CSCI 1510

This Lecture:

- Block Cipher Modes of Operation (continued)
- Practical Constructions of Hash Functions
- Midterm Review
- Selected Problems from Homework

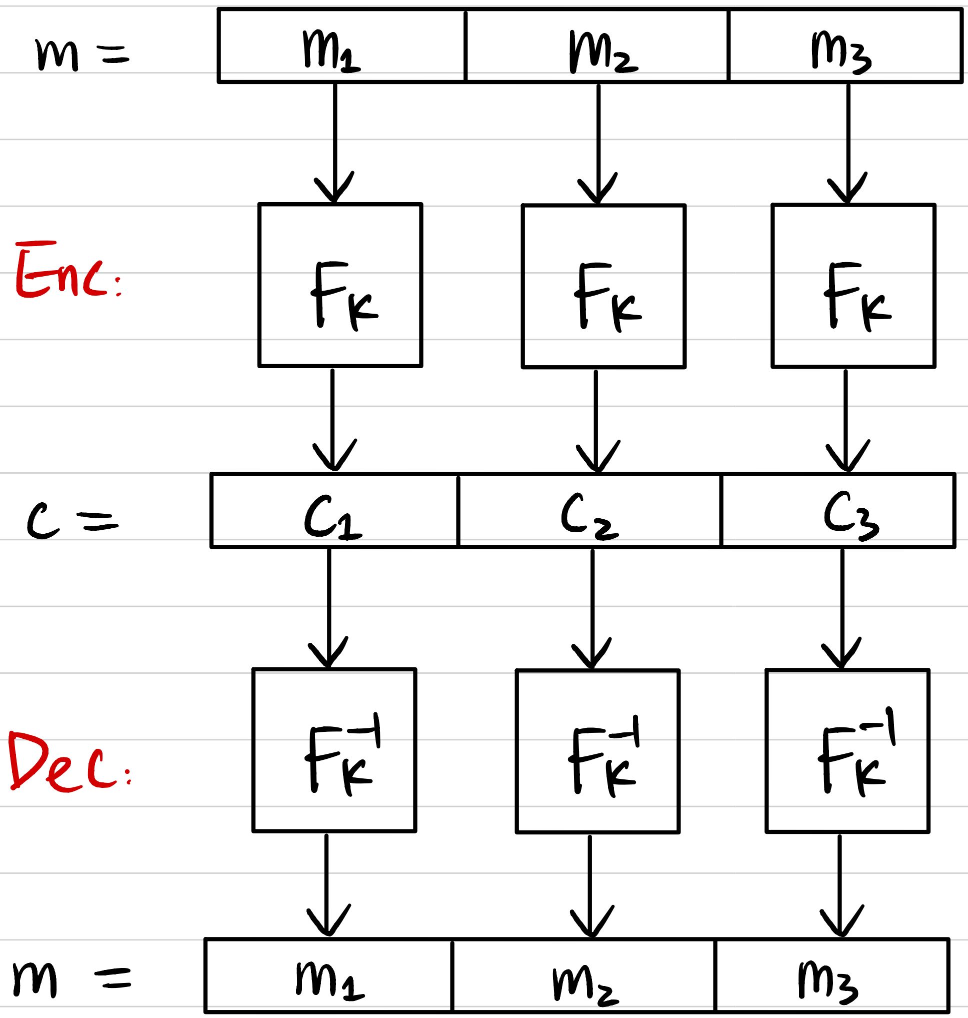
Block Cipher Modes of Operation

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

Assumed to be a pseudorandom permutation (PRP).

