

CSCI 1515 Applied Cryptography

This Lecture:

- Applications of Hash Functions
- Putting it Together: Secure Communication
- Pseudorandom Generator (PRG)
Pseudorandom Function (PRF)
Pseudorandom Permutation (PRP)
- Block Cipher and Modes of Operation

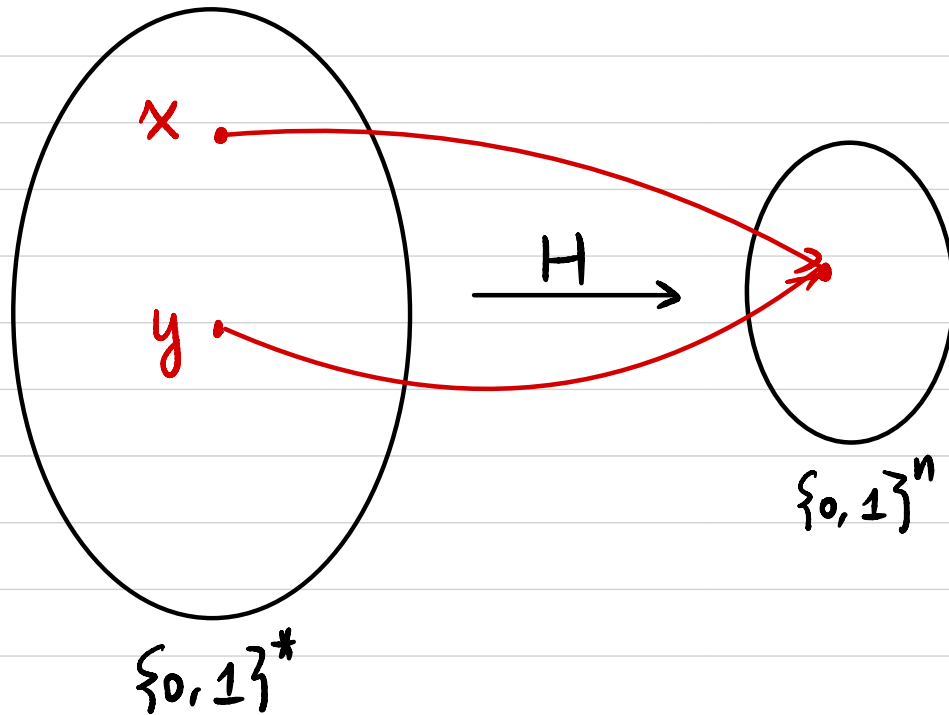
ANNOUNCEMENT: Cipher due extended to 2/9.

Please contact us ASAP if you're not on Gradescope/EdStem.

Hash Function

$$\lambda = 128$$
$$n = 256$$

$$H: \{0, 1\}^* \rightarrow \{0, 1\}^n$$



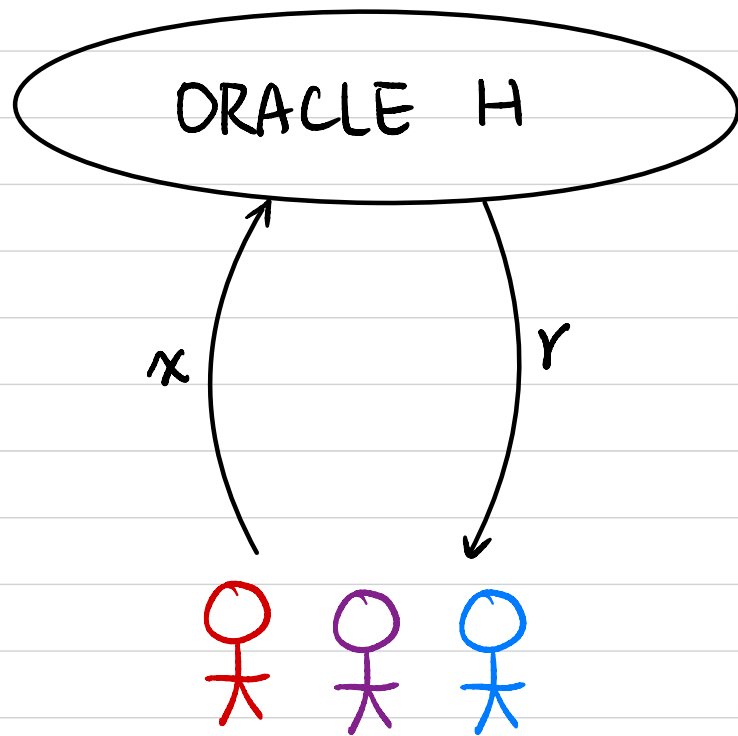
Collision-Resistant Hash Function (CRHF):

It's computationally hard to find $x, y \in \{0, 1\}^*$ s.t.

