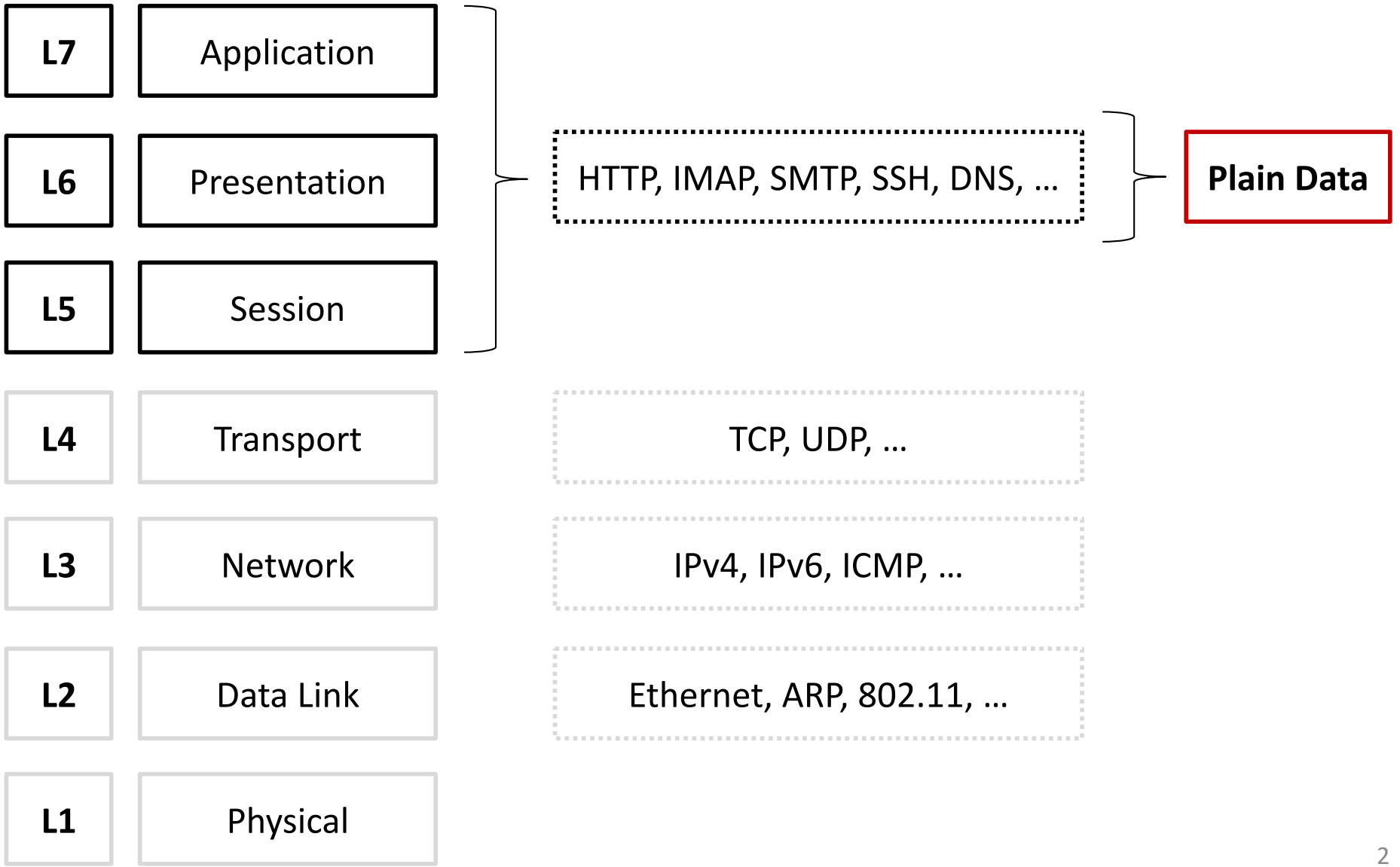
CS326 – Systems Security

Lecture 19

