.org/rfc/rfc793.txt

# TCP 3-way handshake

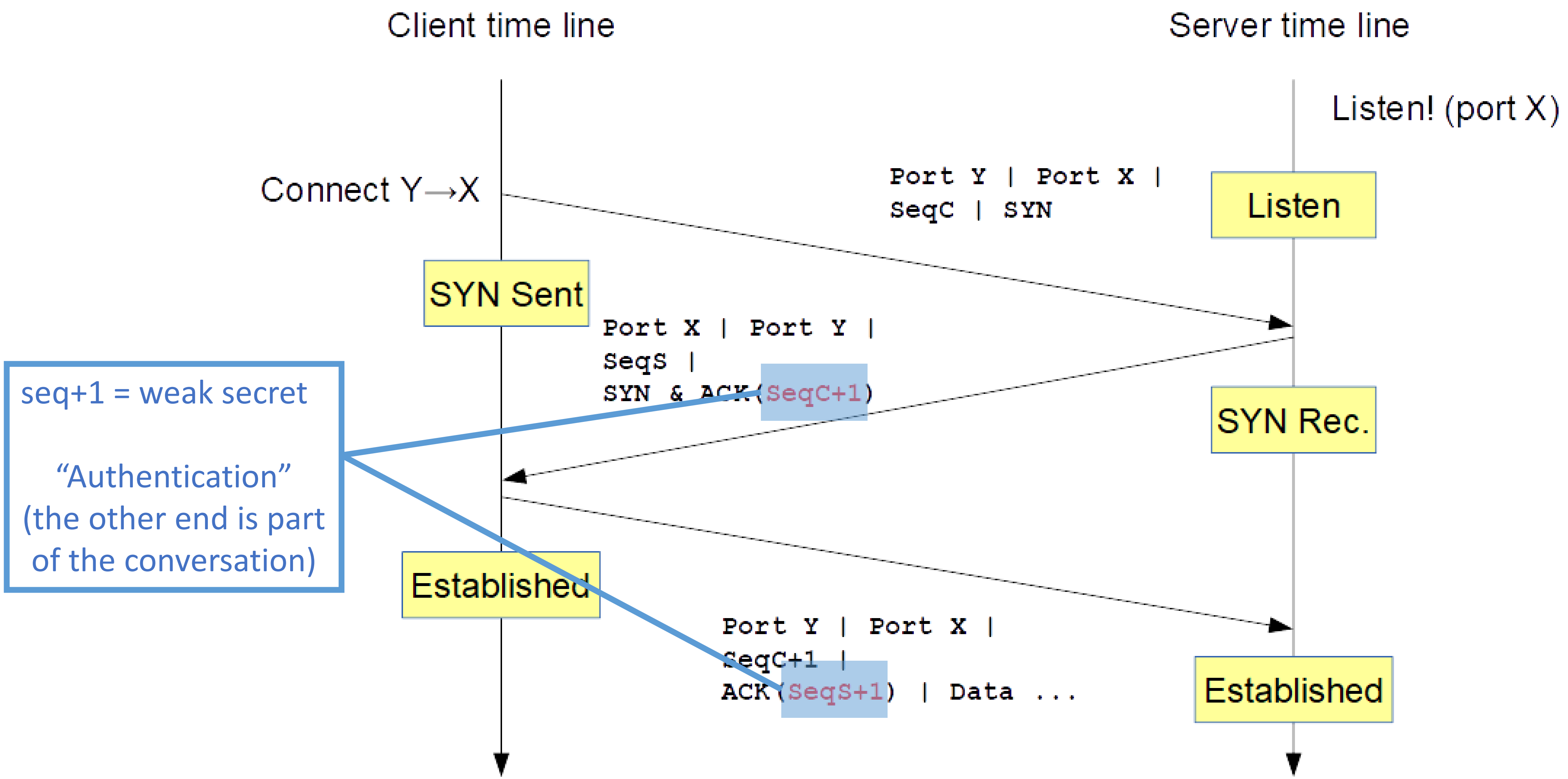
Refresher



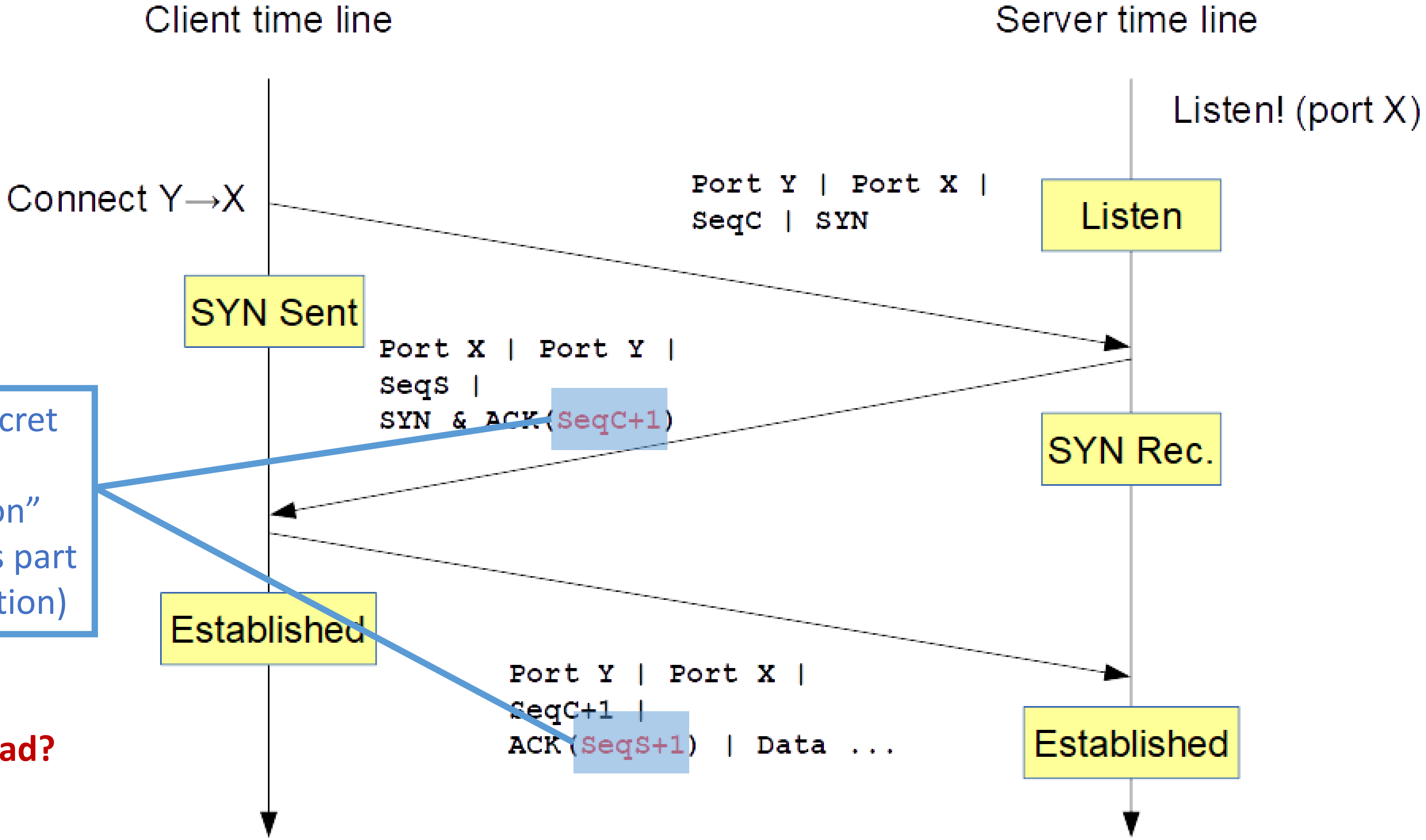
# TCP 3-way handshake – Security considerations



# TCP 3-way handshake – Security considerations



# TCP 3-way handshake – Security considerations

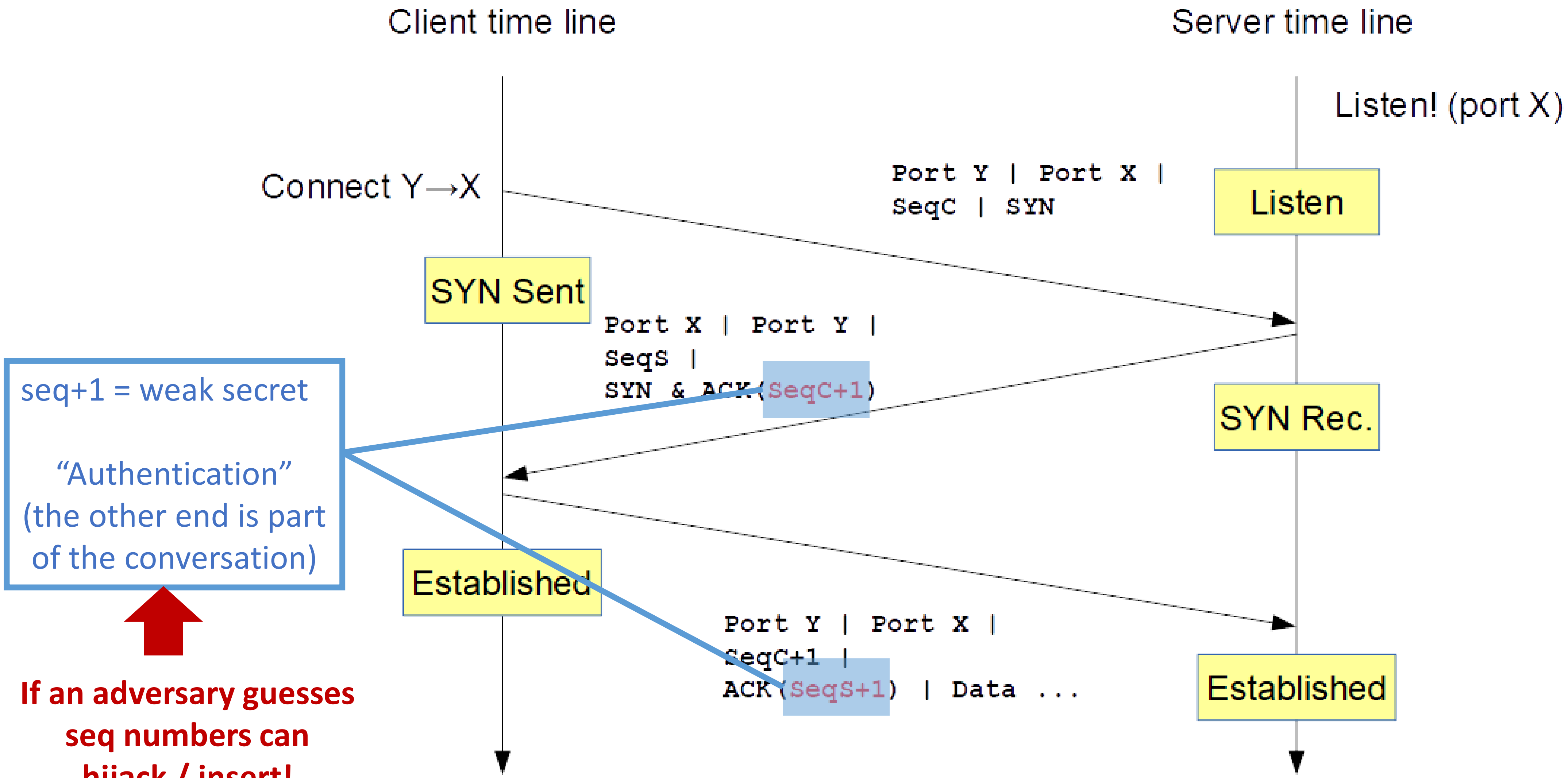



seq+1 = weak secret

"Authentication"
(the other end is part of the conversation)

**Why is this bad?**

# TCP 3-way handshake – Security considerations




**If an adversary guesses seq numbers can hijack / insert!**

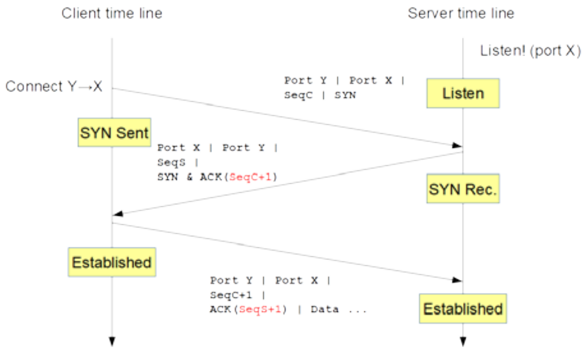
https://www.techrepublic.com/article/tcp-hijacking/

# TCP 3-way handshake – Security considerations

**Can the adversary guess???**

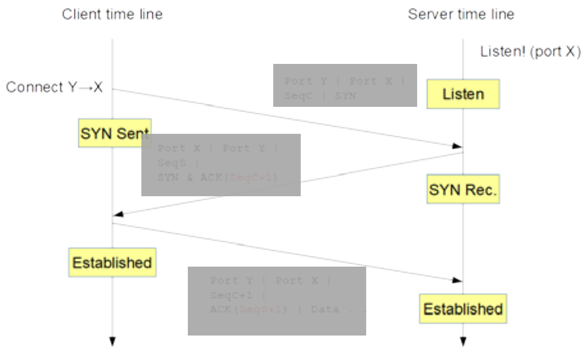
**NO NEED TO GUESS**

If he is in the same network and the communication is in the clear

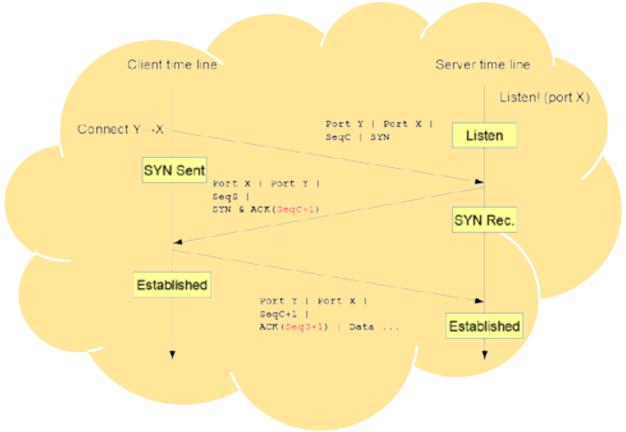



**NEED TO GUESS**

If he is in the same network and the communication is encrypted




If he is in a different network and has no direct view of the exchange



# TCP 3-way handshake – Security considerations

Can the adversary guess???