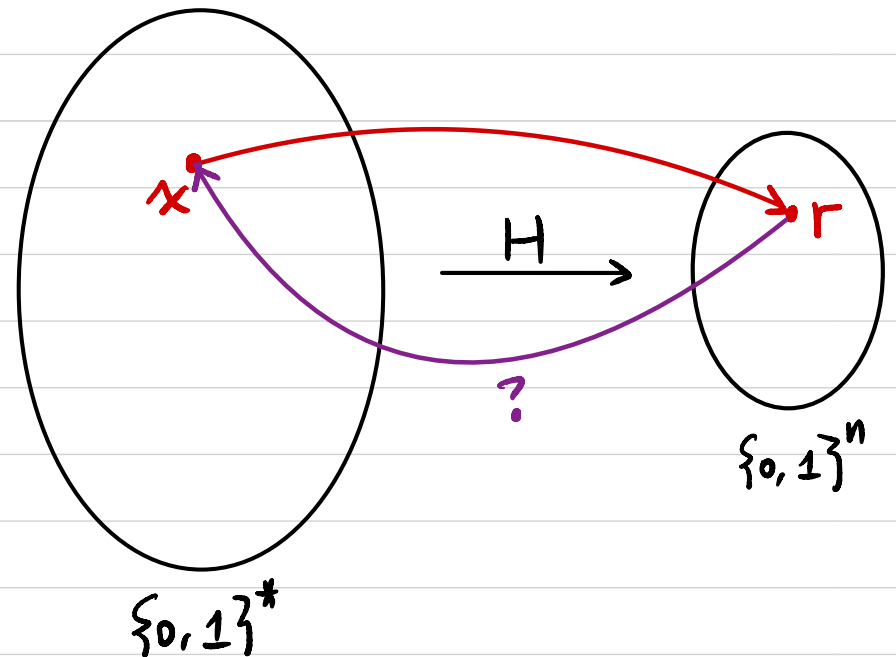
$$x \neq y, \quad H(x) = H(y) \quad (\text{collision})$$

Random Oracle Model

$$H: \{0,1\}^* \rightarrow \{0,1\}^n$$



$$r \leftarrow \{0,1\}^n$$
$$H(x) := r$$



Constructions for Hash Function

Secure Hash Functions (SHA): Standardized by NIST.

- SHA-2: Standardized in 2001
output length 224, 256, 384, 512-bit
- SHA-3: Competition 2007-2012
released in 2015
output length 224, 256, 384, 512-bit

Applications of Hash Functions

- HMAC**

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

SHA-2

$$\text{Mac}_k(m) = H(k \parallel m)$$

$$\text{Mac}_k(m) = H(k \parallel H(k \parallel m))$$

$$\text{HMAC}_k(m) = H(\underbrace{k_2}_{k \oplus \text{opad}} \parallel \underbrace{H(k_1 \parallel m)}_{k \oplus \text{ipad}})$$

public

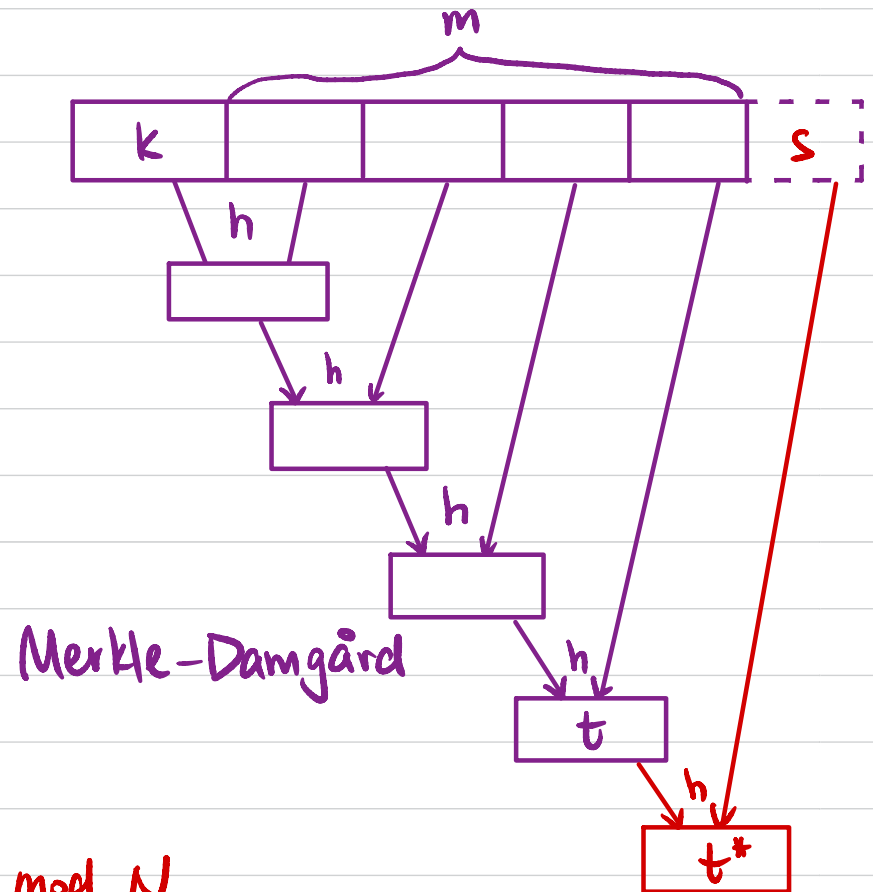
- Hash-and-Sign**: $\text{Sign}_{sk}(H(m))$

RSA Signature: $\text{Sign}_{sk}(m) = H(m)^d \bmod N$

- Password-based Authentication**

username, $H(\text{password})$

Given m , $t = H(k \parallel m)$
Forge $m^* = m \parallel s$, $t^* = H(t \parallel s)$



Applications of Hash Functions

- Deduplication

$$\begin{aligned} H(D_1) &\rightarrow h_1 \\ H(D_2) &\rightarrow h_2 \end{aligned}$$

