

RSA-OAEP IND-CCA2 증명

김동현(wlswudpdlf31@kookmin.ac.kr), 국민대 FDL

April 29, 2025

Contents

1	논문정보	3
2	보안 개념	4
2.1	OW trapdoor permutation	4
2.2	Partial-domain OW trapdoor permutation	4
2.3	Set partial-domain OW trapdoor permutation	5
2.4	IND security against CCA2	5
2.5	IND security against CCA2 in ROM	6
3	RSA-OAEP	8
4	증명	9
4.1	증명: Reduction and simulation	9
4.2	증명: 사건 정의	10
4.3	증명: Analysis of the Decryption Oracle Simulation	10
5	Journal of Cryptology	13
5.1	랜덤 오라클 시뮬레이터	13
5.2	평문 추출기	13
5.3	게임 구성	15
5.3.1	0 번째 게임	15
5.3.2	1 번째 게임	16
5.3.3	2 번째 게임	18
5.3.4	3 번째 게임	20
5.3.5	4 번째 게임	21
5.3.6	5 번째 게임	22
5.3.7	6 번째 게임	23
5.3.8	7 번째 게임	24
5.3.9	8 번째 게임	25
6	박종환 교수님 발표자료 검토	28
6.1	Structure	28
6.2	Questions	28

7 t가 그대로 내려온다면	29
7.1 평문추출기	30
7.2 게임 변환	31
7.2.1 게임 0	31
7.2.2 게임 1	32
7.2.3 게임 2	33
7.2.4 게임 3	34
7.2.5 게임 4	35
7.2.6 게임 5	36
7.2.7 게임 6	37
7.2.8 게임 7	38
7.2.9 게임 8	39
7.3 리덕션	40
8 ??	41
8.1 Random Self-Reducibility	41
A 보조정리	43

1 논문정보

- 제목: RSA-OAEP is Secure under the RSA Assumption
- 저자: Eiichiro Fujisaki1, Tatsuaki Okamoto, David Pointcheval, and Jacques Stern
- 년도: 2001년
- 초록: 최근 Victor Shoup은 적응적 선택 암호문 공격에 대한 OAEP의 보안성에 관한 널리 받아 들여진 결과에 틈이 있음을 지적하였다. 더욱이, 그는 기본 트랩도어 치환의 단방향성만으로는 OAEP의 보안성을 증명할 수 없을 것으로 예상된다는 점을 보였다. 본 논문은 OAEP의 보안성에 대한 또 다른 결과를 제시한다. 즉, 본 논문에서는 무작위 오라클 모델에서, 기본 치환의 부분 영역 단방향성(partial-domain one-wayness) 하에서, OAEP가 적응적 선택 암호문 공격에 대해 의미론적 보안성을 제공함을 증명한다. 따라서, 이는 형식적으로 더 강한 가정을 사용한다. 그럼에도 불구하고, RSA 함수의 부분 영역 단방향성이 (전체 영역) 단방향성과 동치이므로, RSA-OAEP의 보안성은 단순한 RSA 가정만으로도 증명될 수 있음을 알 수 있다. 다만, 그 축소(reduction)는 타이트하지 않다.
- 2장에서는 증명에 필요한 보안 개념을 다룬다.
- 3장에서는 RSA-OAEP에 대한 설명을 다룬다.
- 4장과 5장에서는 RSA-OAEP가 IND-CCA2 보안을 만족함을 다룬다. 4장은 Crypto2001에 나온 논문에 대한 증명을, 5장은 Journal of Cryptology에 나온 논문에 대한 증명을 다룬다. 4장은 증명을 마무리 하기는 했으나, 빈틈이 매우 많으므로, 4장 대신 5장을 확인한다.

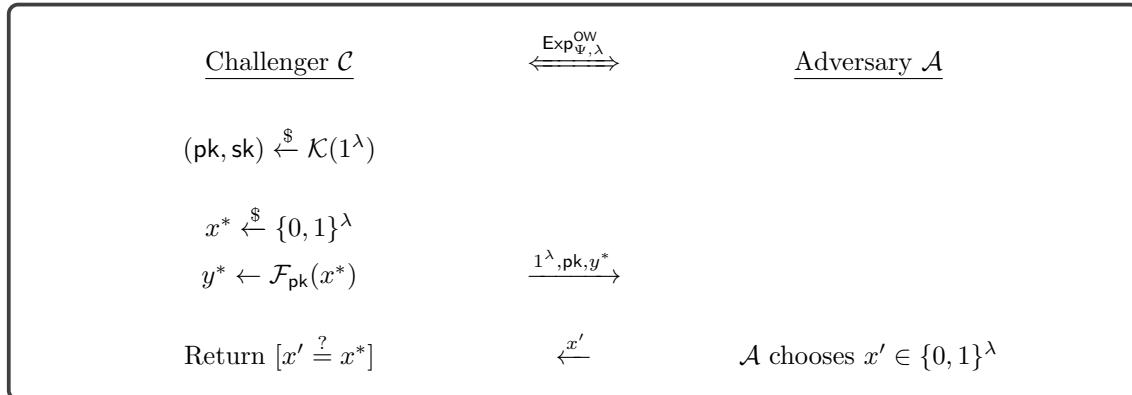
2 보안 개념

2.1 OW trapdoor permutation

트랩도어 치환 체계(Trapdoor permutation scheme) $\Psi = (\mathcal{K}, \mathcal{F}, \mathcal{I})$ 를 다음과 같이 정의한다.

- $\mathcal{K}(1^\lambda)$: 확률론적 키 생성 알고리즘으로, 1^λ 를 입력 받아 (pk, sk) 를 생성한다.
- $\mathcal{F}_{\text{pk}}(x)$: 결정론적 알고리즘으로, pk 와 $x \in \{0, 1\}^\lambda$ 를 입력 받아 $y \in \{0, 1\}^\lambda$ 를 출력한다.
- $\mathcal{I}_{\text{sk}}(y)$: 결정론적 알고리즘으로, sk 와 $y \in \{0, 1\}^\lambda$ 를 입력 받아 $x \in \{0, 1\}^\lambda$ 를 출력한다. $\mathcal{K}(1^\lambda)$ 로 생성한 모든 (pk, sk) 와 모든 $x \in \{0, 1\}^\lambda$ 에 대해, $\mathcal{I}_{\text{sk}}(\mathcal{F}_{\text{pk}}(x)) = x$ 를 만족한다.

동작시간(Running time) τ 를 가지는 공격자 \mathcal{A} 와 트랩도어 치환 체계 Ψ 에 대한 일방향성(One-wayness) 실험 $\text{Exp}_{\Psi, \lambda}^{\text{OW}}(\mathcal{A}; \tau)$ 을 다음과 같이 정의한다.



\mathcal{A} 의 능력치 $\text{Adv}_{\mathcal{A}; \Psi}^{\text{OW}}(\lambda, \tau)$ 을 다음과 같이 정의한다.

$$\text{Adv}_{\Psi, \lambda}^{\text{OW}}(\mathcal{A}; \tau) := \Pr[\text{Exp}_{\Psi, \lambda}^{\text{OW}}(\mathcal{A}; \tau) = 1].$$

2.2 Partial-domain OW trapdoor permutation

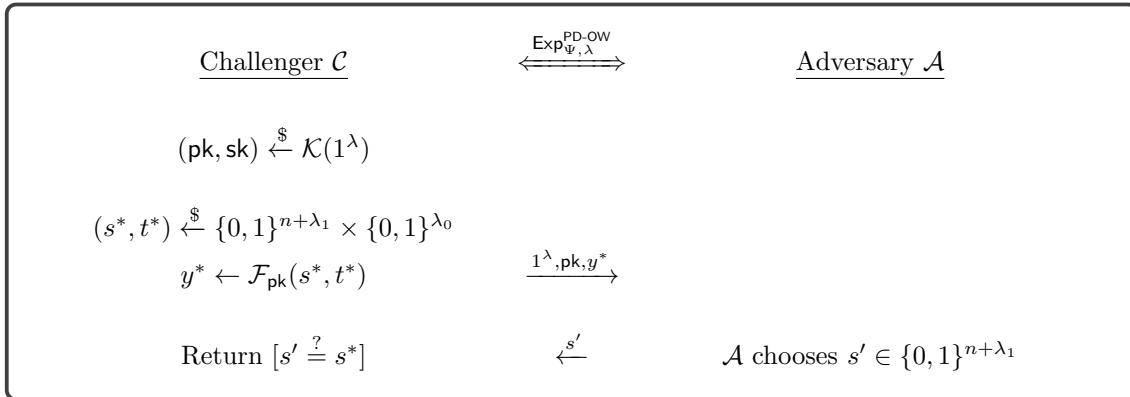
트랩도어 치환 $\Psi = (\mathcal{K}, \mathcal{F}, \mathcal{I})$ 에서, $\mathcal{F}_{\text{pk}}(x) : \{0, 1\}^\lambda \rightarrow \{0, 1\}^\lambda$ 를 다음과 같이 표현한다.

$$\mathcal{F}_{\text{pk}} : \{0, 1\}^{n+\lambda_1} \times \{0, 1\}^{\lambda_0} \rightarrow \{0, 1\}^{n+\lambda_1} \times \{0, 1\}^{\lambda_0}.$$

이때 $\lambda = n + \lambda_0 + \lambda_1$ 이다.

메모. 예를 들어, $x = s \parallel t$ 라고 할 때, $y \leftarrow \mathcal{F}_{\text{pk}}(x)$ 대신 $y \leftarrow \mathcal{F}_{\text{pk}}(s \parallel t)$ 로 표현할 수 있다.

동작시간 τ 를 가지는 공격자 \mathcal{A} 와 트랩도어 함수 체계 Ψ 에 대한 부분 일방향성(Partial-domain one-wayness) 실험 $\text{Exp}_{\Psi, \lambda}^{\text{PD-OW}}(\mathcal{A}; \tau)$ 을 다음과 같이 정의한다.

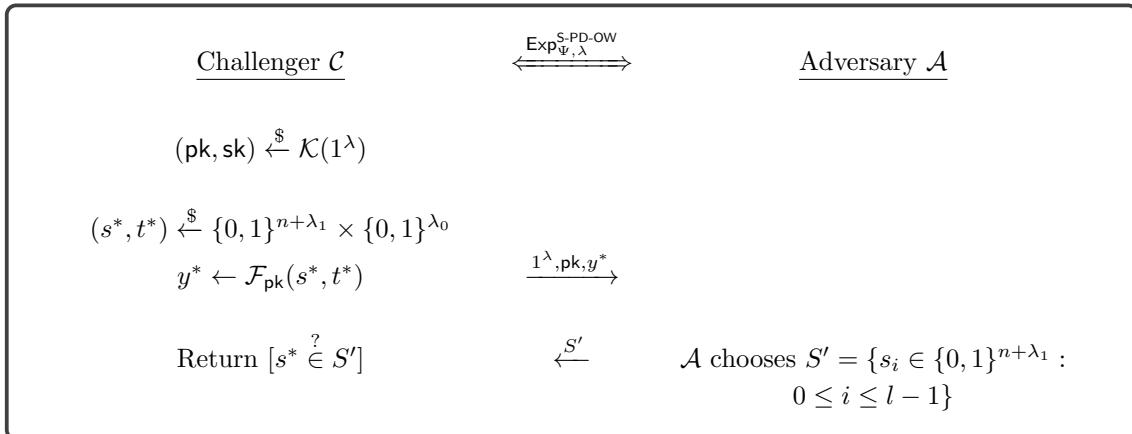


공격자 \mathcal{A} 의 능력치 $\text{Adv}_{\Psi, \lambda}^{\text{PD-OW}}(\mathcal{A}; \tau)$ 를 다음과 같이 정의한다.

$$\text{Adv}_{\Psi, \lambda}^{\text{PD-OW}}(\mathcal{A}; \tau) := \Pr[\text{Exp}_{\Psi, \lambda}^{\text{PD-OW}}(\mathcal{A}; \tau) = 1].$$

2.3 Set partial-domain OW trapdoor permutation

동작시간 τ 를 가지고 l 개의 원소를 출력하는 공격자 \mathcal{A} 와 트랩도어 함수 체계 Ψ 에 대한 집합 부분 일방향성(Set partial-domain one-wayness) 실험 $\text{Exp}_{\Psi, \lambda}^{\text{S-PD-OW}}(\mathcal{A}; \tau, l)$ 을 다음과 같이 정의한다.



공격자 \mathcal{A} 의 능력치 $\text{Adv}_{\Psi, \lambda}^{\text{S-PD-OW}}(\mathcal{A}; \tau, l)$ 를 다음과 같이 정의한다.