- If clear observable connection (no need to guess)
- Weak random numbers generation (e.g., time dependent)

Example attack

- The (historical) “rsh” UNIX utility that provides a remote shell implemented authentication and authorization on the basis of remote IP address only! (**Bad idea**)

https://www.techrepublic.com/article/tcp-hijacking/

# TCP 3-way handshake – Security considerations

Can the adversary guess???

- If clear observable connection (no need to guess)
- Weak random numbers generation (e.g., time dependent)

Example attack

- The (historical) "rsh" UNIX utility that provides a remote shell implemented authentication and authorization on the basis of remote IP address only! (**Bad idea**)
- The Robert Morris Attack:

Remote: cannot directly observe connection

1) Send a SYN packet **spoofed** as if it was from authorized host.
2) **Guess** server SeqS and send an ACK with SeqS+1 and some data.
3) The data is interpreted as a shell command and executed!

https://www.techrepublic.com/article/tcp-hijacking/

# Basic steps of TCP hijacking

**Who**: a man in the middle adversary (MITM)

- can observe communication
- can intercept and inject packets

**What**:

1- Wait for TCP session to be established between client and server

2- Wait for authentication phase to be over

3- Only then use knowledge of sequence numbers to take over the session and inject malicious traffic.

4- Use malicious traffic to execute commands, …

5- The genuine connection gets cancelled (desynchronized or reset).

https://www.techrepublic.com/article/tcp-hijacking/

# Basic steps of TCP hijacking

**How can we solve this?**

Cryptographically authenticate all exchanges! Not only at the start

But TCP cannot do that…

**Who**: a man in the middle adversary (MITM)
- can observe communication
- can intercept and inject packets

**What**:

1- Wait for TCP session to be established between client and server

2- Wait for authentication phase to be over

3- Only then use knowledge of sequence numbers to take over the session and inject malicious traffic.

4- Use malicious traffic to execute commands, …

5- The genuine connection gets cancelled (desynchronized or reset).

https://www.techrepublic.com/article/tcp-hijacking/

# The Transport Layer Security (TLS)

- Cryptographic protocols above TCP/IP -- “middlelayer”

- **Goal**: providing communications security:
    - Confidentiality: symmetric encryption
    - Authentication (One or two-side): public key cryptography
    - Integrity: MAC and signatures

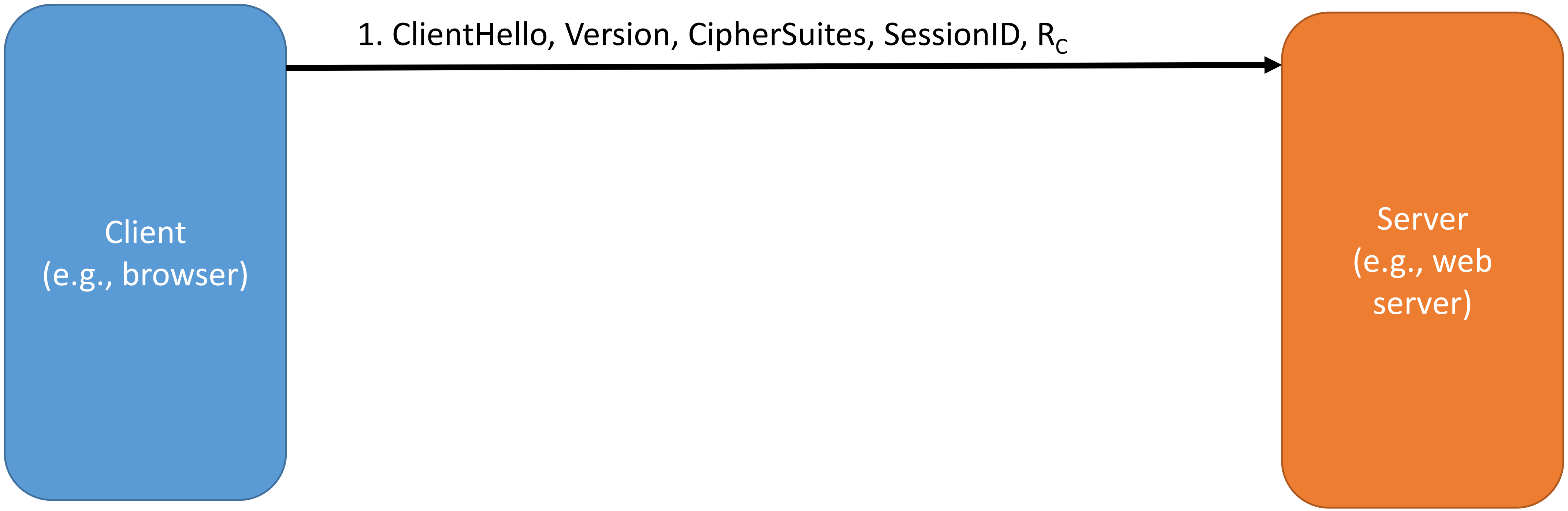
- Provides **forward secrecy**
    - Learning a secret at one point in time does not reveal anything about the past

- State of the art: TLS v3
    - Reality: a zoo in the Internet (it is difficult to upgrade a huge number of computers)
    - SSL, same principles but many vulnerabilities -- deprecated!

https://www.youtube.com/watch?v=_mE_JmwFi1Y

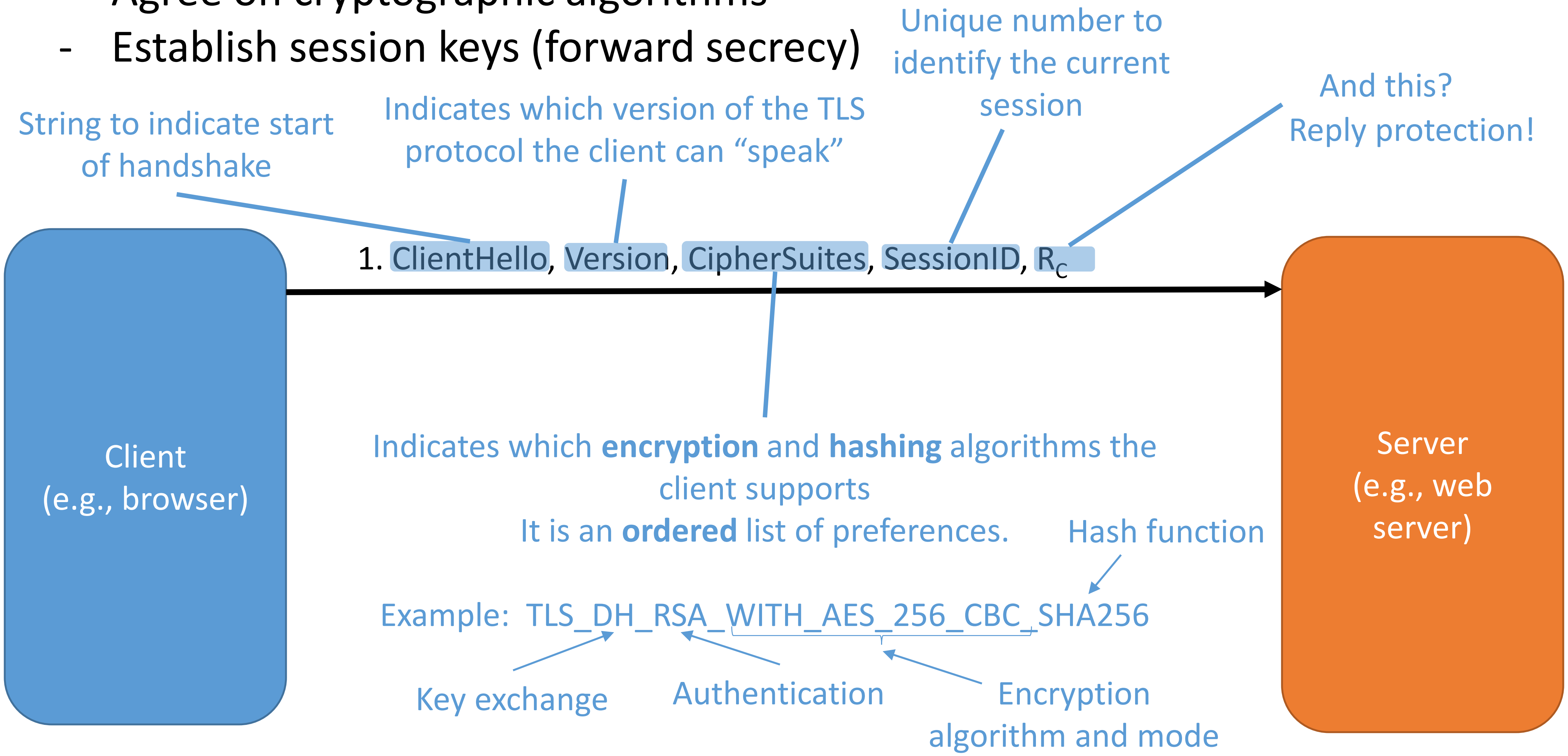
# The TLS handshake

- **Goal**: bootstrap the communication
  - Agree on cryptographic algorithms
  - Establish session keys (forward secrecy)



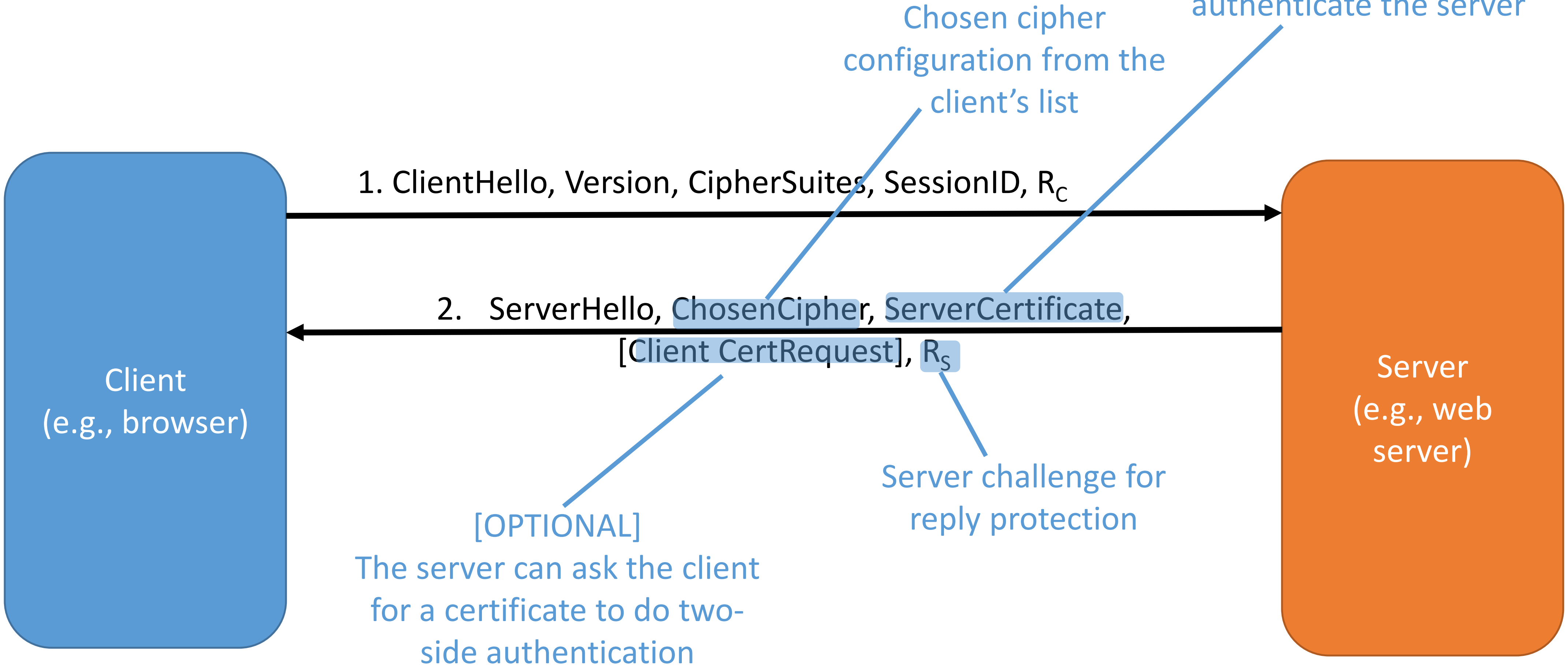
# The TLS handshake

- **Goal**: bootstrap the communication
  - Agree on cryptographic algorithms
  - Establish session keys (forward secrecy)



