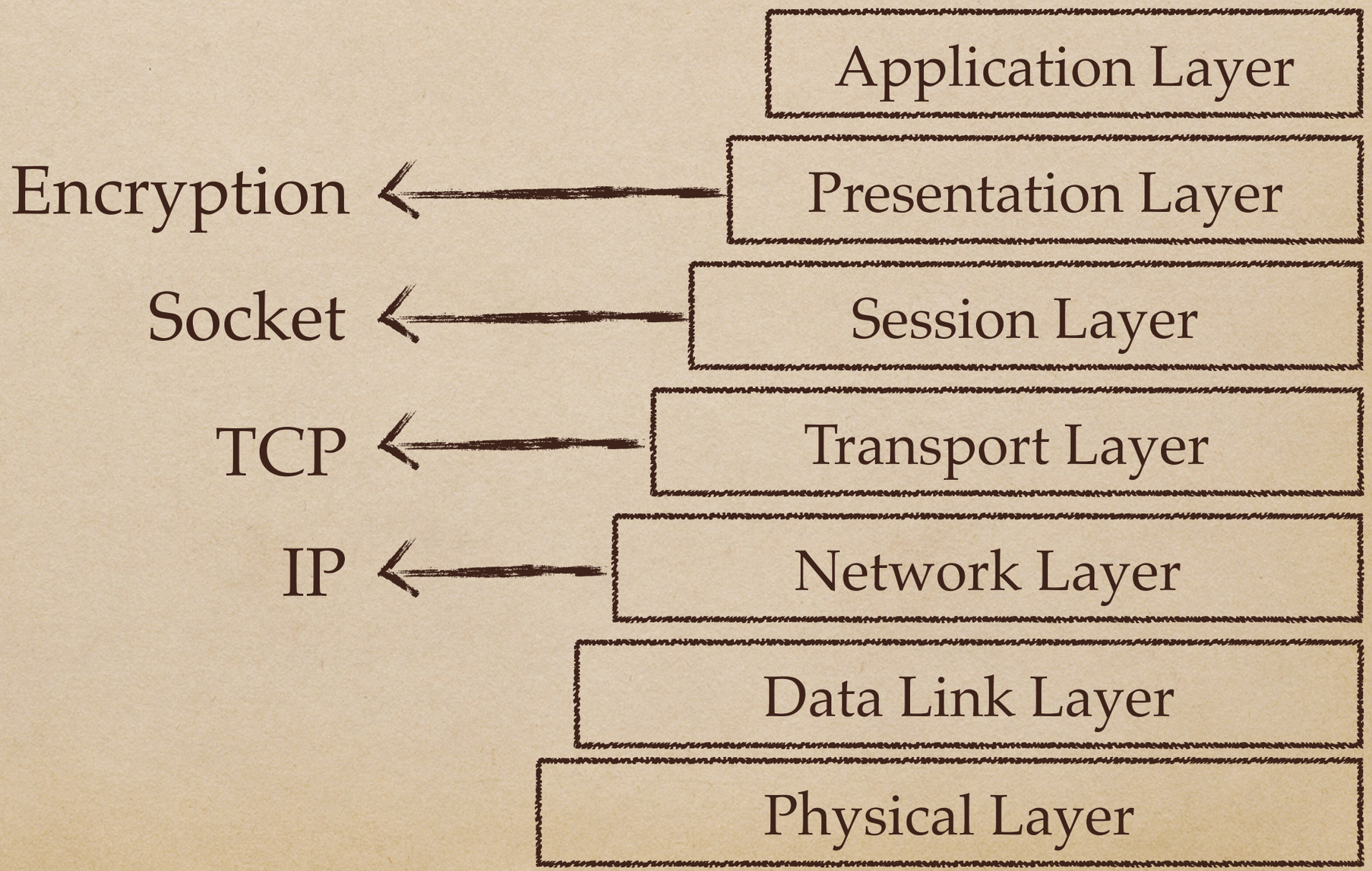
HW2

Network Security

TLS/SSL Implementation

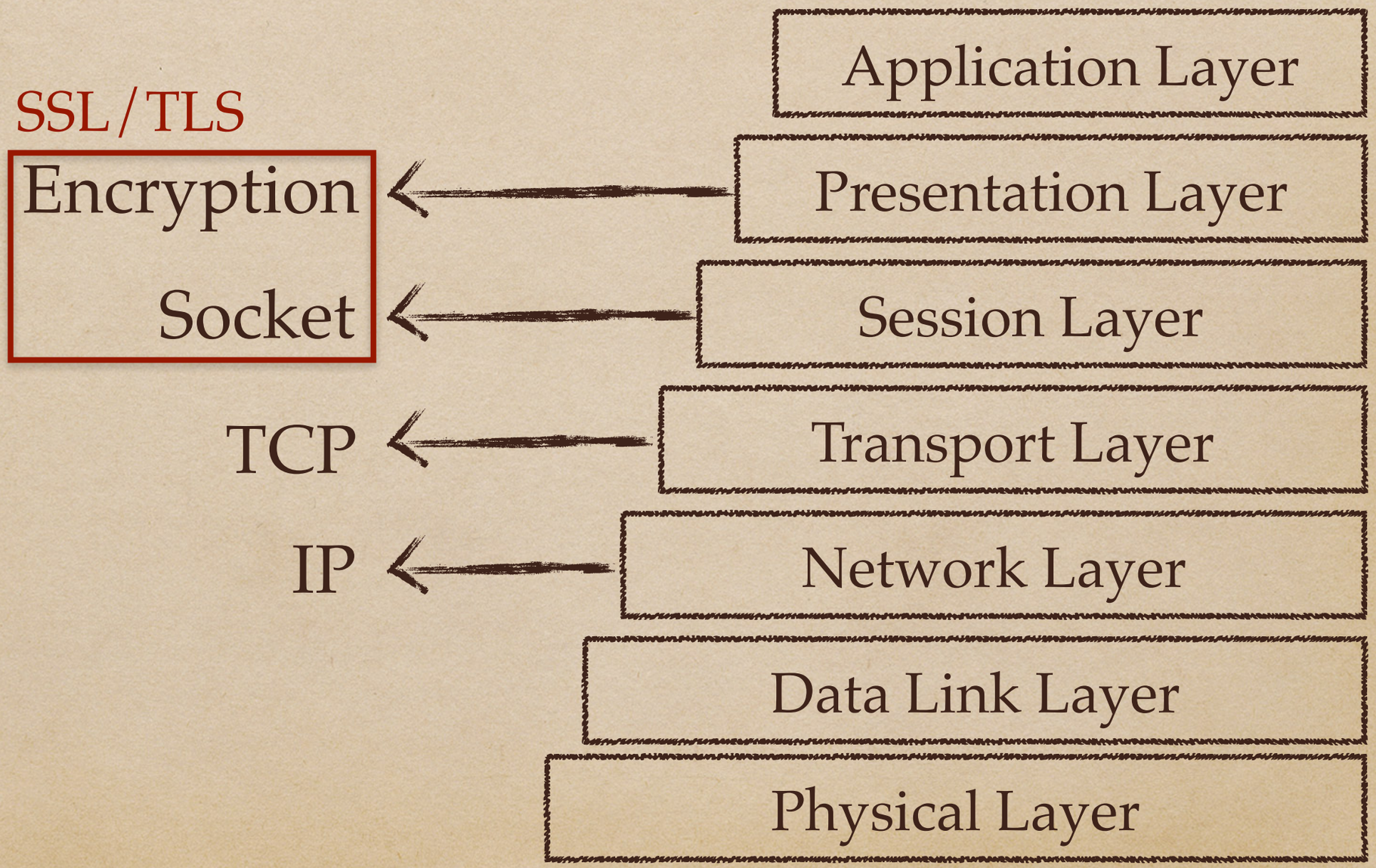
TLS/SSL

Transport Layer Security/Secure Sockets Layer



TLS/SSL

Transport Layer Security/Secure Sockets Layer



SSL and TLS

SSL was a first of its kind of cryptographic protocol.

TLS was a recent upgraded version of SSL.