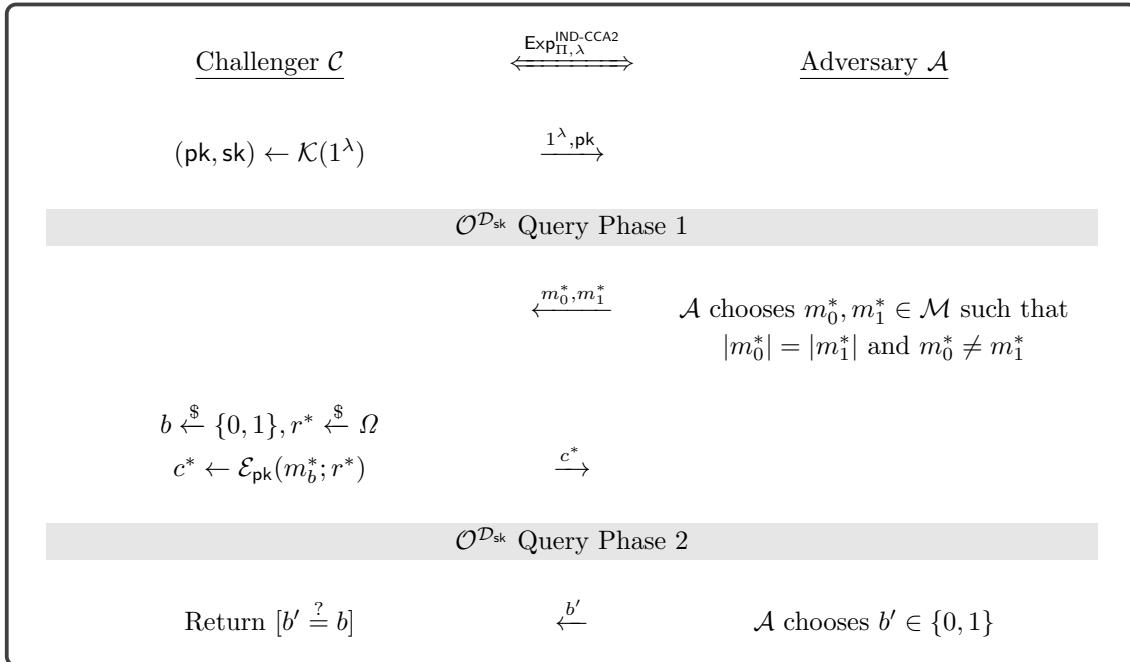
$$\text{Adv}_{\Psi, \lambda}^{\text{S-PD-OW}}(\mathcal{A}; \tau, l) := \Pr[\text{Exp}_{\Psi, \lambda}^{\text{S-PD-OW}}(\mathcal{A}; \tau, l) = 1].$$

2.4 IND security against CCA2

공개키 암호 체계(Public-key encryption scheme) $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ 를 다음과 같이 정의한다.

- $\mathcal{K}(1^\lambda)$: 확률론적 키 생성 알고리즘으로, 1^λ 를 입력 받아 (pk, sk) 를 생성한다.
- $\mathcal{E}_{\text{pk}}(m)$: 암호화 알고리즘으로, pk 와 $m \in \mathcal{M}$ 를 입력 받아 $c \in \mathcal{C}$ 를 출력한다. 확률론적 알고리즘으로, $r \xleftarrow{\$} \Omega$ 를 추가로 입력 받아 $\mathcal{E}_{\text{pk}}(m; r)$ 으로 표현할 수도 있다.
- $\mathcal{D}_{\text{sk}}(c)$: 결정론적 복호화 알고리즘으로, sk 와 $c \in \mathcal{C}$ 를 입력 받아 $m \in \mathcal{M}$ 를 출력한다.

동작시간 τ 를 가지고 복호화 오라클에 q 회 질의하는 공격자 \mathcal{A} 와 공개키 암호 체계 Π 에 대해, 선택 암호문 공격(Adaptive chosen ciphertext attack, 이하 CCA2)에 대한 구별불가능성(Indistinguishability) 실험 $\text{Exp}_{\Pi, \lambda}^{\text{IND-CCA2}}(\mathcal{A}; \tau, q)$ 을 다음과 같이 정의한다.

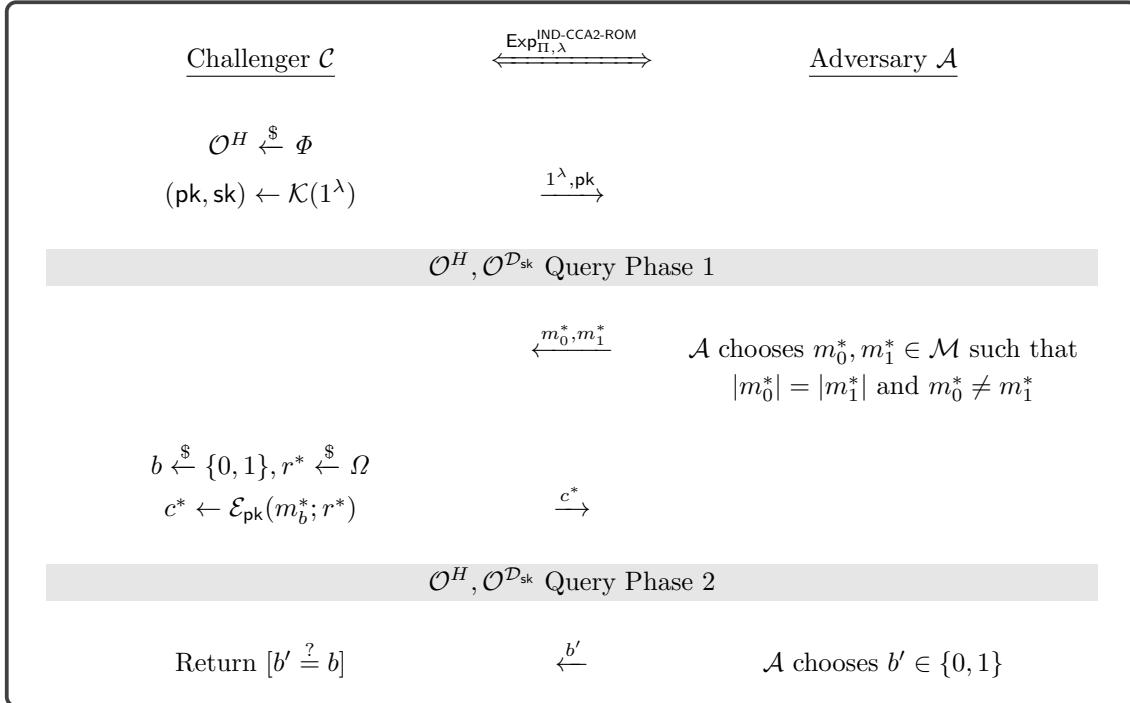


공격자 \mathcal{A} 의 능력치 $\text{Adv}_{\Pi, \lambda}^{\text{IND-CCA2}}(\mathcal{A}; \tau, q)$ 를 다음과 같이 정의한다.

$$\text{Adv}_{\Pi, \lambda}^{\text{IND-CCA2}}(\mathcal{A}; \tau, q) = 2 \cdot \Pr[\text{Exp}_{\Pi, \lambda}^{\text{IND-CCA2}}(\mathcal{A}; \tau, q) = 1] - 1.$$

2.5 IND security against CCA2 in ROM

동작시간 τ 를 가지고 복호화 오라클에 q_D 회, 랜덤 오라클에 q_H 회 질의하는 공격자 \mathcal{A} 와 공개키 암호체계 Π 에 대해, 랜덤 오라클 모델(Random oracle model)에서의 CCA2에 대한 구별불가능성 실험 $\text{Exp}_{\Pi, \lambda}^{\text{IND-CCA2-ROM}}(\mathcal{A}; \tau, q_D, q_H)$ 을 다음과 같이 정의한다.



공격자 \mathcal{A} 의 능력치 $\text{Adv}_{\Pi,\lambda}^{\text{IND-CCA2-ROM}}(\mathcal{A}; \tau, q_D, q_H)$ 를 다음과 같이 정의한다.

$$\text{Adv}_{\Pi,\lambda}^{\text{IND-CCA2-ROM}}(\mathcal{A}; \tau, q_D, q_H) = 2 \cdot \Pr[\text{Exp}_{\Pi,\lambda}^{\text{IND-CCA2-ROM}}(\mathcal{A}; \tau, q_D, q_H) = 1] - 1.$$

3 RSA-OAEP

다음과 같은 트랩도어 치환 \mathcal{F} 를 고려한다.

$$\mathcal{F}_{\text{pk}} : \{0, 1\}^{n+\lambda_1} \times \{0, 1\}^{\lambda_0} \rightarrow \{0, 1\}^{n+\lambda_1} \times \{0, 1\}^{\lambda_0}.$$

그리고 두 해시 함수 H, G 를 다음과 같이 준비한다.

$$H : \{0, 1\}^{\lambda_0} \rightarrow \{0, 1\}^{\lambda-\lambda_0}, \quad G : \{0, 1\}^{\lambda-\lambda_0} \rightarrow \{0, 1\}^{\lambda_0}.$$

트랩도어 치환 체계 $\Psi = (\mathcal{K}, \mathcal{F}, \mathcal{I})$ 를 포함하는 OAEP 변환 $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ 는 다음과 같이 동작한다.

- $\mathcal{K}(1^\lambda)$: (pk, sk)를 생성한다. pk 는 이후 트랩도어 치환 \mathcal{F} 에서 사용하며, sk 는 \mathcal{I} 에서 사용한다.
- $\mathcal{E}_{\text{pk}}(m; r)$: $m \in \{0, 1\}^n$ 과 $r \xleftarrow{\$} \{0, 1\}^{\lambda_0}$ 가 주어졌을 때, s, t 를 다음과 같이 계산한다.

$$s = (m \parallel 0^{\lambda_1}) \oplus G(r), \quad t = r \oplus H(s).$$

s, t 를 계산하는 과정을 도식화하면 그림 1와 같다. 이후 암호문 $c = \mathcal{F}_{\text{pk}}(s, t)$ 을 출력한다.

- $\mathcal{D}_{\text{sk}}(c)$: $(s, t) = \mathcal{I}_{\text{sk}}(c)$ 을 계산한 후, r, M 을 다음과 같이 계산한다.

$$r = t \oplus H(s), \quad M = s \oplus G(r).$$

만약 $[M]_{\lambda_1} = 0^{\lambda_1}$ 이면 $[M]^n$ 을 출력하고, 아니라면 “Reject”를 출력한다. 이 때, $[M]_{\lambda_1}$ 은 M 의 마지막 λ_1 비트 LSB를 의미하고, $[M]^n$ 은 M 의 첫 n 비트 MSB를 의미한다.