← unique identifier

$$\text{If } h_1 \neq h_2 \Rightarrow D_1 \neq D_2$$

$$\text{If } h_1 = h_2 \Rightarrow D_1 = D_2 \text{ (why?)}$$

Virus Scan

$$H(F) \stackrel{?}{=} H(F^*)$$

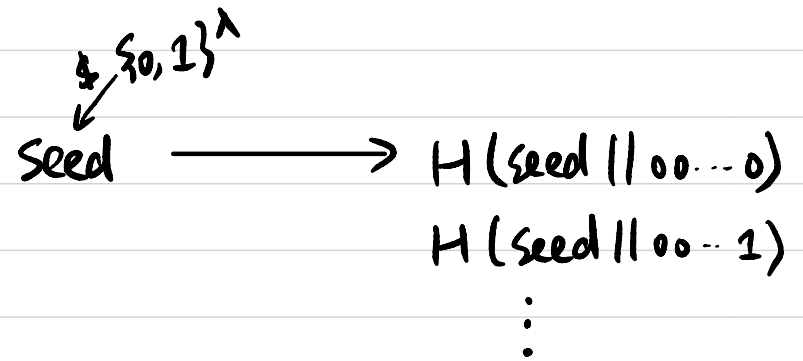
Video Deduplication

$$H(V_1) \stackrel{?}{=} H(V_2)$$

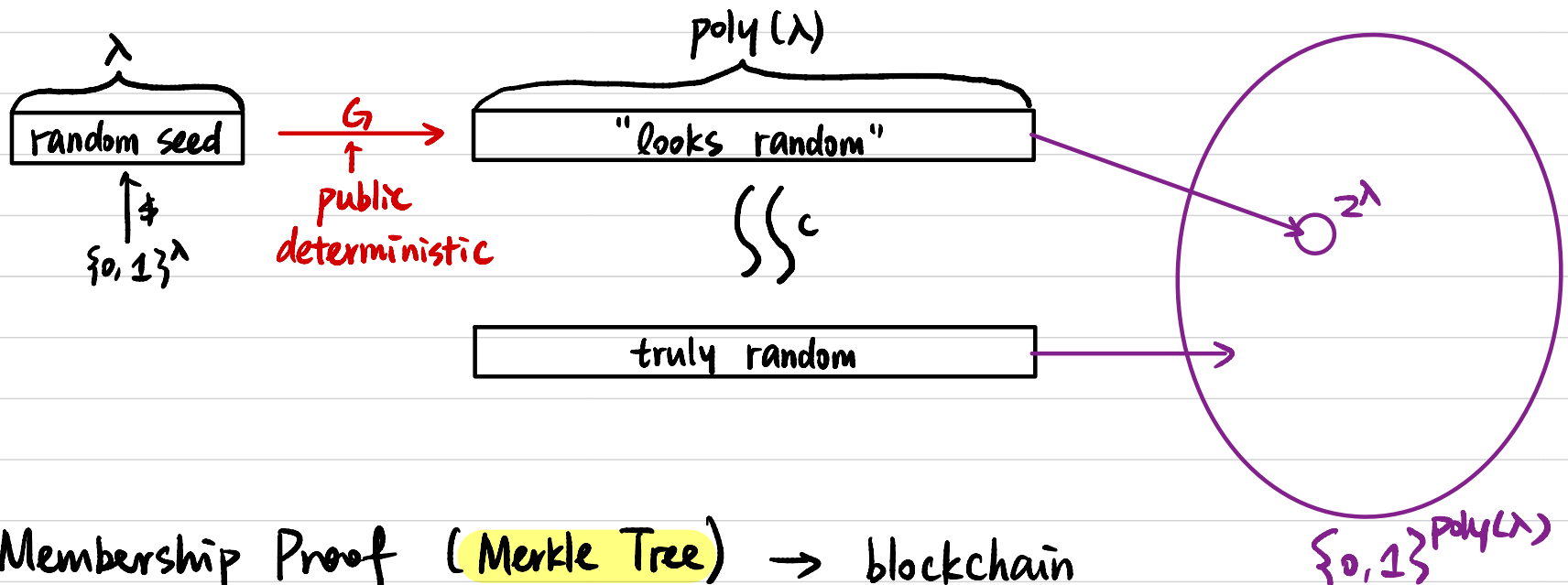
Applications of Hash Functions

- HKDF (Key Derivation Function):

HMAC(g^a , salt)



Pseudorandom Generator (PRG)