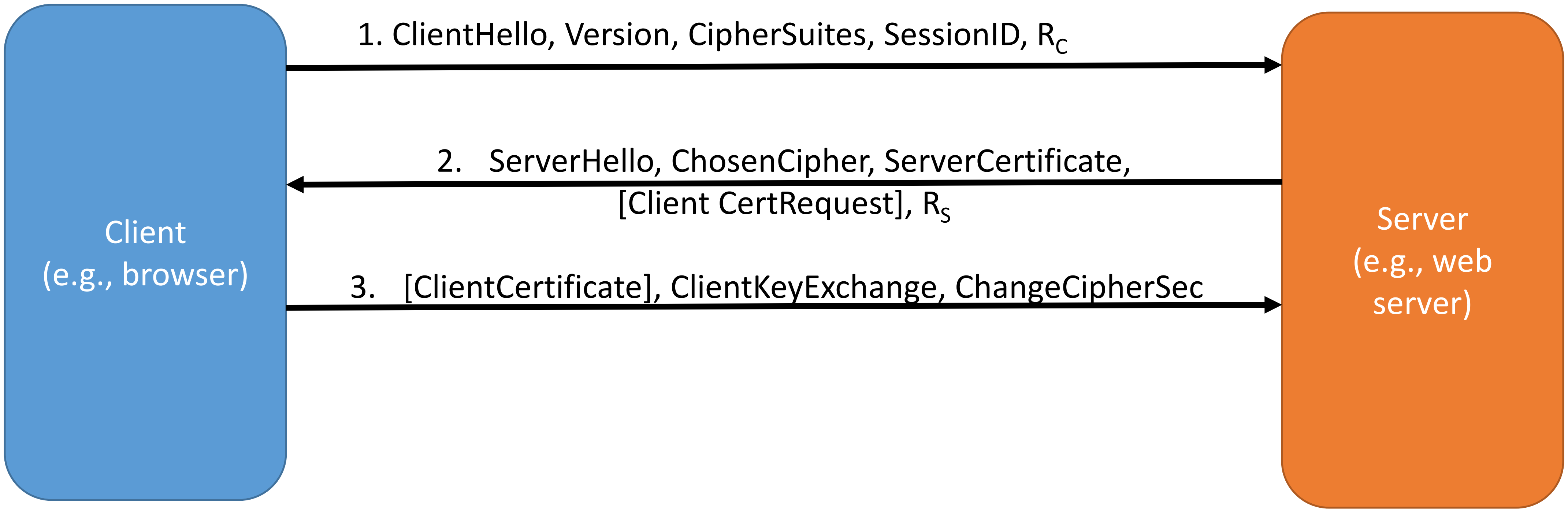
# The TLS handshake

- **Goal**: bootstrap the communication
  - Agree on cryptographic algorithms
  - Establish session keys (forward secrecy)



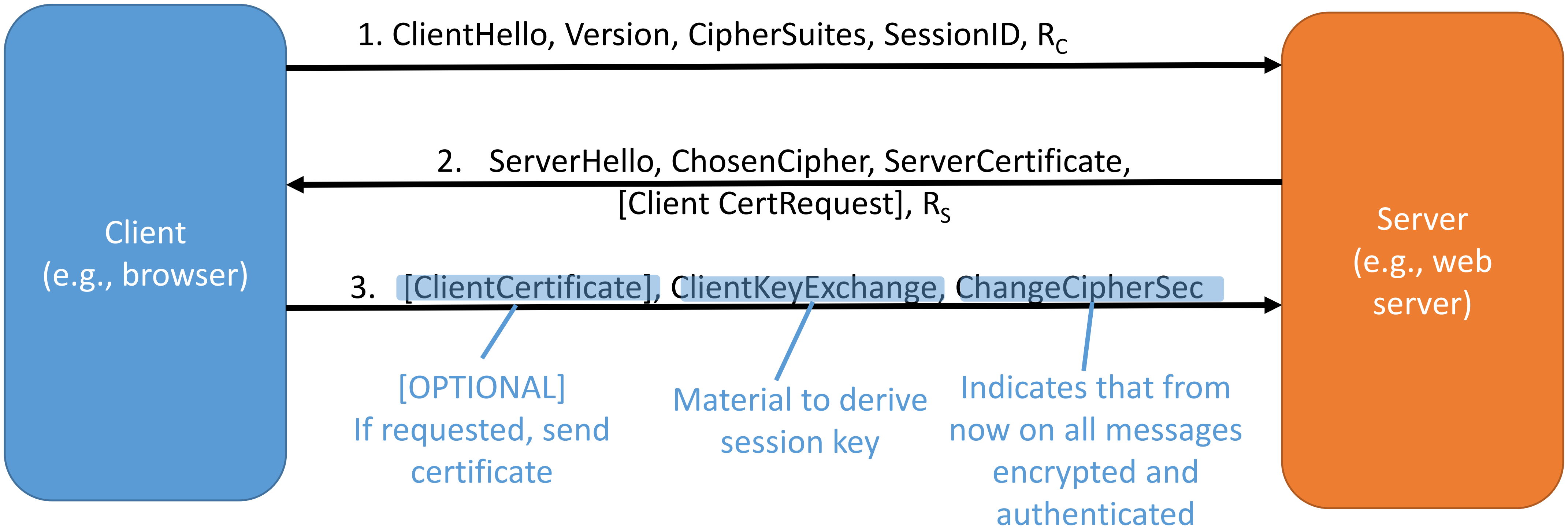
# The TLS handshake

- **Goal**: bootstrap the communication
  - Agree on cryptographic algorithms
  - Establish session keys (forward secrecy)



# The TLS handshake

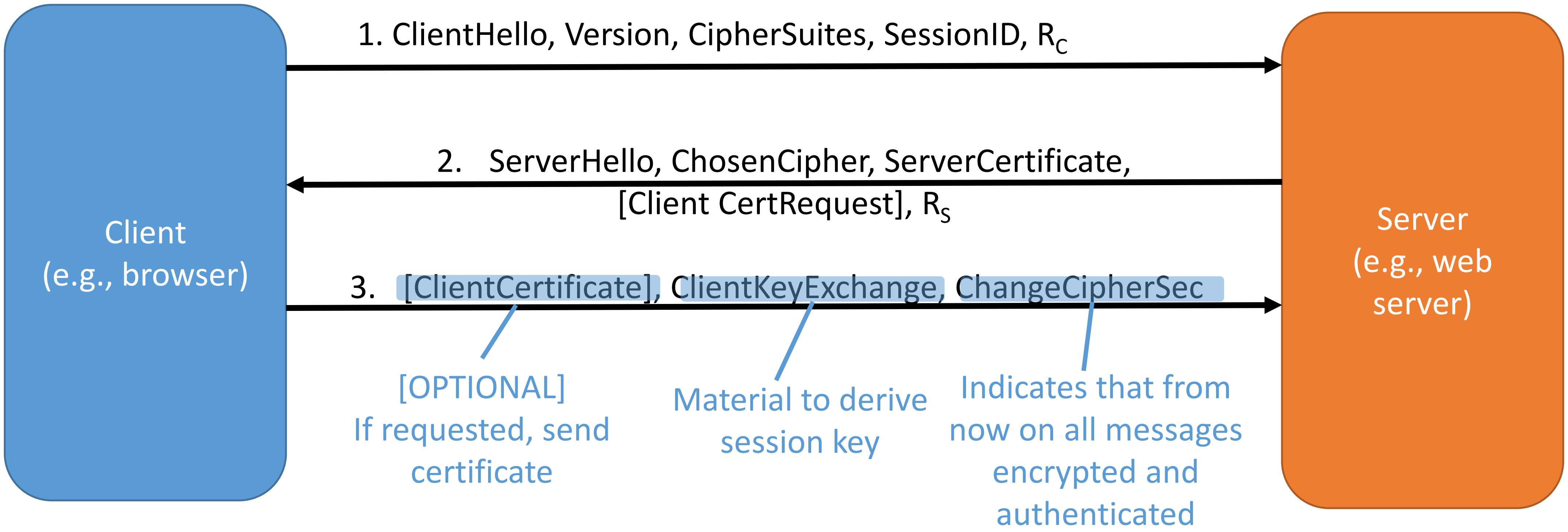
- **Goal**: bootstrap the communication
  - Agree on cryptographic algorithms
  - Establish session keys (forward secrecy)



# The TLS handshake

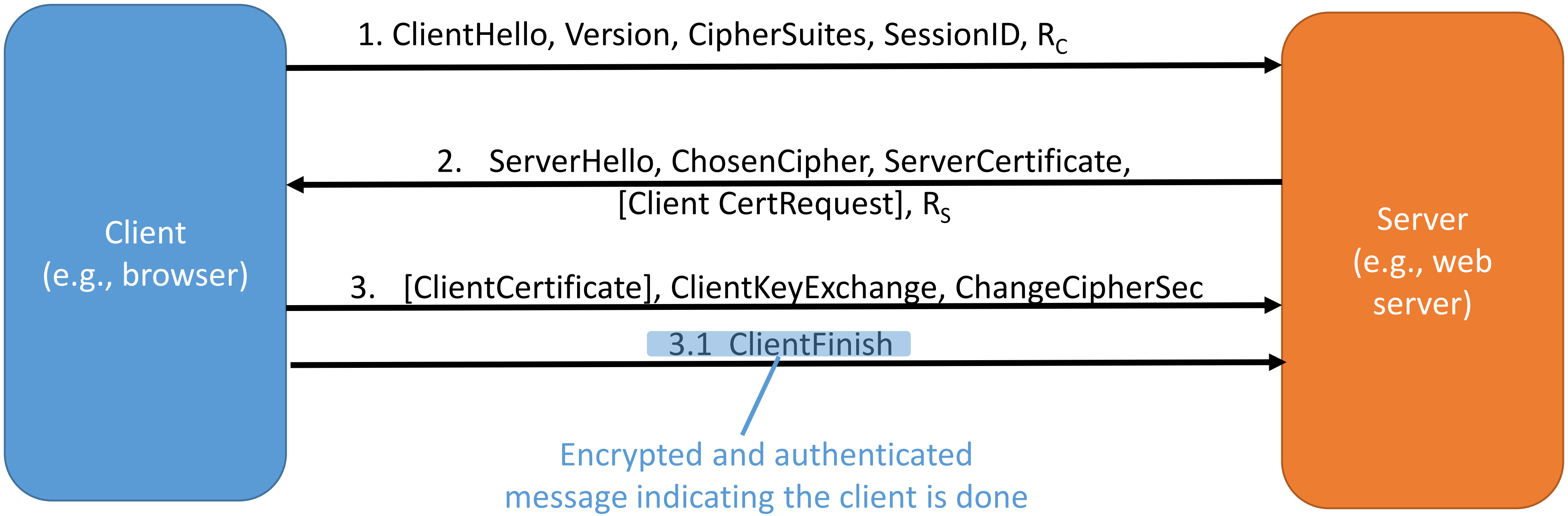
- **Goal**: bootstrap the communication
  - Agree on cryptographic algorithms
  - Establish session keys (forward secrecy)

**After step 3 Client and server have a shared session key!!!**



# The TLS handshake

- **Goal**: bootstrap the communication
  - Agree on cryptographic algorithms
  - Establish session keys (forward secrecy)