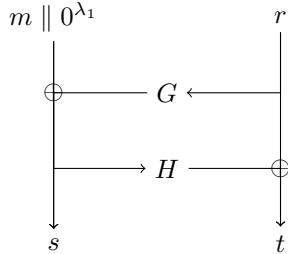


Figure 1: $\mathcal{E}_{\text{pk}}(m; r)$ 에서 s, t 를 계산하는 과정

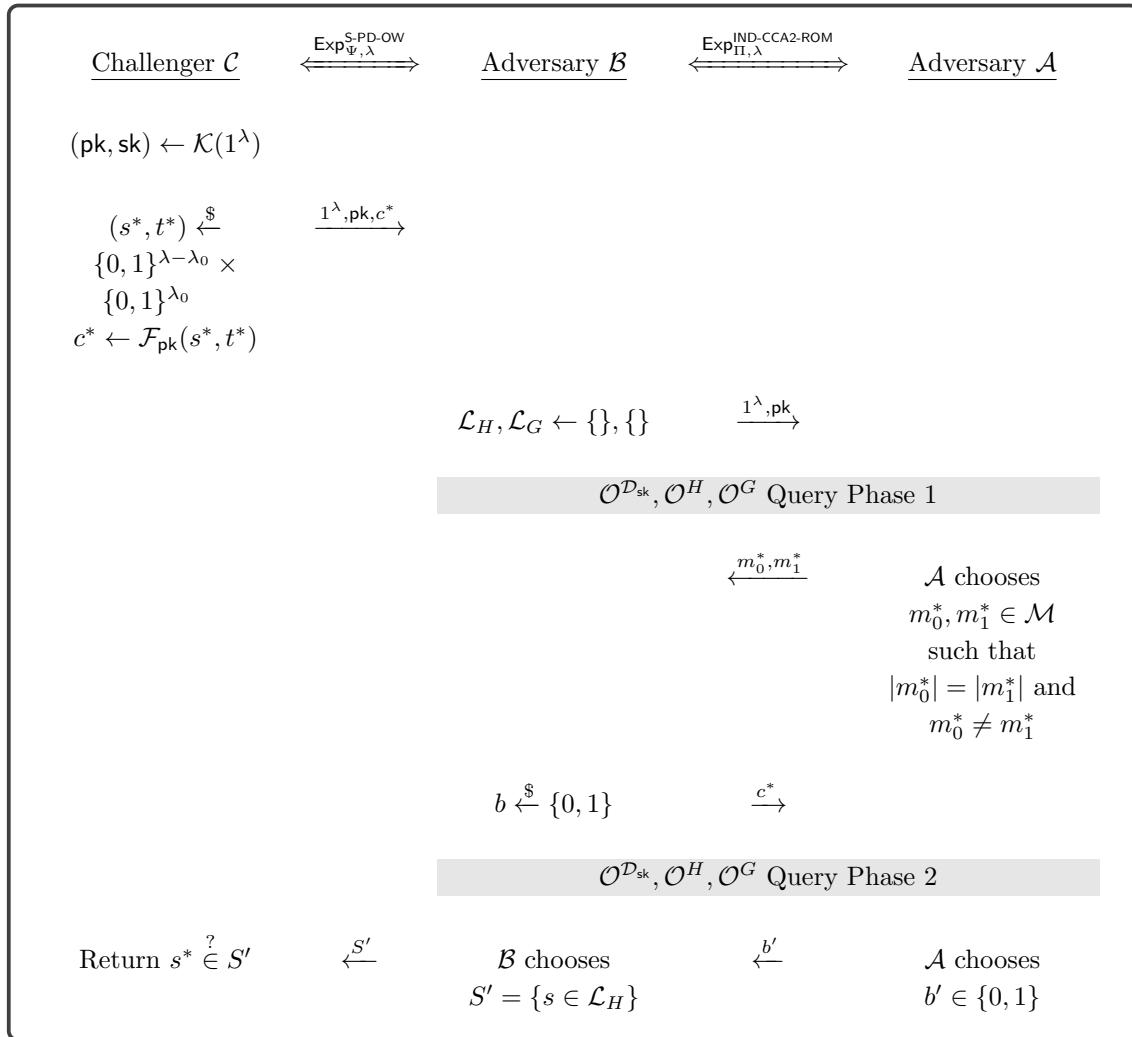
4 증명

보조정리 1. 공격자 \mathcal{A} 를 OAEP 변환 $(\mathcal{K}, \mathcal{E}, \mathcal{D})$ 에 대해 동작시간 τ 를 가지고, 복호화 오라클 \mathcal{O}^D 와 랜덤 오라클 $\mathcal{O}^H, \mathcal{O}^G$ 에 각각 q_D, q_H, q_G 회 질의하는 IND-CCA2 공격자라 하자. 이때, 다음을 만족하는 S-PD-OW 공격자 \mathcal{B} 가 존재한다.

$$\text{Adv}_{\Psi, \lambda}^{\text{S-PD-OW}}(\mathcal{B}; \tau', q_H) \geq \frac{\text{Adv}_{\Pi, \lambda}^{\text{IND-CCA2}}(\mathcal{A}; \tau, q_D, q_H, q_G)}{2} - \frac{2q_D q_G + q_D + q_G}{2^{\lambda_0}} - \frac{2q_D}{2^{\lambda_1}}.$$

여기서, $\tau' \leq \tau \cdot q_H \cdot q_G \cdot (T_{\mathcal{F}} + O(1))$ 이고, $T_{\mathcal{F}}$ 는 트랩도어 치환 \mathcal{F} 의 시간 복잡도를 의미한다.

4.1 증명: Reduction and simulation



먼저, 공격자 \mathcal{B} 가 \mathcal{O}^H 를 동작시키는 시뮬레이션을 정의한다. 공격자 \mathcal{A} 가 랜덤 오라클 \mathcal{O}^H 에 δ 를 질의했다고 하자. 공격자 \mathcal{B} 는 다음과 같이 H_δ 를 응답한다.

- 만약 δ 가 \mathcal{L}_H 에 있다면, δ 에 대응하는 H_δ 를 응답한다. (즉, $(\delta, H_\delta) \in \mathcal{L}_H$)
- 만약 δ 가 \mathcal{L}_H 에 없다면, $H_\delta \leftarrow \{0, 1\}^{\lambda_0}$ 을 수행한 후 H_δ 를 응답한다. 이후 $\mathcal{L}_H \leftarrow \mathcal{L}_H \cap (\delta, H_\delta)$ 를 수행한다.

다음으로, 공격자 \mathcal{B} 가 \mathcal{O}^G 를 동작시키는 시뮬레이션을 정의한다. 공격자 \mathcal{A} 가 랜덤 오라클 \mathcal{O}^G 에 γ 를 질의했다고 하자. 공격자 \mathcal{B} 는 다음과 같이 G_γ 를 응답한다.

1. 만약 γ 가 \mathcal{L}_G 에 있다면, γ 에 대응하는 G_γ 를 응답한다.
2. 만약 γ 가 \mathcal{L}_G 에 없다면, 다음 과정을 진행한다.
 - (a) 어떤 $(\delta, H_\delta) \in \mathcal{L}_H$ 에 대해, 만약 $c^* = \mathcal{F}_{\text{pk}}(\delta, \gamma \oplus H_\delta)$ 라면, 우리는 여전히 G 를 올바르게 시뮬레이션할 수 있다. 이 때 응답은 $G_\gamma \leftarrow \delta \oplus (m_b \parallel 0^{\lambda_1})$ 이다. $\delta = s^*$ 이고 s^* 가 균등하게 분포하므로 G_γ 는 균등 분포된 값이 된다.
 - (b) 모든 $(\delta, H_\delta) \in \mathcal{L}_H$ 에 대해, 만약 $c^* \neq \mathcal{F}_{\text{pk}}(\delta, \gamma \oplus H_\delta)$ 라면, $G_\gamma \leftarrow \{0, 1\}^{n+\lambda_1}$ 를 수행한다.
 - (c) G_γ 를 응답한 후, $\mathcal{L}_G \leftarrow \mathcal{L}_G \cap (\gamma, G_\gamma)$ 를 수행한다.

마지막으로, 공격자 \mathcal{B} 가 \mathcal{O}^D 를 동작시키는 시뮬레이션을 정의한다. 공격자 \mathcal{A} 가 랜덤 오라클 \mathcal{O}^D 에 $c = \mathcal{F}_{\text{pk}}(s, t)$ 를 질의했다고 하자. 공격자 \mathcal{B} 는 다음과 같이 응답한다.

1. \mathcal{L}_G 의 질의 응답 쌍 $(\gamma, G_\gamma) \in \mathcal{L}_G$ 및 \mathcal{L}_H 의 $(\delta, H_\delta) \in \mathcal{L}_H$ 를 조회하고, 각 리스트에서 선택된 쌍에 대해 다음과 같이 정의한다.

$$\sigma = \delta, \quad \tau = \gamma \oplus H_\delta, \quad \mu = G_\gamma \oplus \delta.$$

만약 $c = \mathcal{F}_{\text{pk}}(\sigma, \tau)$ 이면서 $[\mu]_{\lambda_1} = 0^{\lambda_1}$ 라면, $[\mu]^n$ 을 응답한다.

2. 그 외에는 Reject를 응답한다.

4.2 증명: 사건 정의

Table 1: 오라클 관련 사건 정의

AskG	r^* 가 \mathcal{O}^G 에 질의되었을(has been asked) 사건.
AskH	s^* 가 \mathcal{O}^H 에 질의되었을 사건.
GBad	\mathcal{O}^G 에 r^* 를 질의했지만, \mathcal{O}^G 의 응답이 $s^* \oplus (m_b \parallel 0^{sk})$ 가 아닌 사건. GBad가 발생하면, AskG도 발생한다.
DBad	CPA-시나리오에서 복호화가 실패하는 사건.
Bad	$\text{GBad} \vee \text{DBad}$.

공격자 \mathcal{A} 는 복호화 오라클 \mathcal{O}^D 에 암호문 $c = \mathcal{F}_{\text{pk}}(s, t)$ 를 질의할 수 있다. 질의한 암호문 c 와 관련된 사건을 다음 표와 같이 정의한다.

4.3 증명: Analysis of the Decryption Oracle Simulation

보조정리 2. s^* 가 \mathcal{O}^H 에 질의되지 않았을 때, \mathcal{O}^D 는 질의된 암호문 c ($c \neq c^*$)에 대해 출력을 정확히 생성할 수 있으며, 이 확률은 다음보다 크거나 같다.

$$1 - \left(\frac{2}{2^{k_1}} + \frac{2q_G + 1}{2^{k_0}} \right).$$

또한, 시간 제한 $t' \leq q_G \cdot q_H \cdot (T_F + O(1))$ 내에서 이를 수행할 수 있다.

Table 2: 복호화 시뮬레이션 관련 사건 정의

SBad	$s = s^*$ 인 사건.
RBad	$r = r^*$ 인 사건. 즉, $H(s) \oplus t = H(s^*) \oplus t^*$ 인 사건.
CBad	$\text{SBad} \vee \text{RBad}$.
AskR	$r \in \mathcal{O}^G$ 에 질의되었을 사건. 즉, $H(s) \oplus t$ 이 질의되었을 사건
AskS	s 가 \mathcal{O}^H 에 질의되었을 사건.
AskRS	$\text{AskR} \wedge \text{AskS}$
Fail	복호화 오라클이 질의 c 에 대해 잘못 응답하는 사건. i 번째 질의 c_i 에 대해서는 Fail_i 로 나타낸다. 여기서 $i = 1, \dots, q_D$ 이다. 어떤 i 에 대해서도 Fail_i 의 확률을 균등하게 평가(evaluate)할 수 있으므로, 여기서는 사용하지 않는다. Fail 사건은 평문 추출기(plaintext extractor)가 실제 복호화 오라클에서는 허용될 암호문을 거부하는 경우로 제한된다. 실제로, 추출기가 암호문을 허용하는 순간, 해당 암호문이 유효하며 출력 평문과 일치함을 알 수 있다.

Proof. 본 증명에서는 다음이 참임을 보인다.

$$\Pr[\text{Fail} \mid \neg \text{AskH}] \leq \frac{2}{2^{\lambda_1}} + \frac{2q_G + 1}{2^{\lambda_0}}.$$

$\Pr[\text{Fail} \mid \neg \text{AskH}]$ 는 다음과 같이 표현 가능하다.

$$\Pr[\text{Fail} \wedge \text{CBad} \mid \neg \text{AskH}] + \Pr[\text{Fail} \wedge \neg \text{CBad} \mid \neg \text{AskH}].$$

본 증명에서는 먼저 $\Pr[\text{Fail} \wedge \text{CBad} \mid \neg \text{AskH}]$ 를 구한다. $\text{CBad} = \text{SBad} \vee (\text{RBad} \wedge \neg \text{SBad})$ 를 이용하여, 이 확률을 다음과 같이 표현한다.

$$\begin{aligned} & \Pr[\text{Fail} \wedge \text{CBad} \mid \neg \text{AskH}] \\ &= \Pr[\text{Fail} \wedge (\text{SBad} \vee (\text{RBad} \wedge \neg \text{SBad})) \mid \neg \text{AskH}] \\ &= \Pr[(\text{Fail} \wedge \text{SBad}) \vee (\text{Fail} \wedge \text{RBad} \wedge \neg \text{SBad}) \mid \neg \text{AskH}] \\ &\leq \Pr[\text{Fail} \wedge \text{SBad} \mid \neg \text{AskH}] + \Pr[\text{Fail} \wedge \text{RBad} \wedge \neg \text{SBad} \mid \neg \text{AskH}] \\ &\leq \Pr[\text{Fail} \wedge \text{SBad} \mid \neg \text{AskH}] + \Pr[\text{RBad} \wedge \neg \text{SBad} \mid \neg \text{AskH}] \\ &\leq \Pr[\text{Fail} \mid \text{SBad} \wedge \neg \text{AskH}] + \Pr[\text{RBad} \mid \neg \text{SBad} \wedge \neg \text{AskH}]. \\ &= \Pr[\text{Fail} \wedge \text{AskR} \mid \text{SBad} \wedge \neg \text{AskH}] + \Pr[\text{Fail} \wedge \neg \text{AskR} \mid \text{SBad} \wedge \neg \text{AskH}] + \Pr[\text{RBad} \mid \neg \text{SBad} \wedge \neg \text{AskH}]. \\ &\leq \Pr[\text{AskR} \mid \text{SBad} \wedge \neg \text{AskH}] + \Pr[\text{Fail} \mid \neg \text{AskR} \wedge \text{SBad} \wedge \neg \text{AskH}] + \Pr[\text{RBad} \mid \neg \text{SBad} \wedge \neg \text{AskH}]. \end{aligned}$$