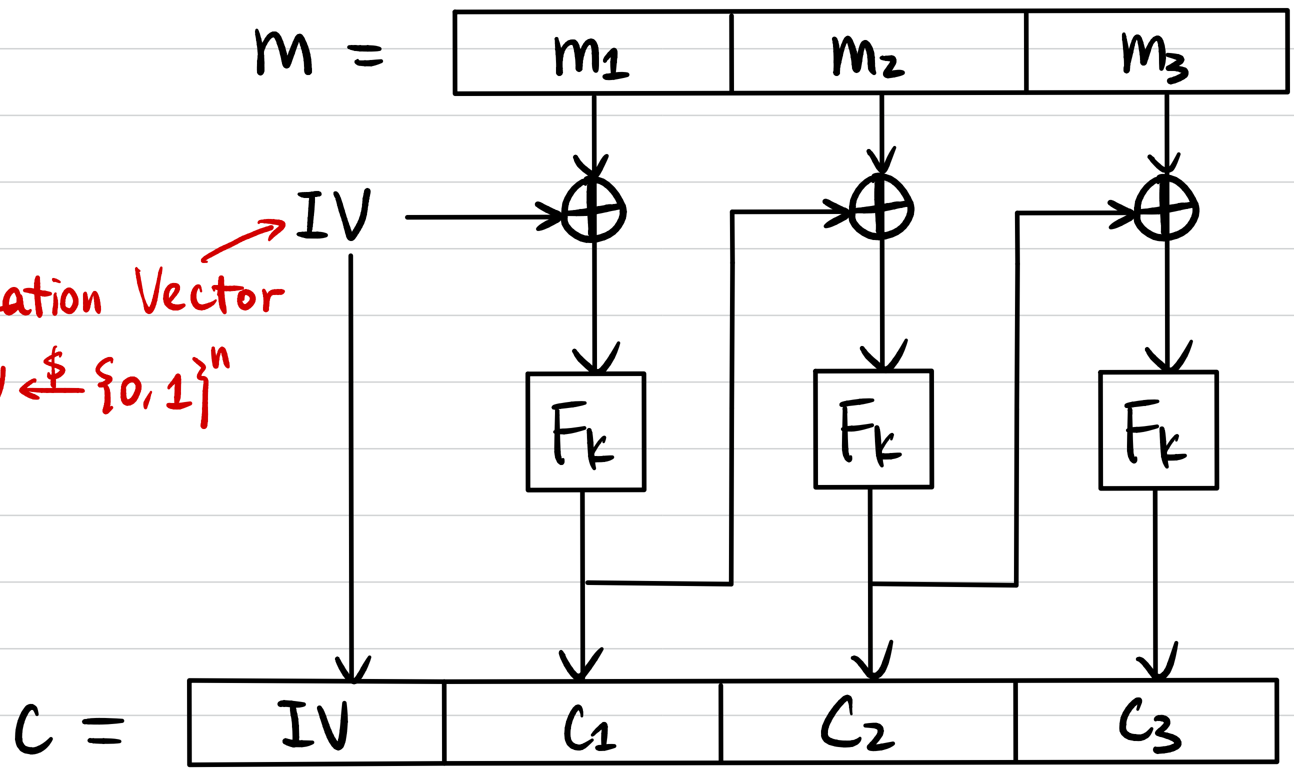
Goal: Construct a CPA-secure encryption scheme for arbitrary-length messages.