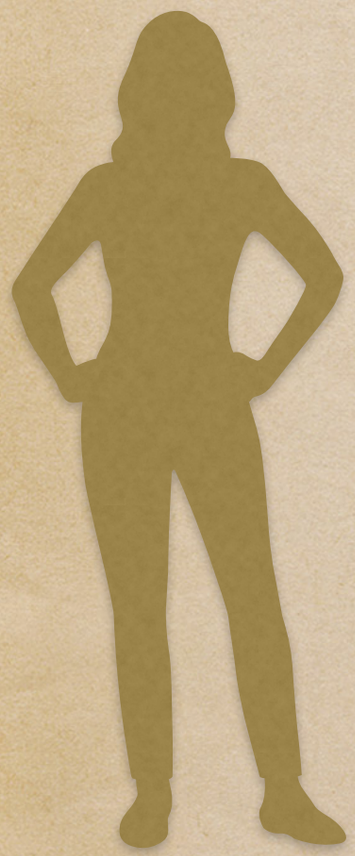
SSL and TLS protocols

Protocol	Published	Status
SSL 1.0	Unpublished	Unpublished
SSL 2.0	1995	Deprecated in 2011 (RFC 6176)
SSL 3.0	1996	Deprecated in 2015 (RFC 7568)
TLS 1.0	1999	Deprecated in 2020[11][12][13]
TLS 1.1	2006	Deprecated in 2020[11][12][13]
TLS 1.2	2008	
TLS 1.3	2018	

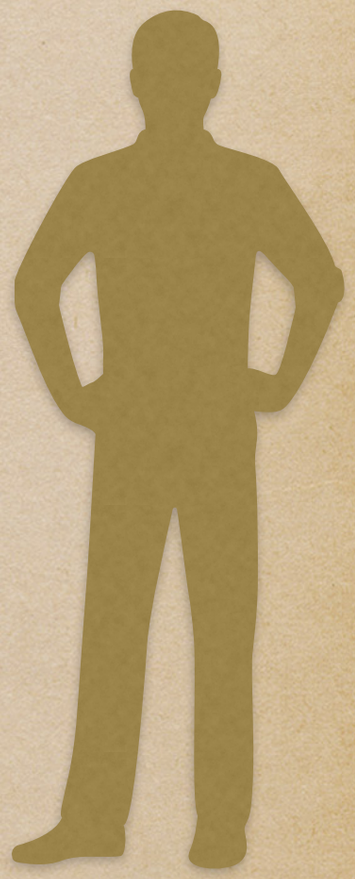
https://zh.wikipedia.org/wiki/傳輸層安全性協定
https://www.ssl2buy.com/wiki/ssl-vs-tls