세 번째 사건은 $s \neq s^*$ 이고 공격자 \mathcal{A} 가 s^* 에 대해 \mathcal{O}^H 에 질의하지 않았을 때 RBad가 발생함을 의미한다. s^* 가 \mathcal{O}^H 에 질의되지 않았고 $s \neq s^*$ 일 때, $H(s^*)$ 는 예측 불가능(unpredictable)하며 $H(s)$ 뿐 아니라 t , t^* 와도 독립적이다. 이때 RBad 사건, $H(s^*) = H(s) \oplus t \oplus t^*$ 는 최대 $2^{-\lambda_0}$ 의 확률로 발생한다. 즉, 다음과 같다.

$$\Pr[\text{RBad} \mid \neg \text{SBad} \wedge \neg \text{AskH}] \leq 2^{-\lambda_0}.$$

첫 번째 사건은 $s = s^*$ 이며 $H(s^*)$ 는 예측 불가능할 때, $r \in \mathcal{O}^G$ 에 대해 질의되었을 사건을 의미한다. 이때, $H(s)$ 또한 예측 불가능하다. 즉, $r = H(s) \oplus t$ 가 예측 불가능하므로, $r \in \mathcal{O}^G$ 에 질의되었을 확률은 최대 $q_G \cdot 2^{-\lambda_0}$ 이다. 즉, 다음과 같다.

$$\Pr[\text{AskR} \mid \text{SBad} \wedge \neg \text{AskH}] \leq q_G \cdot 2^{-\lambda_0}.$$

두 번째 사건은 복호화 시뮬레이션에서 $H(s)$ 는 예측 불가능하고 r 은 \mathcal{O}^G 에 질의되지 않았을 때, 유효한 암호문 c 를 거부하는 경우이다. 페이스텔 네트워크(Feistel network)의 일대일 성질에 따라 $s = s^*$ 이면 $r \neq r^*$ 이고, 따라서 $G(r)$ 는 예측 불가능하다. 그러므로 이 경우 중복 조건은 $2^{-\lambda_1}$ 보다 큰 확률로 성립할 수 없다. 즉, 다음과 같다.

$$\Pr[\text{Fail} \mid \neg\text{AskR} \wedge \text{SBad} \wedge \neg\text{AskH}] \leq 2^{-\lambda_1}.$$

세 식을 결합하면, 다음과 같다.

$$\Pr[\text{Fail} \wedge \text{CBad} \mid \neg\text{AskH}] \leq 2^{-k_1} + (q_G + 1) \cdot 2^{-k_0}.$$

다음으로, $\Pr[\text{Fail} \wedge \neg\text{CBad} \mid \neg\text{AskH}]$ 를 계산하고 본 증명을 마친다. 만약 $\neg\text{CBad} \wedge \text{AskRS}$ 가 성립한다면, 복호화 시뮬레이션은 실패하지 않는다. 따라서 이 식은 아래와 같이 표현 가능하다.

$$\begin{aligned} & \Pr[\text{Fail} \wedge \neg\text{CBad} \mid \neg\text{AskH}] \\ &= \underbrace{\Pr[\text{Fail} \wedge \neg\text{CBad} \wedge \text{AskRS} \mid \neg\text{AskH}]}_{=0} + \Pr[\text{Fail} \wedge \neg\text{CBad} \wedge \neg\text{AskRS} \mid \neg\text{AskH}] \\ &= \Pr[\text{Fail} \wedge \neg\text{CBad} \wedge \neg\text{AskRS} \mid \neg\text{AskH}]. \end{aligned}$$

이제 $\neg\text{AskH}$ 를 잠시 고려하지 않고, 위 확률을 다음과 같이 계산한다.

$$\begin{aligned} & \Pr[\text{Fail} \wedge \neg\text{CBad} \wedge \neg\text{AskRS}] \\ &= \Pr[\text{Fail} \wedge \neg\text{RBad} \wedge \neg\text{SBad} \wedge (\neg\text{AskR} \vee \neg\text{AskS})] \\ &= \Pr[\text{Fail} \wedge \neg\text{RBad} \wedge \neg\text{SBad} \wedge (\neg\text{AskR} \vee (\neg\text{AskS} \wedge \text{AskR}))] \\ &= \Pr[(\text{Fail} \wedge \neg\text{RBad} \wedge \neg\text{SBad} \wedge \neg\text{AskR}) \vee (\text{Fail} \wedge \neg\text{RBad} \wedge \neg\text{SBad} \wedge (\neg\text{AskS} \wedge \text{AskR}))] \\ &\leq \Pr[\text{Fail} \wedge \neg\text{RBad} \wedge \neg\text{SBad} \wedge \neg\text{AskR}] + \Pr[\text{Fail} \wedge \neg\text{RBad} \wedge \neg\text{SBad} \wedge \neg\text{AskS} \wedge \text{AskR}] \\ &\leq \Pr[\text{Fail} \wedge \neg\text{RBad} \wedge \neg\text{AskR}] + \Pr[\text{Fail} \wedge \text{AskR} \wedge \neg\text{AskS} \wedge \neg\text{SBad}] \\ &\leq \Pr[\text{Fail} \wedge \neg\text{RBad} \mid \neg\text{AskR}] + \Pr[\text{Fail} \wedge \text{AskR} \mid \neg\text{AskS} \wedge \neg\text{SBad}] \\ &\leq \Pr[\text{Fail} \mid \neg\text{RBad} \wedge \neg\text{AskR}] + \Pr[\text{AskR} \mid \neg\text{AskS} \wedge \neg\text{SBad}]. \end{aligned}$$

첫 번째 사건에서, r 이 \mathcal{O}^G 에 대해 질의되지 않았고, 추가로 $r \neq r^*$ 인 사건을 고려하면, $G(r)$ 는 예측할 수 없으며, 따라서 $[s \oplus G(r)]_{\lambda_1} = 0^{\lambda_1}$ 이 될 확률은 $2^{-\lambda_1}$ 보다 작다. 그리고 두 번째 사건에서, $H(s)$ 에 대한 정보 없이 r 이 \mathcal{O}^G 에 대해 질의될 확률은 $q_G \cdot 2^{-\lambda_0}$ 보다 작다. 또한, 이 사건은 AskH 와 독립적이므로 다음이 성립한다.

$$\Pr[\text{Fail} \wedge \neg\text{CBad} \wedge \neg\text{AskRS} \mid \neg\text{AskH}] \leq 2^{-\lambda_1} + q_G \cdot 2^{-\lambda_0}.$$

그러므로, 다음과 같다.

$$\begin{aligned} \Pr[\text{Fail} \mid \neg\text{AskH}] &= \Pr[\text{Fail} \wedge \text{CBad} \mid \neg\text{AskH}] + \Pr[\text{Fail} \wedge \neg\text{CBad} \mid \neg\text{AskH}] \\ &\leq (2^{-\lambda_1} + (q_G + 1) \cdot 2^{-\lambda_0}) + (2^{-\lambda_1} + q_G \cdot 2^{-\lambda_0}). \\ &= \frac{2}{2^{\lambda_1}} + \frac{2q_G + 1}{2^{\lambda_0}}. \end{aligned}$$

이 시뮬레이터의 실행 시간은 가능한 모든 쌍에 대해 $\mathcal{F}_{\text{pk}}(\sigma, \tau)$ 를 계산하는 시간만 포함되며, 따라서 그 시간은 다음과 같이 상한된다.

$$q_G \cdot q_H \cdot (T_{\mathcal{F}} + \mathcal{O}(1)).$$

□

5 Journal of Cryptology

본 절에서는 Journal of Cryptology의 RSA-OAEP is Secure under the RSA Assumption 논문 내용을 정리했다. 기존 논문과 달리 여기서는 여러 개의 GAME을 정의해, 각 GAME의 성공 확률 차이를 활용하여 공격자의 능력치 관계를 표현한다.

5.1 랜덤 오라클 시뮬레이터

랜덤 오라클 시뮬레이터 H 를 다음과 같이 정의한다.

H 에 대해 새로운 질의 δ 가 들어오면, 시뮬레이터는 무작위 값을 H_δ 로 출력하고, 쌍 (δ, H_δ) 를 리스트 \mathcal{L}_H 에 추가한다.

랜덤 오라클 시뮬레이터 G 를 다음과 같이 정의한다.

- G 에 대해 새로운 질의 γ 가 들어오면, 시뮬레이터는 H 의 리스트 \mathcal{L}_H 를 살펴보고, H 에 대해 어떤 δ 가 질의되었는지를 확인한다.
- 각 H_δ 에 대해 $z = \gamma \oplus H_\delta$ 를 계산하고, $c^* = \mathcal{F}_{pk}(\delta, z)$ 가 성립하는지 확인한다. 만약 어떤 δ 에 대해 이 관계가 성립한다면, 시뮬레이터는 G 의 응답을 다음과 같이 줄 수 있다.

$$G_\gamma = \delta \oplus (m_b \parallel 0^{\lambda_1}).$$

이때 $\delta = s^*$ 이고, s^* 는 균등하게 분포된 값이므로, G_γ 도 균등 분포된(random) 값이다.

- 그렇지 않은 경우에는 G_γ 를 랜덤한 값으로 정의한다. 두 경우 모두 (γ, G_γ) 를 G 의 리스트 \mathcal{L}_G 에 추가한다.

메모. 만약 도전 암호문 c^* 가 결정되기 전에 G 에 질의한다면, G_γ 를 무작위 값으로 설정하여 응답 한다.

5.2 평문 추출기

메모. 평문 추출기(*plaintext extractor*)를 이해하기 위해서는 *Plaintext Awareness* 개념을 알아야 하는 듯 하다. 여기서는 간단하게 이해하자면, 평문 추출기는 시뮬레이터가 그 오라클을 흉내 내기 위해 만든 도구이다. 실제 복호를 하는 것이 아니고, 복호 결과를 추출하는 의미로 평문 추출기로 부른다.

평문추출기 \mathcal{PE} 를 다음과 같이 정의한다.

평문 추출기 \mathcal{PE} 의 입력은 다음과 같다.

- 무작위 오라클 G, H 에 대한 질의 응답 쌍을 모아 놓은 두 개의 리스트 $\mathcal{L}_G, \mathcal{L}_H$.
- 유효한 암호문 c^* .
- 후보 암호문 c . 이때, $c \neq c^*$ 이다.

추출기 \mathcal{PE} 의 동작 방식은 다음과 같다.

- 암호문 $c = \mathcal{F}_{\text{pk}}(s \parallel t)$ 가 주어지면, \mathcal{L}_G 에 있는 모든 (γ, G_γ) 와 \mathcal{L}_H 에 있는 모든 (δ, H_δ) 에 대해 다음을 계산한다.

$$\sigma = \delta, \quad \theta = \gamma \oplus H_\delta, \quad \mu = G_\gamma \oplus \delta.$$

- 그리고 다음 조건을 검사한다.

$$c = \mathcal{F}_{\text{pk}}(\sigma \parallel \theta) \quad \text{and} \quad [\mu]_{\lambda_1} = 0^{\lambda_1}.$$

- 조건이 만족되면, \mathcal{PE} 는 μ 의 앞부분, 즉 $[\mu]^n$ 을 평문으로 출력하고 종료한다. 조건을 만족하는 조합이 없다면, \mathcal{PE} 는 Reject 메시지를 반환한다.

리스트의 순서에 관계없이, \mathcal{PE} 의 출력은 항상 유일하게 정의된다는 것을 쉽게 확인할 수 있다. 함수 \mathcal{F} 가 순열이므로, $\sigma = s$ 는 유일하게 결정되고, 따라서 δ 도 유일하게 결정된다. 또한 \mathcal{L}_G 와 \mathcal{L}_H 는 각각 함수 G 와 H 에 대한 입력 출력 쌍들이며, 하나의 입력에 대해 대응되는 출력은 최대 하나이기 때문에, H_δ 역시 유일하게 결정된다. 마찬가지로 $\theta = t$ 도 유일하게 결정되며, 따라서 γ 와 G_γ 도 유일하게 결정된다. 결국 선택될 수 있는 μ 는 최대 하나이며, 그 출력 여부는 조건 $[\mu]_{\lambda_1} = 0^{\lambda_1}$ 을 만족하는지에 따라 결정된다.

