Provable Security

수학문제연구회 36기 한승우

2025년 4월 28일

Notations

- $\{0,1\}^n$: 0, 1로 이루어진 n-tuple의 집합
  - e.g. $\{0,1\}^2 = \{(0,0),(0,1),(1,0),(1,1)\}$
- $\{0,1\}^*$: 0, 1로 이루어진 finite sequence의 집합
  - e.g. $\{0,1\}^* = \bigcup_{n=0}^{\infty}\{0,1\}^n$
- $u, v \in \{0,1\}^n$를 $\mathbb{Z}_2^n$ 위의 원소로 볼 수 있는데 이때 $u \oplus v$를 element-wise addition으로 정의 (XOR)
  - e.g. $(0,1,0) \oplus (1,1,0) = (1,0,1)$.
- $\mathcal{F}_n =$ 함수 $\{0,1\}^n \to \{0,1\}^n$의 집합
  - $|\mathcal{F}_n| = 2^{n2^n}$
- $\mathcal{P}_n =$ 순열(permutation) $\{0,1\}^n \to \{0,1\}^n$의 집합
  - $|\mathcal{P}_n| = (2^n)!$

Probabilistic Algorithm

- Probabilistic algorithm = probabilistic Turing machine = "동전을 던질 수 있는 컴퓨터"의 수학적 모델링
- $\Pr[y \leftarrow \mathcal{D}(x)]$ = probabilistic algorithm $\mathcal{D}$에 입력 x를 주었을 때 출력이 y일 확률 (x가 고정)
- $\Pr[x \xleftarrow{\$} X : y \leftarrow \mathcal{D}(x)] = X$에서 x를 uniform하게 뽑아 probabilistic algorithm $\mathcal{D}$에 입력으로 주었을 때 출력이 y일 확률

Encryption

암호화?

- $\mathcal{P}$ = plaintext의 집합 (암호화하고 싶은 것들)
- $\mathcal{C}$ = ciphertext의 집합 (암호화된 것들)
- $\mathcal{K}$ = key들의 집합 (i.e., 비밀번호의 집합)
- $\mathsf{Enc}: \mathcal{K} \times \mathcal{P} \to \mathcal{C}$: 암호화 함수(알고리즘)
- $\mathsf{Dec}: \mathcal{K} \times \mathcal{C} \to \mathcal{P}$: 복호화 함수(알고리즘)

암호화가 "작동"하기 위해서 Enc와 Dec가 만족해야 할 성질?

- $\mathsf{Dec}(k, \mathsf{Enc}(k, x)) = x$ for all $x \in \mathcal{P}$ and $k \in \mathcal{K}$.
- 충분한가? Enc와 Dec가 identity function이면 안 될 것 같은데...

"암호화"가 작동하기 위해서 Enc와 Dec가 만족해야 할 성질?

- plaintext x에 대한 정보를 $\mathsf{Enc}(k, x)$에서 알 수 없어야 함
- 이걸 어떻게 수학적으로 표현할까?

Perfect Security

Perfect Security (Information-Theoretic Security)

암호화 스킴이 “완벽히 안전”하다는 것은:

- 모든 $x_0, x_1 \in \mathcal{P}$와 모든 $c \in \mathcal{C}$에 대해

$$\Pr_{k \leftarrow \mathcal{K}}[c = \mathsf{Enc}(k, x_0)] = \Pr_{k \leftarrow \mathcal{K}}[c = \mathsf{Enc}(k, x_1)].$$

암호화 스킴이 완벽히 안전하면 “x의 ciphertext”의 분포와 “random string의 ciphertext”의 분포가 같음

Perfect Security: Another Formulation

어떤 방어자와 (시간이 무제한으로 주어진) 어떤 probabilistic algorithm $\mathcal{D}$가 공격자인 다음 “security game”을 생각해 보자.

1. 방어자는 $x_0, x_1 \in \mathcal{P}$를 고르고 공격자 $\mathcal{D}$에게 보낸다.
2. 방어자는 임의의 key $k \in \mathcal{K}$와 $b \in \{0, 1\}$를 uniform하게 고른다.
3. 방어자는 $c = \text{Enc}(k, x_b)$를 계산해서 공격자 $\mathcal{D}$에게 입력으로 보낸다.
4. $\mathcal{D}(x_0, x_1, c)$의 출력이 b과 같으면 공격자가 이기고, 다르면 방어자가 이긴다.

