CSCI 1515 Applied Cryptography

This Lecture:

- Block Cipher Modes of Operation
- Public Key Infrastructure (PKI)

Block Cipher Modes of Operation

Given: $F: \{0,1\}^\lambda \times \{0,1\}^\lambda \to \{0,1\}^\lambda$ ← 128

Assumed to be a pseudorandom permutation (PRP).

Goal: Construct a symmetric-key encryption (SKE) scheme for arbitrary-length messages with Chosen-Plaintext Attack (CPA) Security.

- $k \xleftarrow{\$} \{0,1\}^\lambda$
- $Enc_k(m)$ $\quad |m| = \alpha \cdot \lambda$ (If not, pad it to a multiple of λ)
- $Dec_k(c)$

Pseudorandom Permutation (PRP)

$k \xleftarrow{\$} \{0,1\}^\lambda$

$F_k : \{0,1\}^n \underset{F_k^{-1}}{\overset{F_k}{\rightleftarrows}} \{0,1\}^n$ bijective

How many possible F_k's ?

2^λ

$\approx_c$ (not knowing k)

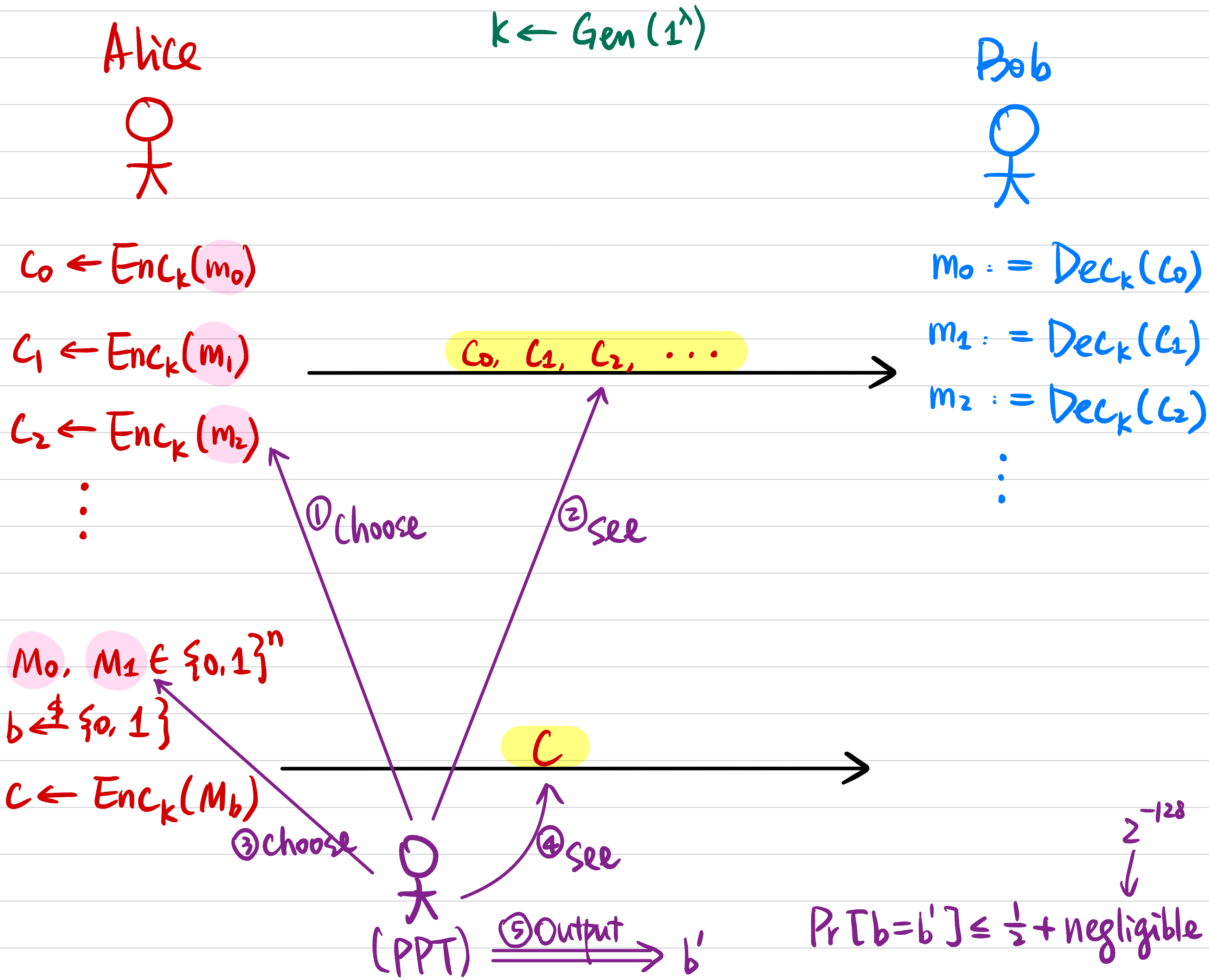
$f \xleftarrow{\$} \{ F \mid F : \{0,1\}^n \to \{0,1\}^n, \text{ F is bijective} \}$