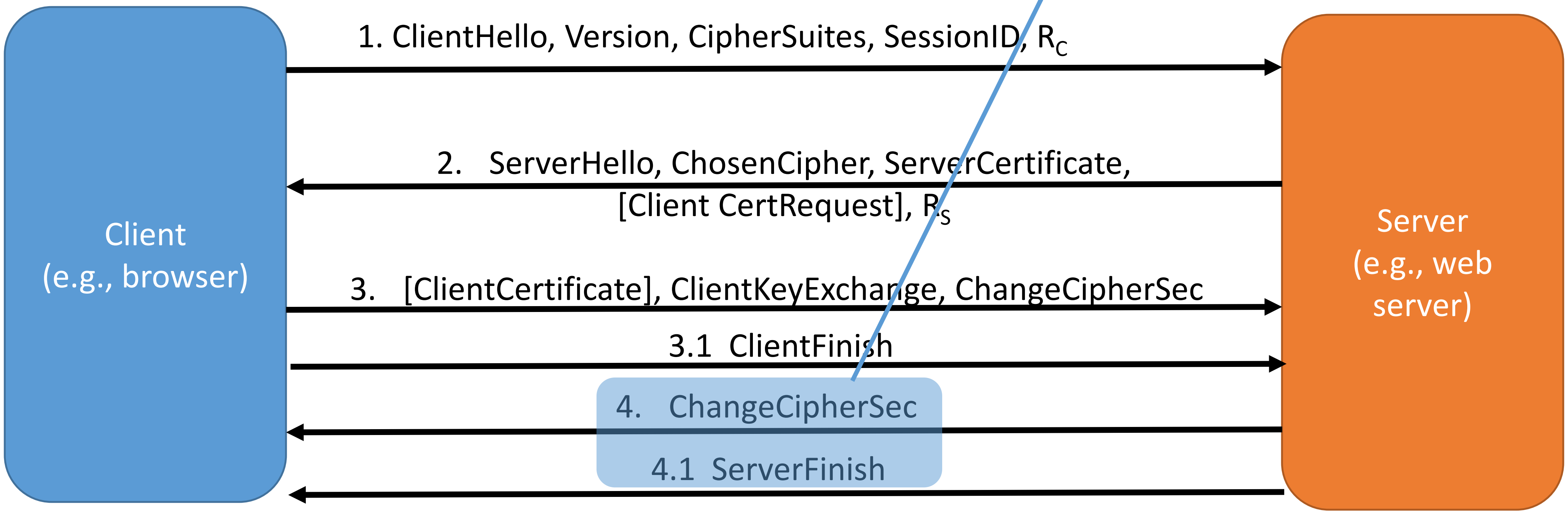


# The TLS handshake

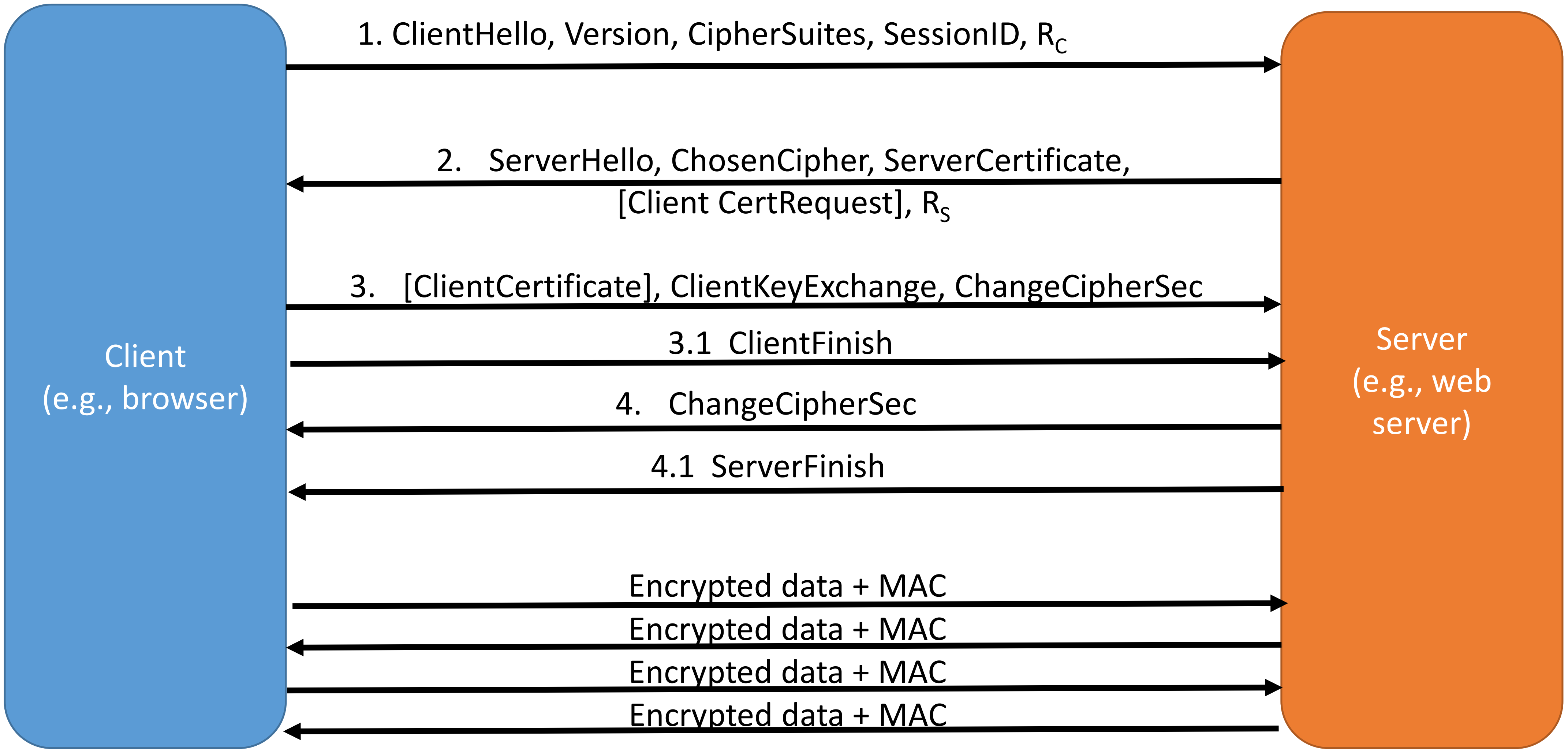
- **Goal**: bootstrap the communication
  - Agree on cryptographic algorithms
  - Establish session keys (forward secrecy)

Server does the same: indicates that from now on everything will be encrypted and authenticated

And sends an authenticated encrypted Finished message

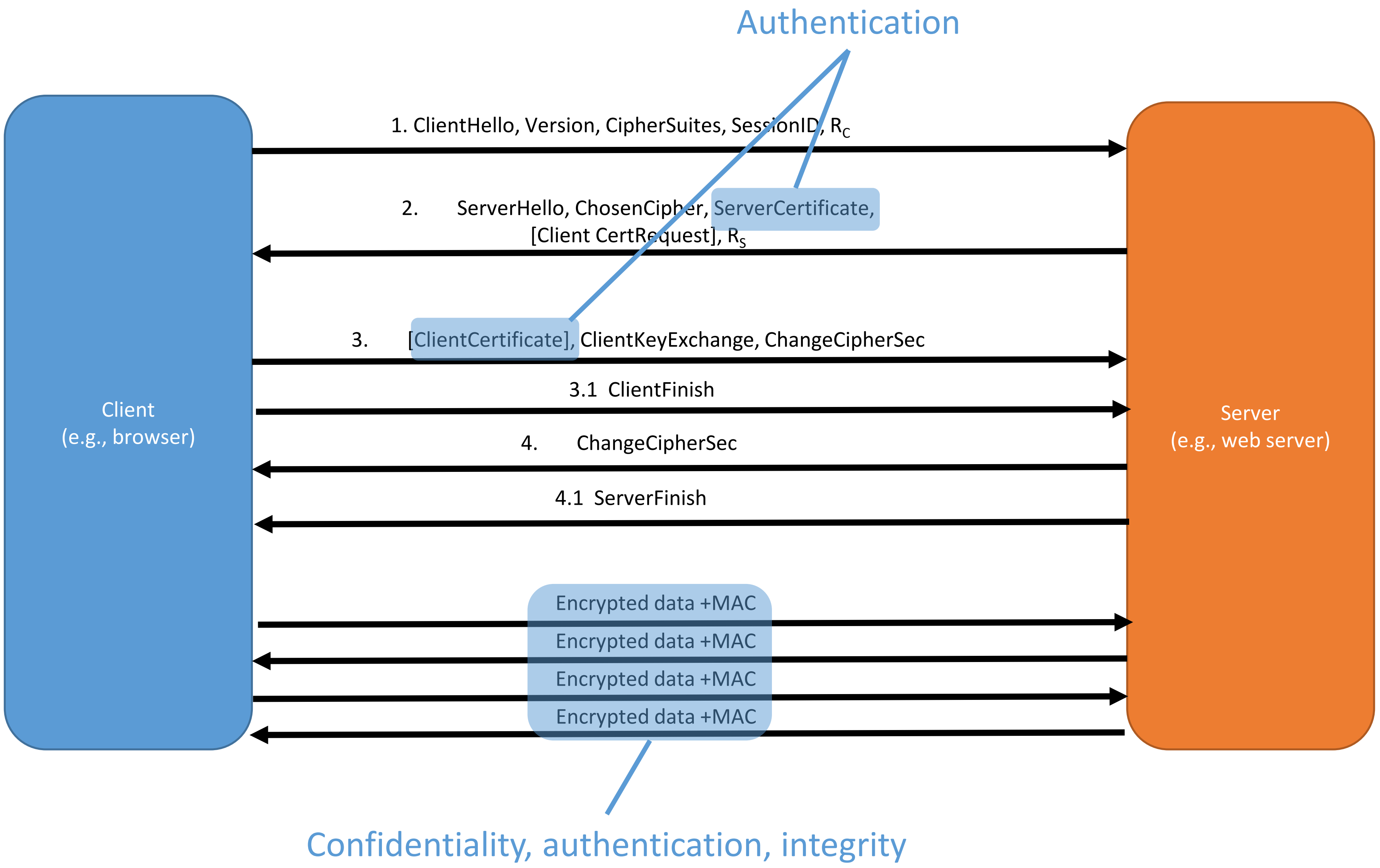


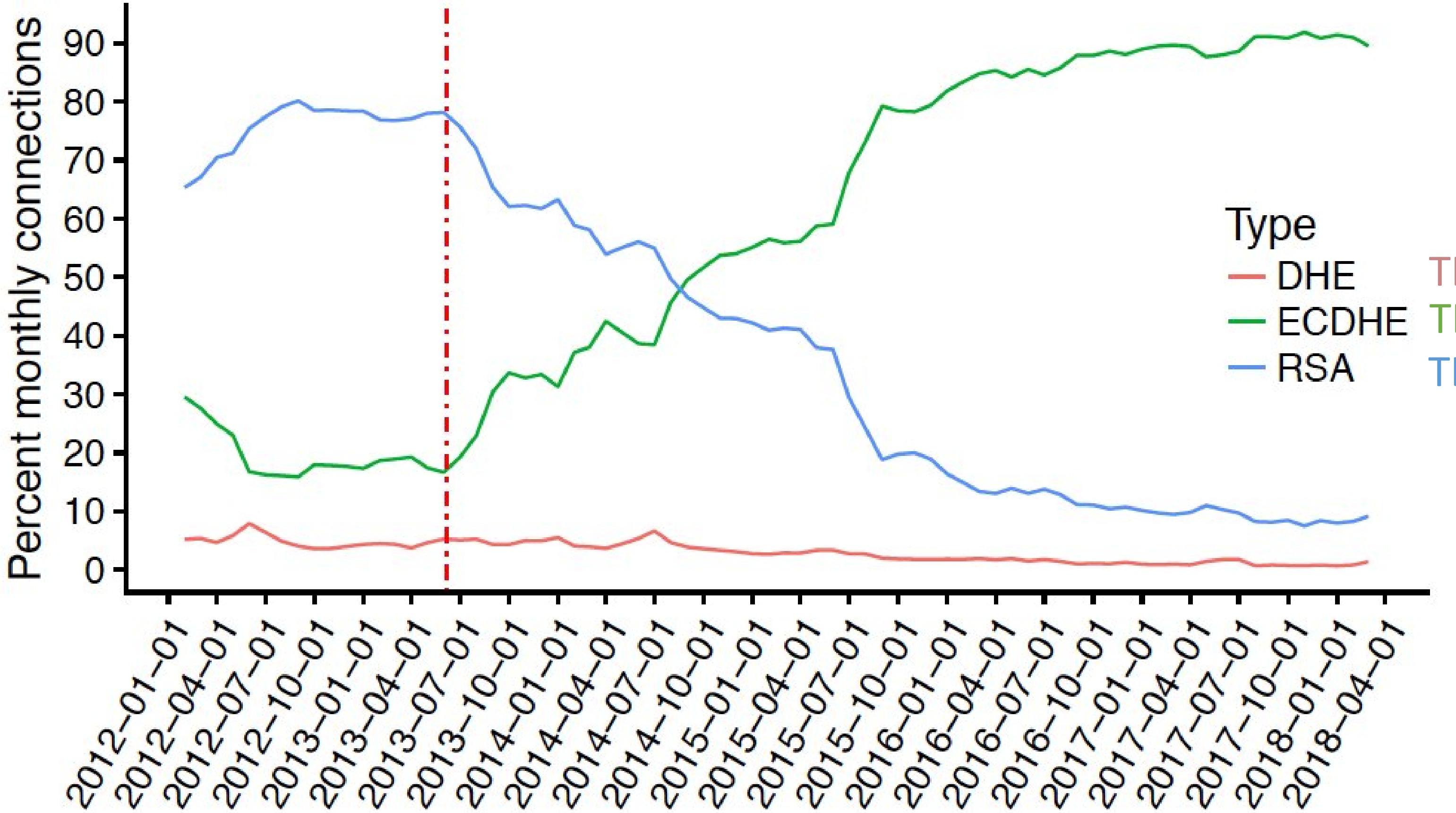
# TLS session



Client
(e.g., browser)
Server
(e.g., web server)
1. ClientHello, Version, CipherSuites, SessionID, $R_C$
2. ServerHello, ChosenCipher, ServerCertificate, [Client CertRequest], $R_S$
3. [ClientCertificate], ClientKeyExchange, ChangeCipherSec
3.1 ClientFinish
4. ChangeCipherSec
4.1 ServerFinish
Encrypted data + MAC
Encrypted data + MAC
Encrypted data + MAC
Encrypted data + MAC

