Pseudorandom Function

CS 5158/6058 Data Security and Privacy

Spring 2018

Instructor: Boyang Wang

A Fixed-Length Encryption

- A Fixed-Length Encryption with PRG
  - Similar as OTP, use PRG instead of random key
  - Secure (indistinguishable) if G is a PRG

Let $G(\{0,1\}^n) \to \{0,1\}^{l(n)}$ be a PRG, build an encryption scheme Π for messages of length $l(n)$:

- $\mathsf{KeyGen}(1^n) : k \leftarrow \{0,1\}^n$, return k.
- $\mathsf{Enc}_k(m) : c \leftarrow G(k) \oplus m$, return c
- $\mathsf{Dec}_k(c)\ m \leftarrow G(k) \oplus c$, return m

Comparison with OTP

- $\mathsf{KeyGen}(1^n) : k \leftarrow \{0,1\}^n$
- $\mathsf{Enc}_k(m) : c \leftarrow k \oplus m$

- $\mathsf{KeyGen}(1^n) : k \leftarrow \{0,1\}^n$
- $\mathsf{Enc}_k(m) : c \leftarrow G(k) \oplus m$

	One-Time Pad	Fixed-Length
Function	deterministic	deterministic
Key size	n bits	n bits
Message size	n bits	l(n) bits
Security	perfect	computational
Is it practical?	no	yes

Revisit Security Game

- So far, only consider <u>one ciphertext</u> (i.e., a single pair of messages) in the game

1. Adversary $\mathcal{A}$ outputs $m_0, m_1 \in \mathcal{M}$ with $|m_0| = |m_1|$
2. Challenger flips a coin $b \in \{0,1\}$, computes $c_b \leftarrow \mathsf{Enc}_k(m_b)$, where $k \leftarrow \mathsf{KeyGen}(1^n)$, and returns c_b to $\mathcal{A}$
3. $\mathcal{A}$ guesses a bit b'
4. Outputs 1 if $b' = b$, otherwise 0; $\mathcal{A}$ wins if it is 1

Multiple Encryption

- In practice, two parties share a key, and send <u>multiple ciphertexts</u>. E.g., iMessage, Emails

1. Adversary $\mathcal{A}$ outputs $\vec{M}_0 = (m_{0,1}, ..., m_{0,t})$, $\vec{M}_1 = (m_{1,1}, ..., m_{1,t})$, with $|m_{0,i}| = |m_{1,i}|$ for all i.
2. Challenger flips a coin $b \in \{0,1\}$, computes $c_b \leftarrow \mathsf{Enc}_k(m_{b,i})$ for all i, where $k \leftarrow \mathsf{KeyGen}(1^n)$, and returns $\vec{C} = (c_{b,1}, ..., c_{b,t})$ to $\mathcal{A}$
3. $\mathcal{A}$ guesses a bit b'
4. Outputs 1 if $b' = b$, otherwise 0; $\mathcal{A}$ wins if it is 1

Security on Multiple Encryption

- Adversary A distinguishes two sequences M_0 and M_1 with at most a negligible probability

Def. A symmetric-key encryption Π has indistinguishable multiple encryption if for all PPT adversaries $\mathcal{A}$ there is a negligible function s.t.

$$\Pr[\mathsf{PrivK}^{\mathsf{mult}}_{\mathcal{A},\Pi}(n) = 1] \leq \frac{1}{2} + \mathtt{negl}(n)$$

Chosen-Plaintext Attack (CPA)

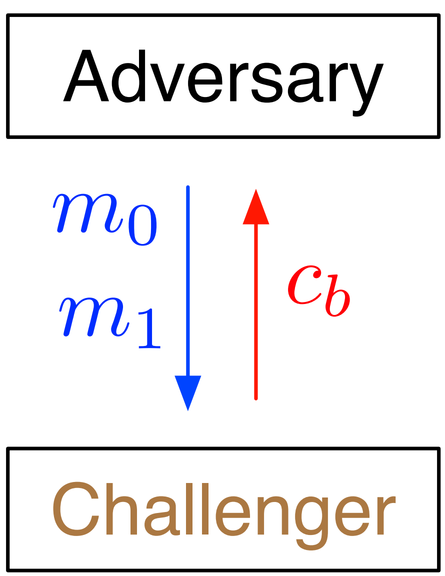
- Is the model of multiple ciphertexts sufficient?
  - Two sequences M_0, M_1, each with t messages