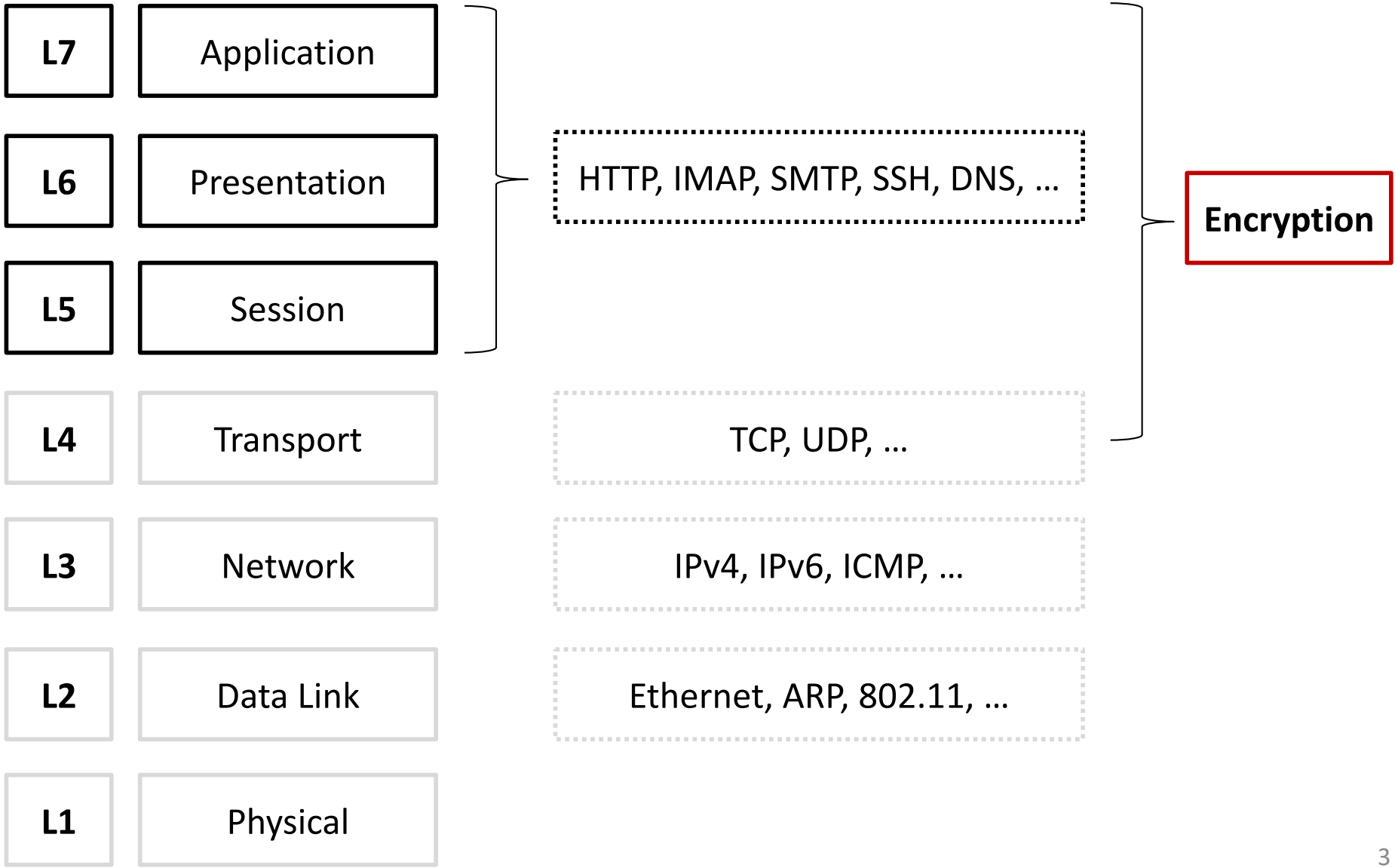
Transport Layer Security (TLS)