# TLS properties



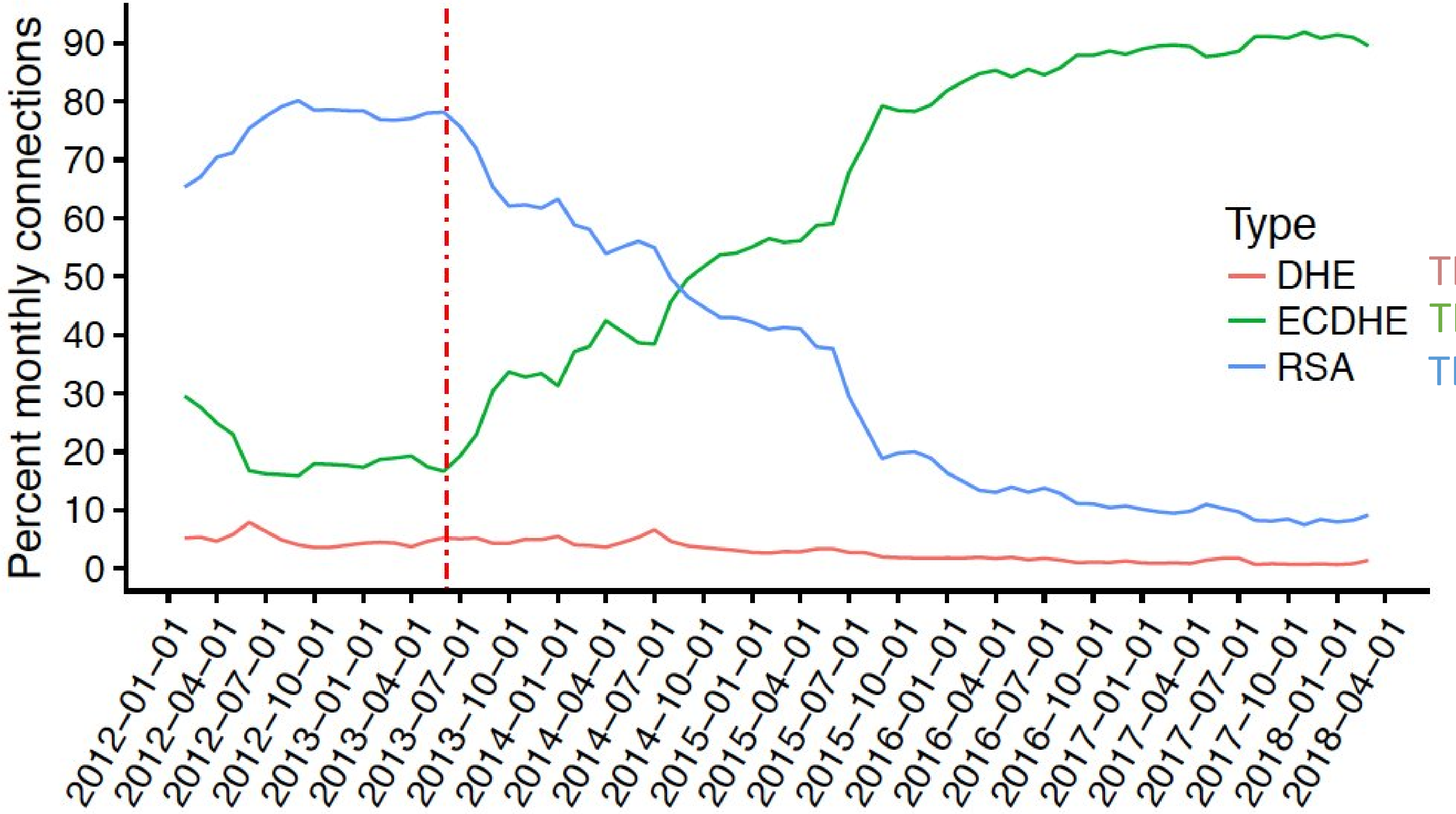



TLS_DH_RSA_WITH_AES_256_CBC_SHA256

TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA256

TLS_RSA_WITH_AES_256_CBC_SHA256

**Figure 8: Negotiated RSA and forward secret connections. Dotted line shows the date of first Snowden revelations.**




TLS_DH_RSA_WITH_AES_256_CBC_SHA256

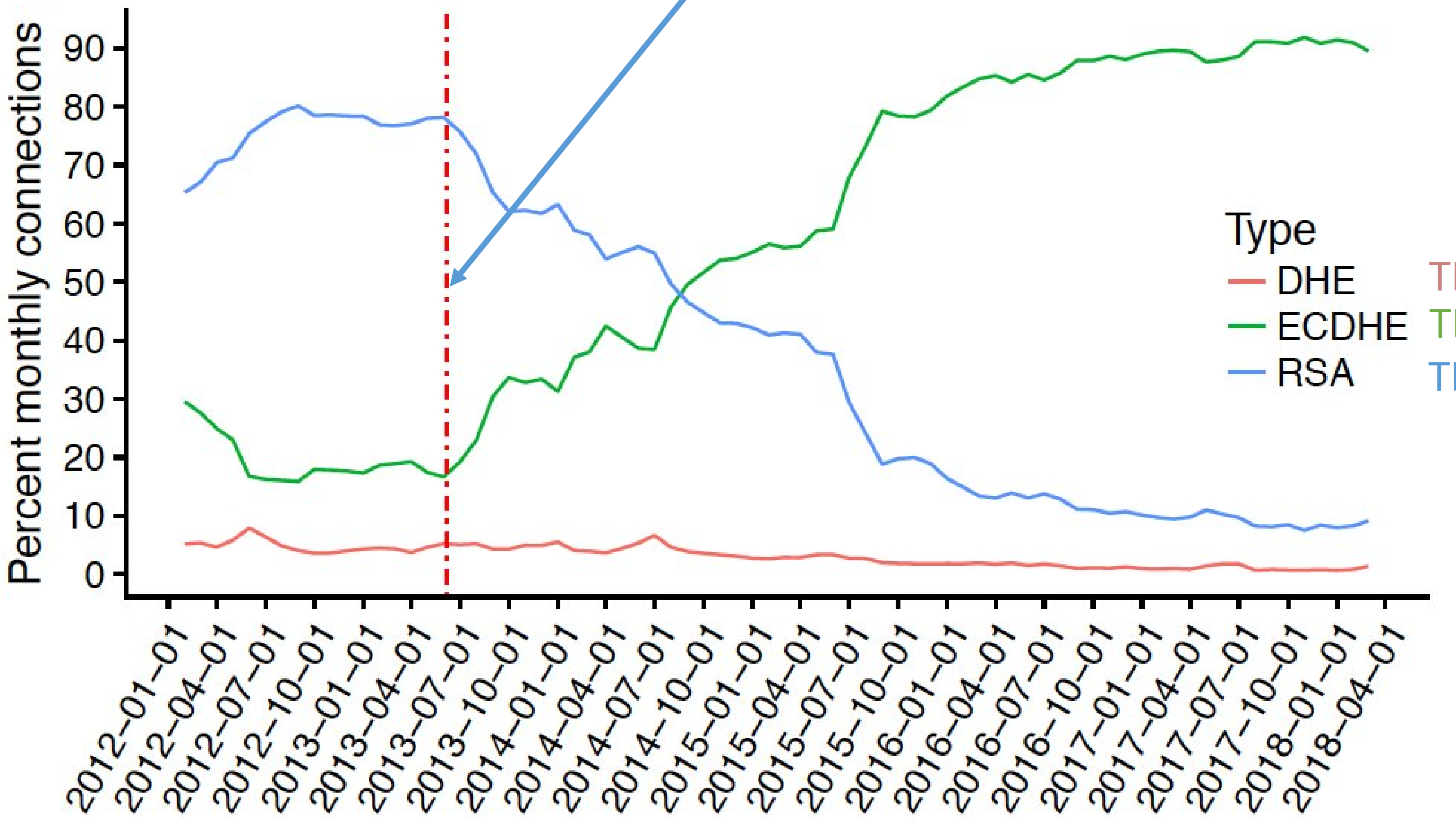
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA256

TLS_RSA_WITH_AES_256_CBC_SHA256

Key transport (as in the previous slides) does not provide **forward secrecy** in case the secret RSA key is stolen

**Figure 8: Negotiated RSA and forward secret connections. Dotted line shows the date of first Snowden revelations.**

After Snowden revealed that the NSA could brute force RSA keys, huge change to Diffie Hellmann based key exchange.
**The use of ephemeral keys provides forward secrecy**




TLS_DH_RSA_WITH_AES_256_CBC_SHA256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA256
TLS_RSA_WITH_AES_256_CBC_SHA256

Key transport (as in the previous slides) does not provide **forward secrecy** in case the secret RSA key is stolen

**Figure 8: Negotiated RSA and forward secret connections. Dotted line shows the date of first Snowden revelations.**

# Attacks on TLS

**Advanced**
**Not for exam**

Downgrading attacks (CVE-2014-3511): implementation flaw that enables the adversary to force server and client to use a less secure version of TLS/SSL.

BEAST (CVE-2011-3389): exploits an implementation weakness in TLS 1.0 implementation of Cipher Block Chaining (CBC) which results in predictable initialization vectors. This allowed to decrypt parts of a packet, and specifically to decrypt HTTP cookies when HTTP is run over TLS

Padding Oracle: because of the MAC-then-encrypt design, TLS is vulnerable to padding oracle attacks. These use block padding as an "oracle" to find out whether a decryption is right or wrong

- Lucky Thirteen (CVE-2013-0169): timing side-channel attack that allows the attacker to decrypt arbitrary ciphertext

Renegotiation attacks: exploit the "renegotiation" feature of TLS that enable users to have new parameters. The adversary can inject his own packets at the beginning of a connection

Many more… DoS, more crypto problems, more protocol problems… Nowadays provable security in TLS 1.3

https://tools.ietf.org/html/rfc7457

# Denial of Service

**Goal**: prevent legitimate users from accessing a service

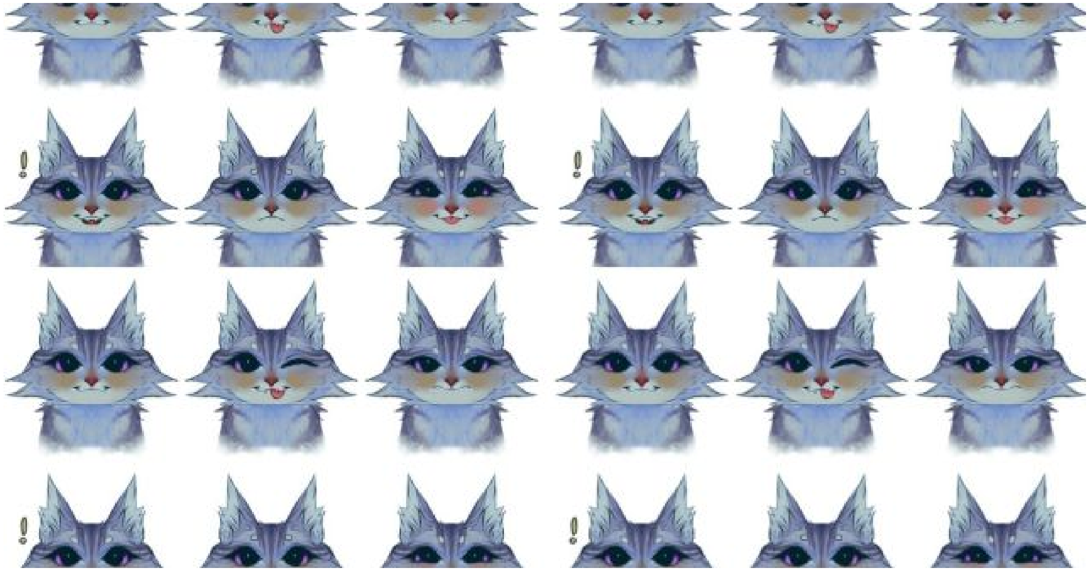
**Option A - Crash victim**: exploit software flaws to make it stop

**Option B – Exhaust victim's resources**

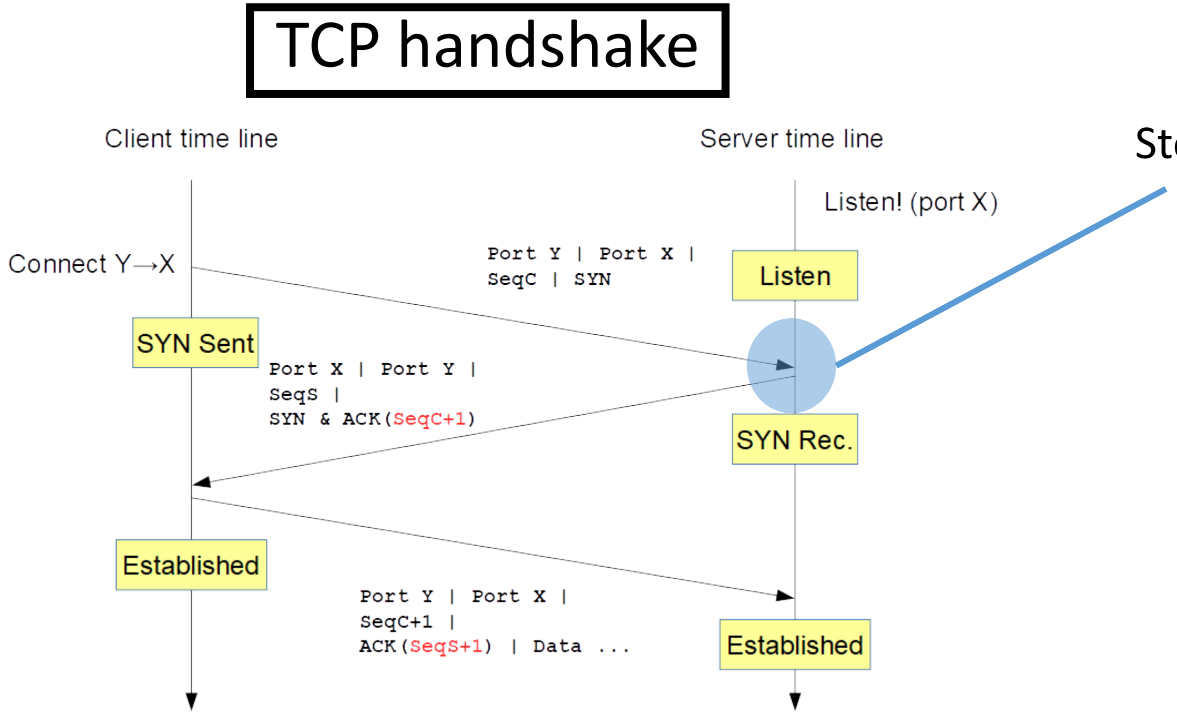
- Network: Bandwidth
- Host
  - Kernel: TCP connection state tables, etc.
  - Application: CPU, memory, etc.