$(2^n)!$

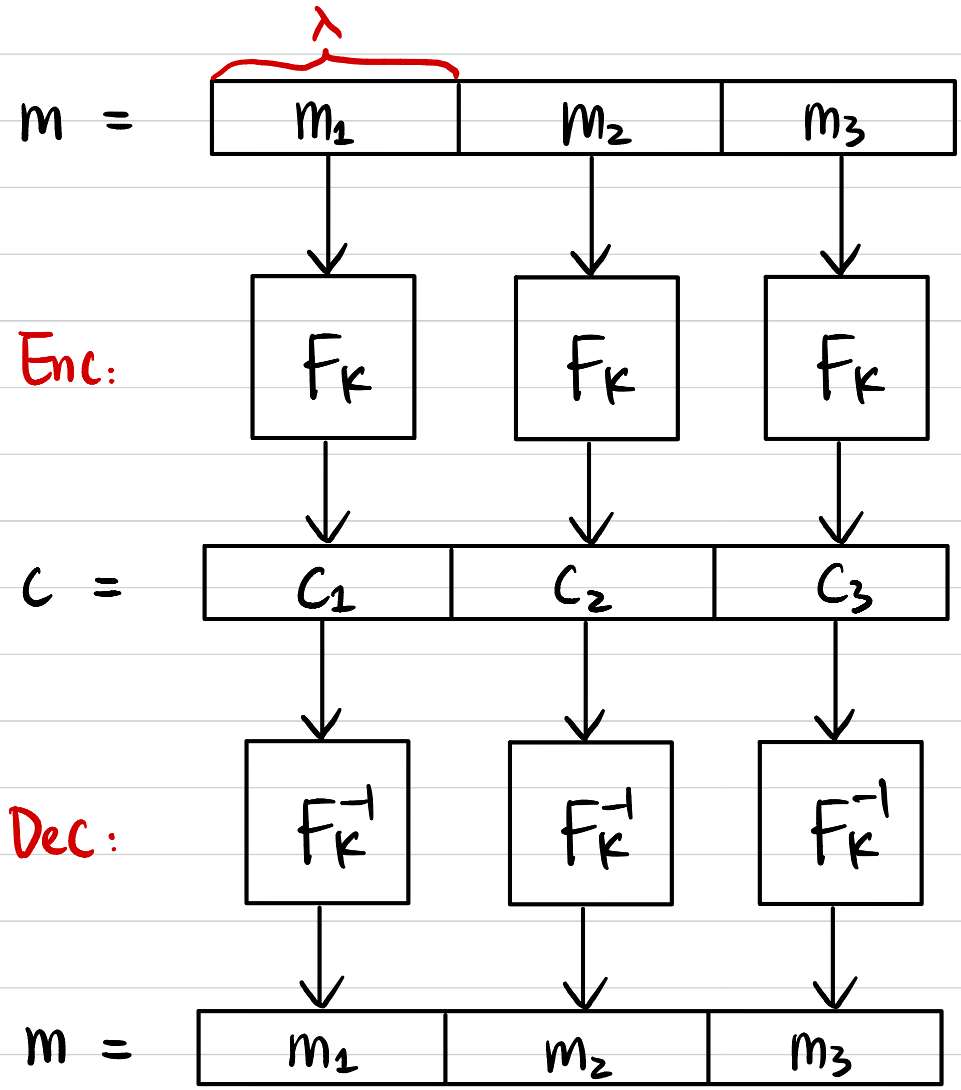
$f : \{0,1\}^n \underset{f^{-1}}{\overset{f}{\rightleftarrows}} \{0,1\}^n$ bijective

How many possible f's ?

