Introduction to TLS

jong.sh/tlsfcu

jong.sh/tlsfcu

我是誰

- 吳忠憲
- 台大快速密碼學實驗室
- `js <at> jong.sh`

大綱

- 密碼學工具
- TLS 簡介
- TLS handshake
- TLS cipher suites
- Forward secrecy
- Homework preview

大綱

- 密碼學工具
- TLS 簡介
- TLS handshake
- TLS cipher suites
- Forward secrecy
- Homework preview
- Python programming
- TLS records
- Decoding TLS
- More info on TLS

密碼學工具

- Authenticated Encryption
- Public Key Encryption
- Public Key Digital Signature
- Authenticated Key Exchange
- Public Key Infrastructure

電腦網路通訊之威脅

- 阻斷服務攻擊
- 針對系統漏洞進行利用
- 釣魚攻擊、社交工程攻擊
- ...

電腦網路通訊之威脅

- 如果有個敵人「想知道」或者「想竄改」你的筆電與 www.google.com 之間的通訊內容, 他可以怎麼辦？

電腦網路通訊之威脅

- 如果有個敵人「想知道」或者「想竄改」你的筆電與 `www.google.com` 之間的通訊內容, 他可以怎麼辦？

- 試圖攻進你的筆電
- 試圖攻進 Google 的主機
- 試圖執行「中間人」(man-in-the-middle) 攻擊

中間人攻擊

- 網路節點可以竊聽、竄改經過它的封包
- 敵人可能已經掌控部分的網路節點了:
  - 假的、遭入侵的 Wi-Fi AP / Router
  - 假的手機訊號基地台

對抗中間人最有效的手段:加密

- 然而正確的實作並不容易
  “Don't Roll Your Own Cryptography/Security.”

- TLS 是個很通用的選擇
- TLS 通常操作於 TCP 之上 (HTTP to HTTPS)
- TLS 最常見的用途是單向的身份認證 (server)

TLS 的功能

- 通訊內容的秘密性 (confidentiality)
- 通訊內容的完整性 (integrity)

TLS 的功能

- 通訊內容的秘密性 (confidentiality)
- 通訊內容的完整性 (integrity)

- TLS 並不提供匿名性
- TLS 在任何一方遭入侵時就沒有功效

TLS 的各個版本

SSL 2.0	IETF RFC 6176	1995; deprecated in 2011
SSL 3.0	IETF RFC 7568	1996; deprecated in 2015
TLS 1.0	IETF RFC 2246	1999
TLS 1.1	IETF RFC 4346	2006
TLS 1.2	IETF RFC 5246	2008
TLS 1.3	working draft	to be determined

到底是 SSL 還是 TLS？

- SSL 2.0 與 3.0 版並不安全, 已經被廢棄了
- 現在大家都使用 TLS 1.0 或 1.1 或 1.2 版

- SSL 這個名詞用久了, 習慣改不掉
- 如今 SSL 和 TLS 往往指的是同一回事

TLS 加密提供通訊內容的秘密性、完整性

- 靠的是 Authenticated Encryption
- E.g., AES-128 + CBC mode + SHA-256

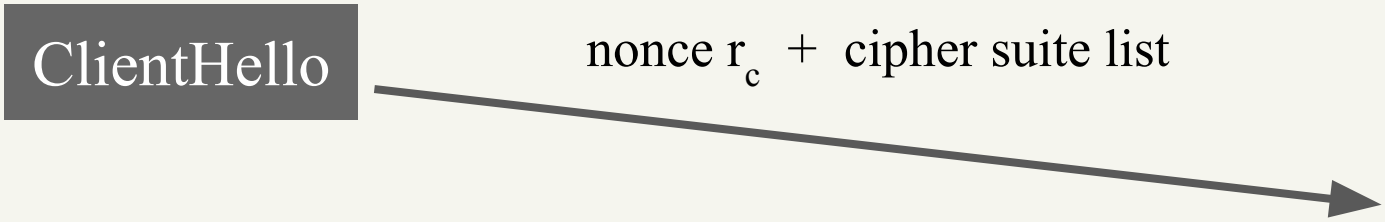
- 問題一:要用什麼演算法?
- 問題二:秘密金鑰哪裡來?