Elias Athanasopoulos
athanasopoulos.elias@ucy.ac.cy

Network Layers



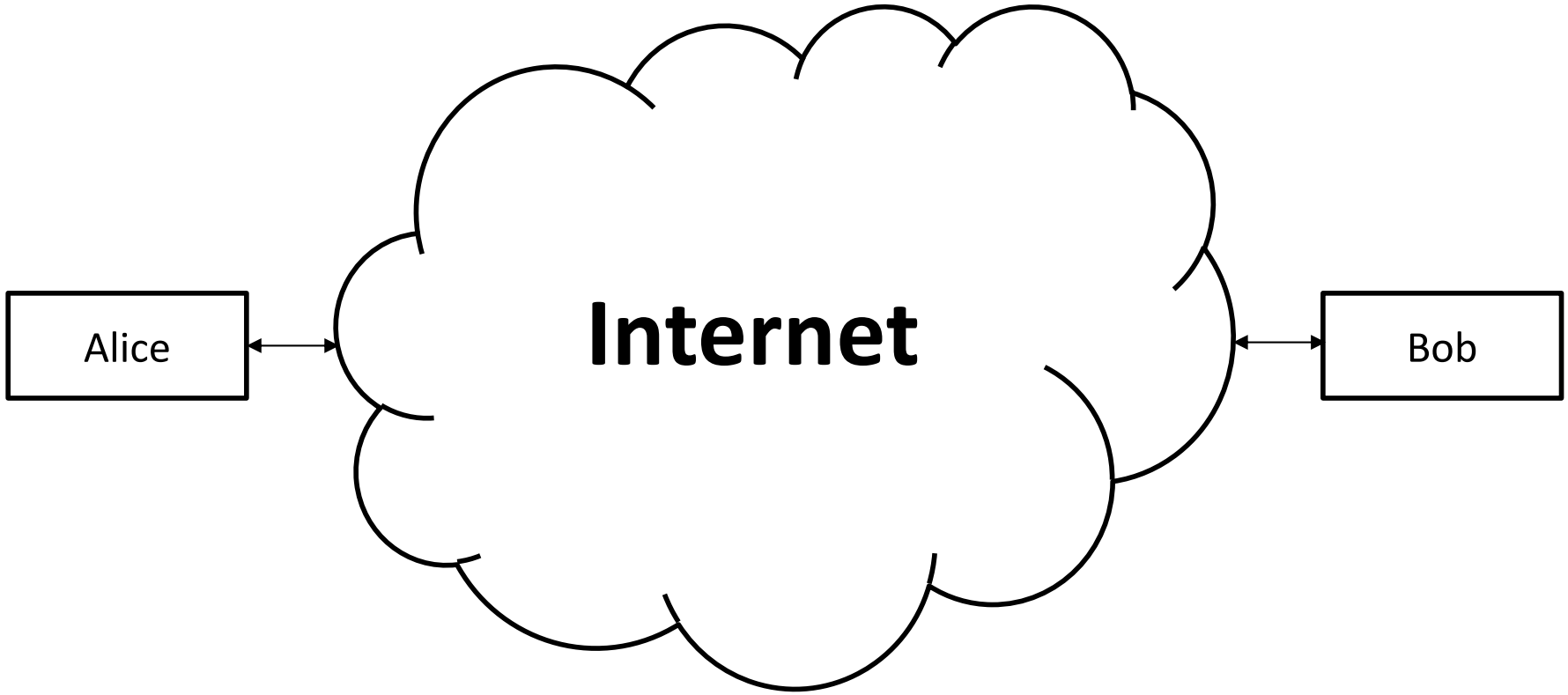
Network Layers and TLS



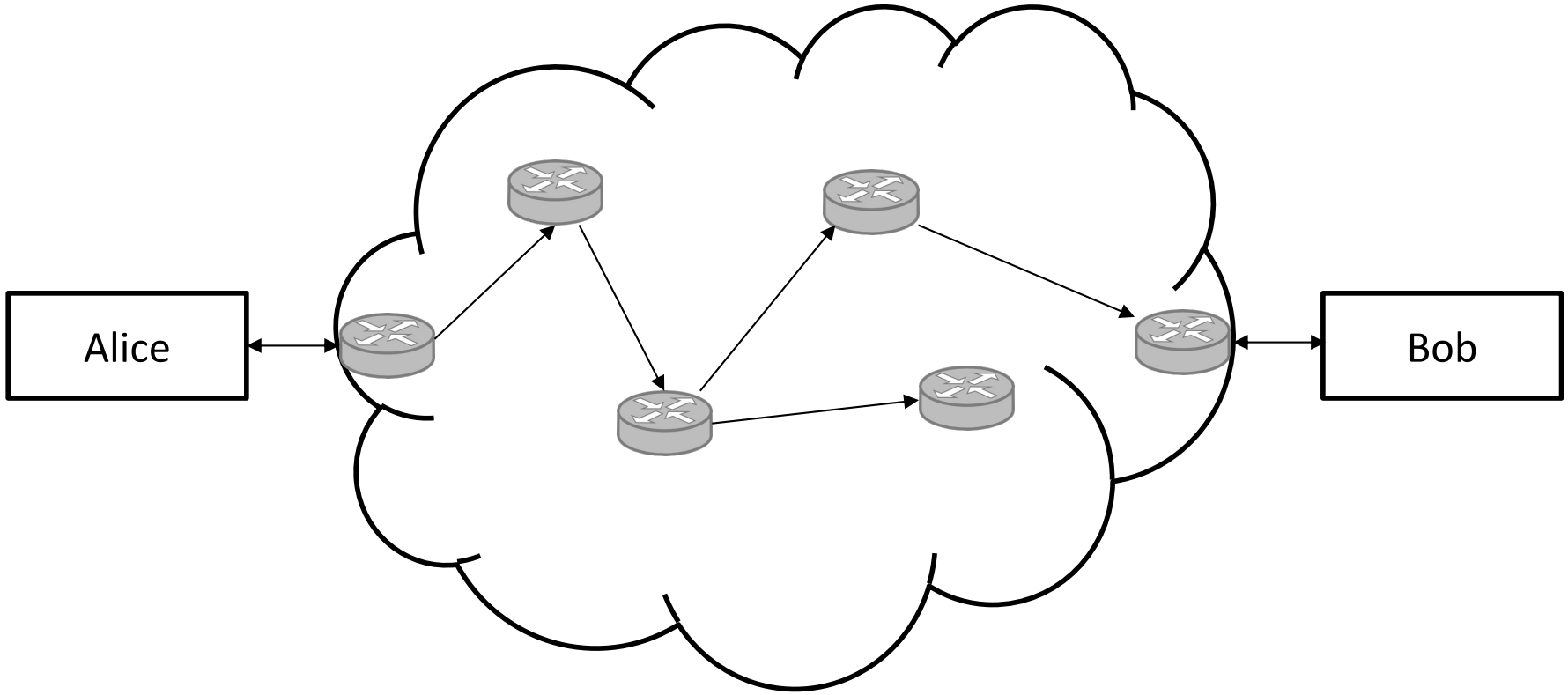
Transport Layer Security (TLS)