TLS 1.2

TCP Handshake



Client



Server

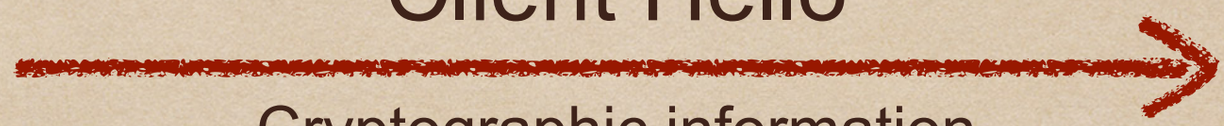
Piv

Pub

TLS 1.2

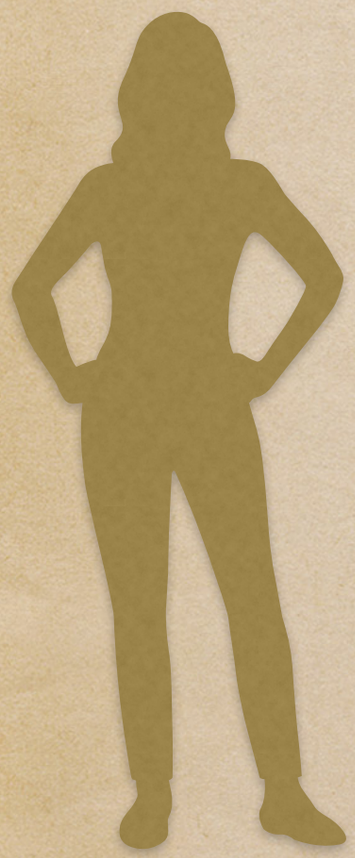
TCP Handshake

Client Hello

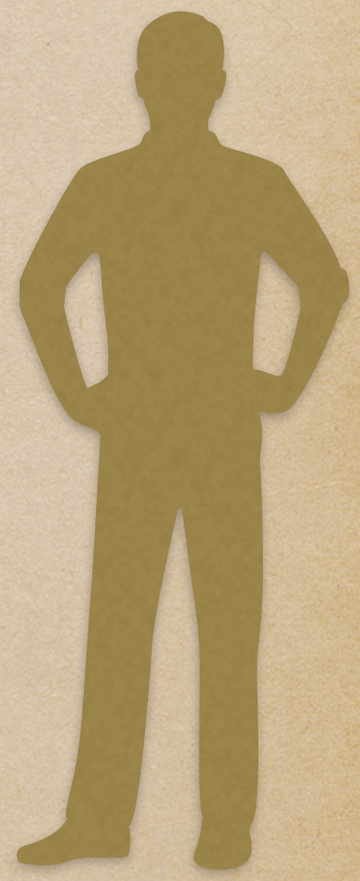


Cryptographic information

TLS_ECDHE_RSA_WITH_AES_256
_GCM_SHA384



Client



Server

Piv

Pub

https://tls.ulfheim.net
https://tools.ietf.org/html/rfc5246

TLS 1.2

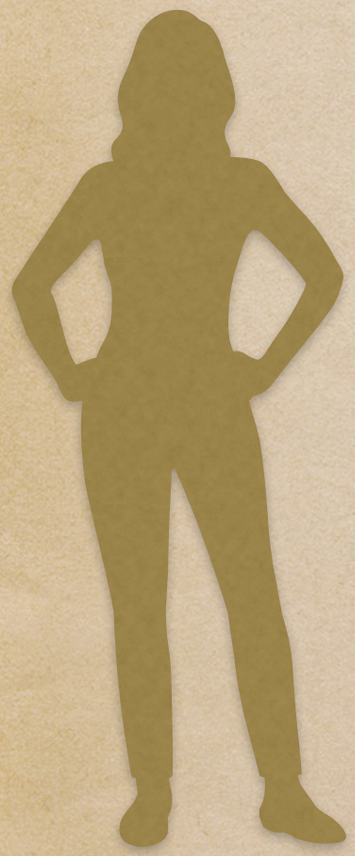
TCP Handshake

Client Hello

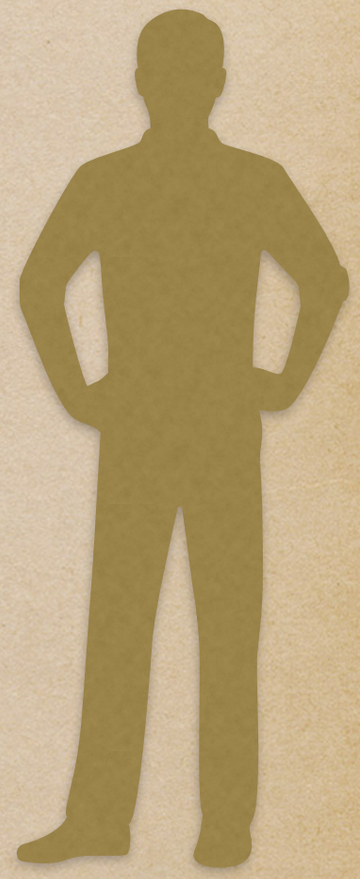
Server Hello

Cryptographic information

Server Certificate