메모. 원래 복호화 오라클은 출력이 항상 유일하게 정의되기 때문에, \mathcal{PE} 의 출력도 항상 유일하게 정의됨을 보여야 한다. 공격자가 \mathcal{PE} 에 같은 입력을 넣었을 때, 다른 출력을 얻으면 안된다. 순열이 아니여도 \mathcal{F} 가 일대일 함수라면 여전히 \mathcal{PE} 의 출력이 유일할 것 같다.

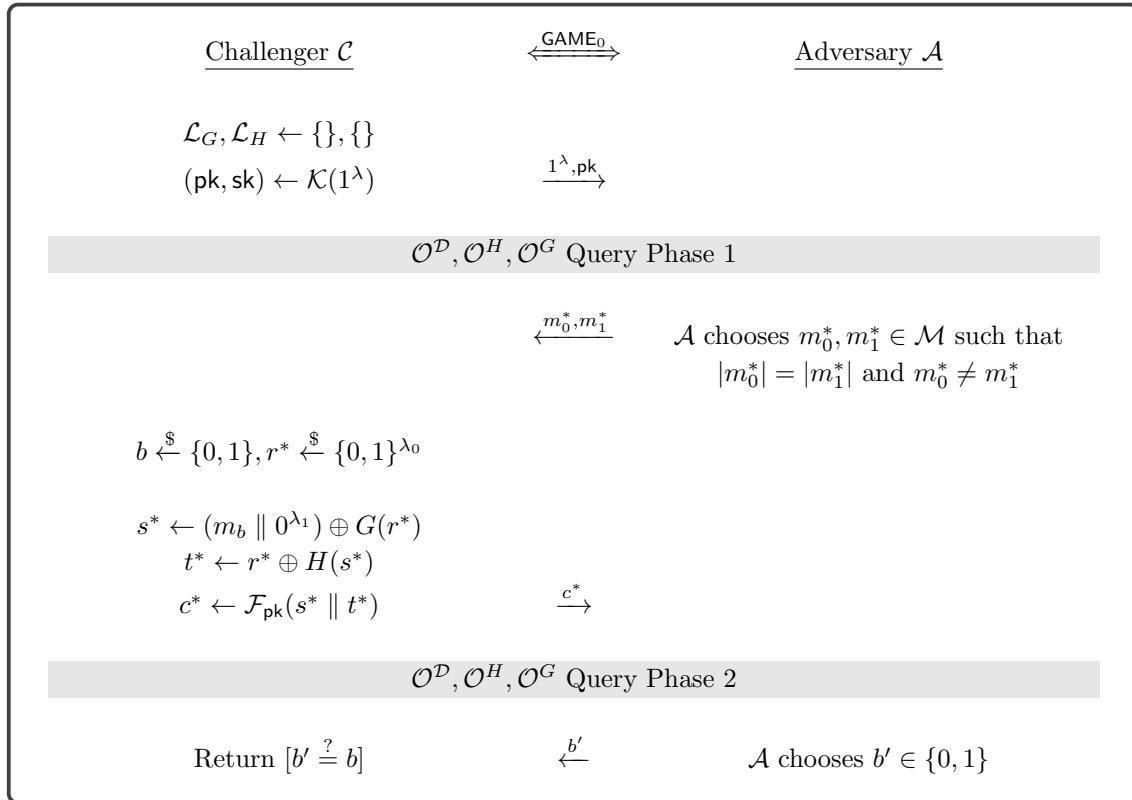
5.3 게임 구성

이후에서 c^* 는 암호화 오라클로부터 얻은 도전 암호문을 의미한다. 우리는 복호화 오라클 대신 평문 추출기를 사용하는 상황을 상정하고 있으며, 의미론적 보안을 모순시키려는 맥락에서, c^* 가 메시지 m_b 에 대한 암호문이라고 가정한다. 또한 c^* 의 난수 시드를 r^* 라고 표기한다. 이때 다음 관계가 성립한다.

$$r^* = H(s^*) \oplus t^* \quad \text{and} \quad G(r^*) = s^* \oplus (m_b \parallel 0^{\lambda_1})$$

이후의 모든 별표가 없는 변수들은 복호화 질의에 해당한다. 우리는 이제 이전 증명의 간단한 확장으로서, 복호화 오라클을 활용하는 완전한 증명을 제시한다. 이 증명에서는 앞서 정의한 평문 추출기가 실패할 수 있는 모든 경우를 순차적으로 배제해가며 논리를 전개한다.

5.3.1 0 번째 게임

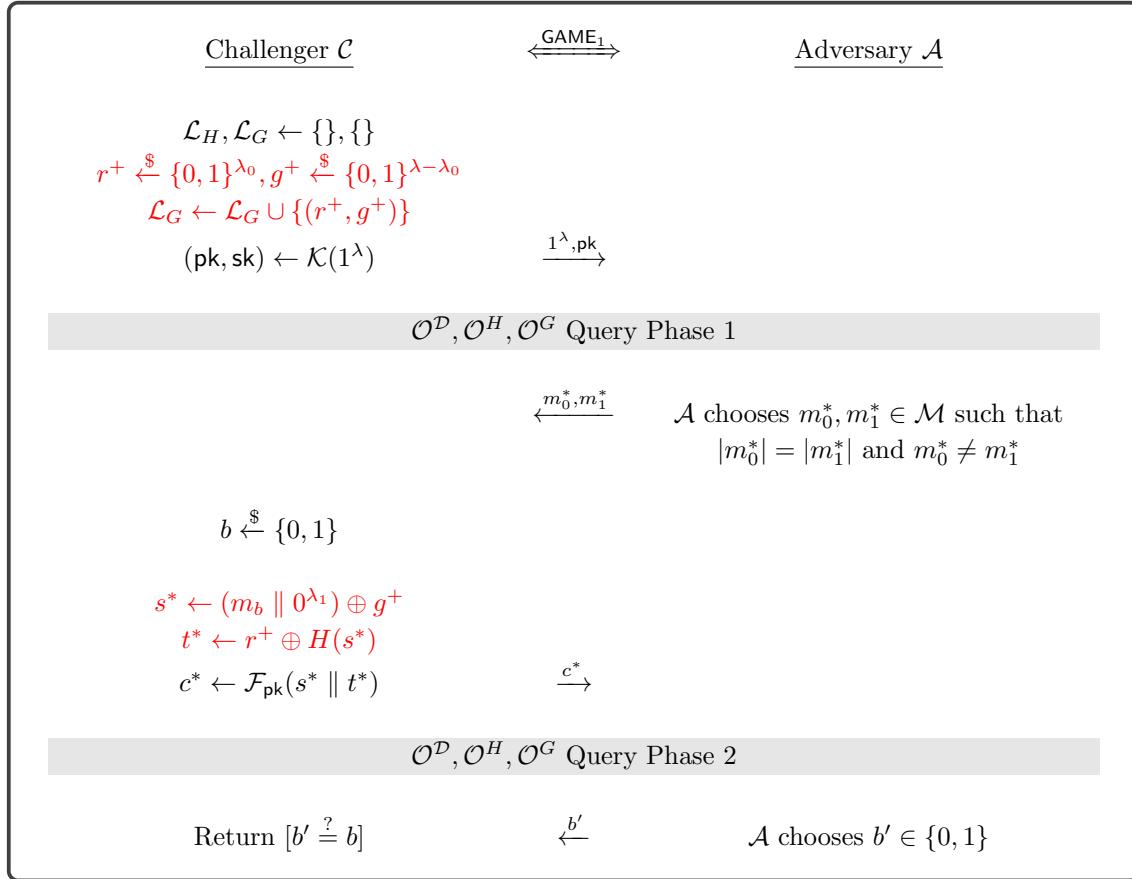


메모. GAME_0 는 IND-CCA2 실험과 동일하다. 여기서 사용하는 복호화 오라클 \mathcal{O}^D 는 평문 추출기가 아니라는 점에 주의한다.

GAME_0 에서 공격자는 도전 암호문을 복호화 오라클에 질의할 수 없다. 이벤트 S_0 는 GAME_0 가 1을 반환하는 사건을 의미하며, 이후 게임 단계에서도 유사하게 S_i 로 표기한다. 정의에 따라, 다음이 성립한다.

$$\text{Adv}_{\Pi,\lambda}^{\text{IND-CCA2}}(\mathcal{A}; \tau, q_D, q_H, q_G) = 2 \cdot \left| \Pr[S_0] - \frac{1}{2} \right|.$$

5.3.2 1 번째 게임



GAME₁에서는 난수 시드 r^* 의 값을 명시적으로 드러내고, 그 생성을 게임 초반으로 이동시키는 것이다. 즉, 사전에 무작위로 다음 값을 선택한다:

$$r^+ \xleftarrow{\$} \{0,1\}^{\lambda_0}, \quad g^+ \xleftarrow{\$} \{0,1\}^{\lambda-\lambda_0}.$$

그리고 이후부터는 r^* 대신 r^+ , $G(r^*)$ 대신 g^+ 를 사용한다.

메모. GAME₀에서 s^* 를 만들 때, r^* 를 G 에 통과시켜 $G(r^*)$ 를 구하고, $G(r^*)$ 를 이용하여 s^* 를 만든다. 그러나 GAME₀에서는 $G(r^*)$ 를 계산하지 않는다. 대신 g^+ 를 무작위로 생성하고, g^+ 를 이용하여 s^* 를 만든다.

GAME₁은 다음 두 규칙을 따른다.

- $r^* = r^+, s^* = (m_b \parallel 0^{\lambda_1}) \oplus g^+$ 이고, 이로부터 다음이 유도된다.

$$t^* = r^+ \oplus H(s^*), \quad c^* = f(s^* \parallel t^*).$$

- 무작위 오라클 G 에 대해 r^+ 로 질의가 들어오면, 응답은 항상 g^+ 이다.

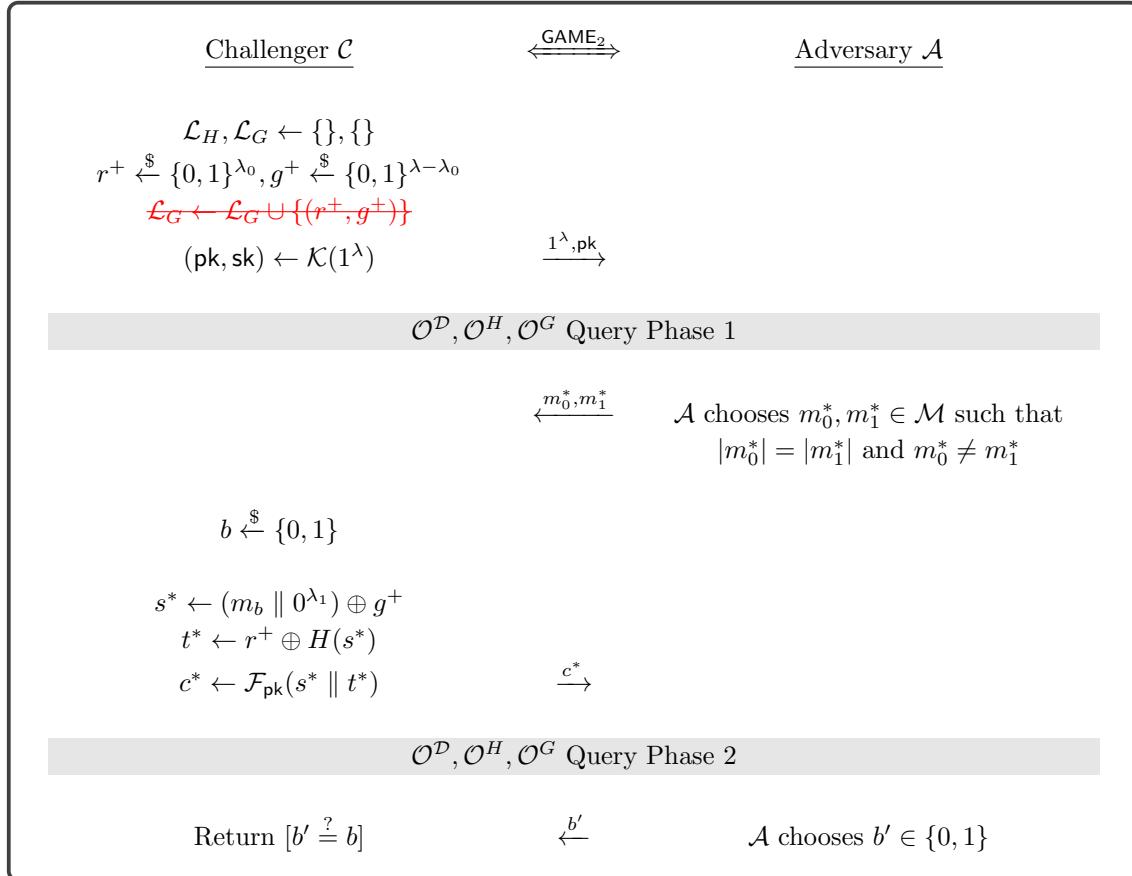
메모. 즉, GAME₀에서 GAME₁으로 바뀌면서 달라진 것은, $G(r^*)$ 대신 무작위 값을 사용하여 s^* 를 만든다는 점 뿐이라고 생각하면 된다.