Electronic Code Book (ECB) Mode



CPA Secure? No! Deterministic Enc

Cipher Block Chaining (CBC) Mode



Initialization Vector

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

$\mathcal{H}_0$: $Enc(m_0)$

$\mathcal{H}_1$: $F_k \to f$

$\mathcal{H}_2$: $m_0 \to m_1$

$\mathcal{H}_3$: $f \to F_k$

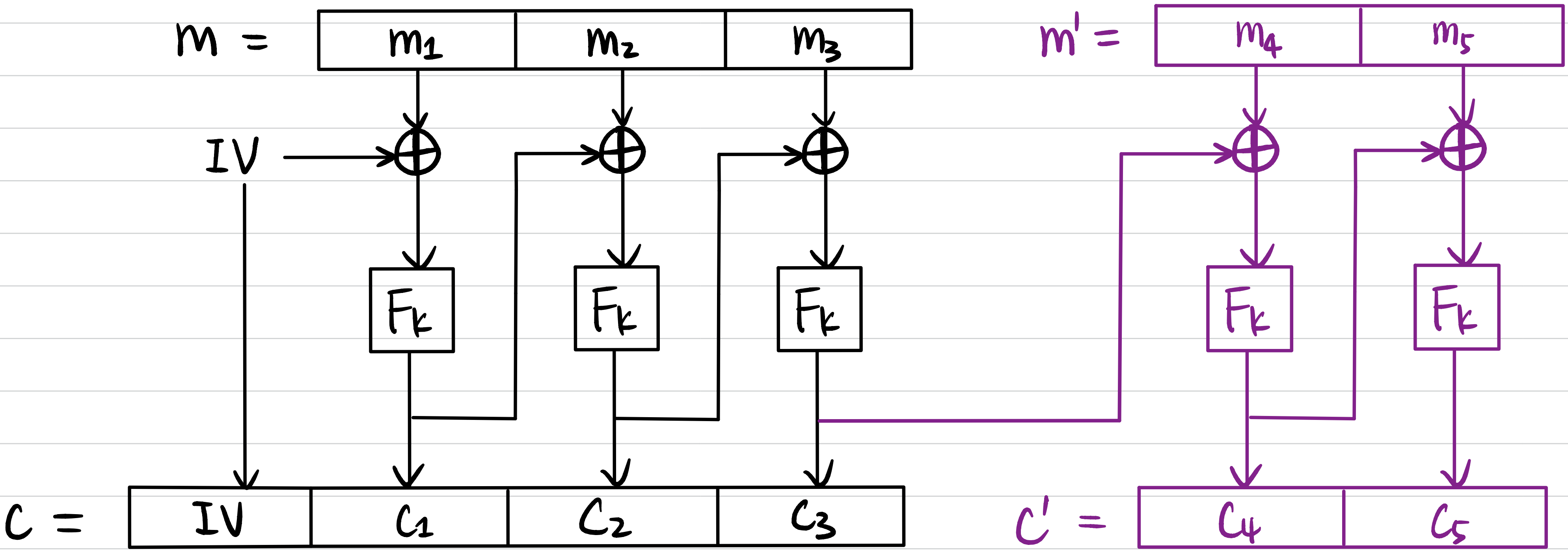
$Enc(m_1)$

How to decrypt? $F_k^{-1}(c_i) \oplus c_{i-1} \to m_i$

CPA secure? Yes!

Can we parallelize the computation? No for Enc, Yes for Dec.

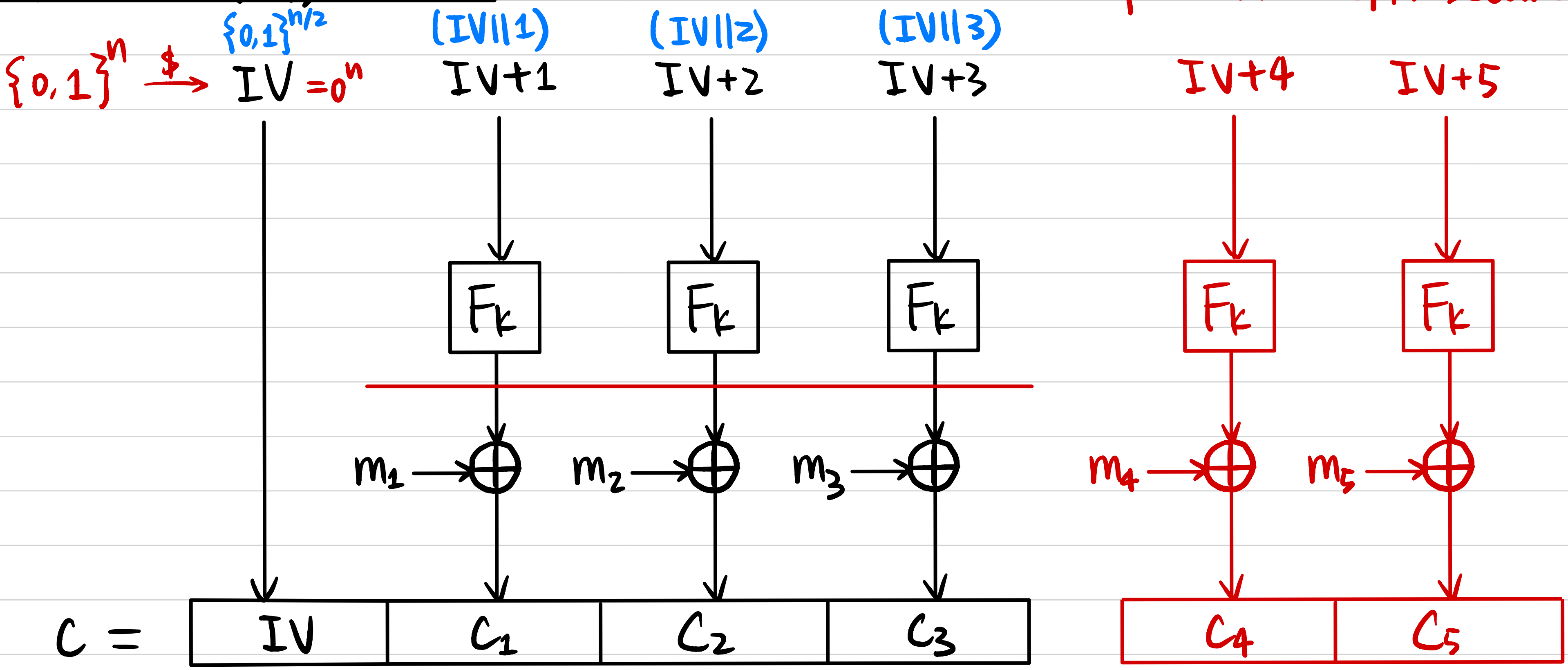
Chained Cipher Block Chaining (CBC) Mode