Client

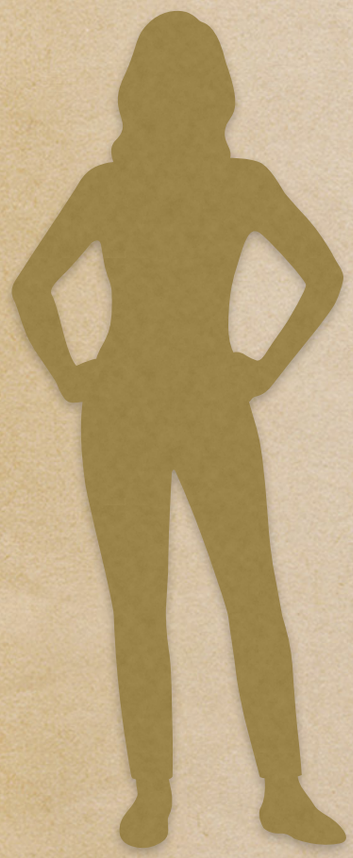


Server

Piv

Pub

TLS 1.2



TCP Handshake

Client Hello

Server Hello

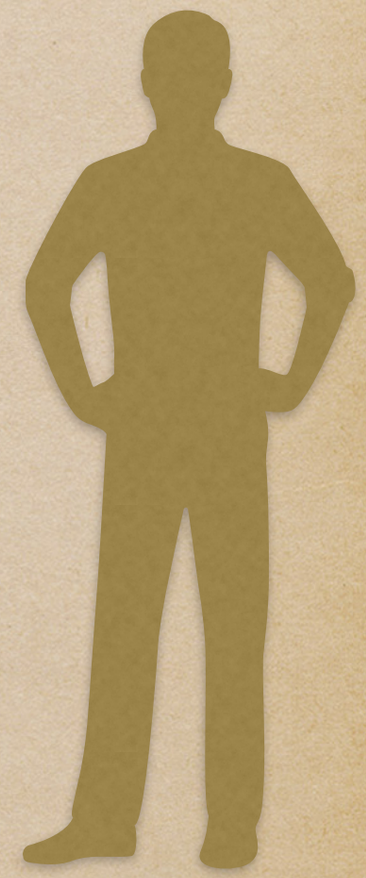
Server Certificate

Server Key Exchange

Server Hello Done

Client Key Exchange

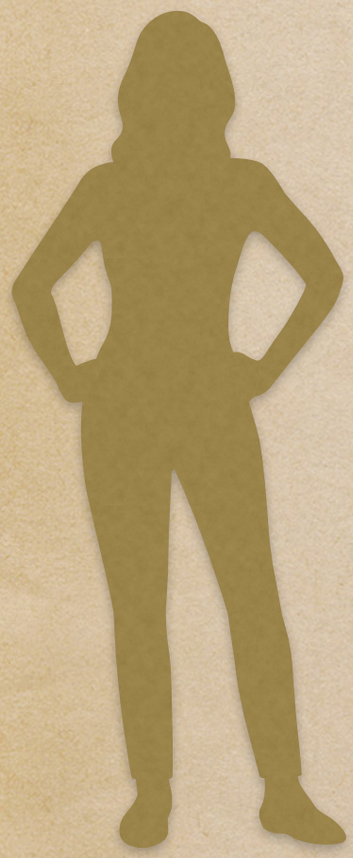
Client Hello Done



Client

Server

TLS 1.2



TCP Handshake

Client Hello

Server Hello

Server Certificate

Server Key Exchange

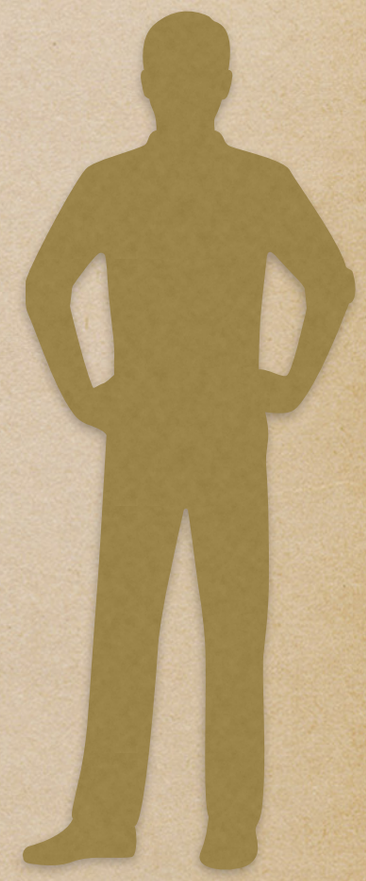
Server Hello Done

Client Key Exchange

Client Hello Done

Change Cipher Spec