- Adversary can be stronger
  - May submit <u>unlimited</u> messages
  - May <u>adaptively</u> submit a later message based on previous messages/ciphertexts it has observed

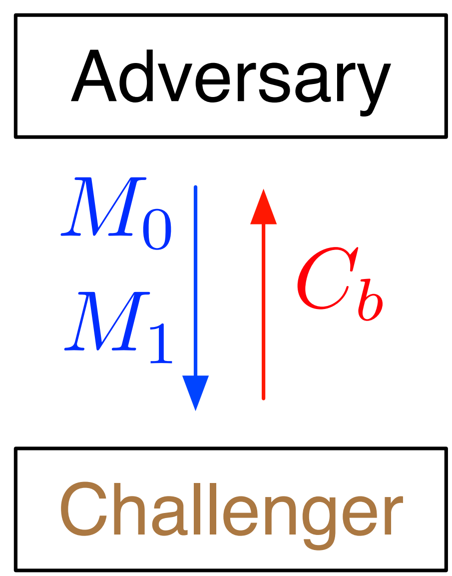
- Chosen-Plaintext Attack (CPA)
  - Most of the enc. we use are secure under CPA

Previous Security Games

Single ciphertext

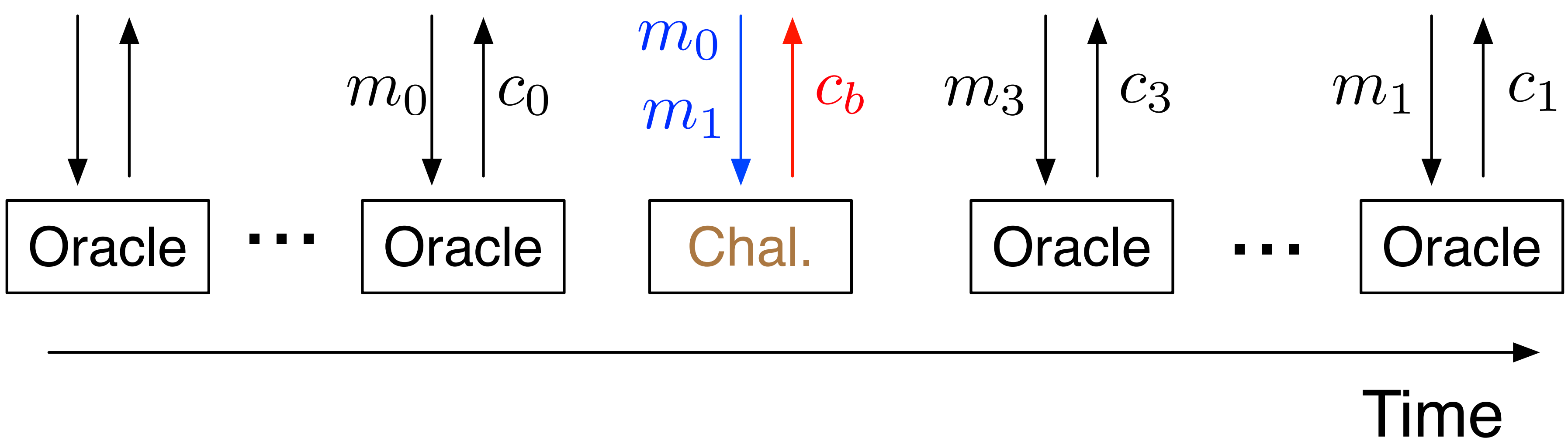


Multiple ciphertext



CPA Security Game

- Encryption Oracle
  - Informally, a black-box for encryption
  - Adversary does not know the key
  - Submit a message, return a ciphertext