CPA Secure ? No!

Counter (CTR) Mode

Stateful CTR: CPA Secure



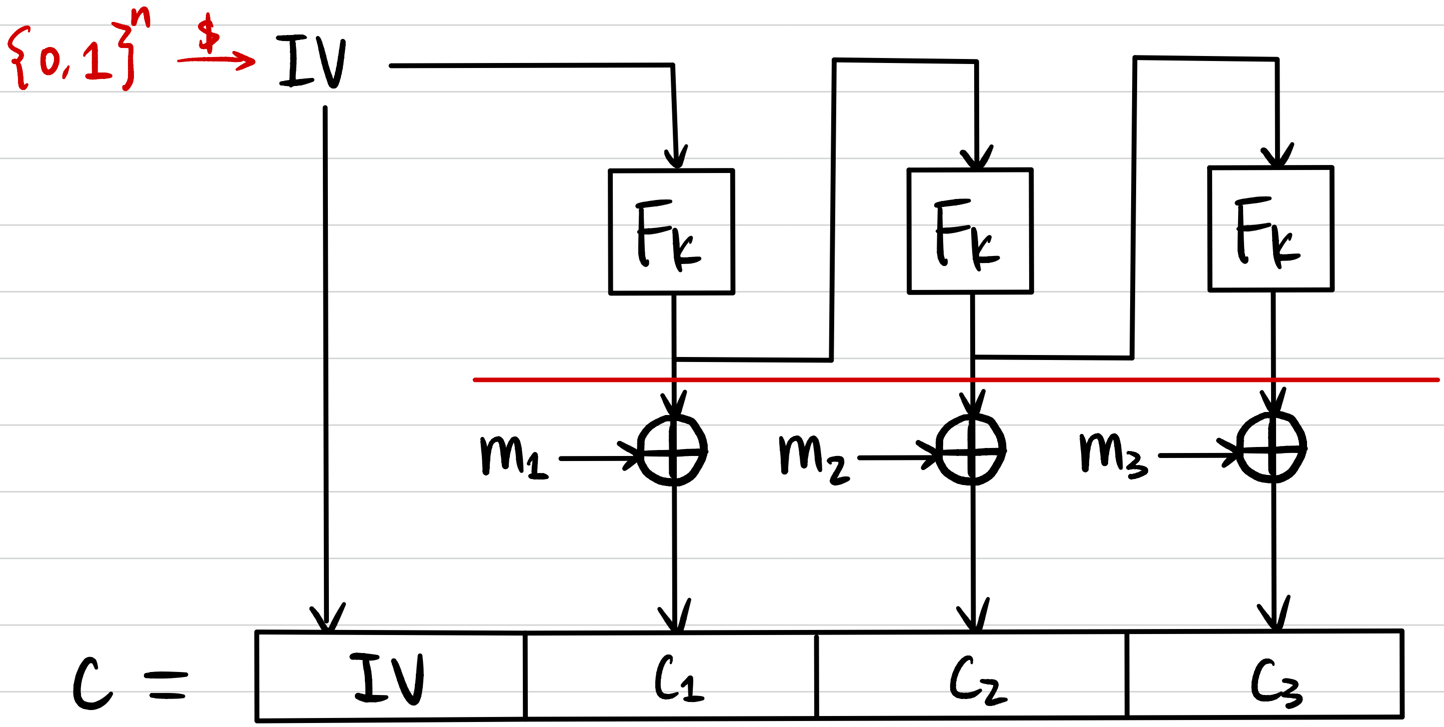
How to decrypt? $F_k(IV+i) \oplus c_i \Rightarrow m_i$