Chosen-Plaintext Attack (CPA) Security

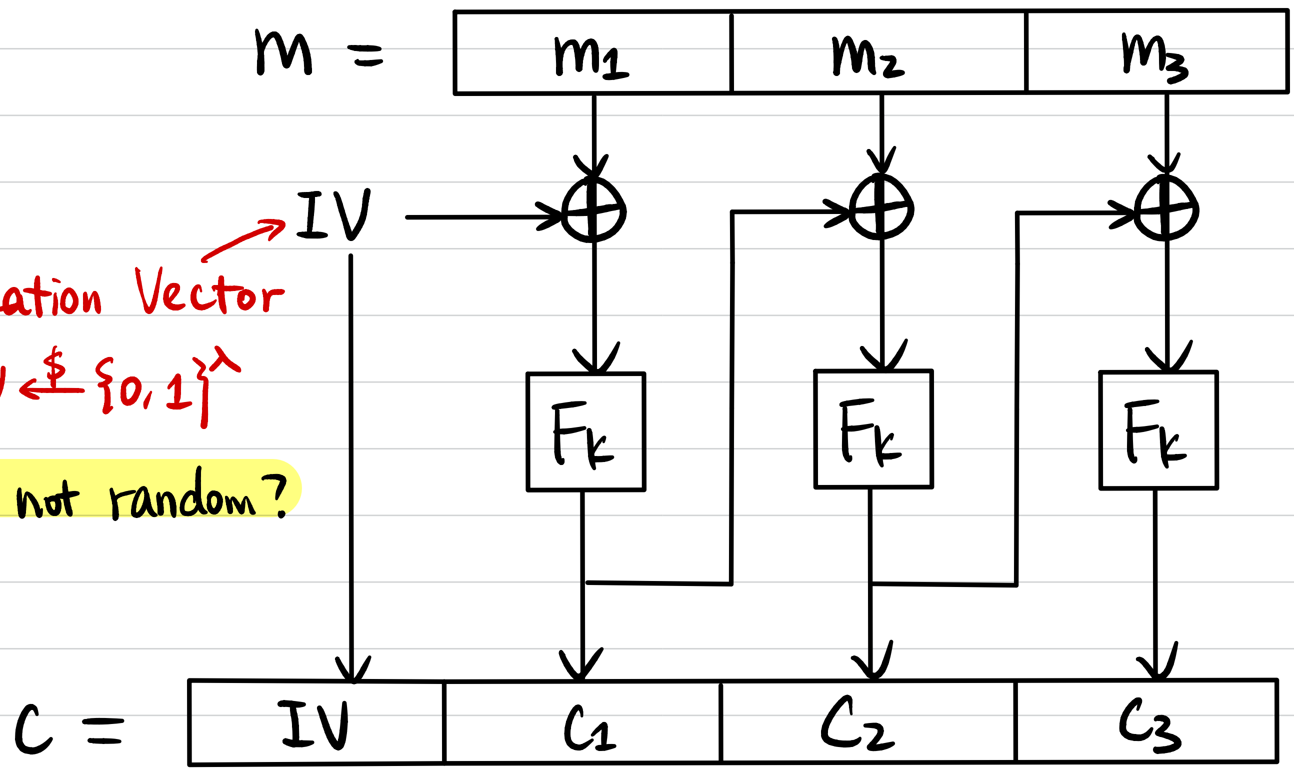


Electronic Code Book (ECB) Mode



CPA Secure?