Perfect Security: Another Formulation

암호화 스킴이 “완벽히 안전”하다는 것은:

- 모든 $x_0, x_1 \in \mathcal{P}$에 대해 위 security game에서 공격자가 이길 확률은 항상 1/2이다.
- 다른 말로, 모든 $x_0, x_1 \in \mathcal{P}$에 대해 위 security game에서 공격자의 최선의 전략은 “찍기”이다.

Example: One-Time Pad

$\mathcal{P} = \mathcal{C} = \mathcal{K} = \{0,1\}^\ell$인 다음 암호화 스킴을 살펴보자.

One-Time Pad (OTP)

- $\mathsf{Enc}(k, x) = k \oplus x$
- $\mathsf{Dec}(k, c) = k \oplus c$

- 당연히 $\mathsf{Dec}(k, \mathsf{Enc}(k, x)) = k \oplus (k \oplus x) = x$.
- 모든 $x \in \mathcal{P}$에 대해서 $\mathsf{Enc}(k, x)$의 분포는 모두 같다.

따라서 **One-Time Pad는 완벽히 안전하다**. 그런데 왜 아무도 One-Tiem Pad를 실제로 안 쓸까?

Keyed Function

Keyed Function

함수

$$f : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$$

을 생각해 보자.이 함수의 첫 번째 parameter $k \in \mathcal{K}$를 고정하여 얻은 새 함수

$$\begin{aligned} f_k : \{0,1\}^n &\longrightarrow \{0,1\}^n \\ x &\longmapsto f(k,x) \end{aligned}$$

는 $\mathcal{F}_n$의 원소로 볼 수 있을 것이다. 이런 함수 f를 **keyed function**이라고 부른다.

Pseudorandom Function (PRF)

Keyed function $f : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$가 주어져 있다. 전과 비슷하게, 어떤 방어자와 (시간이 **다항 시간**으로 제한된) 어떤 probabilistic algorithm $\mathcal{D}$가 공격자인 다음 "security game"을 생각해 보자.

1. 방어자는 $b \in \{0,1\}$를 uniform하게 고른다.
2. $b = 0$이면, 방어자는 임의의 key $k \in \mathcal{K}$를 골라 $f^* = f_k$로 둔다.
3. $b = 1$이면, 방어자는 임의의 $f^* \xleftarrow{\$} \mathcal{F}_n$을 고른다.
4. 공격자는 여러 번의 "질의"를 할 수 있다. 각 질의 x_i에 대해 방어자는 $y_i = f^*(x_i)$를 공격자에게 알려준다.
5. 모든 질의의 마지막에 공격자는 $b' \in \{0,1\}$을 결정한다. 만약 $b = b'$이면 공격자가 이기고, $b \neq b'$이면 방어자가 이긴다.

Pseudorandom Function (PRF)

Advantage of Distinguisher

공격자 $\mathcal{D}$에 대해,

$$P_{\mathsf{re}} := \Pr[k \xleftarrow{\$} \mathcal{K} : 1 \leftarrow \mathcal{D}^{f_k}]$$

$$P_{\mathsf{id}} := \Pr[F \xleftarrow{\$} \mathcal{F}_n : 1 \leftarrow \mathcal{D}^{F}]$$

이라고 하면,

$$\mathsf{Adv}_f(D) = |P_{\mathsf{id}} - P_{\mathsf{re}}|$$

라고 하자.

$$(\mathcal{D}\text{가 이길 확률}) = \frac{1}{2}P_{\mathsf{id}} + \frac{1}{2}(1 - P_{\mathsf{re}}) = \frac{1}{2} + \frac{1}{2}(P_{\mathsf{id}} - P_{\mathsf{re}})$$

우리는 $\mathcal{D}$가 이길 확률을 줄이는 게 아니라 정확히 1/2 근처로 맞춰야 한다. (Why?)

Pseudorandom Function (PRF)

Pseudorandom Function (PRF)

만약 모든 $\mathcal{D}$에 대해 $\mathrm{Adv}_f(\mathcal{D}) \leq$ (negligible function on n) 이하이면 f를 **pseudorandom function (PRF)**라고 부른다.