CPA Secure? Yes!

Can we parallelize the computation? Yes!

PRG from PRF $G(s) := F_s(1) \| F_s(2) \| \cdots$

Output Feedback (OFB) Mode



How to decrypt?

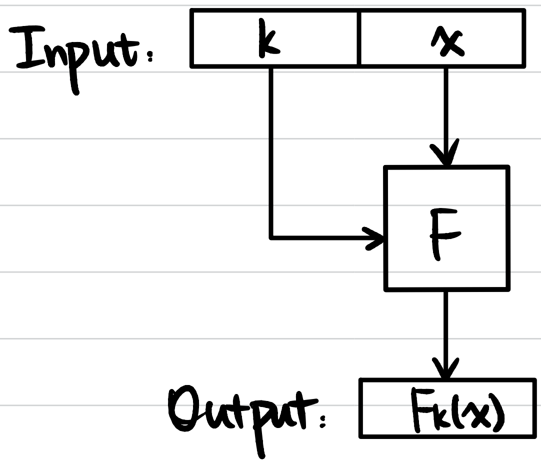
CPA secure?

Can we parallelize the computation?

PRG from PRF $\quad G(s) := F_s(0) \,\|\, F_s(F_s(0)) \,\|\, \cdots$

Compression Function from Block Cipher

Block Cipher $\xRightarrow{\text{Davies-Meyer}}$ Compression Function (fixed-length hash function) $\xRightarrow{\text{Merkle-Damgård}}$ Arbitrary-length hash function



If F is model as an "ideal cipher", then Davies-Meyer Construction is Collision-resistent.

Practical Constructions of Hash Function

MD5: output length 128-bit
best known attack 2^{16}
Collision found in 2004

Secure Hash Functions (SHA): Standardized by NIST.

- SHA-0: Standardized in 1993
  output length 160-bit
  best known attack 2^{39}

- SHA-1: Standardized in 1995
  output length 160-bit
  best known attack 2^{63}
  Collision found in 2017

Practical Constructions of 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

Midterm Review

- Symmetric-Key Encryption
  - Syntax
  - Kerckhoff's Principle

- Perfect Security
  - Definition
  - Construction: One-Time Pad
  - Limitations: $|K| \geq |M|$

- Computational Security
  - Negligible function & Asymptotic approach

Midterm Review

- Computational Security for Message Secrecy
  - * Semantic Security
    - Definition
    - Construction: Pseudo-OTP from PRG ← Definition
    - Proof by reduction
    - Limitations: Cannot reuse key
  - * CPA Security
    - Definition
    - Construction from PRF ← Definition
    - Proof by hybrid argument + reduction
    - Limitations: Cannot query for decryption
  - * CCA Security
    - Definition

Midterm Review

- Message Integrity
  * Message Authentication Code (MAC)
    - Syntax
    - Definitions: Secure / Strongly secure
    - Constructions

      Fixed-length MAC of length n from PRF

      Fixed-length MAC of length $\ell(n) \cdot n$ from PRF : CBC-MAC

      Arbitrary-length MAC : extension of CBC-MAC

  * Unforgeability of Encryption Scheme
    - Definition

- Authenticated Encryption: Secrecy & Integrity
  - Definition: CCA Secure & Unforgeable
  - Constructions : CPA-secure encryption + MAC