Cipher Block Chaining (CBC) Mode



Initialization Vector

$IV \xleftarrow{\$} \{0,1\}^\lambda$

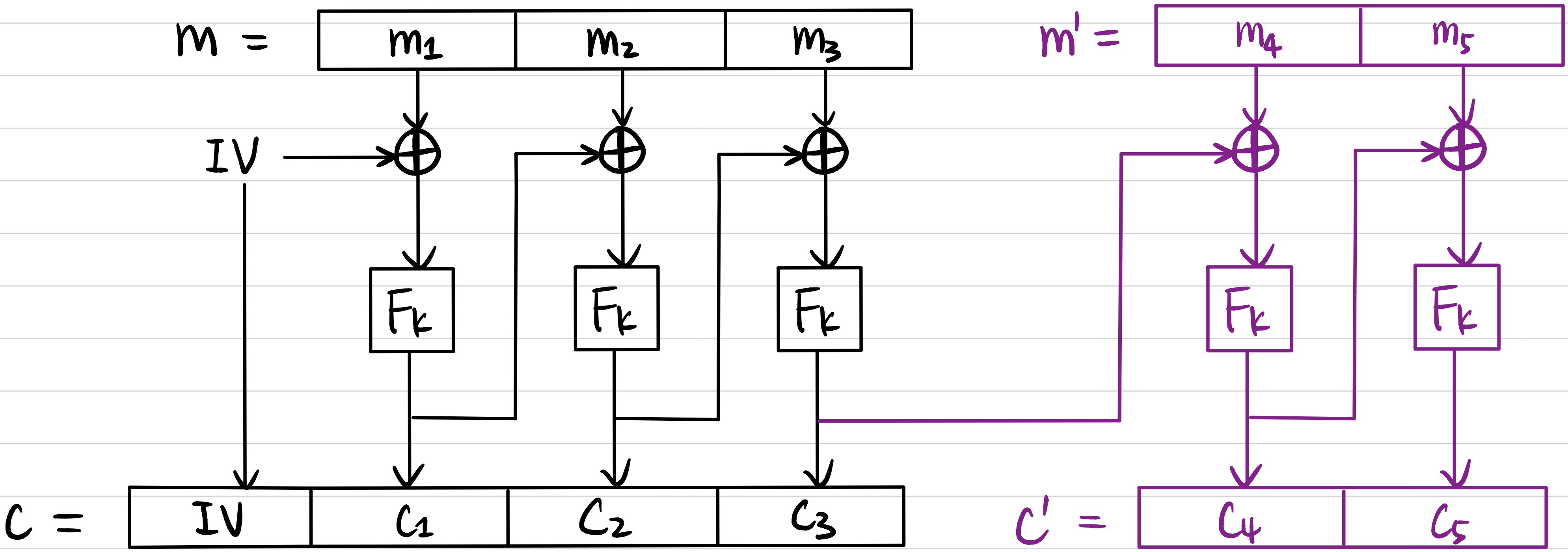
What if not random?

How to decrypt?

CPA secure?

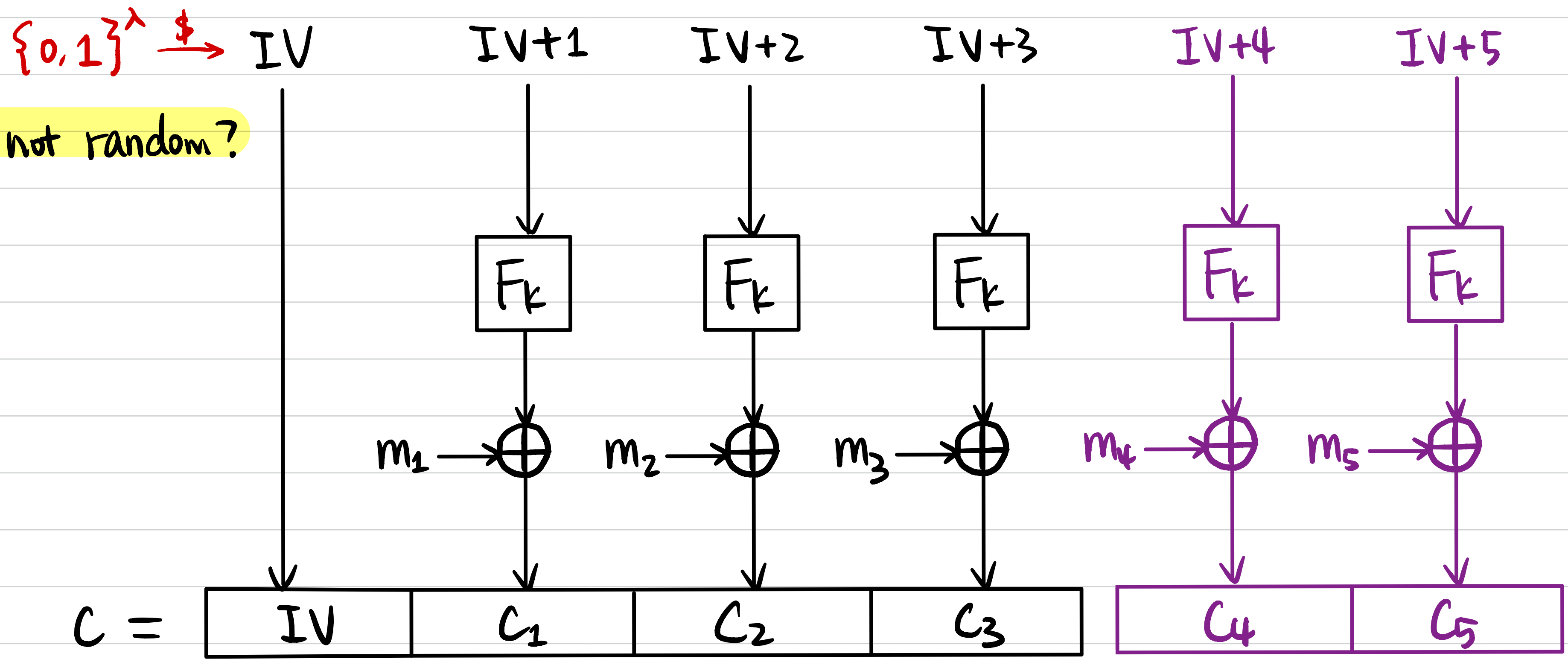
Can we parallelize the computation?