CPA Security Game

1. A key $k \leftarrow \mathsf{KeyGen}(1^n)$ is generated.
2. Adversary $\mathcal{A}$ has access to encryption oracle $\mathsf{Enc}_k(\cdot)$, and outputs two messages m_0, m_1 with $|m_0| = |m_1|$
3. Challenger flips a fair coin $b \in \{0, 1\}$, computes $c_b \leftarrow \mathsf{Enc}_k(m_b)$, and returns c_b.
4. Adversary $\mathcal{A}$ continues to have oracle access to $\mathsf{Enc}_k(\cdot)$.
5. $\mathcal{A}$ guesses a bit b'
6. Outputs 1 if $b' = b$, otherwise 0; $\mathcal{A}$ wins if it is 1

CPA Security Game

Def. A symmetric-key encryption Π is indistinguishable under chosen-plaintext attacks, or is CPA-secure, if for all PPT adversaries $\mathcal{A}$ there is a negligible function s.t.

$$\Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n) = 1] \leq \frac{1}{2} + \mathtt{negl}(n)$$

- Is CPA model really necessary?
  - U.S. knew **`AF`** was the target, suspected Midway
  - U.S. sent "Midway is low on water."
  - Japan sent "**`AF`** is low on water."
  - <u>Practice</u>: Who was Adversary in CPA?

Deterministic v.s. Probabilistic

- Deterministic enc. is not secure under multiple ciphertexts or under CPA
  - $M_0 = (m_{0,1}, m_{0,2})$ and $M_1 = (m_{1,1}, m_{1,2})$
    - $m_{0,1} == m_{0,2}$ and $m_{1,1} \mathrel{!=} m_{1,2}$
    - Return $C_b = (c_{b,1}, c_{b,2})$
    - If $c_{b,1} == c_{b,2}$, $b' = 0 = b$; else $b' = 1 = b$

- Need <u>probabilistic encryption</u>
  - Output different ciphertexts from a same message

Other Attacks

- Ciphertext-only attack:
  - An attacker knows a set of ciphertexts, $c_1, \ldots, c_n$, it attempts to determine the messages of those ciphertexts.

- Known-plaintext attack:
  - An attacker knows a set of message-ciphertext pairs, $(m_1, c_1), \ldots, (m_n, c_n)$, it attempts to determine the message of a ciphertext c_{n+1}

Pseudorandom Function

- Func(m,n): a function family includes all the mappings from $\mathcal{D}=\{0,1\}^m \longrightarrow \mathcal{R}=\{0, 1\}^n$
  - E.g., m =3 and n =2, one f(d) from Func(3,2)

d	000	001	010	011	100	101	110	111
f(d)	10	11	11	00	10	01	11	01

- $|\text{Func}(m,n)| = 2^{n \cdot 2^m}$
  - 2^n outputs, each output has 2^m inputs

Function Family

- |Func(m,n)| = $2^{n \cdot 2^m}$
- <u>Example:</u> m = 2, n = 1, then |Func(2,1)| = 16

d	00	01	10	11
f1(d)	0	0	0	0
f2(d)	0	0	0	1
f3(d)	0	0	1	0
f4(d)	0	0	1	1

Function Family

- |Func(m,n)| = $2^{n \cdot 2^m}$
- Example: m = 2, n = 1, then |Func(2,1)| = 16

d	00	01	10	11
f5(d)	0	1	0	0
f6(d)	0	1	0	1
f7(d)	0	1	1	0
f8(d)	0	1	1	1

Function Family

- |Func(m,n)| = $2^{n \cdot 2^m}$
- Example: m = 2, n = 1, then |Func(2,1)| = 16

d	00	01	10	11
f9(d)	1	0	0	0
f10(d)	1	0	0	1
f11(d)	1	0	1	0
f12(d)	1	0	1	1

Function Family

- |Func(m,n)| = $2^{n \cdot 2^m}$
- Example: m = 2, n = 1, then |Func(2,1)| = 16

d	00	01	10	11
f13(d)	1	1	0	0
f14(d)	1	1	0	1
f15(d)	1	1	1	0
f16(d)	1	1	1	1

Function Family

- |Func(m,n)| = $2^{n \cdot 2^m}$
- Practice:
  - m = 3, n = 1, then |Func(3,1)| = ??
    - Give one function that is from Func(3,1)