Midterm Review

- Practical Constructions
  - Block Cipher: PRP ← Definition
  - Constructions: SPN / Feistel Network / DES / AES
  - Attacks on reduced rounds
  - Modes of Operation

Midterm Review

- Hash Function
  - Definition: Collision-Resistant
  - Birthday Attack & Implications
  - Merkle-Damgård Transform
  - Applications
  - Practical Constructions: Davies-Meyer / SHA

c. Alice and Bob are arguing in class. Bob insists that an encryption scheme with message space $\mathcal{M}$ is perfectly secure if and only if for every probability distribution over $\mathcal{M}$ and every pair of ciphertexts $c_0, c_1 \in \mathcal{C}$, it is the case that any computed ciphertext C must be equally likely to be c_0 or c_1, i.e. that $\Pr[C = c_0] = \Pr[C = c_1]$.

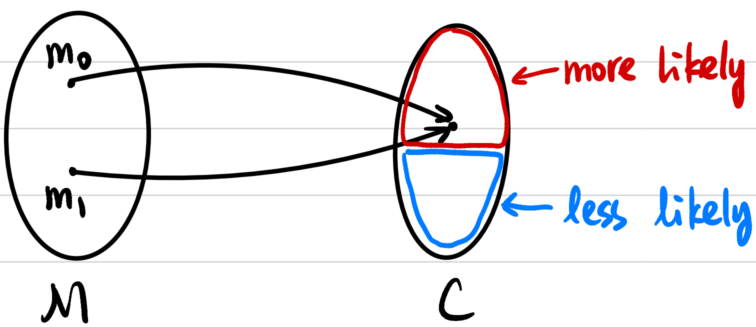
If you think Bob is correct, help him out by writing a proof of the statement. Otherwise, help Alice convince him that he is wrong by providing a counterexample.

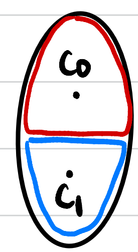
Π is perfectly secure $\;\not\Longrightarrow\;$ $\forall$ prob. dist. over $\mathcal{M}$

$\forall c_0, c_1 \in \mathcal{C}$, $\Pr[C=c_0] = \Pr[C=c_1]$

$\forall m_0, m_1, c$,

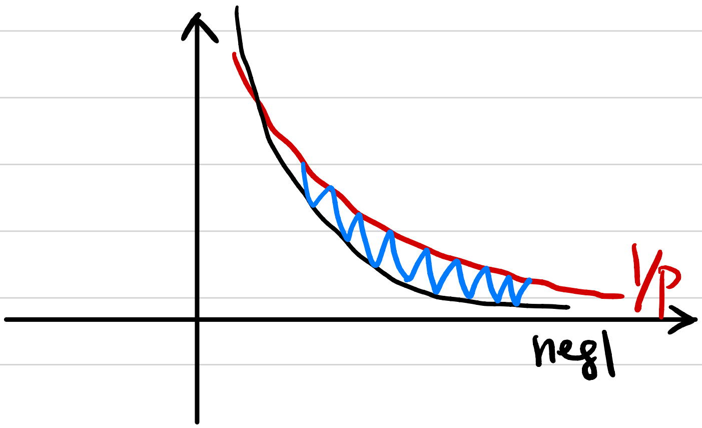
$\Pr[C=c \mid M=m_0] = \Pr[C=c \mid M=m_1]$





$\mathrm{Enc}_k(m) := m \oplus k \,\|\, b$ — 60% 0, 40% 1

c. Suppose that $\varepsilon : \mathbb{N} \to [0,1]$ is *not* a negligible function. Is the following statement true: There exists a polynomial p where $p(k) > 0$ for all k, and some $k_0 \geq 1$, such that $\varepsilon(k) > 1/p(k)$ for all $k > k_0$. In other words, is ε necessarily asymptotically greater than some inverse polynomial? If you think the statement is true for every non-negligible function ε, prove it. Otherwise, provide a counterexample.



$$\varepsilon(n) := \begin{cases} 2^{-n} & \text{if } n \text{ is even} \\ 1/n^2 & \text{if } n \text{ is odd} \end{cases}$$