Chained Cipher Block Chaining (CBC) Mode



CPA Secure ?

Counter (CTR) Mode



What if not random?

How to decrypt?

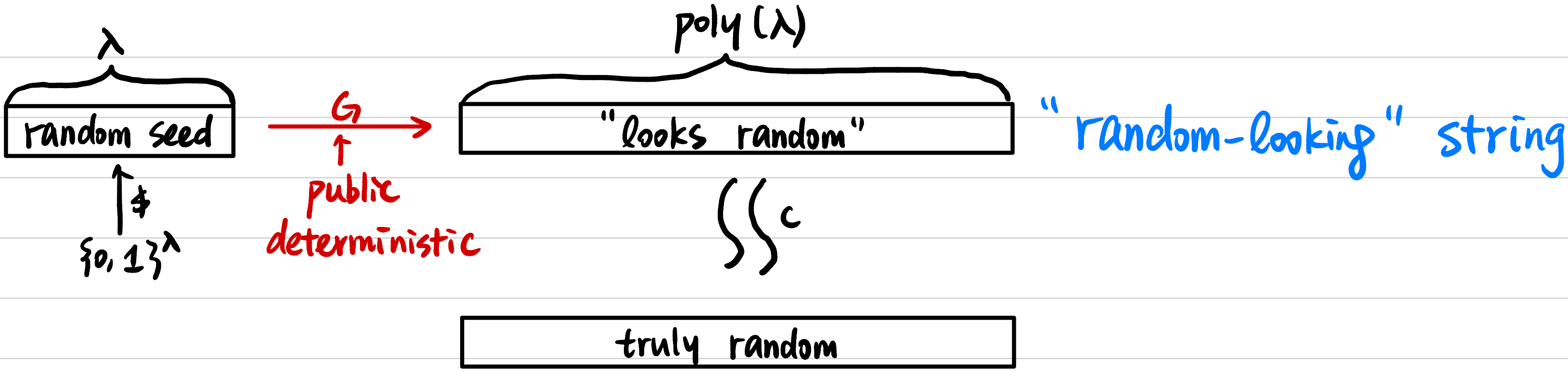
CPA Secure?

"Stateful" CTR Mode?

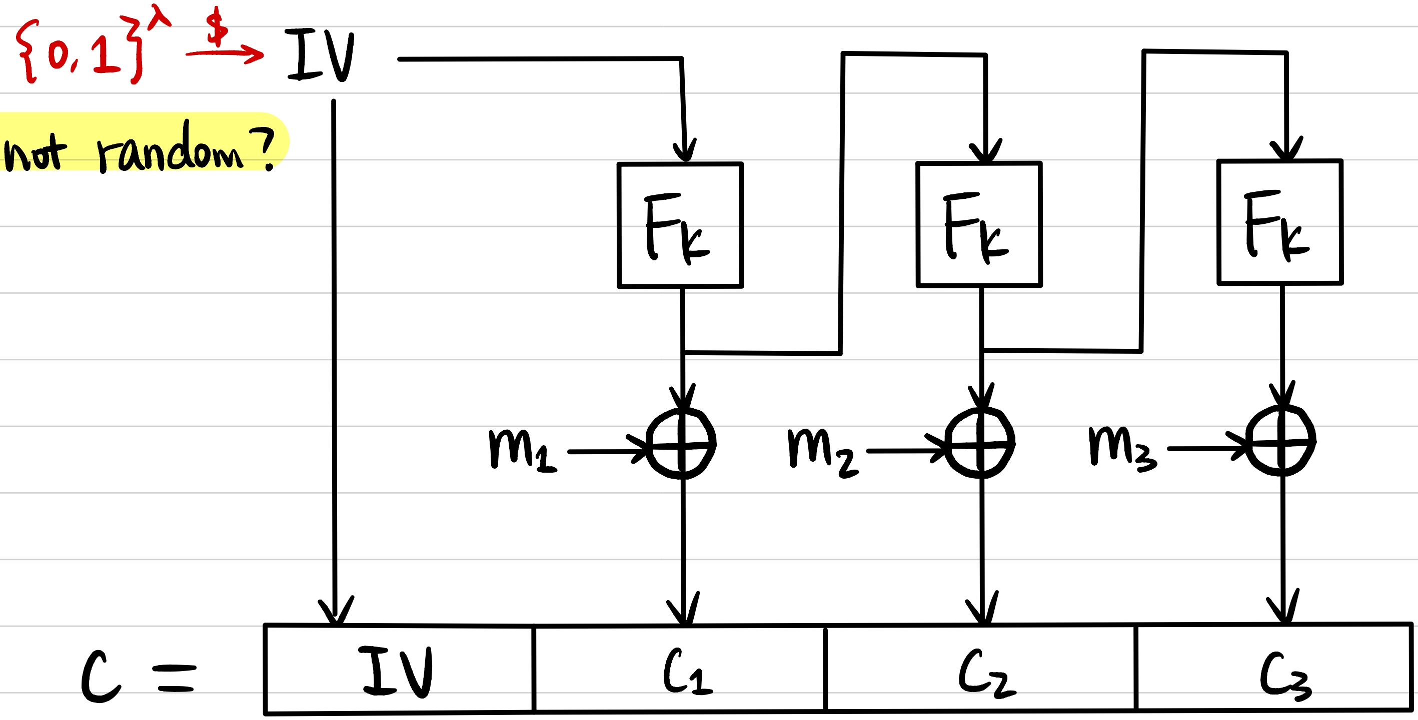
Can we parallelize the computation?