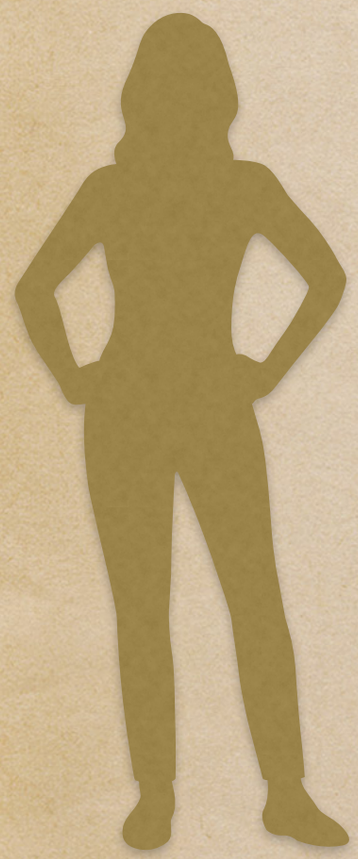
Start to use Symmetric Encryption



Client

Server

TLS 1.2



TCP Handshake

Client Hello

Server Hello

Server Certificate

Server Key Exchange

Server Hello Done

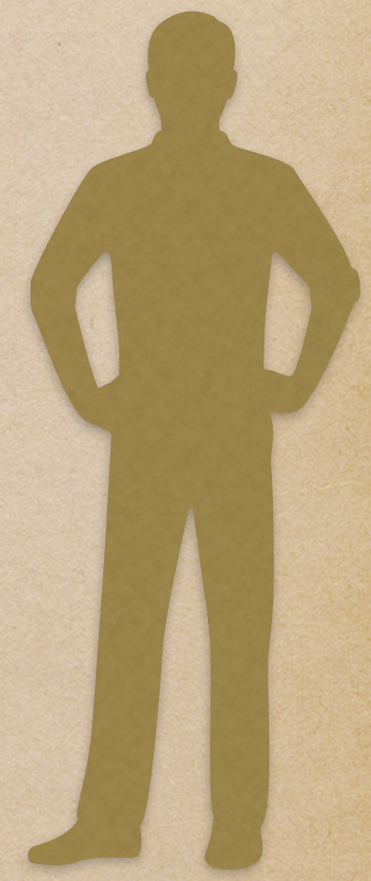
Client Key Exchange

Client Hello Done

Change Cipher Spec

Finished Message

Verify that the handshake was successful



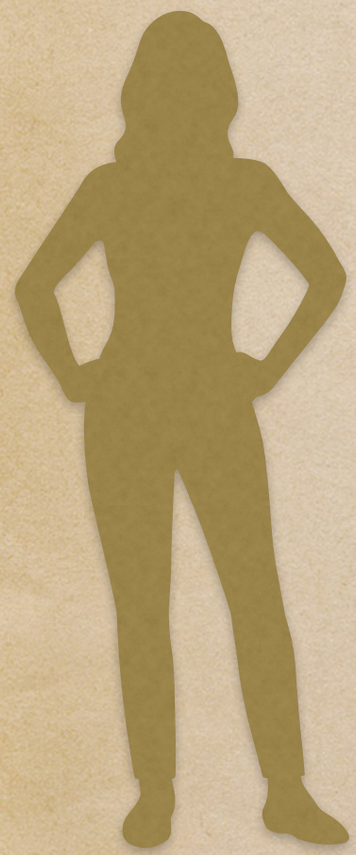
Client

Server

Piv

Pub

TLS 1.2



TCP Handshake

Client Hello

Server Hello

Server Certificate

Server Key Exchange

Server Hello Done