  - m = 3, n = 1, then |Func(3,1)| = 2^8 = 256

d	000	001	010	011	100	101	110	111
f(d)	0	0	0	0	0	0	0	0

Keyed Function

- A <u>keyed function</u> *F* mapping from $\mathcal{D}$={0,1}^{m} —> $\mathcal{R}$={0, 1}^{n}:

$$F : \mathcal{K} \times \mathcal{D} \rightarrow \mathcal{R}$$
$$\mathcal{K} = \{0,1\}^l, \mathcal{D} = \{0,1\}^m, \mathcal{R} = \{0,1\}^n$$

  - First input is called the key *k*
  - *k* is <u>chosen uniformly</u> from $\mathcal{K}$

$$F_k(x) = F(k, x) = y$$

  - *F* is efficient (i.e., polynomial time)
  - Given a key *k*, F_k is **<u>deterministic</u>**

Keyed Function

$$F : \mathcal{K} \times \mathcal{D} \rightarrow \mathcal{R}$$
$$\mathcal{K} = \{0,1\}^l, \mathcal{D} = \{0,1\}^m, \mathcal{R} = \{0,1\}^n$$

- Example of keyed function *F*: l = 2, m = 2, n = 1

d	00	01	10	11
k=00,f(d)	1	1	0	0
k=01,f(d)	0	0	0	1
k=10,f(d)	0	1	1	0
k=11,f(d)	0	1	1	1

Keyed Function

$$F : \mathcal{K} \times \mathcal{D} \to \mathcal{R}$$
$$\mathcal{K} = \{0,1\}^l, \mathcal{D} = \{0,1\}^m, \mathcal{R} = \{0,1\}^n$$

- $|F|$: the number of functions in keyed function *F*:
  - Given each key, F_k is <u>deterministic</u>
  - $|F|$ is equal to the number of keys 2^l
  - E.g., m = 2, n = 1, l = 2, then $|F| = 2^l = 2^2 = 4$
- Function family: Func(m,n) $\mathcal{D}=\{0,1\}^m$ —> $\mathcal{R}=\{0, 1\}^n$
  - |Func(m, n)| = $2^{n \cdot 2^m}$
  - E.g., m = 2, n = 1, then |Func(2, 1)| = 16

Keyed Function

$$F : \mathcal{K} \times \mathcal{D} \to \mathcal{R}$$
$$\mathcal{K} = \{0,1\}^l, \mathcal{D} = \{0,1\}^m, \mathcal{R} = \{0,1\}^n$$

- Practice: Given $|F| = 2^l$, m = 3, n = 1, **l = 1**
  - $|F|$ = ??; give one example of F

d	000	001	010	011	100	101	110	111
0,f(d)	1	1	0	0	0	0	1	0
1,f(d)	0	0	0	1	1	0	1	1

Keyed Function

$$F : \mathcal{K} \times \mathcal{D} \rightarrow \mathcal{R}$$

$$\mathcal{K} = \{0,1\}^l, \mathcal{D} = \{0,1\}^m, \mathcal{R} = \{0,1\}^n$$

- <u>Practice:</u> Given $|F| = 2^l$, m = 3, n =1, **l = 2**
  - $|F|$ = ??; give one example of F

d	000	001	010	011	100	101	110	111
00,f(d)	1	1	0	0	0	0	1	0
01,f(d)	0	0	0	1	1	0	1	1
10,f(d)	0	1	1	0	1	0	1	1
11,f(d)	0	1	1	1	1	0	0	1