兩個問題都在「 TLS Handshake 」解決

兩個問題都在「 TLS Handshake 」解決

1. Cipher suite negotiation
2. Authenticated key exchange

TLS Handshake Protocol



Value	Cipher Suite Description
0000	TLS_NULL_WITH_NULL_NULL
0001	TLS_RSA_WITH_NULL_MD5
0002	TLS_RSA_WITH_NULL_SHA
0003	TLS_RSA_EXPORT_WITH_RC4_40_MD5
0004	TLS_RSA_WITH_RC4_128_MD5
0005	TLS_RSA_WITH_RC4_128_SHA
0006	TLS_RSA_EXPORT_WITH_RC2_CBC_40_MD5
0007	TLS_RSA_WITH_IDEA_CBC_SHA
0008	TLS_RSA_EXPORT_WITH_DES40_CBC_SHA
0009	TLS_RSA_WITH_DES_CBC_SHA
000A	TLS_RSA_WITH_3DES_EDE_CBC_SHA
000B	TLS_DH_DSS_EXPORT_WITH_DES40_CBC_SHA
000C	TLS_DH_DSS_WITH_DES_CBC_SHA
000D	TLS_DH_DSS_WITH_3DES_EDE_CBC_SHA
000E	TLS_DH_RSA_EXPORT_WITH_DES40_CBC_SHA
000F	TLS_DH_RSA_WITH_DES_CBC_SHA

TLS Cipher Suite

- 每個 cipher suite 都是一個長 2 bytes 的值
- 取名格式「 TLS_PartA_WITH_PartB_PartC 」

TLS Cipher Suite

- 每個 cipher suite 都是一個長 2 bytes 的值
- 取名格式「 TLS_PartA_WITH_PartB_PartC 」

- PartA 決定如何交換密鑰、驗證伺服器身份
- PartB 決定如何對資料加密、認證(對稱式)
- PartC 決定這次的 PRF 要基於哪一個雜湊函數

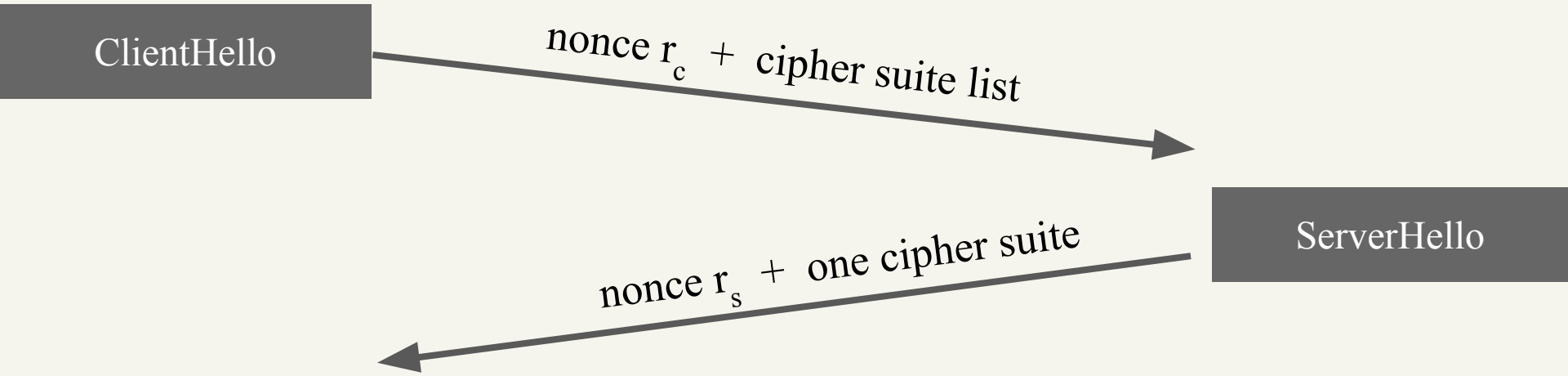
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256