우리는 $(r^*, G(r^*))$ 쌍을 정확히 동일한 분포를 가지는 (r^+, g^+) 로 대체한 것이므로 다음을 만족한다.

$$\Pr[S_1] = \Pr[S_0].$$

메모. $GAME_1$ 으로 바뀌면서 달라진 것은 $G(r^*)$ 대신 무작위 값 g^+ 을 사용한다는 점이다. g^+ 는 균등분포, $G(r^*)$ 도 무작위 오라클 정의에 의해 균등분포, 즉, 동일한 확률분포를 가진다. 그 외 모든 구조가 동일하므로, 공격자는 $GAME_0$ 와 $GAME_1$ 에서 동일한 전략을 사용한다. 따라서, $\Pr[S_1] = \Pr[S_0]$ 을 만족한다.

5.3.3 2 번째 게임



이 게임에서는 위에서 정의한 두 번째 규칙을 제거하고, 무작위 오라클 G 에 대한 질의를 원래대로 복원한다. 따라서 g^+ 는 s^* 를 구성할 때만 사용되고, 그 이후 계산에는 전혀 등장하지 않는다. 이로 인해, \mathcal{A} 의 입력은 비트 b 에 의존하지 않는 확률 분포를 따르게 된다. 따라서 다음이 성립한다.

$$\Pr[S_2] = \frac{1}{2}$$

메모. g^+ 는 무작위로 생성한 값으로, $s^* = (m_b \parallel 0^{\lambda_1}) \oplus g^+$ 를 계산할 때 외에는 전혀 등장하지 않는다. 원래는 r^+ 를 G 에 질의하면 g^+ 가 나오는 두 번째 룰에 의해 g^+ 가 등장했었기 때문에, 공격자가 G 에 r^+ 를 질의하면 정보를 얻을 수 있었으나, 이젠 얻을 수 없다. 따라서 공격자가 어떤 m_0, m_1 을 보내더라도 이와 무관하게 무작위 값 g^+ 가 사용된다. 그래서 \mathcal{A} 가 입력 c^* 를 받았을 때 이 게임의 성공확률은 정확히 1/2이다.

GAME₁과 GAME₂는 r^* 가 오라클 G 에 질의되는 경우에 한해 서로 다를 수 있다.

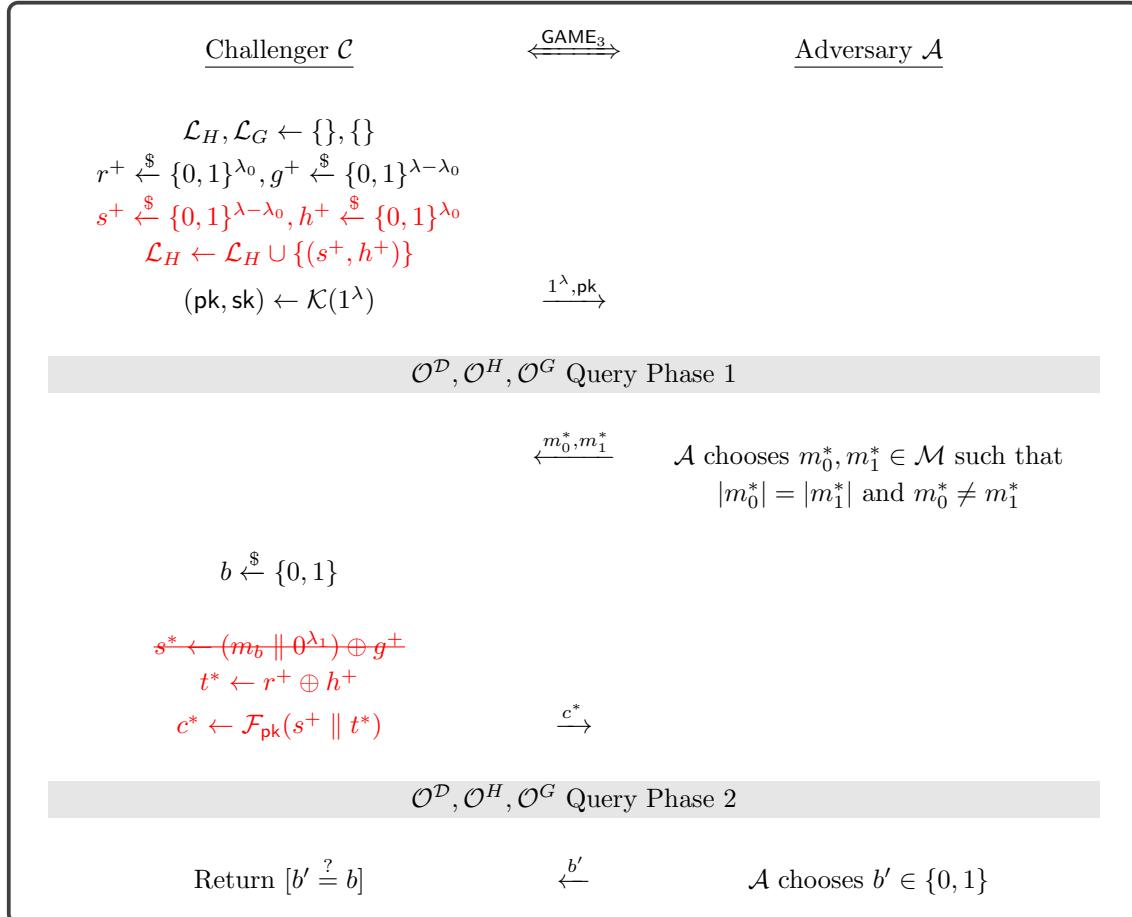
메모. GAME₁에서는 $G(r^*)$ 가 g^+ 로 설정되어 r^* 를 질의할 때 g^+ 를 응답하지만, GAME₂에서는 r^* 를 질의하면, 무작위 값을 응답한다. 응답한 무작위 값이 우연히 g^+ 일 수 있지만, 거의 다르다. r^* 가 아닌 다른 값을 질의하는 경우는 두 게임에서 오라클 G 는 동일하게 동작하지만, r^* 가 질의하면 다르게 동작한다. 따라서, r^* 가 오라클 G 에 질의되는 경우에 한해 두 게임이 서로 다르게 동작할 수 있다.

AskG₂는 GAME₂에서 r^* 가 공격자에 의해 오라클 G 에 질의되는 사건이다. 이후에서 우리는 모든 GAME_i에 대해 동일한 표기 AskG_i를 사용한다. 보조정리(Appendix 참고)에 의해 다음 부등식이 성립한다.

$$|\Pr[S_2] - \Pr[S_1]| \leq \Pr[\text{AskG}_2].$$

메모. 오라클 G 에 질의하는 사건을 고려할 때, 공격자 뿐만 아니라 복호화 오라클이 G 에 질의하는 것도 고려한다. 즉, Ask_{G_2} 는 공격자 및 복호화 오라클에 의해 r^* 가 G 에 질의되는 사건을 의미한다. 복호화 오라클은 $(s, t) \leftarrow g(c)$ 를 계산한 뒤 $r \leftarrow t \oplus H(s)$ 및 $M \leftarrow G(r) \oplus s$ 를 계산하는데, 이 때 $r \circ \rfloor G$ 에 질의된다.

5.3.4 3 번째 게임

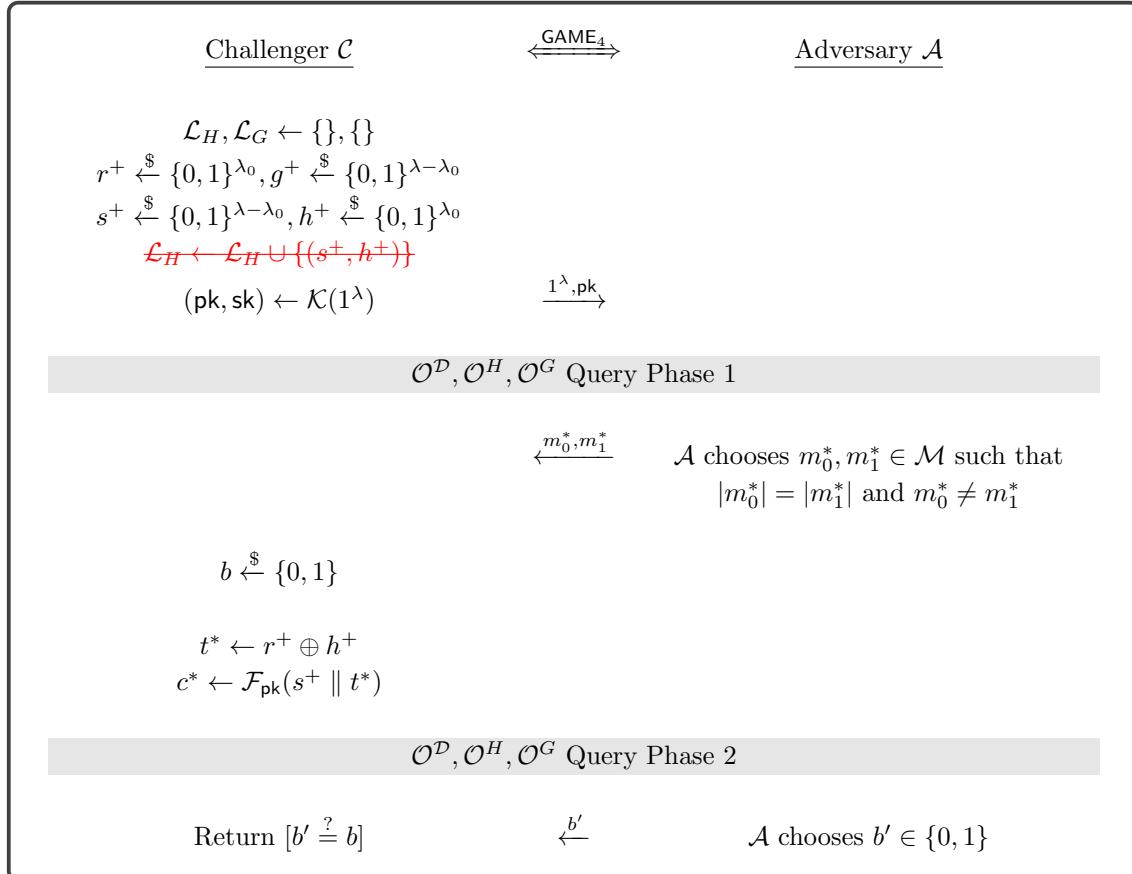


이번에는 $s^+ \xleftarrow{\$} \{0,1\}^{\lambda-\lambda_0}, h^+ \xleftarrow{\$} \{0,1\}^{\lambda_0}$ 를 무작위로 선택하고, s^* 대신 $s^+, H(s^*)$ 대신 h^+ 를 사용한다. 게임의 규칙은 GAME₂와 유사하다. 이 때, 다음을 만족한다.

$$\Pr[\text{AskG}_3] = \Pr[\text{AskG}_2].$$

메모. GAME₂에서 GAME₃로 변경하는 과정은 GAME₀에서 GAME₁으로 변경하는 과정과 유사하다. GAME₁에서 설명한 것과 마찬가지로, $(s^*, H(s^*))$ 와 s^+, h^+ 의 확률분포는 동일하다. 즉, 공격자는 두 게임에서 동일하게 동작하며, G 에 질의하는 동작도 동일하다. 따라서, $\Pr[\text{AskG}_3] = \Pr[\text{AskG}_2]$ 를 만족한다.