- Allows applications to communicate over the network using encryption
  - Sockets that send encrypted data
- Designed for the following requirements
  - Confidentiality, Integrity, Authentication
- Many applications support it
  - HTTPS, e-mail protocols, SSH, etc.
  - Usually TLS is supported in a different port (e.g., HTTP is on 80, and HTTPS is on 443)

Why?



Many Intermediate Nodes (routers)



Man-in-the-Middle Attack (MitM)

- Intermediate nodes can be attacker controlled
  - Routers
  - Gateways
  - Wireless access points
  - Proxies
- Plain traffic can be compromised
  - Monitored (confidentiality), leak passwords, credit cards, etc.
  - Modified (integrity), change the contents of an e-mail, of a financial transaction, etc.

History of TLS

Netscape

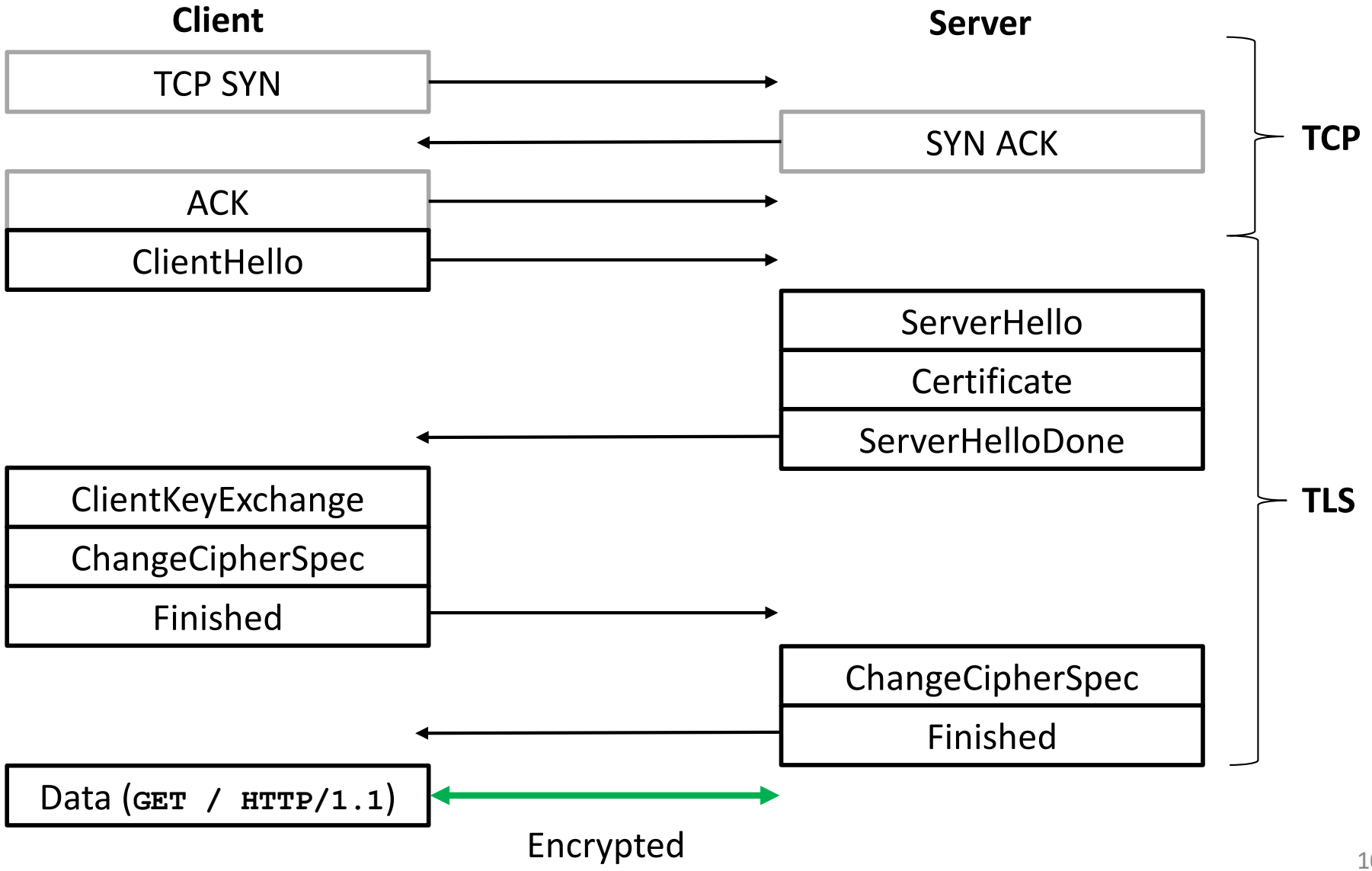
Protocol	Year
SSL 1.0	n/a
SSL 2.0	1995
SSL 3.0	1996
TLS 1.0	1999
TLS 1.1	2006
TLS 1.2	2008
TLS 1.3	**draft/working**