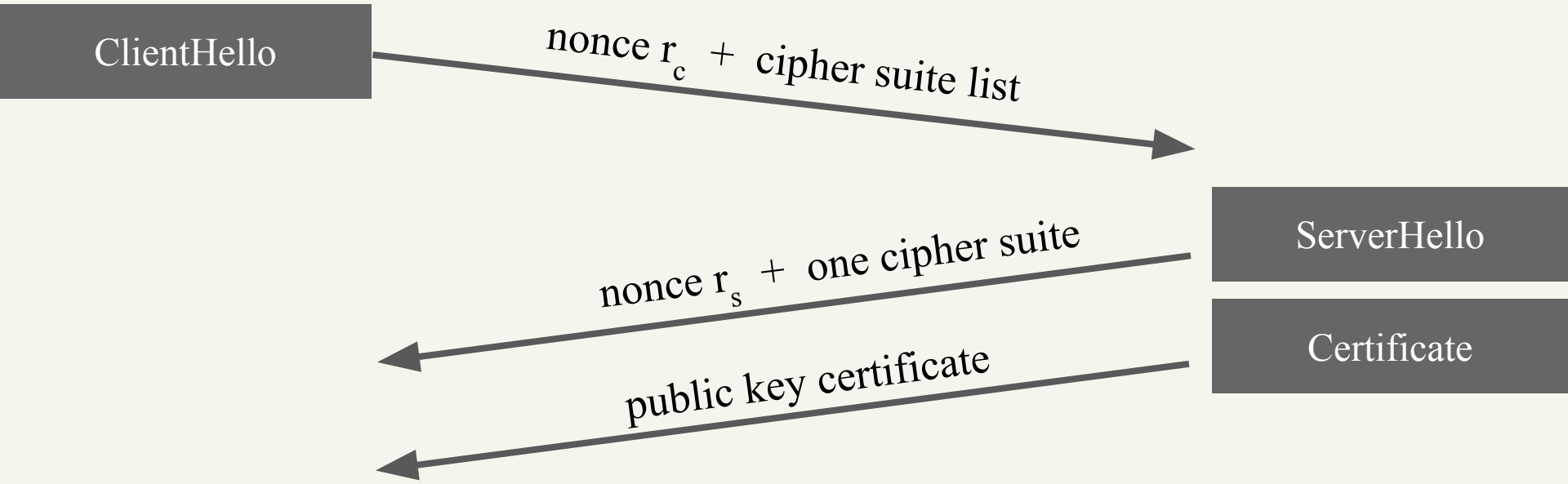
- DHE_RSA：使用動態 Diffie-Hellman 做密鑰交換，並且搭配 RSA 數位簽章做身份認證(公鑰從伺服器的憑證取得)
- AES_128_GCM：雙方協議出這個 session 專用的 master secret 金鑰以後，使用 AES-128 操作於 GCM 模式來對每一個 TLS record 加密
- SHA256：使用 SHA-256 函數構造這個 session 使用的 PRF