$g(n)$ is negligible if $\lim_{n\to\infty} n^k g(n) = 0$ for all $k > 0$.

One-Time Pad with PRF

f가 pseudorandom function이면, 각 $k \in \mathcal{K}$에 대해

$$f_k(0) \parallel f_k(1) \parallel f_k(2) \parallel \cdots$$

는 random sequence과 구별할 수 없다. 그러면...?

One-Time Pad with PRF

- $\mathsf{Enc}(k, x) = x \oplus (f_k(0) \parallel f_k(1) \parallel f_k(2) \parallel \cdots)$
- $\mathsf{Enc}(k, c) = c \oplus (f_k(0) \parallel f_k(1) \parallel f_k(2) \parallel \cdots)$

이런 One-Time Pad with PRF는 더 이상 완전히 안전하지는 않지만, 기존의 One-Time Pad를 “안전하게 구현”한다고 할 수 있다.

Pseudorandom Permutation (PRP)

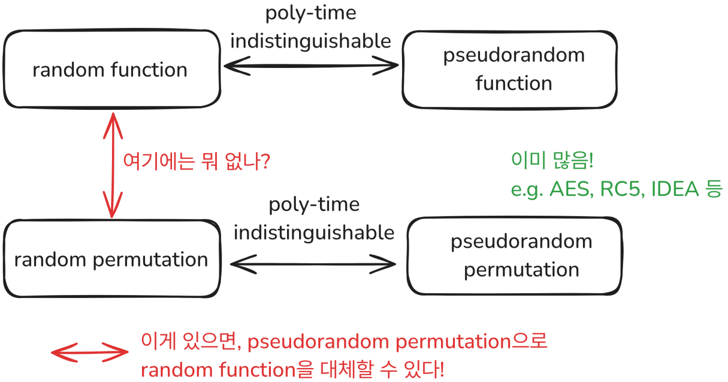
$f_k \in \mathcal{P}_n$인 keyed function $f : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$가 주어져 있다.

1. 방어자는 $b \in \{0,1\}$를 uniform하게 고른다.
2. $b = 0$이면, 방어자는 임의의 key $k \in \mathcal{K}$를 골라 $f^* = f_k$로 둔다.
3. $b = 1$이면, 방어자는 임의의 $f^* \xleftarrow{\$} \mathcal{P}_n$을 고른다.
4. 공격자는 여러 번의 "질의"를 할 수 있다. 각 질의 x_i에 대해 방어자는 $y_i = f^*(x_i)$를 공격자에게 알려준다.
5. 모든 질의의 마지막에 공격자는 $b' \in \{0,1\}$을 결정한다. 만약 $b = b'$이면 공격자가 이기고, $b \neq b'$이면 방어자가 이긴다.

Pseudorandom Function (PRF)

만약 모든 $\mathcal{D}$에 대해 $\mathrm{Adv}_f(\mathcal{D}) \leq$ (negligible function on n) 이하이면 f를 **pseudorandom permutation (PRP)**라고 부른다.

Motivation


poly-time
indistinguishable
random function
pseudorandom function
여기에는 뭐 없나?
이미 많음!
e.g. AES, RC5, IDEA 등
poly-time
indistinguishable
random permutation
pseudorandom permutation
이게 있으면, pseudorandom permutation으로
random function을 대체할 수 있다!

The Security Game

어떤 방어자와 (시간이 무제한인) 어떤 probabilistic algorithm $\mathcal{D}$가 공격자인 다음 security game을 생각해 보자.

1. 방어자는 $b \in \{0,1\}$를 uniform하게 고른다.
2. $b = 0$이면, 방어자는 임의의 $f^* \leftarrow \mathcal{P}_n$을 고른다.
3. $b = 1$이면, 방어자는 임의의 $f^* \leftarrow \mathcal{F}_n$을 고른다.
4. 공격자는 여러 번의 “질의”를 할 수 있다. 각 질의 x_i에 대해 방어자는 $y_i = f^*(x_i)$를 공격자에게 알려준다.
5. 모든 질의의 마지막에 공격자는 $b' \in \{0,1\}$을 결정한다. 만약 $b = b'$이면 공격자가 이기고, $b \neq b'$이면 방어자가 이긴다.