# Example 1 – Skype kittens DoS (CVE-2018-8546)

When receiving about 800 kittens at once, your Skype for Business client will stop responding for a few seconds. If a sender continues sending emojis your Skype for Business client will not be usable until the attack ends.



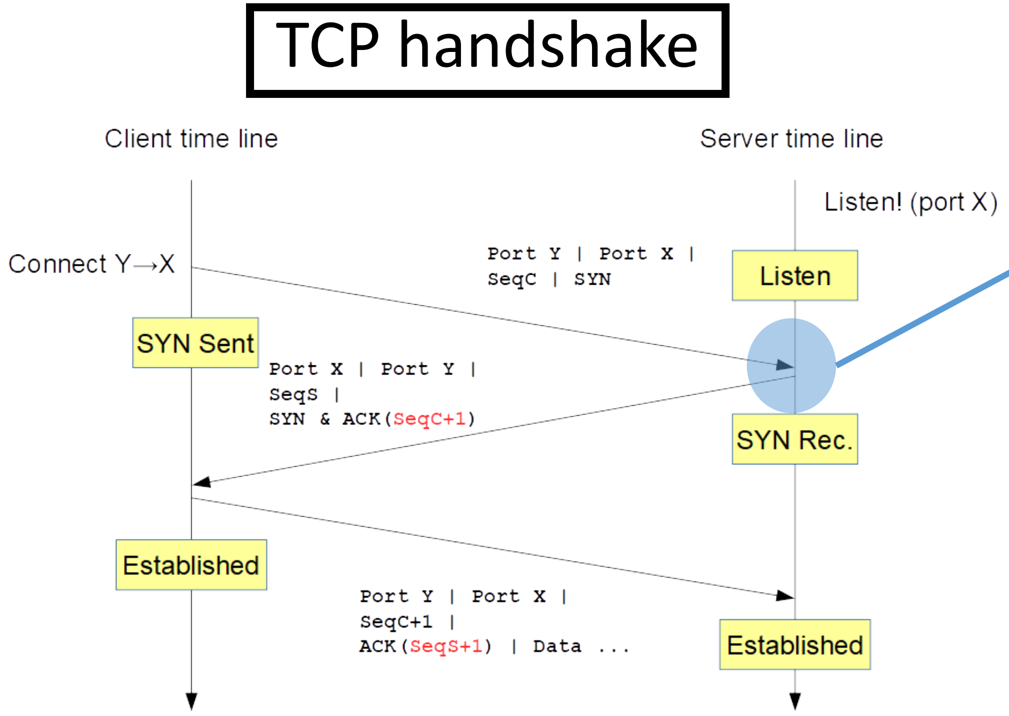
# Example 2 – TCP SYN flood




At this point, the server is listening, waiting for the next ACK in order to establish the connection

Stores a TCP Control Block (TCB) ~ 280 bytes

# Example 2 – TCP SYN flood

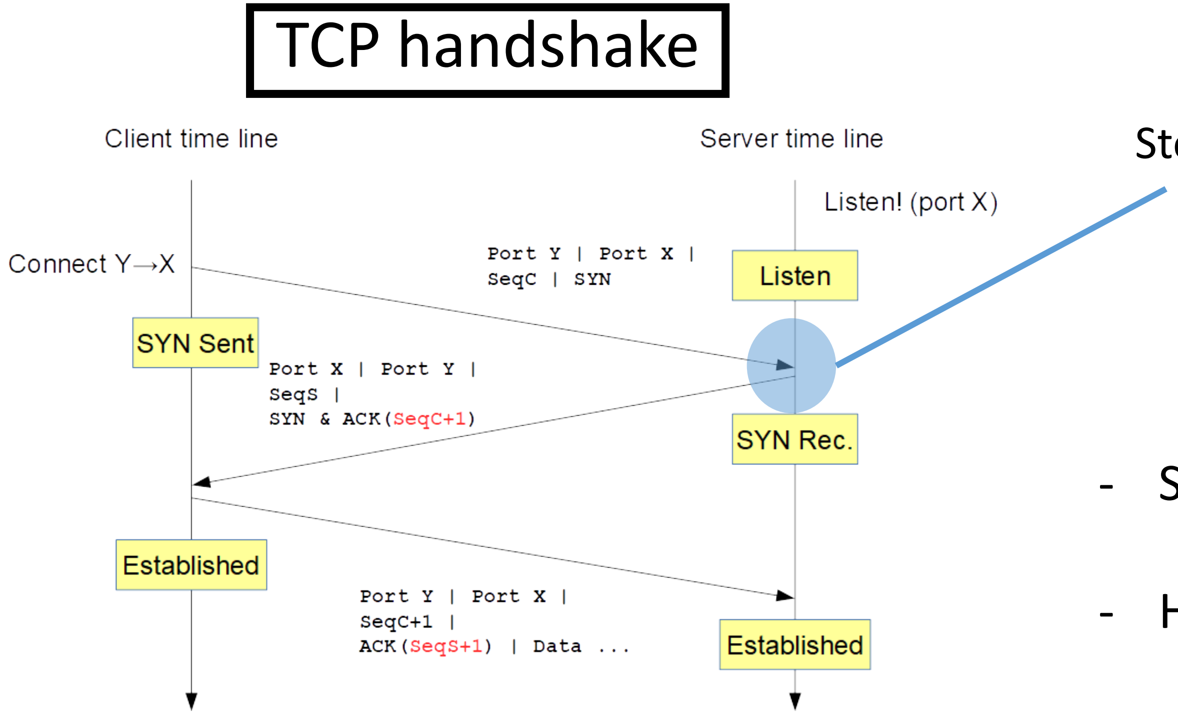



At this point, the server is listening, waiting for the next ACK in order to establish the connection

Stores a TCP Control Block (TCB) ~ 280 bytes

**How does the attack work?**

# Example 2 – TCP SYN flood




At this point, the server is listening, waiting for the next ACK in order to establish the connection

Stores a TCP Control Block (TCB) ~ 280 bytes

**How does the attack work?**

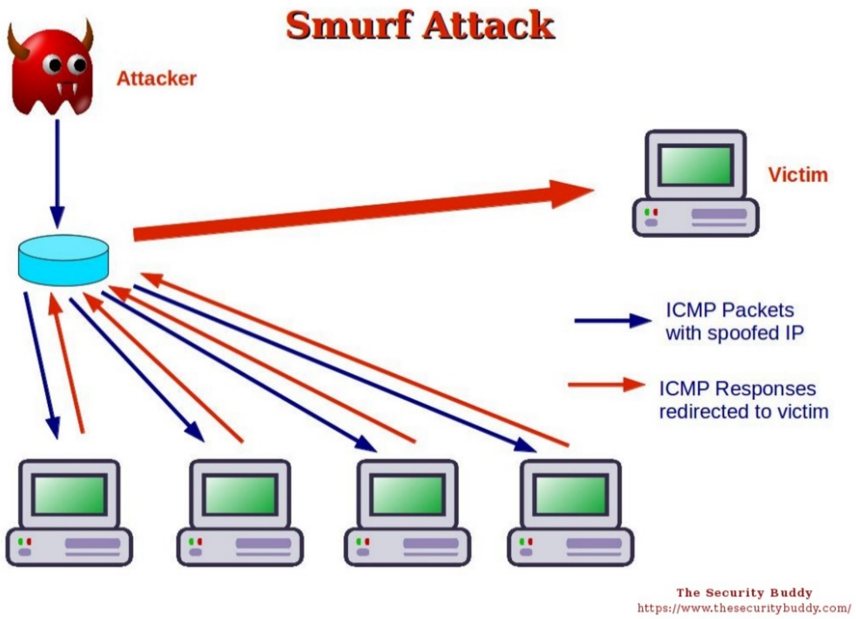
- Send TCP SYN packets with bogus source address
- Half-open TCB entries exist until timeout
- Kernel limits on # of TCBs!!!

**Resources exhausted ⇒ new requests rejected**

# Example 3 – More attacks

Smurf attack

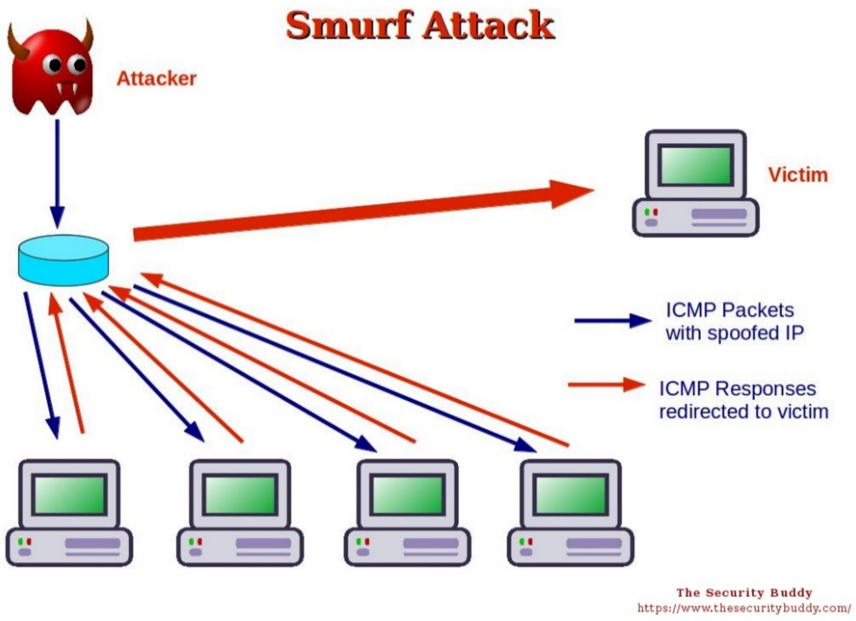
Broadcast Internet Control Message Protocol (ICMP)
Use the target victim's address as the ICMP source address
All the receivers send the response to the victim



# Example 3 – More attacks

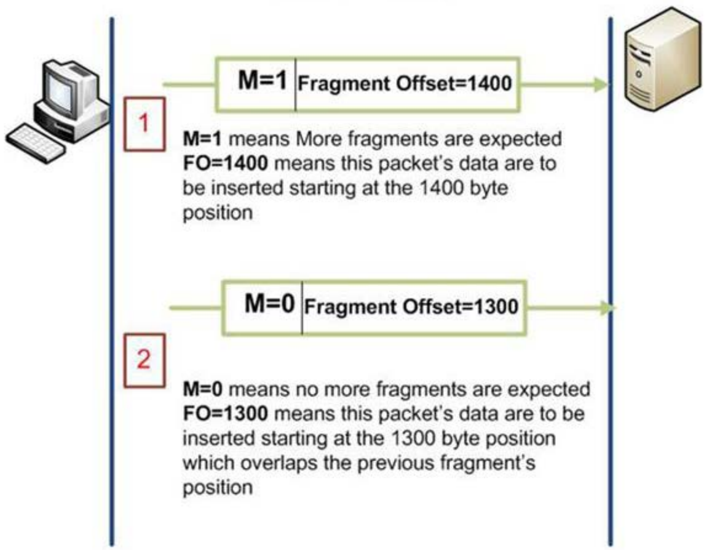
## Smurf attack

Broadcast Internet Control Message Protocol (ICMP)
Use the target victim's address as the ICMP source address
All the receivers send the response to the victim