PRG from PRF

Pseudorandom Generator (PRG)



Output Feedback (OFB) Mode



What if not random?

How to decrypt?

CPA secure?

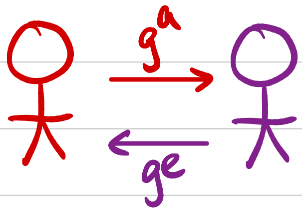
"Stateful" OFB Mode?

Can we parallelize the computation?

PRG from PRF

Putting it All Together: Secure Communication

Alice Eve

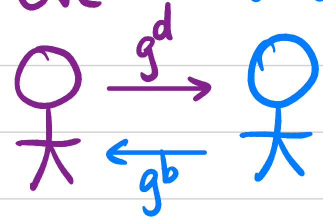


$\xrightarrow{g^a}$

Diffie–Hellman Key Exchange

$\xleftarrow{g^b}$

Eve Bob



Alice: $\Downarrow$ g^{ab} $\Downarrow$ Hash K (k_1, k_2)

Eve: $\Downarrow$ g^{ae}

"Man-in-the-Middle Attack"

Eve: $\Downarrow$ g^{bd}

Bob: $\Downarrow$ g^{ab} $\Downarrow$ Hash K (k_1, k_2)

$\longrightarrow$

Authenticated Encryption

(Encrypt-then-MAC)

$\longleftarrow$

AES HMAC

Authenticated Key Exchange

Signature Scheme

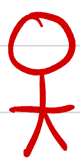
public

$(VK_A, SK_A) \leftarrow Gen(1^\lambda)$

public

$(VK_B, SK_B) \leftarrow Gen(1^\lambda)$

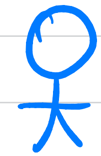
Alice



$\Downarrow$

g^{ab}

Bob



$\Downarrow$

g^{ab}

$\xrightarrow{g^a}$

Diffie-Hellman Key Exchange

$\xleftarrow[g^b]{}$

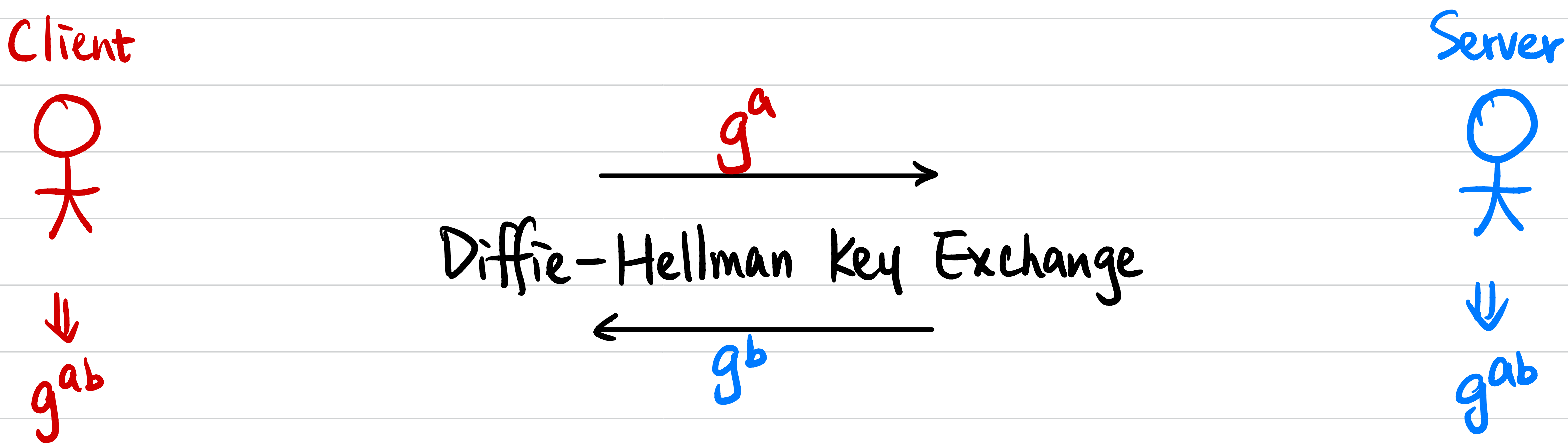
Can we prevent Man-in-the-Middle Attack?