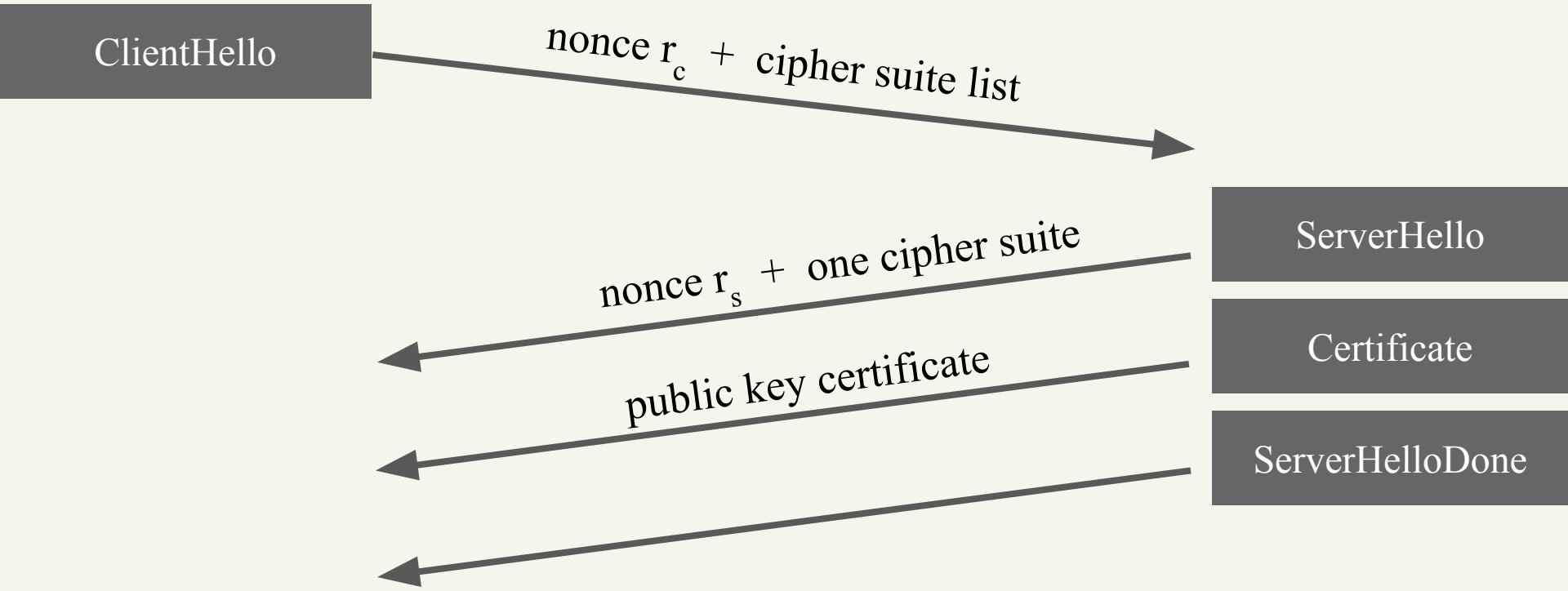
TLS_RSA_WITH_AES_128_CBC_SHA256

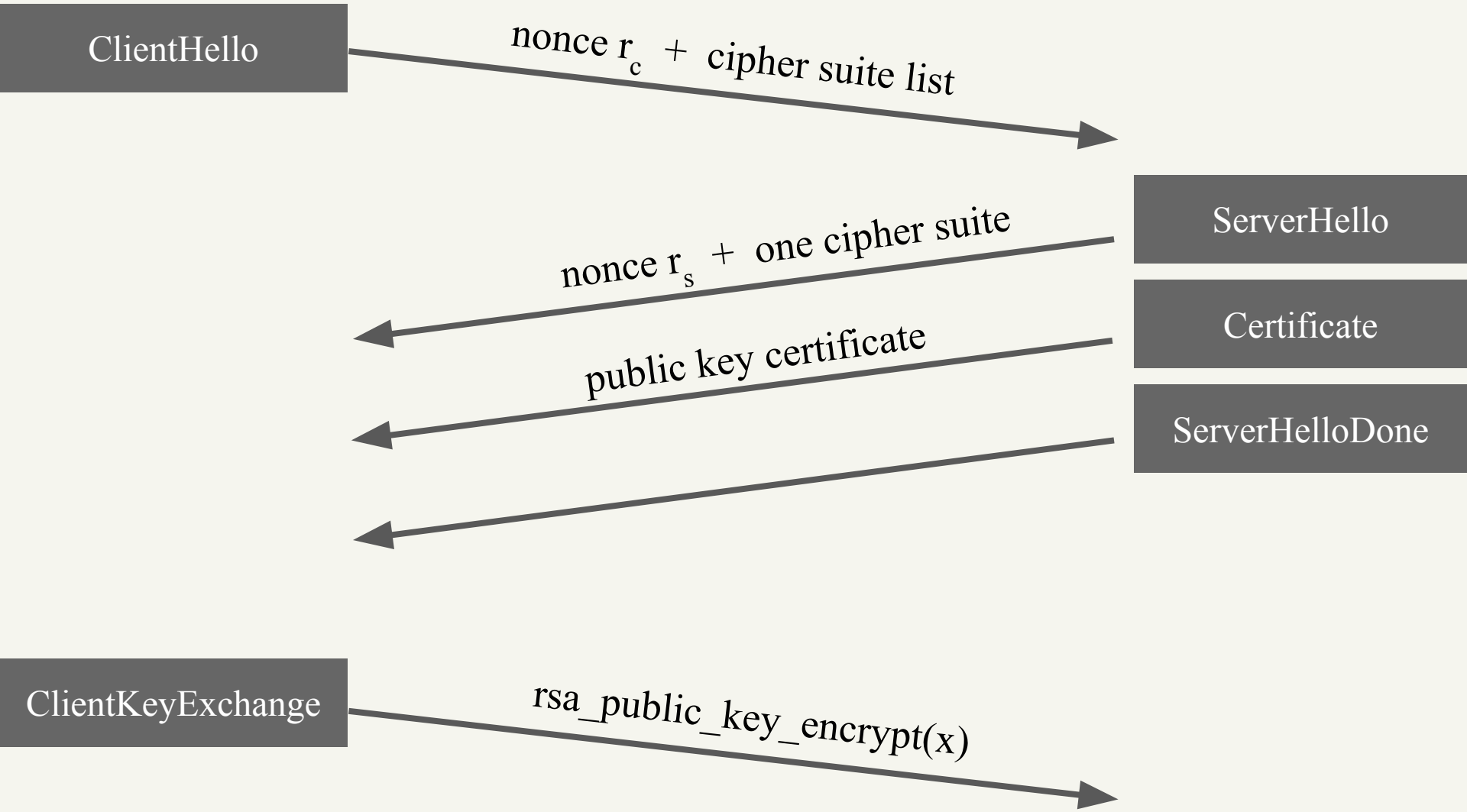
- RSA：使用 RSA 加密做金鑰交換（公鑰從伺服器的憑證取得）
- AES_128_CBC：雙方協議出這個 session 專用的 master secret 金鑰以後，使用 AES-128 操作於 CBC 模式搭配 MAC 來對每一個 TLS record 加密
- SHA256：使用 SHA-256 函數構造這個 session 使用的 PRF