Client Key Exchange

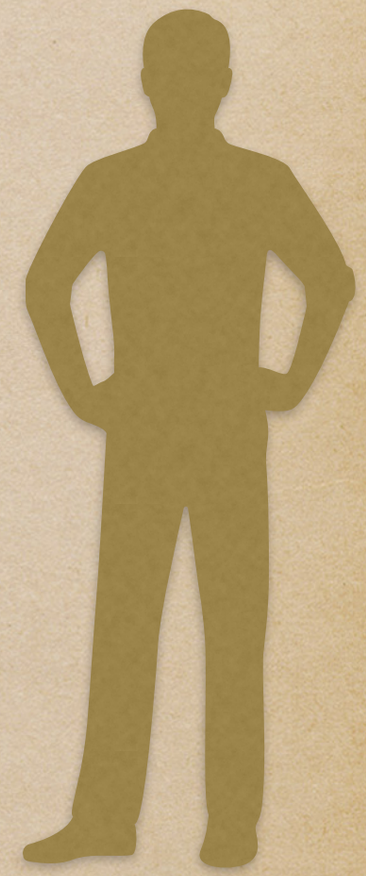
Client Hello Done

Change Cipher Spec

Finished Message

Encrypted Application Data

Start send & recv data



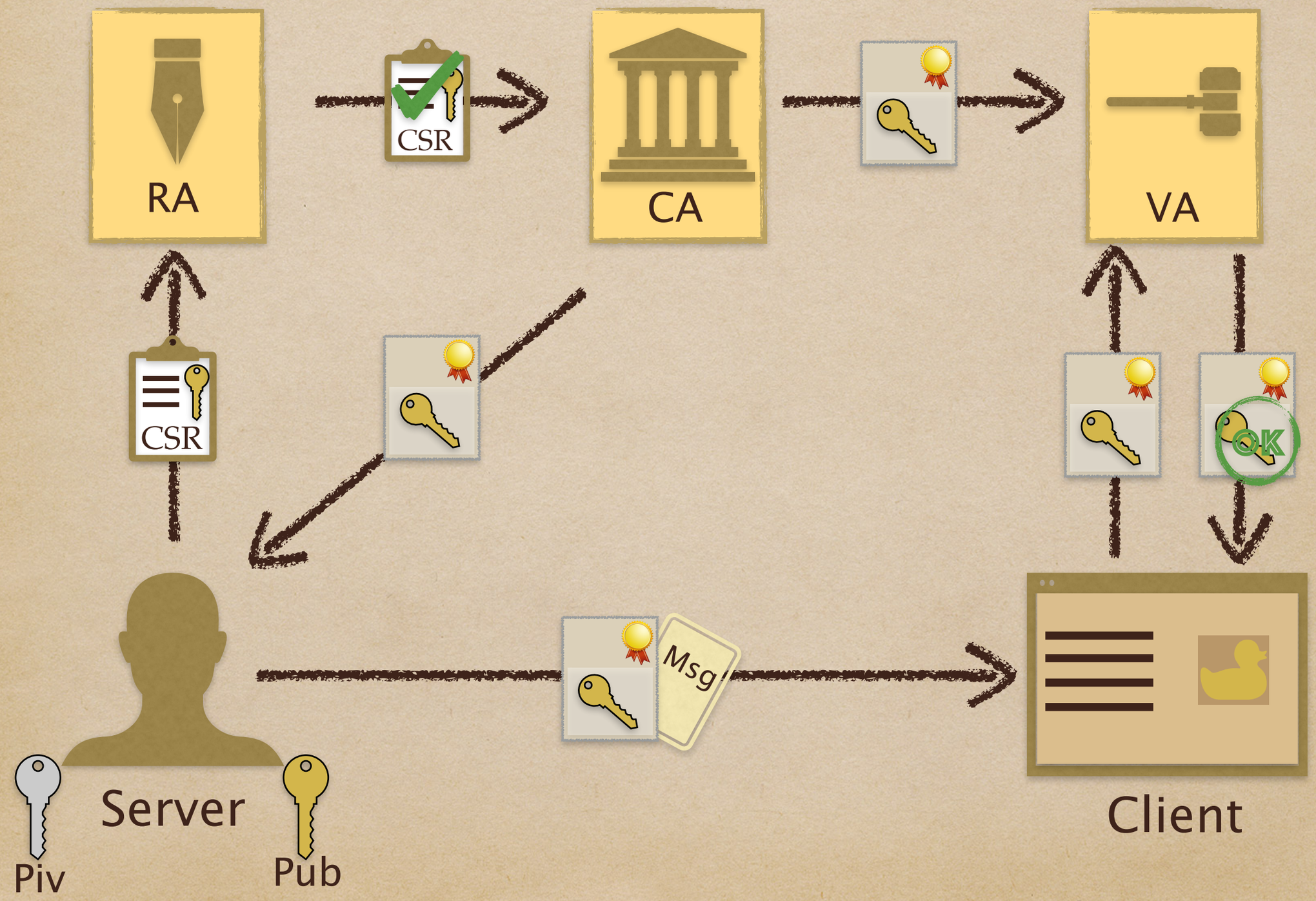
Client

Server

Piv

Pub

SSL certificate



CSR: Certificate Signing Request

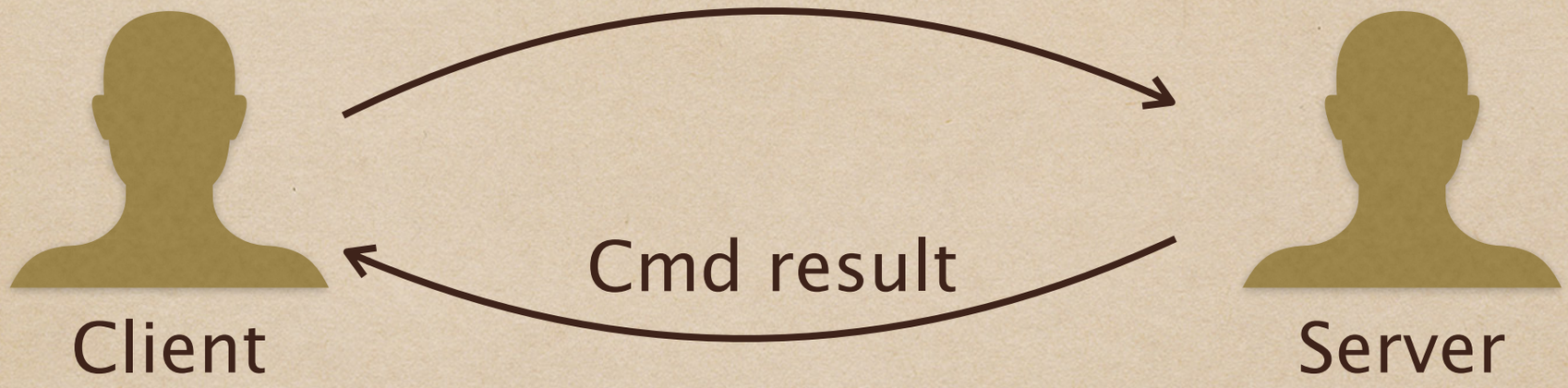
Homework

- 製作一個 Server 和 Client 利用 TLS協定進行資料傳輸
  - 使用自簽憑證
  - 可使用 openssl/ssl.h 的 function 來實作
- Client 端和 Server 端都需要驗證對方的憑證
- 實現的功能(擇1)
  - Remote Shell
  - Remote file copy
  - Message communication

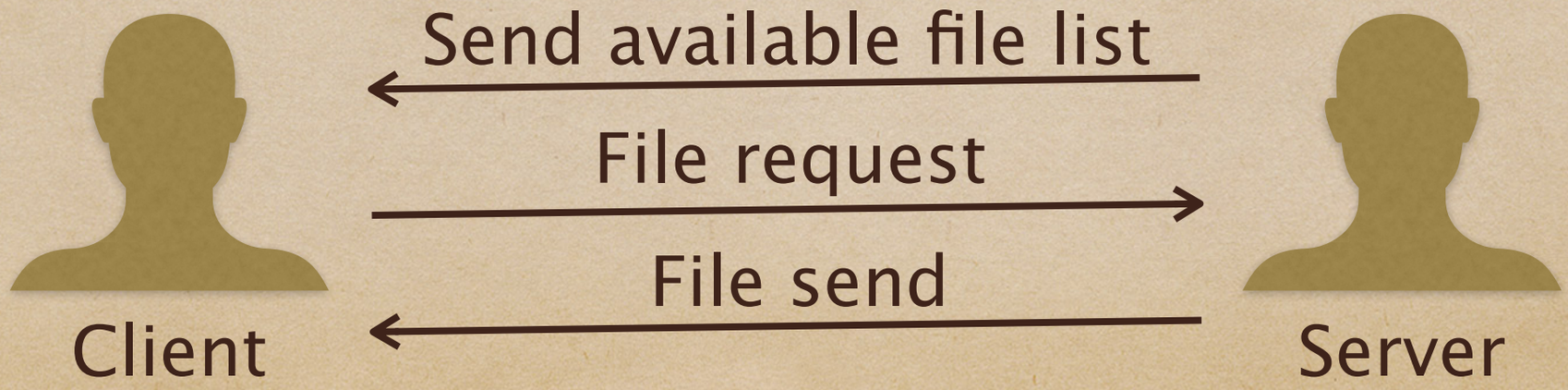
Self-signed certificate

- Openssl
  - sudo apt install libssl-dev (linux)
  - Generate CA
  - Create RSA key-pair
  - Create Certificate Signing Request (CSR)
  - Use our CA to sign the CSR

Remote Shell



Remote file copy



Programing Language

- C & C++
- Python (-10)

作業繳交規定

- 請將程式碼以及報告一起壓縮，並上傳壓縮檔即可
- 壓縮檔請命名為 學號_姓名_HW2
- 報告不限制用什麼編輯軟體但請輸出成 PDF

報告內容

- 建置環境與使用說明
- 重要程式碼說明
- 設計架構與功能說明
- 成果截圖
- 困難與心得

評分

- 報告(30%)
- 程式(60%)，如果使用 python 則為(50%)
- Bonus(10%)

Review

- Socket server & client
- TLS/SSL
- Self-signed certificate