## Teardrop attack

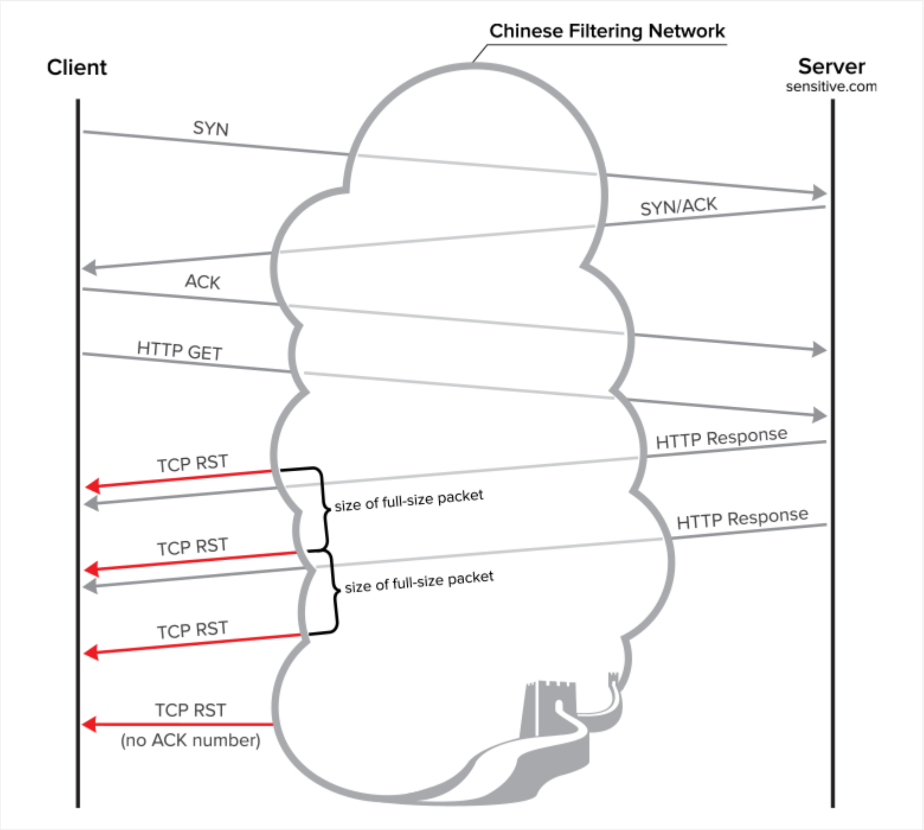
Give the victim fragmented packets with fake information so that it waits indefinitely for packets that never arrive




https://www.pluralsight.com/blog/it-ops/ping-of-death-and-dos-attacks

# Example 4 – DoS without flooding: TCP RST Injection
(e.g., used by the Great Firewall of China)

When the Great Firewall detects an undesired flow, it injects forged TCP resets (with the RST flag bit set) into the data streams so that the endpoints abandon the connection.




https://knowledgebase.paloaltonetworks.com/KCSArticleDetail?id=kA10g000000ClIyCAC

# DoS Prevention – use "cookies"

**Principle**: Minimize the state before you are "authenticated" (i.e., finish 3-way handshake)

Don't create the full TCB

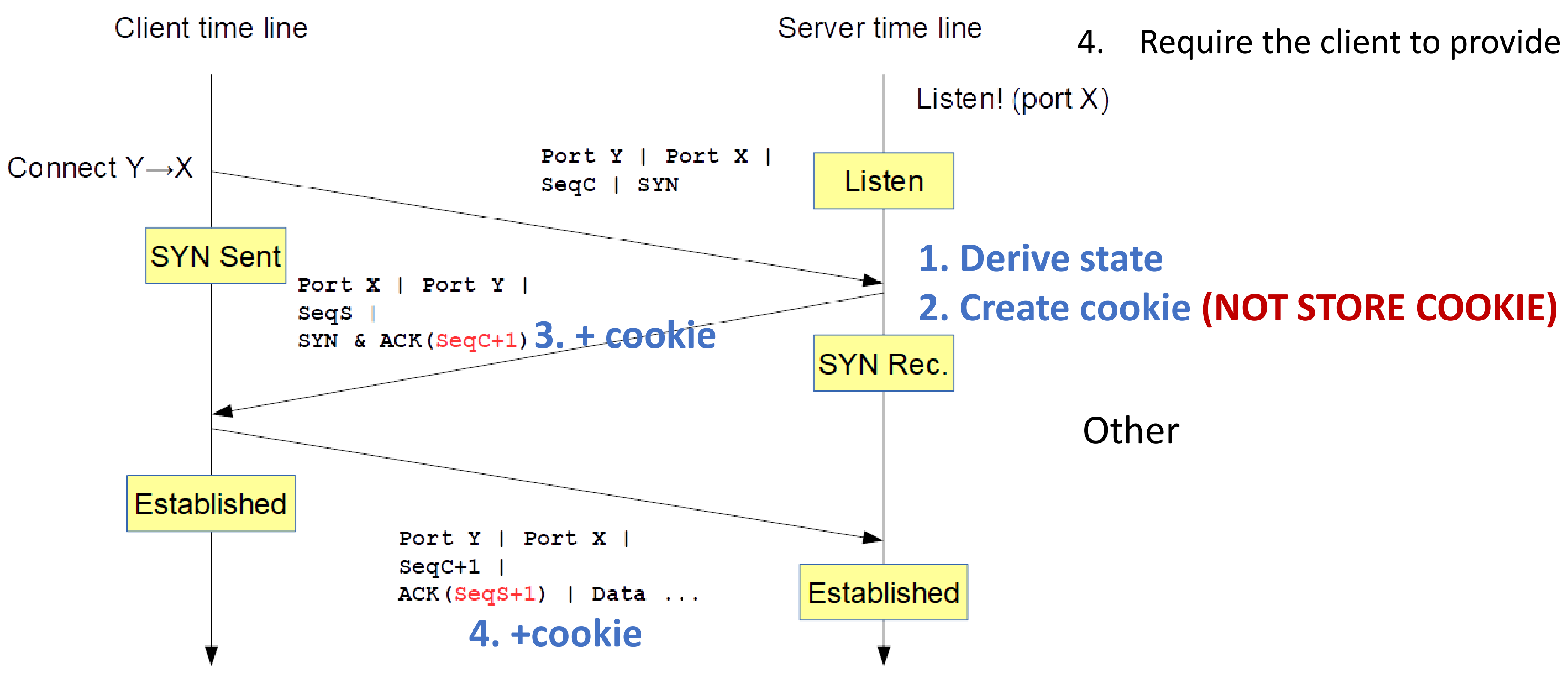
- Compressed TCP state: Very tiny state representation for half-open conns
- A few bytes per connection == can store 100,000s of half-open connections

**Push the state to the client!** "DoS prevention cookies"

- Upon receiving a message, derive the state
- Cryptographically protect the state under a fixed key (confidentiality and integrity)
- Send it back to the client
- Require the client to provide it back to complete the protocol

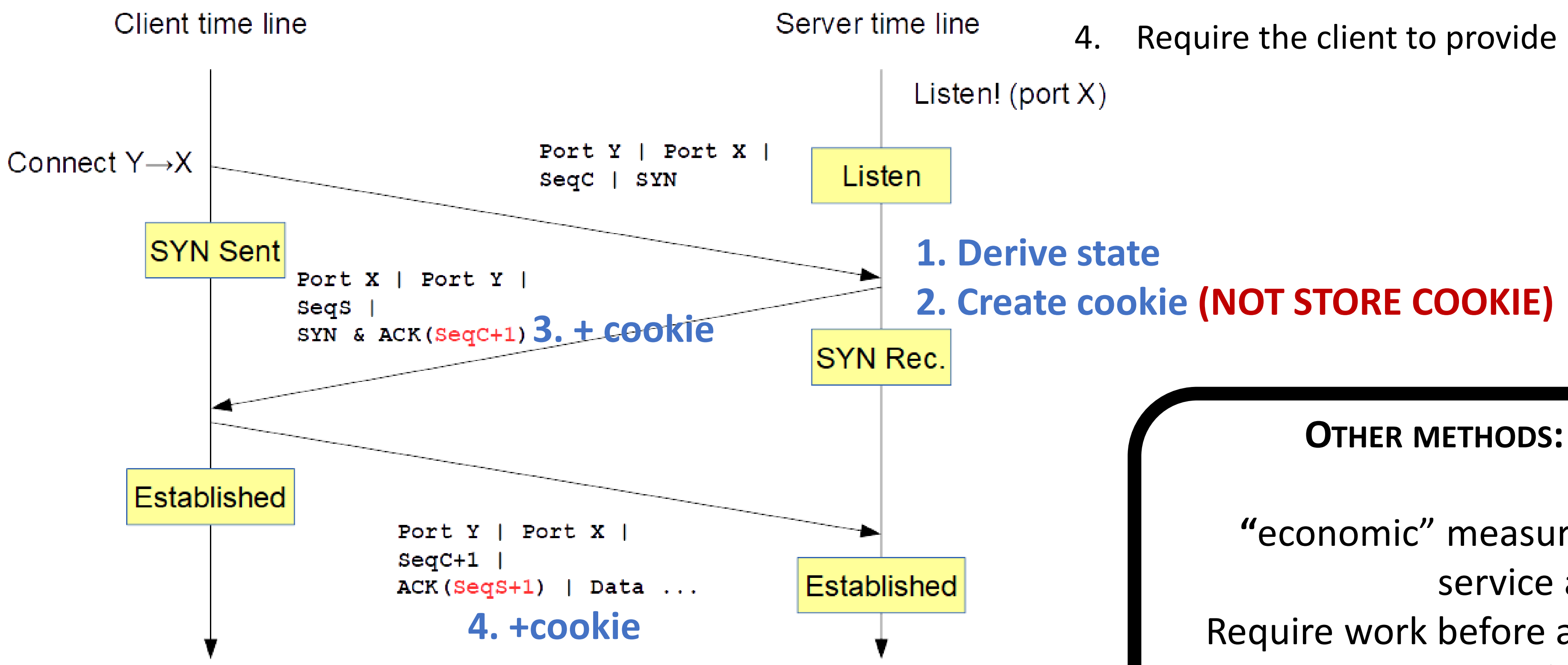
# DoS Prevention – use “cookies”

1. Upon receiving a message, derive the state
2. Cryptographically protect the state under a fixed key (confidentiality and integrity)
3. Send it back to the client
4. Require the client to provide it back to complete the protocol



# DoS Prevention – use “cookies”

1. Upon receiving a message, derive the state
2. Cryptographically protect the state under a fixed key (confidentiality and integrity)
3. Send it back to the client
4. Require the client to provide it back to complete the protocol



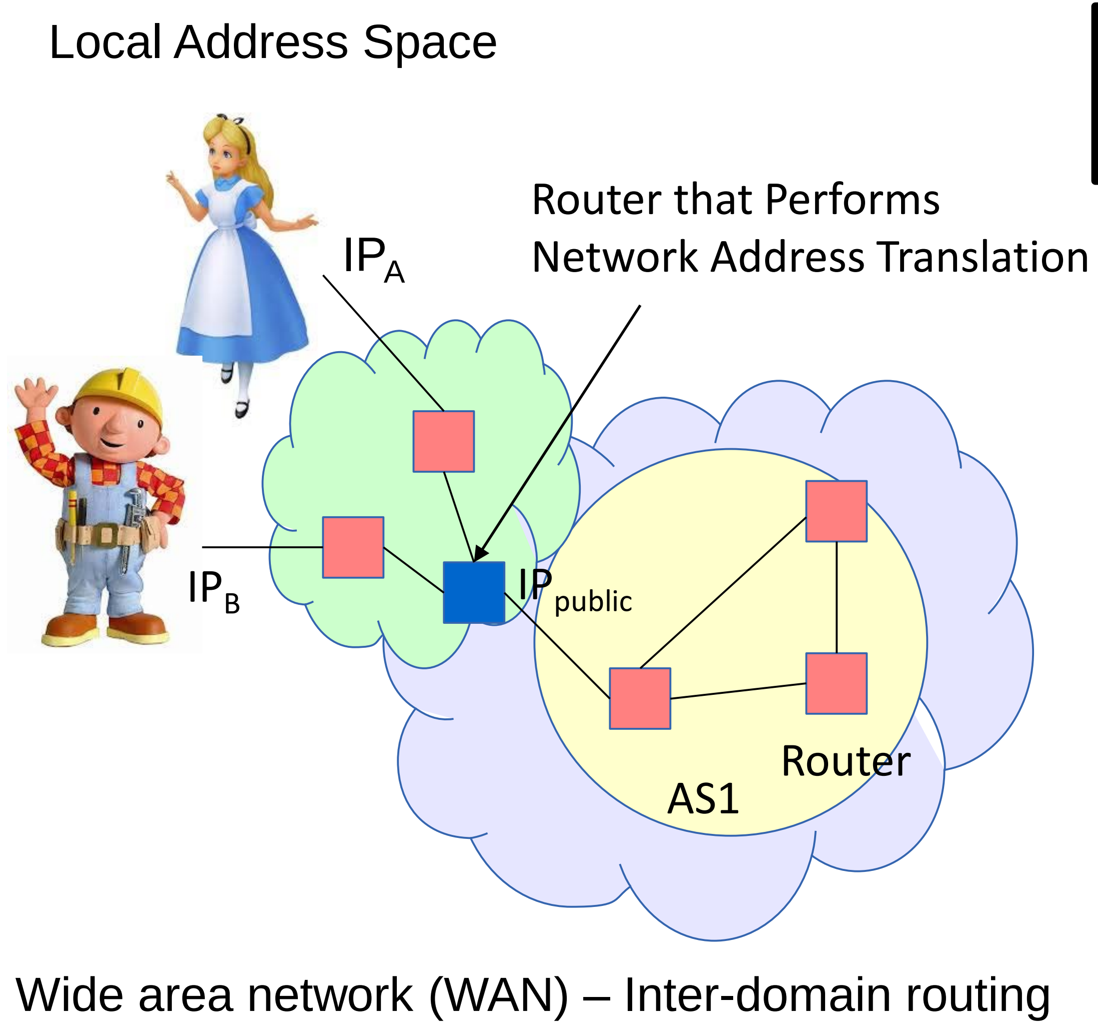

**OTHER METHODS: PROOF OF WORK**

“economic” measure to deter denial of service attacks.
Require work before anything is done (e.g., compute some hashes). Easy to do once and to verify, expensive to do many and DoS