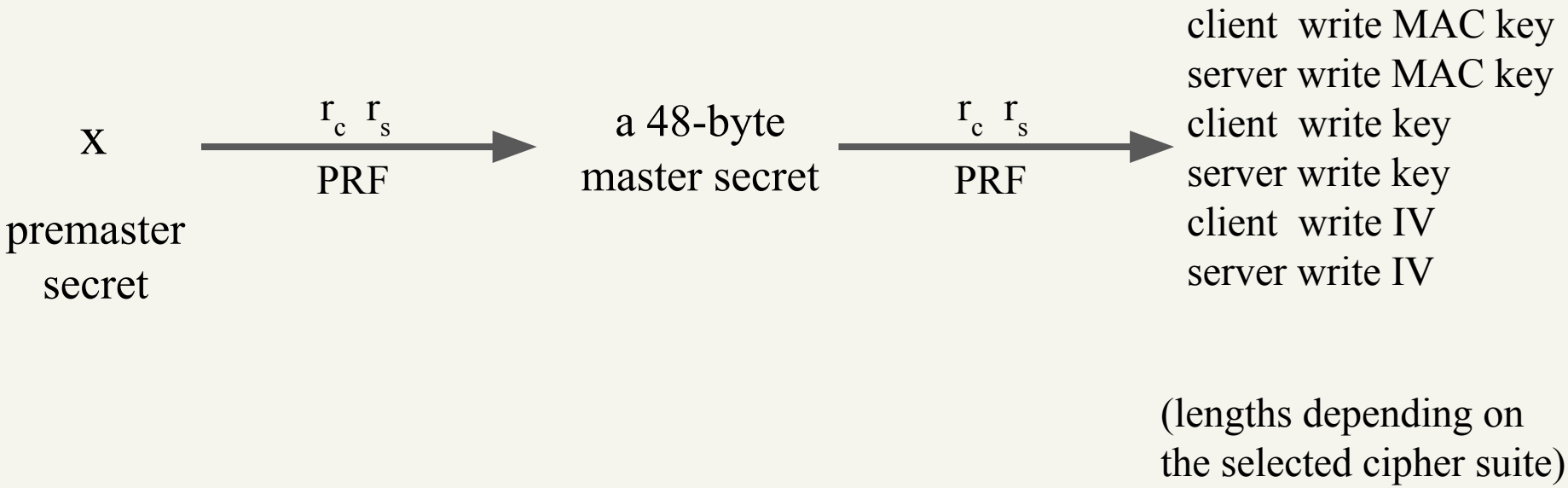

ClientHello
nonce r_c + cipher suite list

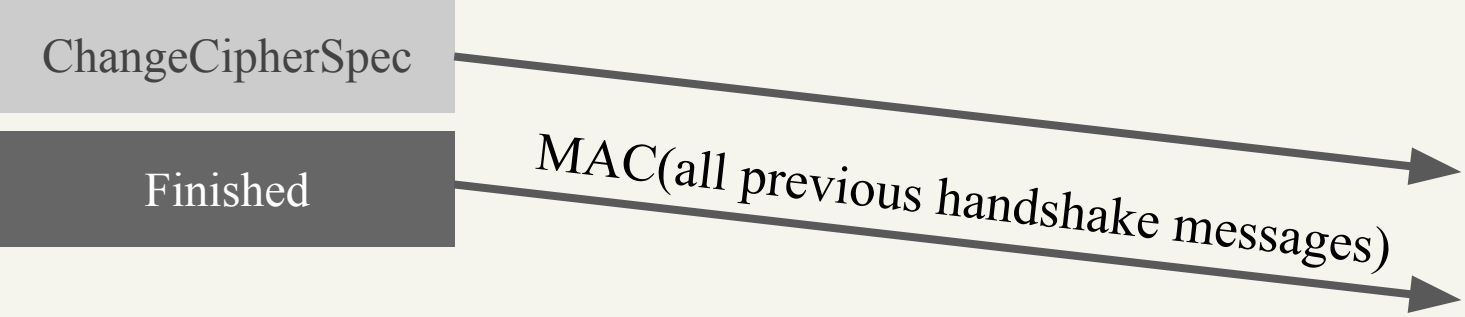

ClientHello
nonce r_c + cipher suite list
ServerHello
nonce r_s + one cipher suite