We are here

Protocol Composition

- TLS Handshake
  - Several slightly different forms based on the *cipher suite* used
  - **Cipher suite**: (a) key-exchange algorithm, (b) bulk encryption algorithm, (c) MAC algorithm
- TLS Record Protocol
  - The part of the protocol that transmits encrypted data

TLS Handshake



`ClientHello`
`ServerHello`

- Client advertises the ciphers it supports

  TLS_RSA_WITH_AES_128_GCM_SHA256
  TLS_RSA_WITH_AES_128_CBC_SHA
  TLS_RSA_WITH_RC4_128_SHA
  …

- Server selects one from the list
- Server sends its certificate
- **`ServerHelloDone`** announces that there are no more messages from the server at this point

`ClientKeyExchange`

- The client encrypts a secret with the server's public key (found in the certificate)
- **`ChangeCipherSpec`** signals that from now on messages will be encrypted
- **`Finished`**, the first message to be encrypted and the client's last message of the handshake, contains a MAC (cryptographic checksum) of all handshake messages exchanged

`ChangeCipherSpec`

- The server decrypts the secret found in the **`ClientKeyExchange`** message using its certificate's private key, and derives the master secret and communication keys
- **`Finished`**, signals a switch to encrypted communication and completes the handshake

TLS Record Protocol

Byte	+0	+1	+2	+3
0	Content type			
1..4	Version		Length	
5..n	*Payload*			
n..m	MAC			
m..p	Padding (block ciphers only)			

The right record size

- Small records have larger CPU overhead due to frequent MAC verification
- Large records will have to be reassembled by the TCP layer before they can be processed by the TLS layer
- Not always possible to tune the record size

Authentication

- During a TLS handshake the server sends a *certificate* to the client
- The certificate is an electronic document (X.509) used to prove the ownership of a public key
- The certificate includes information about the key, the identity of its owner, and the digital signature of an entity that has verified the certificate's contents (called the issuer)
- If the signature is valid and the issuer is trusted then the public key is accepted

SSL Stripping

- Many web sites support both HTTP and HTTPS
- Browser may initially connect to HTTP
- The server then can redirect to HTTPS
- An MtM attacker can modify the server response and change all HTTPS links to point back to HTTP

  `<form action="https://login.site.com">`

  becomes

  `<form action="http://login.site.com">`
- All session is not encrypted!

HSTS

- HTTP Strict Transport Security
  - Policy, which is communicated by the server to the web browser over HTTPS
  - Declared in a field named `Strict-Transport-Security`
  - HSTS Policy specifies a period of time during which the user agent should only access the server using HTTPS
  - Browsers have an internal list of HSTS web sites