The Security Game

Advantage of Distinguisher

- 공격자 $\mathcal{D}$에 대해,

$$P_{\text{re}} := \Pr[P \xleftarrow{\$} \mathcal{P}_n : 1 \leftarrow \mathcal{D}^P]$$
$$P_{\text{id}} := \Pr[F \xleftarrow{\$} \mathcal{F}_n : 1 \leftarrow \mathcal{D}^F]$$

  이라고 하면,

$$\text{Adv}(\mathcal{D}) = |P_{\text{id}} - P_{\text{re}}|$$

  라고 하자.
- 질의를 최대 q번 하는 $\mathcal{D}$의 최대 Adv($\mathcal{D}$)값을 Adv(q)라고 하자.

PRP/PRF Switching Lemma

PRP/PRF Switching Lemma

$$\mathrm{Adv}(q) \le \frac{q^2}{2^n}.$$

뜻?

Assumptions

우리는 다음을 가정할 수 있다.

1. $\mathcal{D}$는 중복된 질의를 하지 않는다.
2. $P_{\text{id}} \geq P_{\text{re}}$
3. $\mathcal{D}$는 “동전을 던지지 않는다”. (probabilistic choice를 하지 않는다.)

Transcripts

Transcript

$\mathcal{D}$가 q회의 질의 $x_1, \cdots, x_q$를 하면, 답변 $y_1, \cdots, y_q$를 얻는다. 이런 tuple

$$(x_1, x_2, \cdots, x_q, y_1, y_2, \cdots, y_q)$$

를 transcript라고 부르고, transcript의 집합을 $\mathcal{T}$라고 하자.

$\mathcal{D}$가 동전을 던지지 않기 때문에, $\mathcal{D}$의 출력은 $\mathcal{D}$가 가진 transcipt $\tau \in \mathcal{T}$에 의해서 결정된다!

Transcripts

T_{re}, T_{id}를 $\mathcal{T}$ 위의 다음과 같은 확률 변수라고 하자.

- $\Pr[T_{\mathsf{re}} = \tau] = \mathcal{D}$가 real world에서 τ를 얻을 확률
- $\Pr[T_{\mathsf{id}} = \tau] = \mathcal{D}$가 ideal world에서 τ를 얻을 확률

그러면,

$$P_{\mathsf{re}} = \Pr[P \xleftarrow{\$} \mathcal{P}_n : 1 \leftarrow \mathcal{D}^P] = \sum_{\tau : \mathcal{D}(\tau) \to 1} \Pr[T_{\mathsf{re}} = \tau]$$

$$P_{\mathsf{id}} = \Pr[F \xleftarrow{\$} \mathcal{F}_n : 1 \leftarrow \mathcal{D}^F] = \sum_{\tau : \mathcal{D}(\tau) \to 1} \Pr[T_{\mathsf{id}} = \tau],$$

즉

$$\mathsf{Adv}(\mathcal{D}) = \sum_{\tau : \mathcal{D}(\tau) \to 1} (\Pr[T_{\mathsf{id}} = \tau] - \Pr[T_{\mathsf{re}} = \tau]).$$

Transcripts

따라서

$$\begin{aligned}
\mathsf{Adv}(\mathcal{D}) &= \sum_{\tau\,:\,\mathcal{D}(\tau)\to 1} \left(\Pr[T_{\mathsf{id}} = \tau] - \Pr[T_{\mathsf{re}} = \tau]\right) \\
&\le \sum_{\tau\,:\,\Pr[T_{\mathsf{id}}=\tau] > \Pr[T_{\mathsf{re}}=\tau]} \left(\Pr[T_{\mathsf{id}} = \tau] - \Pr[T_{\mathsf{re}} = \tau]\right) \\
&\le \sum_{\tau \text{ has duplicate outputs}} \left(\Pr[T_{\mathsf{id}} = \tau] - \Pr[T_{\mathsf{re}} = \tau]\right) \\
&\le \sum_{\tau \text{ has duplicate outputs}} \Pr[T_{\mathsf{id}} = \tau] \\
&\le \binom{q}{2}\frac{1}{2^n} \le \frac{q^2}{2^{n+1}}.
\end{aligned}$$

Questions?