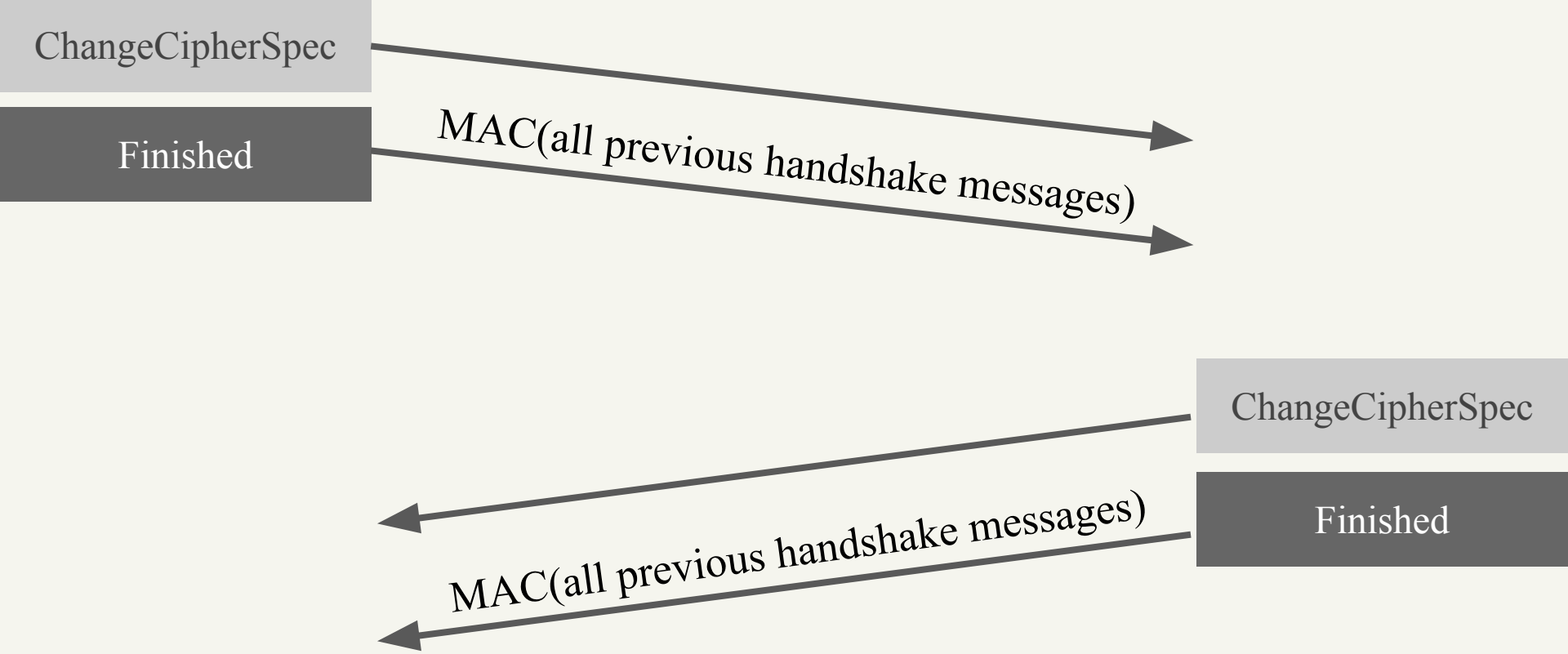

ClientHello
nonce r_c + cipher suite list
ServerHello
nonce r_s + one cipher suite
Certificate
public key certificate