One-Sided Secure Authentication

public

Where does it come from?

$(VK_S, SK_S) \leftarrow Gen(1^\lambda)$



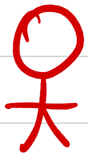
Can we prevent Man-in-the-Middle Attack?

Public Key Infrastructure (PKI)

public

$(VK, SK) \leftarrow Gen(1^\lambda)$

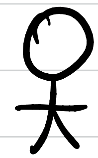
Alice



Cerificate signing request (CSR)

"alice.com"; VK_A $\longrightarrow$

Certificate Authority
(CA)



"alice.com"

$(VK_A, SK_A) \leftarrow Gen(1^\lambda)$

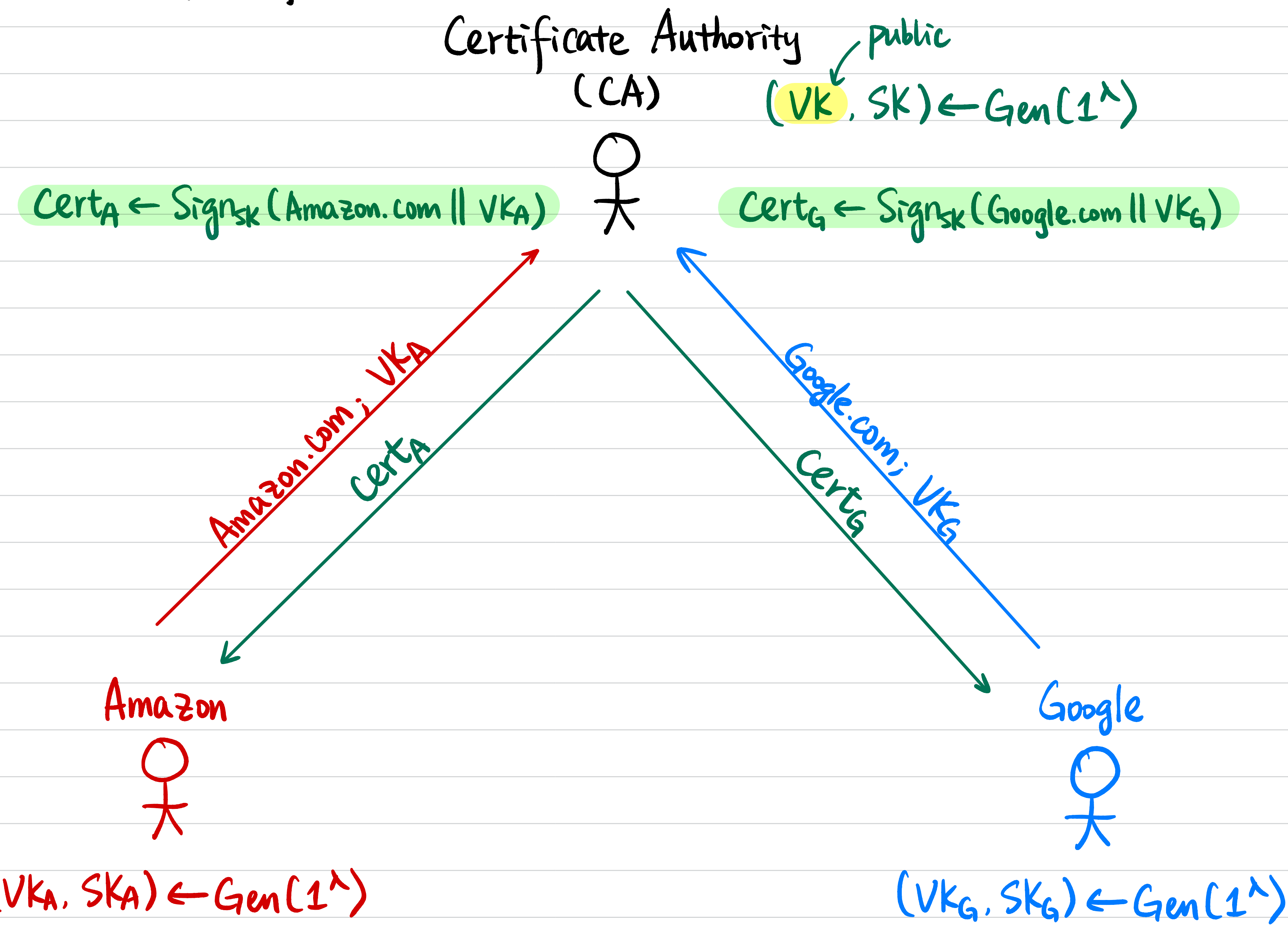
"Certificate"