- Fast Membership Proof (Merkle Tree) $\rightarrow$ blockchain

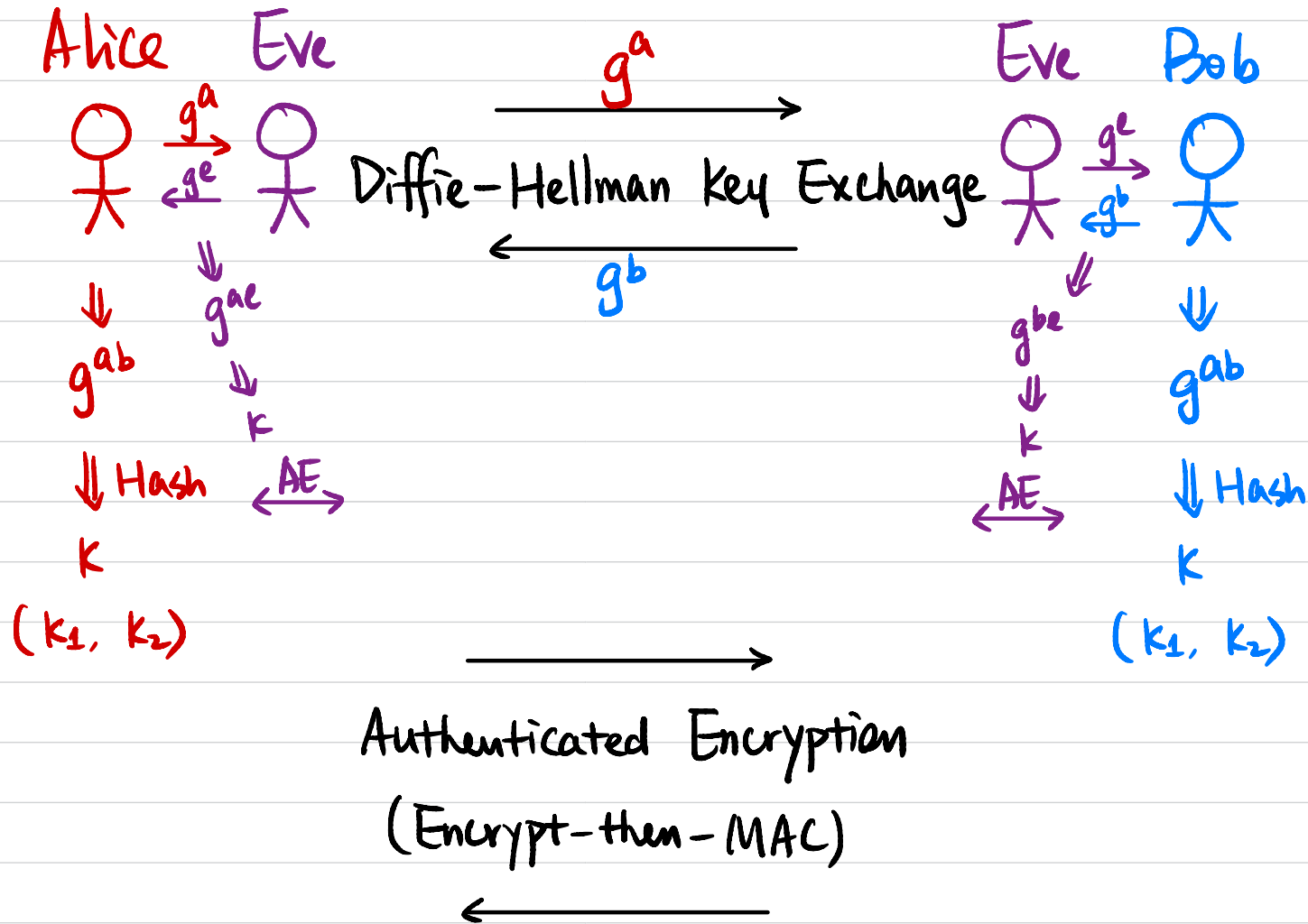
- SKE Scheme?

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

$$\text{Enc}_k(m) = H(k || m) ?$$

Deterministic schemes can't be CPA-secure!

Putting it Together: Secure Communication



Any security issue?

"Man-in-the-Middle" Attack

Signal: Diffie-Hellman Ratchet

Alice



$\Downarrow a_1$
 $g^{a_1 b_1}$
 $\Downarrow$
 k_1

$\Downarrow a_2$
 $g^{a_2 b_1}$
 $\Downarrow$
 k_2

$\Downarrow a_3$
 $g^{a_3 b_2}$
 $\Downarrow$
 k_3

Diffie-Hellman Key Exchange

$\xrightarrow{g^{a_1}}$

$\xleftarrow{g^{b_1}}$
 $Enc_{k_1}(m_B)$

$\xrightarrow{g^{a_2}}$
 $Enc_{k_2}(m_A)$

$\xleftarrow{g^{b_2}}$
 $Enc_{k_3}(m_B)$

Bob



$\Downarrow b_1$
 $g^{a_1 b_1}$
 $\Downarrow$
 k_1

$\Downarrow b_1$
 $g^{a_2 b_1}$
 $\Downarrow$
 k_2

$\Downarrow b_2$
 $g^{a_2 b_2}$
 $\Downarrow$
 k_3

If k_1 is leaked?

If b_1 is leaked?

Summary

Symmetric-Key

Public-Key

Message
Secrecy

Primitive: SKE

Construction: block cipher

Primitive: PKE

Constructions: RSA / ElGamal

Message
Integrity

Primitive: MAC

Constructions: CBC-MAC / HMAC

Primitive: Signature

Constructions: RSA / DSA

Secrecy
& Integrity

Primitive: AE

Construction: Encrypt-then-MAC

Key Exchange

Construction: Diffie-Hellman

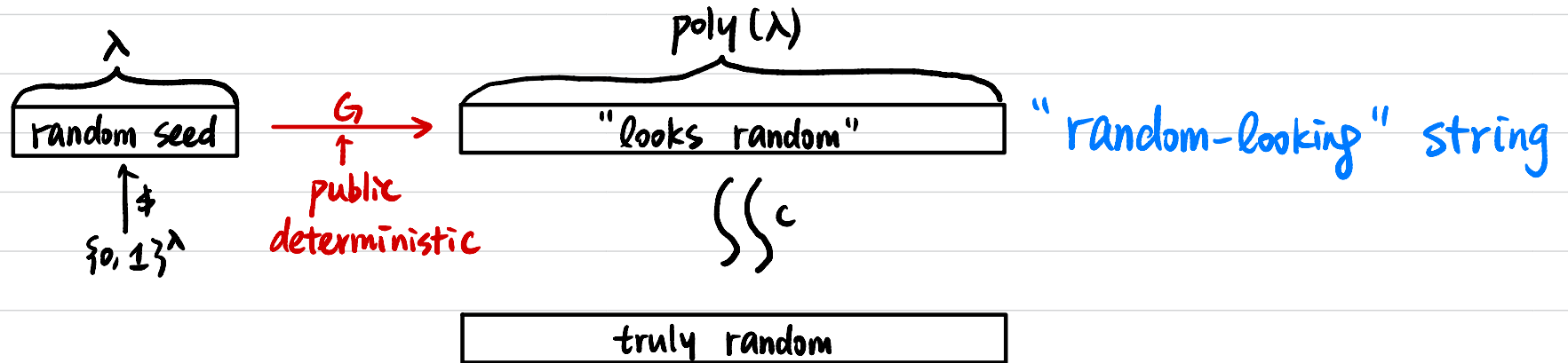
Important
Tool

Primitive: Hash function

Construction: SHA

Pseudorandom Function (PRF)