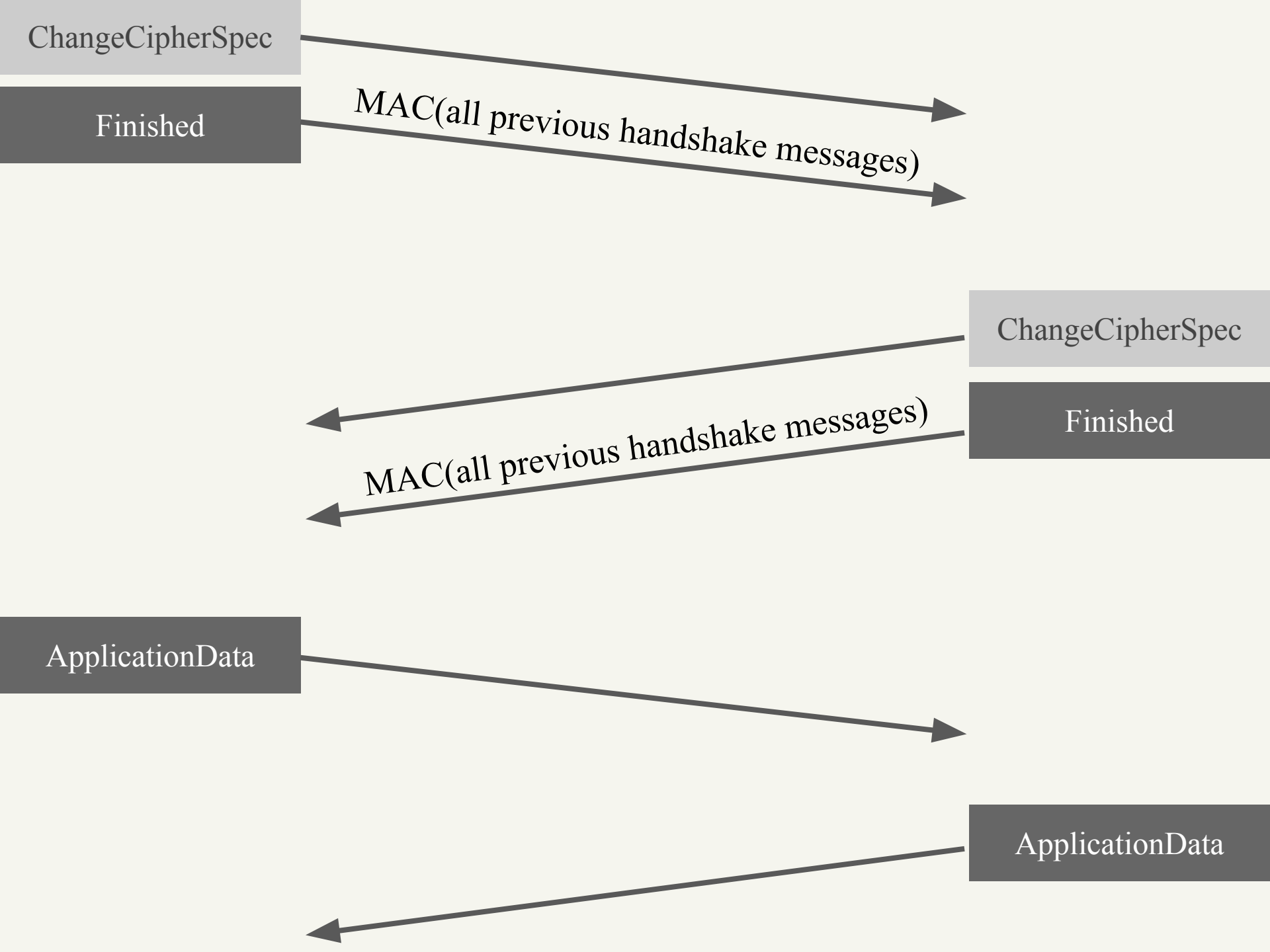

ClientHello
nonce r_c + cipher suite list
ServerHello
nonce r_s + one cipher suite
Certificate
public key certificate
ServerHelloDone


ClientHello
nonce r_c + cipher suite list
ServerHello
nonce r_s + one cipher suite
Certificate
public key certificate
ServerHelloDone
ClientKeyExchange
rsa_public_key_encrypt(x)


x
premaster secret
r_c r_s
PRF
a 48-byte master secret
r_c r_s
PRF
client write MAC key
server write MAC key
client write key
server write key
client write IV
server write IV
(lengths depending on the selected cipher suite)


ChangeCipherSpec
Finished
MAC(all previous handshake messages)


ChangeCipherSpec
Finished
MAC(all previous handshake messages)
ChangeCipherSpec
Finished
MAC(all previous handshake messages)


ChangeCipherSpec
Finished
MAC(all previous handshake messages)
ChangeCipherSpec
Finished
MAC(all previous handshake messages)
ApplicationData
ApplicationData

如果伺服器的 RSA 密鑰被偷了會怎樣？

Forward Secrecy

- https://en.wikipedia.org/wiki/Forward_secrecy

Homework

- 破解沒有 Forward Secrecy 的 TLS 連線
- 可以用 Python 3 (latest release: 3.5.1)
- 也可以用任何其他你喜歡的程式語言
- `jong.sh/tlsfcu`