5.3.5 4 번째 게임



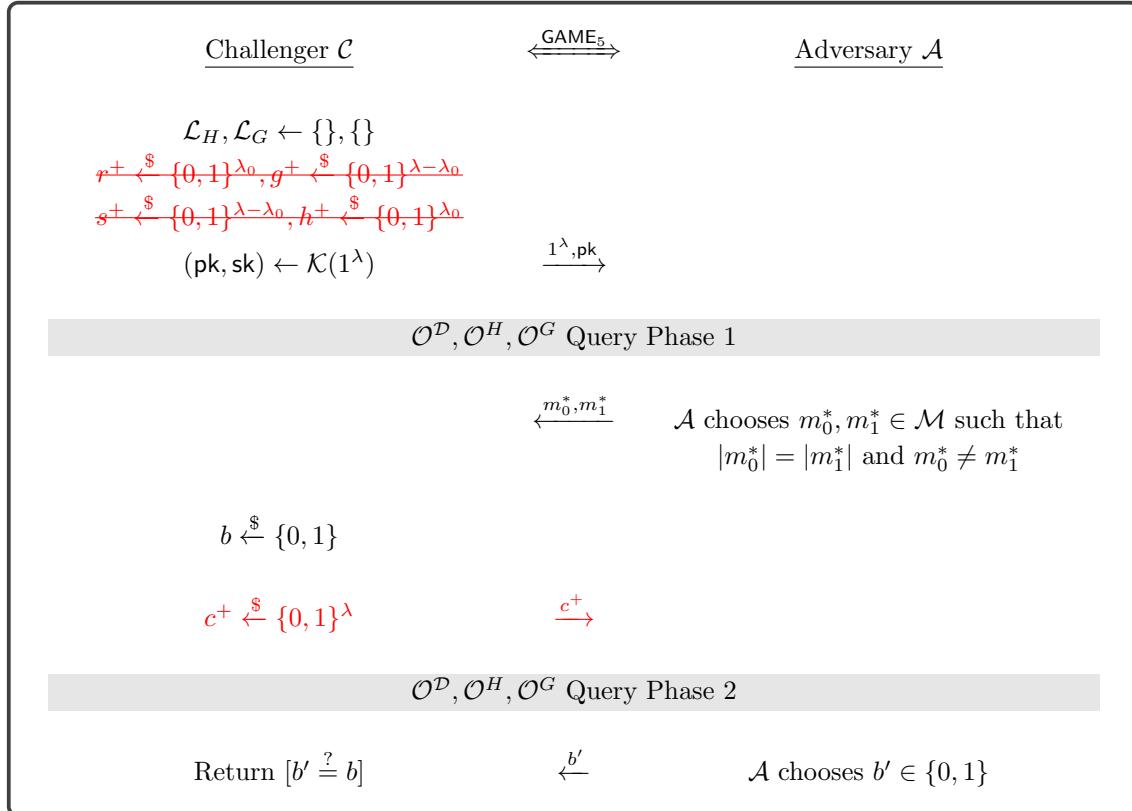
이번 게임에서는 GAME₃의 두 번째 규칙을 제거한다. 그러면 GAME₂와 유사하게, 다음이 성립한다.

$$|\Pr[\text{AskG}_4] - \Pr[\text{AskG}_3]| \leq \Pr[\text{AskH}_4].$$

여기서 AskH₄는 GAME₄에서 공격자 또는 복호화 오라클에 의해 s*가 H 오라클에 질의되는 사건을 나타낸다.

메모. GAME₃와 GAME₄에서 공격자가 H에 s*가 아닌 다른 값을 질의하면 두 게임에서 H는 동일하게 동작하지만, s*를 질의하면, GAME₃의 오라클은 h⁺를, GAME₄의 오라클은 무작위 값을 응답하기 때문에 공격자의 동작이 달라질 수 있다. 따라서 위 부등식을 만족한다. 이 도출 과정은 GAME₂에서 부등식 도출 과정과 유사하다.

5.3.6 5 번째 게임



GAME₅에서는 챌린지 암호문 $c^+ \xleftarrow{\$} \{0,1\}^\lambda$ 을 무작위로 선택하고, 단순히 $c^* = c^+$ 로 설정한다. 이 경우 다음을 만족한다.

$$\Pr[\text{AskH}_5] = \Pr[\text{AskH}_4].$$

메모. GAME₄에서 s^* 와 t^* 은 g^+ 와 h^+ 에 의해 균등분포를 따른다. \mathcal{F} 는 순열이므로, $c^* = \mathcal{F}_{pk}(s^* \parallel t^*)$ 도 균등분포를 따른다. 따라서 c^+ 와 동일한 균등분포를 따르므로, 공격자는 GAME₅에서도 동일하게 동작한다. 그러므로 $\Pr[\text{AskH}_5] = \Pr[\text{AskH}_4]$ 를 만족한다.

5.3.7 6 번째 게임

메모. $GAME_5$ 에서 복호화 오라클 D 는 다음과 같이 동작한다.

- $(s, t) = \mathcal{I}_{sk}(c)$ 을 계산한 후, $r = t \oplus H(s), M = s \oplus G(r)$ 을 계산한다.
- 만약 $[M]_{\lambda_1} = 0^{\lambda_1}$ 이면 $[M]^n$ 을 출력하고, 아니라면 “Reject”를 출력한다.

이번 게임에서는 복호화 오라클이 c 에 대응하는 r 값이 공격자에 의해 이전에 G 오라클에 질의되지 않았을 때, Reject를 반환하게 만든다.

- $(s, t) = \mathcal{I}_{sk}(c)$ 을 계산한 후, $r = t \oplus H(s)$ 을 계산한다.
- $r \circ | G$ 에 질의되지 않았다면(즉, \mathcal{L}_G 에 r 이 없다면), “Reject”를 출력한다.
- $M = s \oplus G(r)$ 을 계산한다.
(여기서 $G(r)$ 은 계산해서 얻는 것이 아닌, \mathcal{L}_G 에서 꺼내온 것)
- 만약 $[M]_{\lambda_1} = 0^{\lambda_1}$ 이면 $[M]^n$ 을 출력하고, 아니라면 “Reject”를 출력한다.

이는 c 가 유효한 암호문인데도 $r \circ | G$ 에 질의되지 않은 경우에만 차이를 발생시킨다.

메모. $GAME_5$ 에서는 공격자가 이전에 r 을 G 에 질의하지 않았을 때, 복호화 오라클에서 r 을 G 에 질의하고, 우연히 조건 $[s \oplus G(r)]_{\lambda_1} = 0^{\lambda_1}$ 가 성립하여 Reject 대신 평문을 반환할 수 있다. 그러나 $GAME_6$ 에서는 $r \circ | G$ 에 질의되지 않았다면 무조건 Reject를 반환한다. 즉, $[s \oplus G(r)]_{\lambda_1} = 0^{\lambda_1}$ 가 성립할 때, 두 게임에서 공격자의 전략이 달라질 수 있다.

$G(r)$ 이 균등분포를 따르므로, 다음과 같은 등식이 성립할 확률은 다음과 같다.

$$\Pr[[s \oplus G(r)]_{\lambda_1} = 0^{\lambda_1}] = \frac{1}{2^{\lambda_1}}.$$

모든 복호화 질의를 고려하면, 다음 부등식을 얻는다.

$$|\Pr[\text{AskH}_6] - \Pr[\text{AskH}_5]| \leq \frac{q_D}{2^{\lambda_1}}.$$

메모. $r \circ | G$ 에 질의되지 않았다는 것은, \mathcal{L}_G 내 모든 (γ, G_γ) 에 대해 $t = \gamma \oplus H(s)$ 를 계산하면 $c = \mathcal{F}_{pk}(s \parallel t)$ 가 성립하지 않는다는 것이다(\mathcal{F} 는 순열). 따라서 $GAME_7$ 의 복호화 오라클을 다음과 같이 해석할 수 있다.

- 암호문 $c = \mathcal{F}_{pk}(s \parallel t)$ 가 주어지면, $(s, t) = \mathcal{I}_{sk}(c)$ 을 계산한다.
- 모든 $(\gamma, G_\gamma) \in \mathcal{L}_G$ 에 대해 다음을 계산한다.

$$\theta = \gamma \oplus H(s), \quad M = s \oplus G_\gamma.$$

- 다음 조건을 검사한다.

$$c = \mathcal{F}_{pk}(s \parallel \theta), \quad [M]_{\lambda_1} = 0^{\lambda_1}.$$

- 조건을 만족하면 $[M]^n$ 을 출력한다. 조건을 만족하는 경우가 없다면 “Reject”를 출력한다.

5.3.8 7 번째 게임

이번 게임에서는, 복호화 오라클이 c 에 대응하는 s 값이 공격자에 의해 이전에 오라클 H 에 질의되지 않은 경우 Reject를 반환하게 만든다.

- $(s, t) = \mathcal{I}_{\text{sk}}(c)$ 을 계산한다.
- s 가 H 에 질의되지 않았다면(즉, \mathcal{L}_H 에 s 가 없다면), Reject를 반환한다.
- $r = t \oplus H(s)$ 을 계산한다.
(여기서 $H(s)$ 은 계산해서 얻는 것이 아닌, \mathcal{L}_H 에서 꺼내온 것)
- $r \circ | G$ 에 질의되지 않았다면(즉, \mathcal{L}_G 에 r 이 없다면), “Reject”를 출력한다.
- $M = s \oplus G(r)$ 을 계산한다.
(여기서 $G(r)$ 은 계산해서 얻는 것이 아닌, \mathcal{L}_G 에서 꺼내온 것)
- 만약 $[M]_{\lambda_1} = 0^{\lambda_1}$ 이면 $[M]^n$ 을 출력하고, 아니라면 “Reject”를 출력한다.

이 변경은 c 가 유효한 암호문이고, r 은 오라클 G 에 이미 질의된 반면, s 는 여전히 H 에 질의되지 않았을 경우에만 차이를 발생시킨다.

메모. 이는 GAME_6 에서의 변경과 비슷하다.

$r = H(s) \oplus t$ 는 균등 분포를 따르므로, $r \circ | G$ 에 질의되었을 확률은 $q_G/2^{\lambda_0}$ 보다 작다. 이전 게임에서는 복호화 오라클이 G 에 추가 질의를 하지 않는다는 점에 유의한다.

메모. GAME_7 에서는 s 가 H 에 질의되지 않았다면, 무조건 Reject를 반환하지만, GAME_6 에서는 복호화 오라클이 s 를 H 에 질의하여, $r = H(s) \oplus t$ 를 계산하고, $r \circ | G$ 에 질의되었는지 확인한다. 즉 $r \circ | G$ 에 질의되었는지에 따라 유효한 암호문을 출력할 수도 있다. 따라서 $r \circ | G$ 에 질의되었는지에 따라 두 게임에서 공격자의 전략이 달라질 수도 있다.

메모. GAME_6 부터 복호화 오라클은 r 을 G 에 질의하지 않는다. 따라서 우리는 공격자의 G 오라클 질의만 고려하는 것이 가능하다. 그러므로 $r \circ | G$ 에 질의되었을 확률은 $q_G/2^{\lambda_0}$ 보다 작다고 할 수 있다.

모든 복호화 질의를 고려하면 다음 부등식을 얻는다.

$$|\Pr[\text{AskH}_7] - \Pr[\text{AskH}_6]| \leq \frac{q_D q_G}{2^{\lambda_0}}.$$

메모. GAME_6 에서 복호화 오라클을 재해석한 것과 같아, GAME_7 에서도 복호화 오라클을 재해석할 수 있다. 즉, $r \circ | G$ 에 질의되었더라도, s 가 H 에 질의되지 않았다는 것은, 모든 $(\delta, H_\delta) \in \mathcal{L}_H$ 에 대해 $c = \mathcal{F}_{pk}(\delta \parallel t)$ 가 성립하지 않는다는 것이다.