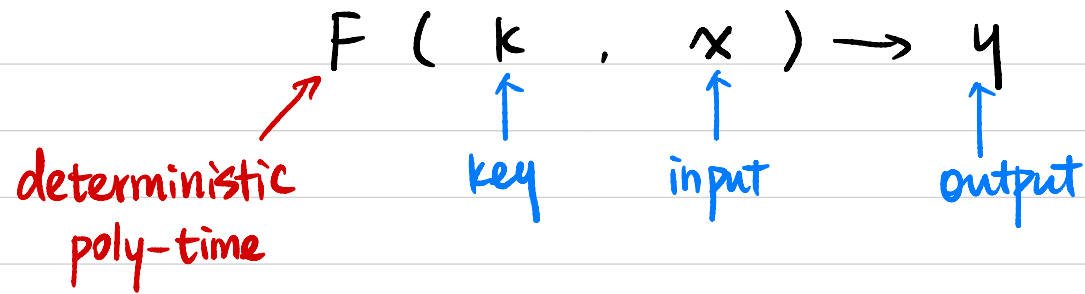
Pseudorandom Generator (PRG)



Pseudorandom Function (PRF): "random-looking" function

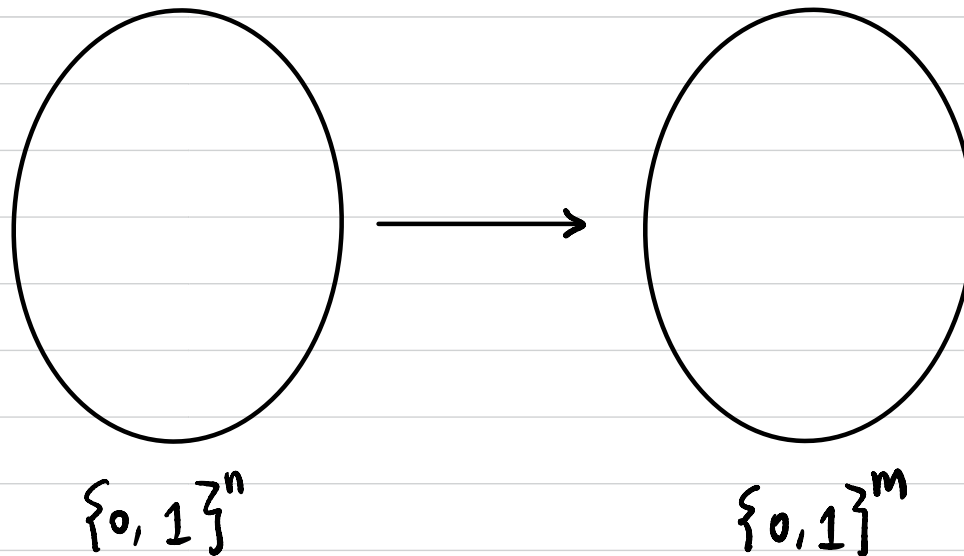
Pseudorandom Function (PRF)

Keyed Function $F: \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^m$



$k \leftarrow \{0,1\}^k$

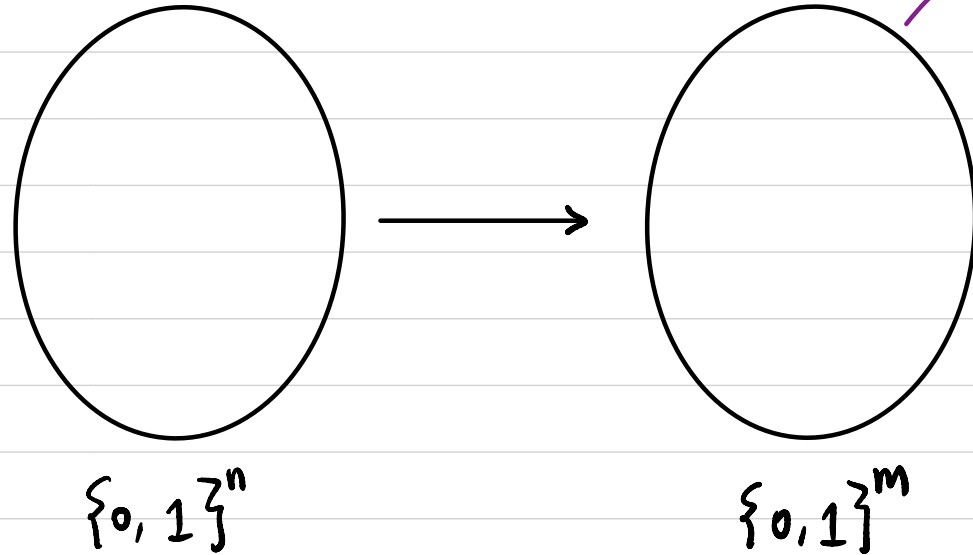
$F_k:$



"looks like a random function"