$cert_A$ $\longleftarrow$

$cert_A \leftarrow Sign_{sk}(alice.com \,||\, VK_A)$

Standard: X.509 certificate

Public Key Infrastructure (PKI)



One-Sided Secure Authentication

public

$(VK, SK) \leftarrow Gen(1^\lambda)$ CA

$Cert_G \leftarrow Sign_{SK}(Google.com \,\|\, VK_G)$

VK_G $Cert_G$

$(VK_G, SK_G) \leftarrow Gen(1^\lambda)$ Google

Client

$\xrightarrow{g^a}$

Diffie-Hellman Key Exchange

$\xleftarrow{}$

$(VK_G, Cert_G)$

(g^b, σ)

$\sigma \leftarrow Sign_{SK_G}(g^b)$

Client: $\Downarrow$ g^{ab}

Google: $\Downarrow$ g^{ab}

$Vrfy_{VK}(Google.com \,\|\, VK_G,\; Cert_G) \stackrel{?}{=} 1$

$Vrfy_{VK_G}(g^b, \sigma) \stackrel{?}{=} 1$

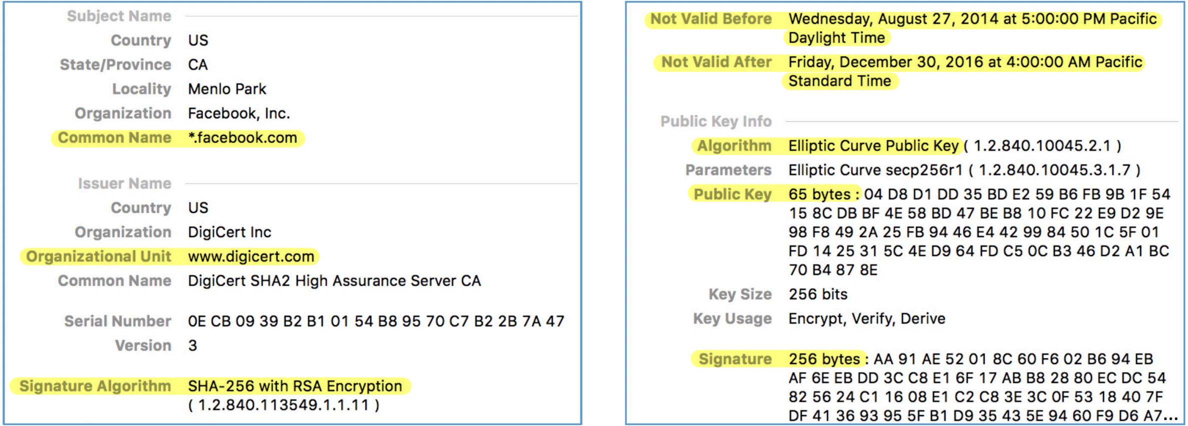


Figure 13.4: An example X.509 certificate

Certificate Chain

Root CA $\qquad$ $(\underbrace{VK}_{\text{public}}, SK) \leftarrow Gen(1^\lambda)$

$\updownarrow$

Intermediate CA1 $\qquad$ $(VK_1, SK_1) \leftarrow Gen(1^\lambda)$

$\updownarrow$

Intermediate CA2 $\qquad$ $(VK_2, SK_2) \leftarrow Gen(1^\lambda)$

$\updownarrow$

Alice $\qquad$ $(VK_A, SK_A) \leftarrow Gen(1^\lambda)$

Certificate Revocation

- Short-lived certificates
- Certificate revocation lists (CRLs)