# Other protection technologies

- **Remember**: cryptography is the key.
- But other solutions can help:
  – NATs
  – Firewalls
  – DMZs
  – Intrustion Detection System

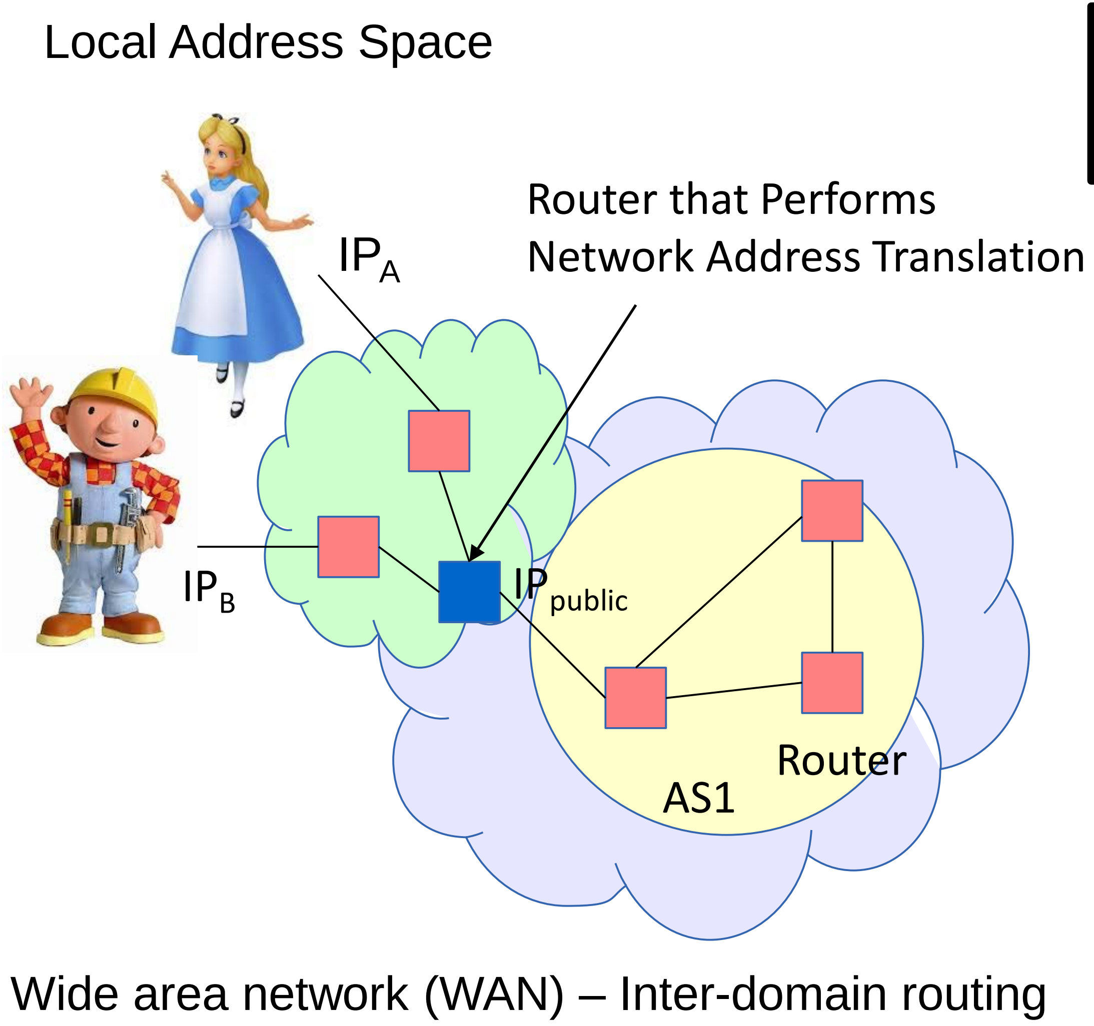
# Network Address Translation (NAT)




**NAT:** router that maintains routing tables of the form: (Internal IP, port) ↔ (External IP, port).

**Why?** Save IPv4 address space (only 32 bits of it!)

# Network Address Translation (NAT)

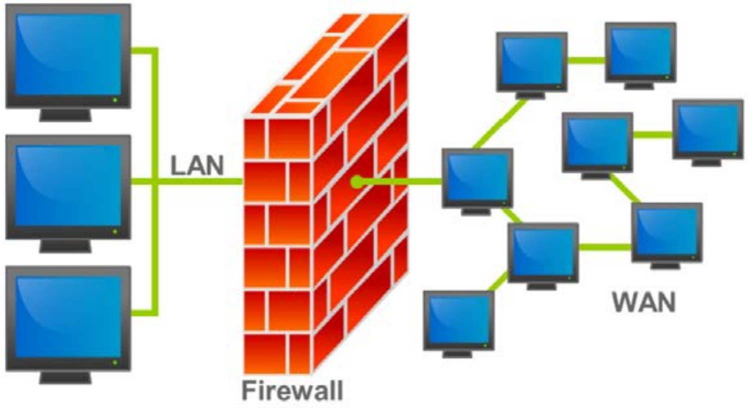



**NAT:** router that maintains routing tables of the form: (Internal IP, port) ↔ (External IP, port).

**Why?** Save IPv4 address space (only 32 bits of it!)

**Security implications:** an external entity *cannot* route into the NAT unless it uses an already mapped port

# Network Firewalls




**FIREWALL:** network router that connects an internal network to an external (public) network. It *mediates* all traffic, and makes “access control” decisions according to a policy.

- Firewall “access control”:
  - Inspects characteristics of the traffic,
  - “Allow” or “deny” traversal across the firewall.
  - Prevent flows that could be dangerous or contravene a security policy in the internal network.

# Firewalls - Simple packet filter (1980s)

Inspect each packet in isolation and Reject/Allow depending on certain “rules”

**Rules:**

- “Equal” or “not equal”, or “in range”.
- Fields: IP Src, IP Dest, Port numbers, Protocol Type.
- **Example**: Force all email traffic to go to a specific mailserver:
  `(Dest IP = mailserver, Dest Port = 25) → Allow`
- `(Src IP = mailserver, Dest Port = 25) → Allow`
  `(Dest IP = *, Dest Port = 25) → Deny`
- `(Src IP = *, Dest Port = 25) → Deny`

**Advantages**: simple to implement, instant decisions

**Disadvantages**: Limited policies can be expressed, limited filtering on content possible

# Stateful Firewalls (1990s)

Understand TCP/UDP semantics → can Reject/Allow depending on the state

**Stateful firewall vs. stateless packet filter – Example**

FTP protocol client opens a connection to the server, and then the server connects back to a high port of the client to transfer the file

- *Simple packet filter (stateless firewall)*: choice between allowing all packets to high ports all the time or none
- *Stateful firewall*: can detect an active FTP session with the server and allow a connection back to a high port from the same server to the same client!

# Application Firewalls (1990s)

**DEEP PACKET INSPECTION** (DPI): evaluate the content, and allow/ reject based rules

can be statefull or stateless!

**Examples**:

- Transparent redirection of HTTP traffic to a local proxy to save bandwidth
- Transparent blocking of certain websites (social networks from a workplace)
- Scanning downloaded executable resources for viruses
- Blocking peer-to-peer protocols, no matter which port they use
- Monitoring traffic to detect leaks of sensitive documents

**And if traffic is encrypted (IPSec, SSL/TLS)?**

- Option 1: block all encrypted traffic.
- Option 2: Install client certificates that allow for decryption & inspection at the firewall.

# Downsides with firewalls

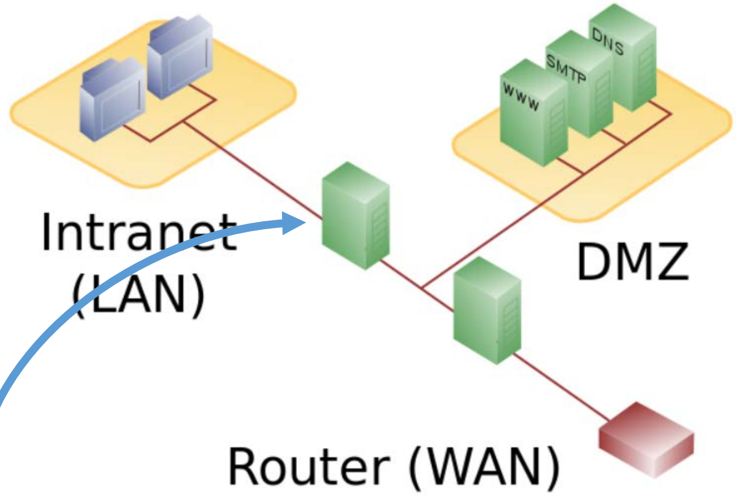
**Key problems**

- Full mediation is **slow** (read/check/write) – observation is cheaper (read/inject).
- Can a firewall authenticate any principals?
- Can a firewall ensure the correctness of the data on which it makes decisions?

**Role of firewall in security engineering**

- Only allow “known good traffic”? ← not possible at this level (what is good traffic in an intranet?)
- Therefore: “filter out definitely bad traffic” and “filter all traffic of a certain class”.
- Remove the noise of background network attacks
- Hosts will still have to implement robust defences to hinder that bad behaviour can be use to “spoof” good characteristics cannot violate the security policy
- **Key lesson**: a firewall **is not** a full substitute for other host and network security mechanisms!

# Defence in depth: the De-Militarized Zone (DMZ)



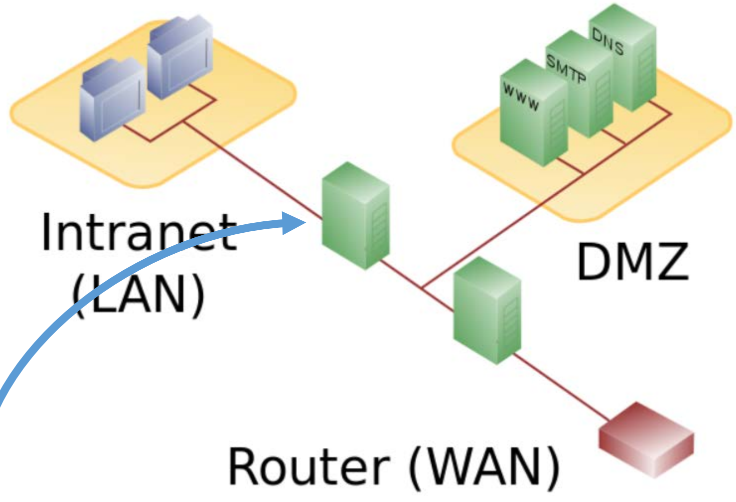

Split “the world” into 3 zones

- **WAN** – outside
- **DMZ** – with public services
- **LAN** – for internal users only

Relies on a firewall to

- Ensure only traffic to well known services traverses outer firewall.
- Ensure only traffic from “**bastion host**” enters LAN from DMZ. Thus the bastion host can perform access control and filtering (eg. VPN/IPSec, Proxy).
- Result: LAN can access DMZ and WAN; DMZ can access WAN. But flows in the other direction are restricted, monitored and authenticated.
- In case a service is compromised internal resources are safe!

# Defence in depth: the De-Militarized Zone (DMZ)




Split "the world" into 3 zones

- **WAN** – outside
- **DMZ** – with public services
- **LAN** – for internal users only

Relies on a firewall to

- Ensure only traffic to well known services traverses outer firewall.
- Ensure only traffic from "**bastion host**" enters LAN from DMZ. Thus the bastion host can perform access control and filtering (eg. VPN/IPSec, Proxy).
- Result: LAN can access DMZ and WAN; DMZ can access WAN. But flows in the other direction are restricted, monitored and authenticated.
- In case a service is compromised internal resources are safe!

# Summary

- The network is **hostile – insiders can be as evil as outsiders**

- **Cryptography is a basic tool** to address network security problems
  - Authenticity, Confidentiality and integrity of traffic content and sessions
  - Authentic binding of names → secure naming (DNS, ARP)
  - Authenticity of routes & routing updates → secure routing (BGP)
  - Strong authentication allows for reliable authorization

- Denial of service **can be defeated**
  - Try to **not keep state** and try to **make the adversary work**

- Other techniques are ultimately **weak(er)**:
  - Firewalls, IDS, filtering, … → weak against strong network adversaries that can MITM
  - They may protect against weak adversaries and/or provide some defence in depth.