Pseudorandom Function (PRF)

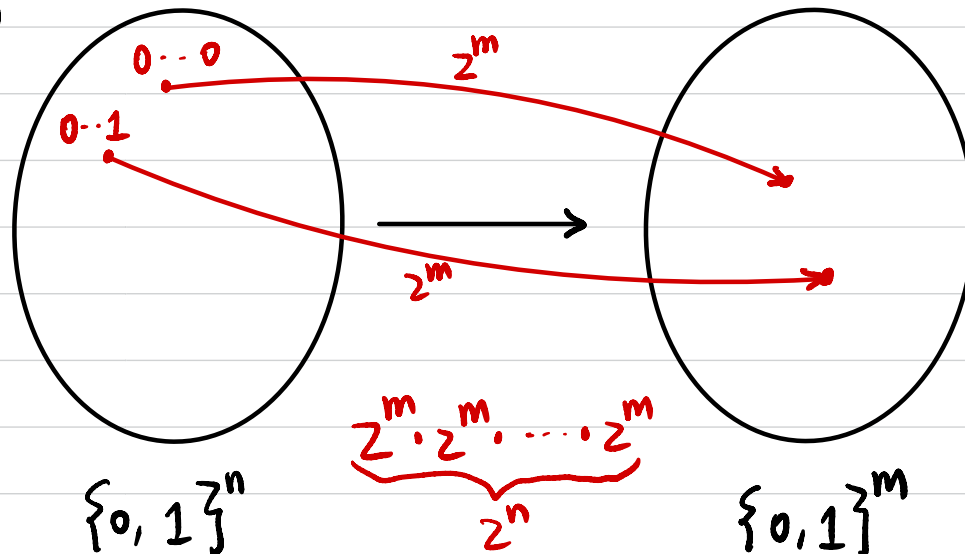
$$k \leftarrow \{0, 1\}^\lambda \quad F_k:$$



How many possible F_k 's?
 2^λ

$$f \leftarrow \{ F \mid F: \{0, 1\}^n \rightarrow \{0, 1\}^m \}$$

$f:$

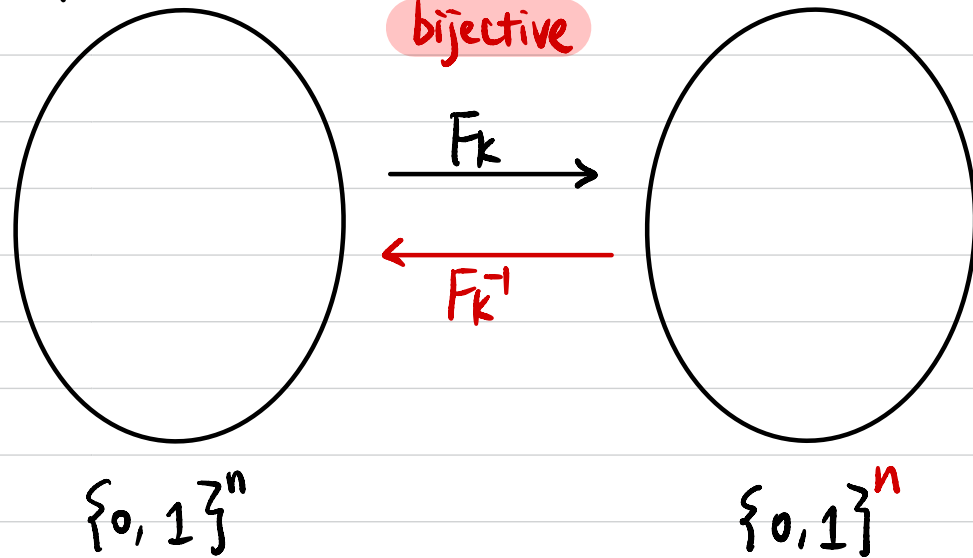


$\sum \sum c$ (not knowing k)

How many possible f 's?
 $(2^m)^{2^n}$

Pseudorandom Permutation (PRP)

$$k \leftarrow \{0, 1\}^n \quad F_k:$$

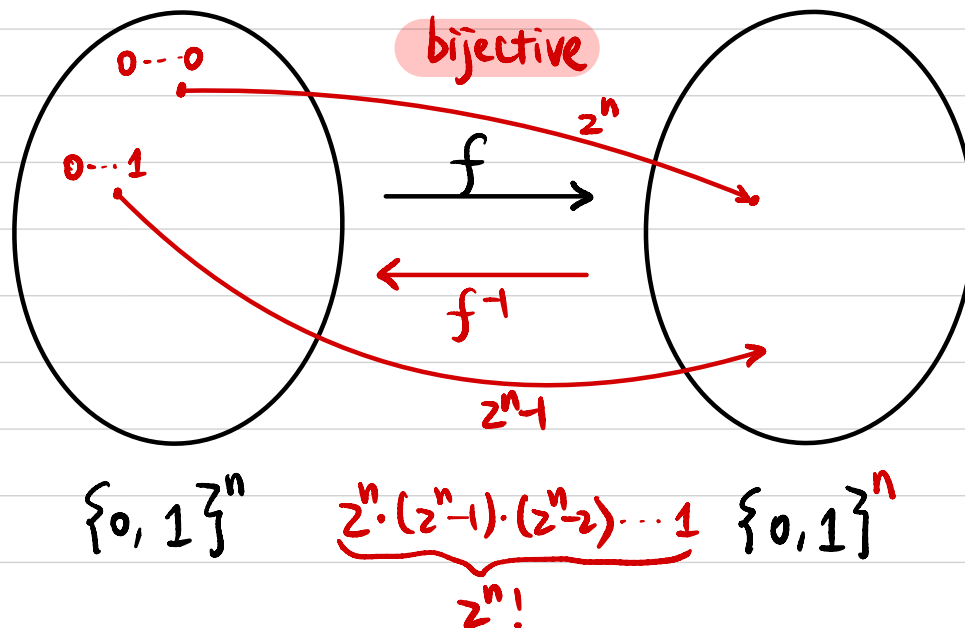


How many possible F_k 's?
 2^n

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

$f:$

$\sum \sum^c$ (not knowing k)