Pseudorandom Function

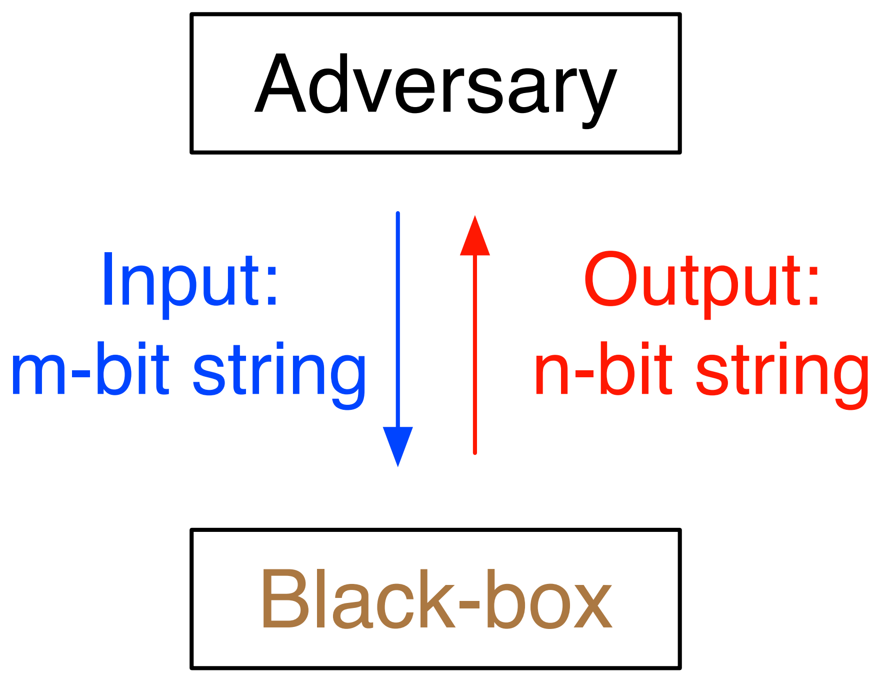
- F is a PRF: if F_k is indistinguishable from f
  - k is chosen uniformly from K
  - f is chosen uniformly from Func(m, n)

Def. Let $F : \{0,1\}^l \times \{0,1\}^m \to \{0,1\}^n$ be an efficient keyed function. F is a PRF is for all PPT adversary $\mathcal{A}$, there is a negligible function s.t

$$|\Pr[\mathcal{A}^{F_k(\cdot)}(1^l) = 1] - \Pr[\mathcal{A}^{f(\cdot)}(1^l) = 1]| \leq \mathtt{negl}(l)$$

Security of PRF

- Adversary A interacts with a black box (either function F_k or a random function f)
  - A cannot tell which one it is, if F is a PRF



PRF v.s. Func(m,n)

$$F_k : \{0,1\}^m \rightarrow \{0,1\}^n, \quad k \in \{0,1\}^l$$

- PRF *F* is not even close to Func(m, n)
  - $|F| = 2^l$
  - $|\text{Func}(m, n)| = 2^{n \cdot 2^m}$

- <u>Practice:</u> if m = 4, l = 2, and n = 2
  - What is |*F*|? and what is |Func(m, n)|?
  - $|F| = 2^2 = 4$; $|\text{Func}(4, 2)| = 2^{32} = 4{,}294{,}967{,}296$

4 v.s. 4,294,967,296

Example

- Is $F_k(m) = k \oplus m$ a PRF?
  - Efficient to compute, deterministic given a key
  - Outputs are uniformly distributed
    - Adversary $\mathcal{A}$ submits m_1, m_2, obtains c_1, c_2
    - If $c_1 \oplus c_2 = m_1 \oplus m_2$, $\mathcal{A}$ outputs 1

$$\Pr[\mathcal{A}^{F_k(\cdot)}(1^l) = 1] = 1, \quad \Pr[\mathcal{A}^{f(\cdot)}(1^l) = 1] = 2^{-n}$$

$$|\Pr[\mathcal{A}^{F_k(\cdot)}(1^l) = 1] - \Pr[\mathcal{A}^{f(\cdot)}(1^l) = 1]| > \mathtt{negl}(l)$$

Not a PRF

Additional Reading

Chapter 3, *Introduction to Modern Cryptography, Drs. J. Katz and Y. Lindell, 2nd edition*