3 GGM and Prefix-Constrained PRFs

A PRF $F: \{0,1\}^k \times \{0,1\}^k \mapsto \{0,1\}^k$ is said to be a prefix-constrained PRF if, given the PRF key, it is possible to generate a *constrained* PRF key K_π which lets you evaluate the PRF only at inputs which have a specific prefix π. More precisely, a prefix-constrained PRF has the following algorithms:

Setup: $\mathsf{Setup}(1^k)$ outputs a key $K \leftarrow \{0,1\}^k$

Constrain: For any string π such that $|\pi| \le k$, $\mathsf{Constrain}(K, \pi)$ outputs a key K_π

Evaluate: $\mathsf{Eval}(K_\pi, x)$ outputs $F_K(x)$ iff. $x = \pi \| t$ for some $t \in \{0,1\}^{k-|\pi|}$, else outputs $\perp$

The security notion for a constrained PRF key K_π is that it should reveal no information about the PRF evaluation at points that do not have the prefix π. For any string π such that $|\pi| \le k$, let X_π denote the set of all $x \in \{0,1\}^k$ that do *not* have π as their prefix. We say $F: \{0,1\}^k \times \{0,1\}^k \mapsto \{0,1\}^k$ is a *spring-break*-secure prefix-constrained PRF if for all PPT $\mathcal{A}$, there exists a negligible function $\nu(\cdot)$ such that

$$\left|\Pr[\mathcal{A}(1^k) \text{ outputs } b' = 0 \text{ in Exp 1}] - \Pr[\mathcal{A}(1^k) \text{ outputs } b' = 0 \text{ in Exp 2}]\right| \le \nu(k)$$

Exp 1	**Exp 2**
Choose key $K \leftarrow \mathsf{Setup}(1^k)$	Choose key $K \leftarrow \mathsf{Setup}(1^k)$ Choose random function $R: \{0,1\}^k \mapsto \{0,1\}^k$
$\mathcal{A}$ chooses a prefix π with $\lvert\pi\rvert \le k$ and obtains $K_\pi = \mathsf{Constrain}(K, \pi)$	$\mathcal{A}$ chooses a prefix π with $\lvert\pi\rvert \le k$ and obtains $K_\pi = \mathsf{Constrain}(K, \pi)$
$\mathcal{A}$ adaptively queries $F_K(\cdot)$ on any inputs $x_1, \dots, x_q \in X_\pi$ and obtains values $F_K(x_i)$ for $1 \le i \le q$	$\mathcal{A}$ adaptively queries $R(\cdot)$ on any inputs $x_1, \dots, x_q \in X_\pi$ and obtains values $R(x_i)$ for $1 \le i \le q$
$\mathcal{A}$ outputs a guess b'	$\mathcal{A}$ outputs a guess b'

In this problem, we will prove that the Goldreich-Goldwasser-Micali (GGM) PRF is also a prefix-constrained PRF. The GGM PRF is obtained as follows: Start with a length-doubling PRG $G: \{0,1\}^k \to \{0,1\}^{2k}$. So $G(s)$ for any $s \in \{0,1\}^k$ outputs a string of length $2k$; we call the first half $G_0(s)$ and second half $G_1(s)$. Let the input be $x = x_1 x_2 \dots x_k$ where each $x_i \in \{0,1\}$. Then, the PRF, with key K is defined as follows:

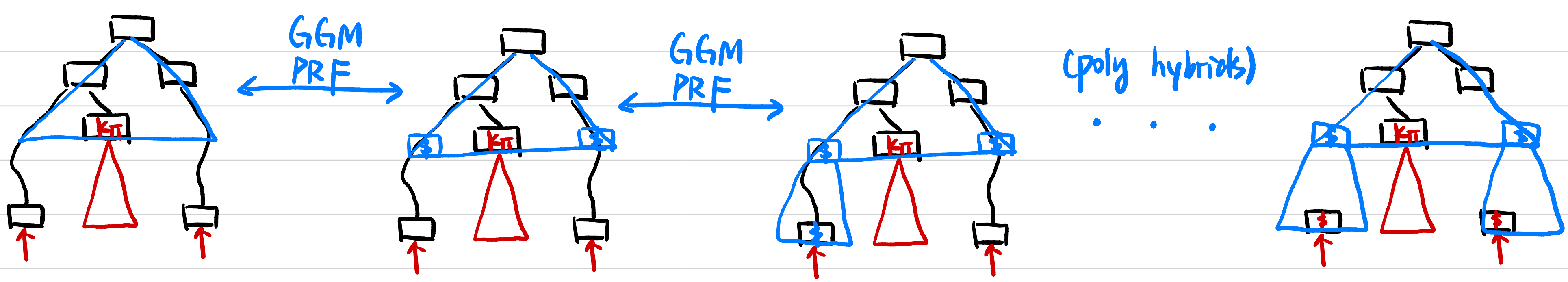
$$F_K(x_1 x_2 \dots x_k) = G_{x_k}(\dots G_{x_2}(G_{x_1}(K))\dots)$$

a. For the GGM PRF, what could be the constrained key K_0 that lets you evaluate $F_K(x)$ for all x starting with a 0? How will you evaluate the PRF with this constrained key?

b. Design the $\mathsf{Constrain}(K, \pi)$ algorithm for any prefix π with $|\pi| \le k$ for the GGM PRF.

c. Describe the corresponding $\mathsf{Eval}(K_\pi, x)$ algorithm.

d. Prove that your prefix-constrained PRF is *spring-break*-secure. You may assume that the GGM PRF $F_K^d(x): \{0,1\}^k \times \{0,1\}^d \to \{0,1\}^k$ is secure for any depth $d = \mathsf{poly}(k)$, not just $d = k$.



4 Leaky PRF

Construct a PRF $F : \{0,1\}^{k+1} \times \{0,1\}^n \mapsto \{0,1\}^n$ with the property that, if an adversary learns the first bit of the secret key of the PRF, then F *is* distinguishable from random. Prove that your construction of F is a PRF and show how the adversary can distinguish F from random if it knows the first bit of the secret key. You may assume that PRFs exist, and use another PRF in your construction.

$F' : \{0,1\}^k \times \{0,1\}^n \longrightarrow \{0,1\}^n$

$$F_s(x) := \begin{cases} s[0] \,\|\, F'_s(x)_{[1:n-1]} & \text{if } x = x^* \\ F'_k(x) & \text{otherwise} \end{cases}$$

1 CPA Security from PRFs and PRGs

Let $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be a PRF and $G : \{0,1\}^n \to \{0,1\}^{n+1}$ be a PRG with expansion factor $\ell(n) = n+1$. Consider the following encryption schemes based on F and G, where in each case, the secret key is a uniform $k \in \{0,1\}^n$.

For each scheme, state 1) whether the scheme is semantically secure and 2) whether it is CPA-secure. Explain your answer **for each security definition** - if you think the scheme is secure under some definition, prove it; otherwise, give an attack.

a. To encrypt a message $m \in \{0,1\}^{n+1}$, choose a uniform $r \in \{0,1\}^n$ and output the ciphertext $\langle r, G(r) \oplus m \rangle$.

b. To encrypt $m \in \{0,1\}^n$, output the ciphertext $m \oplus F_k(0^n)$.

c. To encrypt $m \in \{0,1\}^{2n}$, parse m as $m_1 \| m_2$ with $|m_1| = |m_2|$, then choose uniform $r \in \{0,1\}^n$ and output the ciphertext $\langle r, m_1 \oplus F_k(r), m_2 \oplus F_k(r+1) \rangle$.

CTR mode

4 Secure Arbitrary-Length CBC-MAC

Consider the following modification of the basic CBC-MAC construction. First, $\mathsf{Mac}_k(m)$ computes $k_\ell = F_k(\ell)$, where F is a PRF and ℓ is the length of m. Then, compute the tag using basic CBC-MAC with key k_ℓ. Verify is canonical verification.

Prove that this modification gives a secure MAC for arbitrary-length messages. For simplicity, assume all messages have length a multiple of the block length. You may assume fixed-length CBC-MAC is secure.