How many possible f 's?
 $2^n!$

Block Cipher

$$F: \{0, 1\}^\lambda \times \{0, 1\}^n \rightarrow \{0, 1\}^n$$

λ : key length

n : block length

It is assumed to be a pseudorandom permutation (PRP).

Construction: Advanced Encryption Standard (AES)

- $\lambda = n = 128$
- Standardized by NIST in 2001
- Competition 1997-2000

Before AES: Data Encryption Standard (DES)

- $\lambda = 56, n = 64$

Block Cipher Modes of Operation

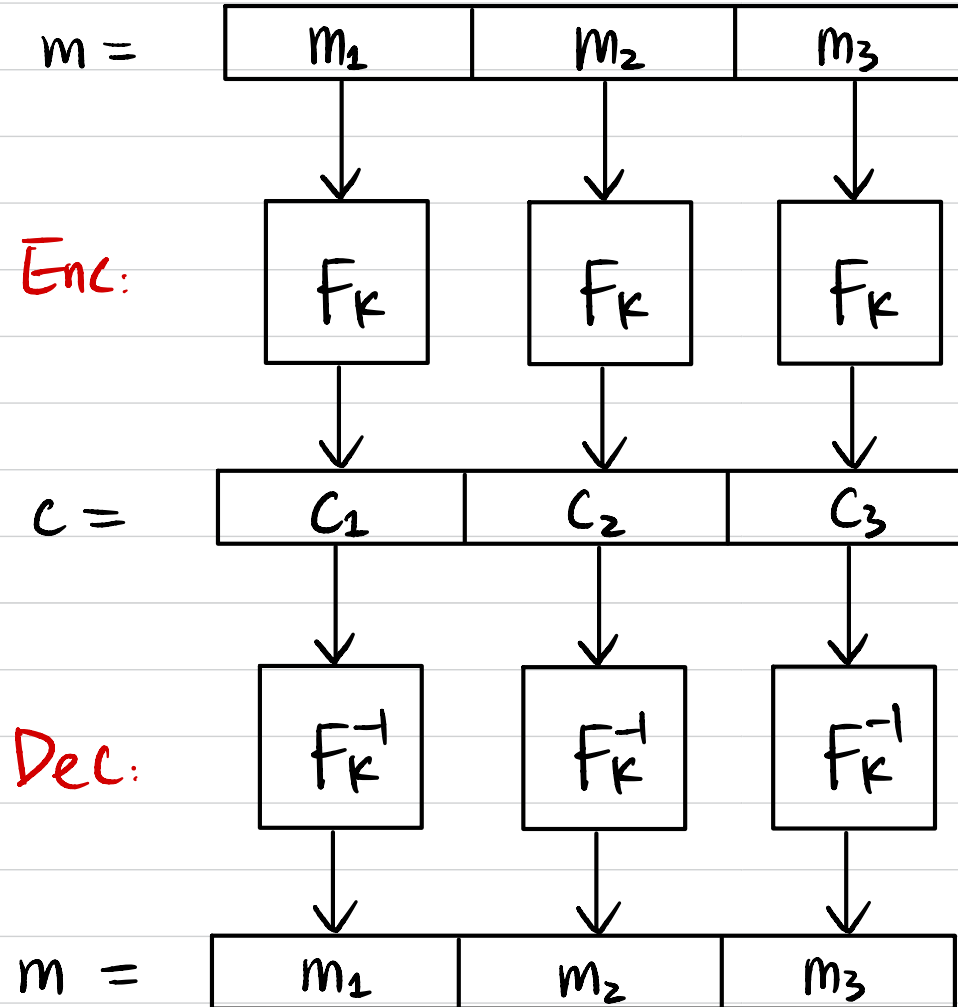
$$F: \{0, 1\}^\lambda \times \{0, 1\}^\lambda \rightarrow \{0, 1\}^\lambda$$

Construct an SKE scheme from F for arbitrary-length messages.

- $k \leftarrow \{0, 1\}^\lambda$
 - $\text{Enc}_k(m)$
 - $\text{Dec}_k(c)$
- $|m| = \alpha \cdot \lambda$ (If not, pad it to a multiple of λ)

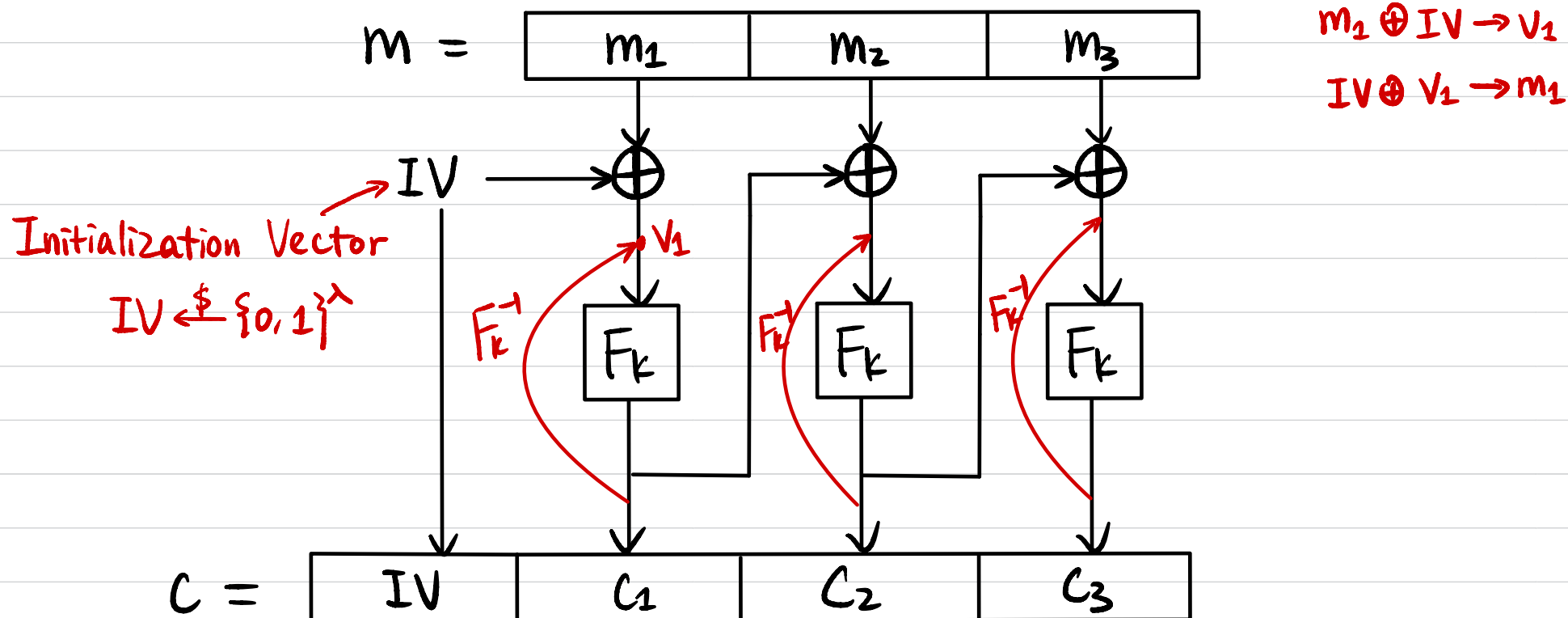
Goal: CPA (Chosen Plaintext Attack) Security

Electronic Code Book (ECB) Mode



CPA Secure? No!

Cipher Block Chaining (CBC) Mode



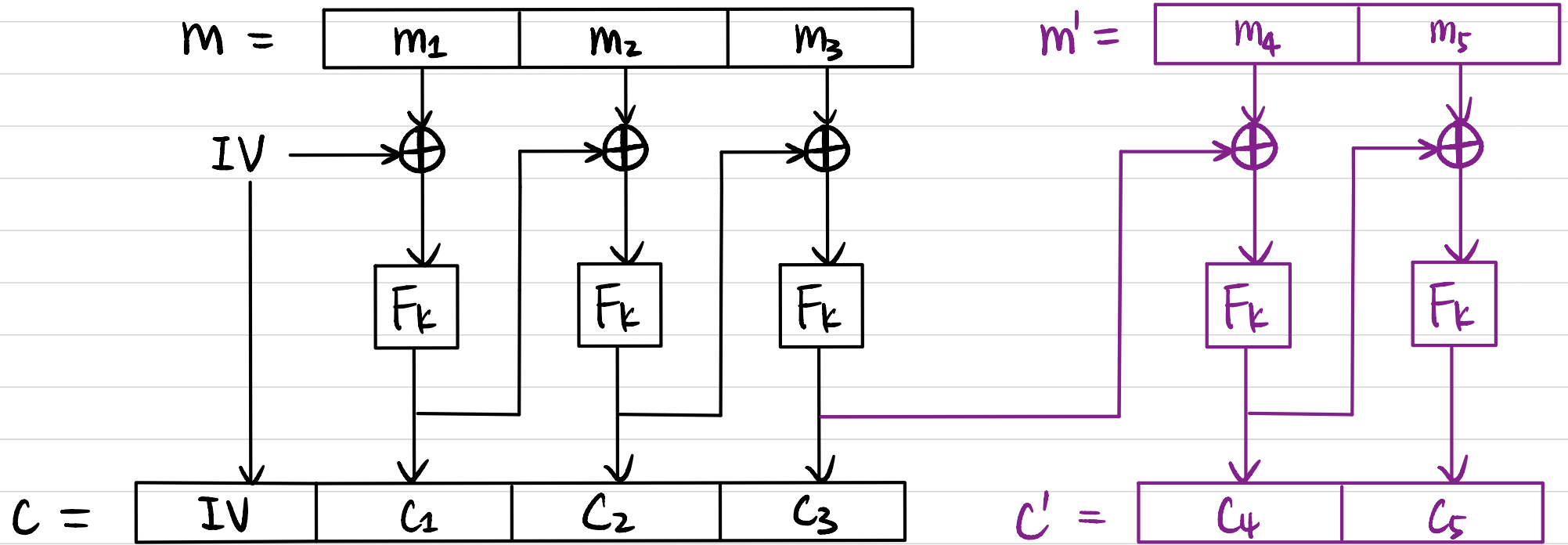
How to decrypt?

CPA Secure? YES!

Can we parallelize the computation?

NO for Enc
YES for Dec

Chained Cipher Block Chaining (CBC) Mode



CPA Secure?