- 암호문 $c = \mathcal{F}_{pk}(s \parallel t)$ 가 주어지면, $(s, t) = \mathcal{I}_{\text{sk}}(c)$ 을 계산한다.
- 모든 $(\gamma, G_\gamma) \in \mathcal{L}_G$ 와 모든 $(\delta, H_\delta) \in \mathcal{L}_H$ 에 대해 다음을 계산한다.

$$\sigma = \delta, \quad \theta = \gamma \oplus H_\delta, \quad M = \delta \oplus G_\gamma.$$

- 다음 조건을 검사한다.

$$c = \mathcal{F}_{pk}(\delta \parallel \theta), \quad [M]_{\lambda_1} = 0^{\lambda_1}.$$

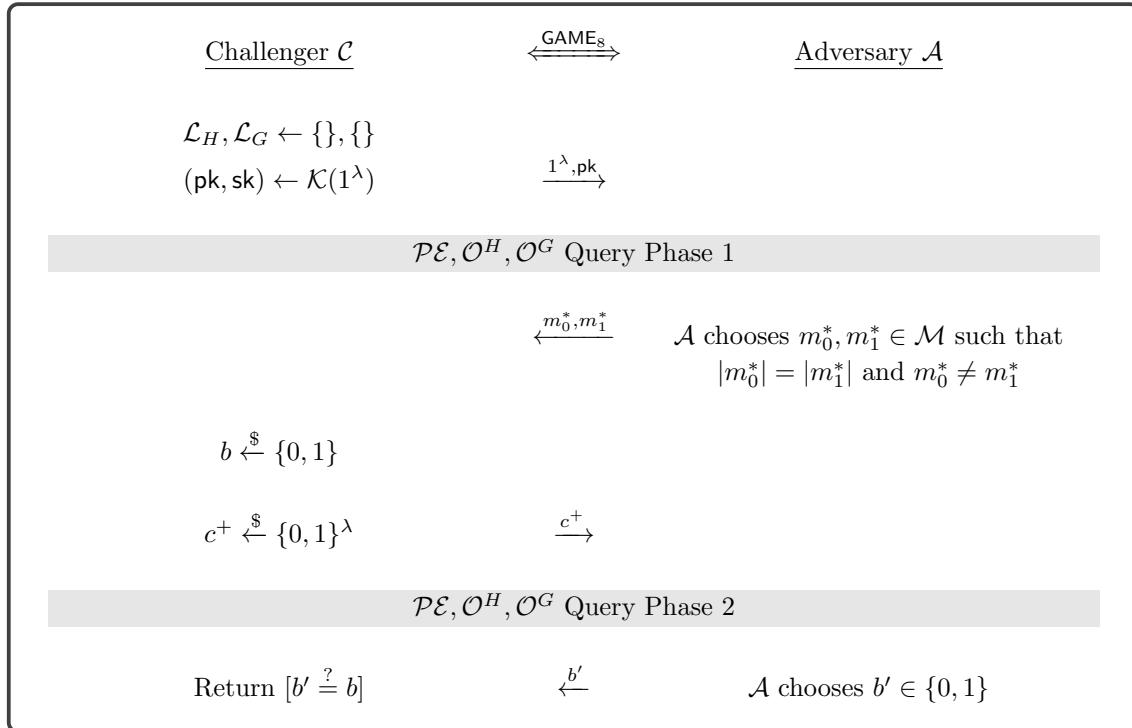
- 조건을 만족하면 $[M]^n$ 을 출력한다. 조건을 만족하는 경우가 없다면 “Reject”를 출력한다.

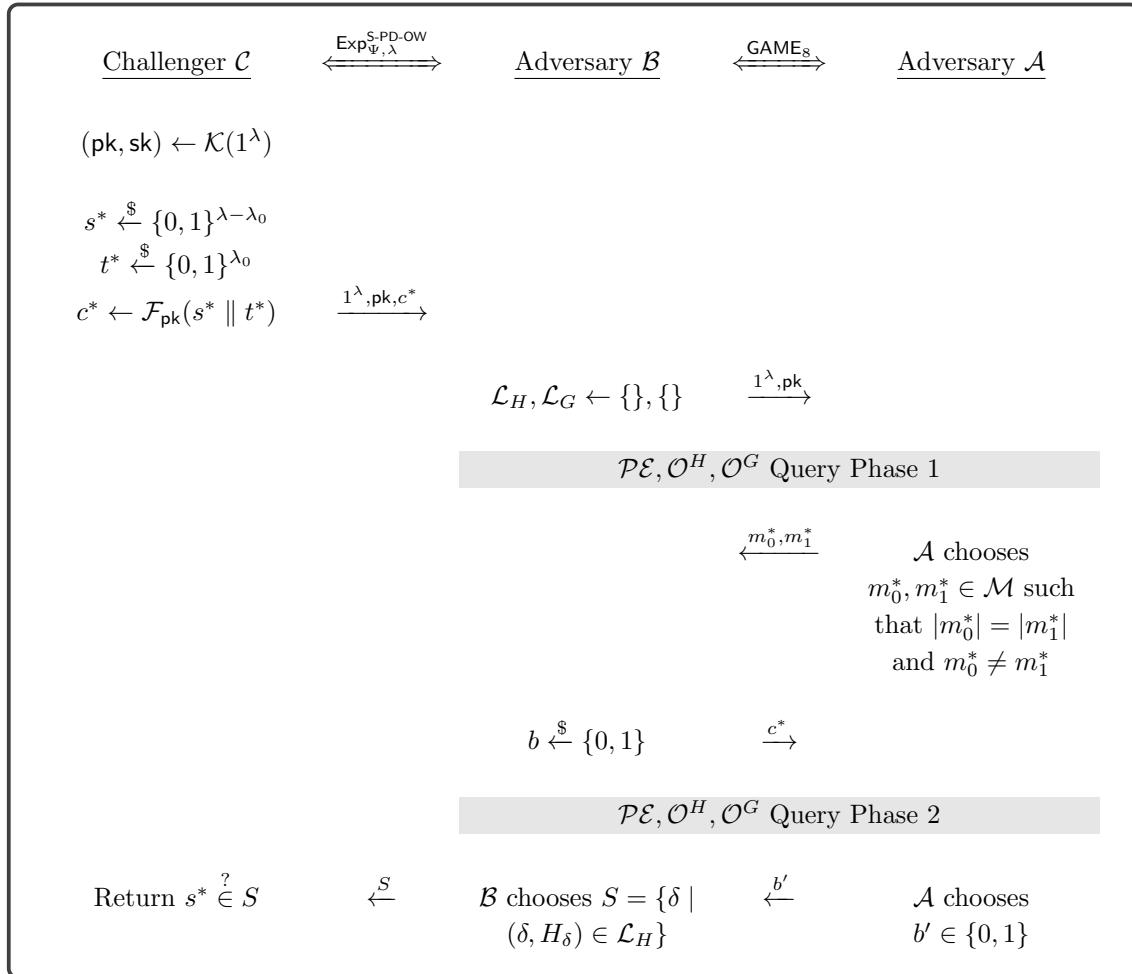
5.3.9 8 번째 게임

GAME₈에서는 복호화 오라클을 평문 추출기로 완전히 대체한다. 평문 추출기는 GAME₇의 복호화 오라클을 완벽하게 시뮬레이션할 수 있다. 따라서, 다음을 만족한다.

$$\Pr[\text{AskH}_8] = \Pr[\text{AskH}_7].$$

메모. 평문 추출기 \mathcal{PE} 는 GAME₇에서 재구성한 복호화 오라클을 완벽하게 시뮬레이션 할 수 있다. 따라서 복호화 오라클을 평문 추출기로 대체하더라도 공격자는 게임 전략을 바꾸지 않는다. 즉, $\Pr[\text{AskH}_8] = \Pr[\text{AskH}_7]$ 이다.





또한, 다음을 만족하는 공격자 \mathcal{B} 가 존재한다.

$$\Pr[\mathsf{AskH}_8] \leq \text{Adv}_{\Psi, \lambda}^{\text{S-PD-OW}}(\mathcal{B}; \tau', q_H).$$

메모. 위 실험에서 공격자 \mathcal{B} 는 $S = \{\delta \mid (\delta, H_\delta) \in \mathcal{L}_H\}$ 를 반환한다. 공격자 \mathcal{A} 가 s^* 를 H 에 질의했다면, $s^* \in S$ 가 성립하여 실험은 1을 반환한다. 즉,

$$\Pr[\mathsf{AskH}_8] \leq \Pr[\mathsf{Exp}_{\Psi, \lambda}^{\text{S-PD-OW}}(\mathcal{B}; \tau', q_H) = 1] = \text{Adv}_{\Psi, \lambda}^{\text{S-PD-OW}}(\mathcal{B}; \tau', q_H).$$

따라서, 다음이 성립한다.

$$\begin{aligned} \frac{1}{2} \cdot \text{Adv}_{\Pi, \lambda}^{\text{IND-CCA2}}(\mathcal{A}; \tau, q_D, q_H, q_G) &= |\Pr[S_0] - \Pr[S_2]| \\ &\leq \Pr[\mathsf{AskG}_2] \\ &\leq \Pr[\mathsf{AskG}_4] + \Pr[\mathsf{AskH}_4] \\ &\leq \frac{q_G + q_D}{2^{\lambda_0}} + \Pr[\mathsf{AskH}_5] \\ &\leq \frac{q_G + q_D}{2^{\lambda_0}} + \frac{q_D}{2^{\lambda_1}} + \Pr[\mathsf{AskH}_6] \\ &\leq \frac{q_G + q_D}{2^{\lambda_0}} + \frac{q_D}{2^{\lambda_1}} + \frac{q_D q_G}{2^{\lambda_0}} + \Pr[\mathsf{AskH}_7] \\ &\leq \frac{q_G + q_D + q_D q_G}{2^{\lambda_0}} + \frac{q_D}{2^{\lambda_1}} + \text{Adv}(\mathcal{B}; q_H, \tau'). \end{aligned}$$

$\frac{\lambda_1}{\lambda_0}$,

$$\frac{1}{2} \cdot \text{Adv}_{\Pi, \lambda}^{\text{IND-CCA2}}(\mathcal{A}; \tau, q_D, q_H, q_G) \leq \text{Adv}_{\Psi, \lambda}^{\text{S-PD-OW}}(\mathcal{B}; \tau', q_H) + \frac{q_G + q_D + q_D q_G}{2^{\lambda_0}} + \frac{q_D}{2^{\lambda_1}}.$$

6 박종환 교수님 발표자료 검토

본 절에서는 박종환 교수님의 발표자료를 검토한다. 먼저 발표자료에서 제공하는 KEM 구조를 설명한다. 여기서는 전부 내 생각이므로, 따로 메모를 적지 않으며, 지금까지 사용한 기호가 다를 수도 있음에 주의한다.

6.1 Structure

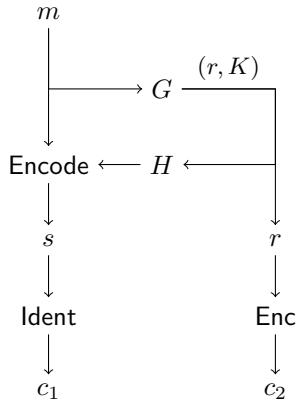


Figure 2: Encapsulation

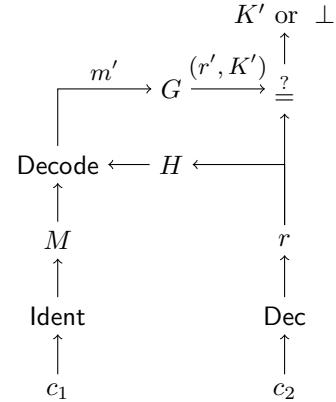


Figure 3: Decapsulation

6.2 Questions

- McEliece와 PALOMA의 Encryption은 Permutation인가?

RSA-OAEP 증명에서는 RSA가 Trapdoor Permutation이라는 가정을 이용한다. 만약 Permutation이 아니라면(그냥 Trapdoor Function이라면), 증명이 성립하지 않을 수 있다.

- Encode, Decode, Ident의 정체가 무엇인가?

Encode, Decode, Ident는 어떤 함수인지 모르겠다. 만약 Encode, Decode가 단순 XOR 연산이고, Ident함수가 없다면, Encap은 다음 구조를 가진다.

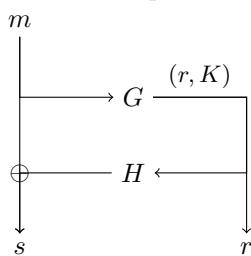


Figure 4: Encapsulation

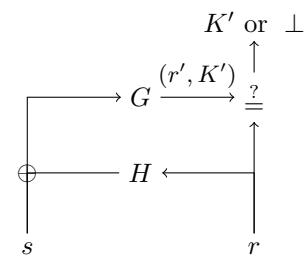


Figure 5: Decapsulation

- OAEP의 구조가 다른데, 위 상황에서는 평문 추출기를 어떻게 정의해야 하는가?

피, 땀, 노력.

7 t가 그대로 내려온다면

OAEP 변환은 다음과 같다. (박종환 교수님 구조랑 유사하게 구성하기 위해, 이전 그림을 좌우대칭하였다. 기호까지 변경하면 헷갈릴 것 같아서, 기호는 그대로 두었다.)

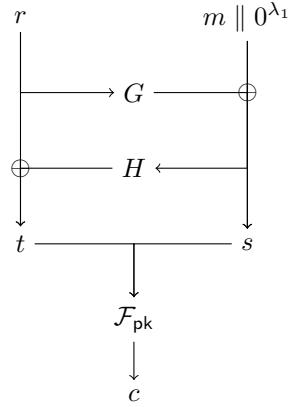


Figure 6: OAEP 변환 (F_{pk} 포함)

이번에는 $F_{pk} : \{0,1\}^{\lambda_0} \times \{0,1\}^{\lambda-\lambda_0} \rightarrow \{0,1\}^{\lambda_0} \times \{0,1\}^{\lambda-\lambda_0}$ 를 $F_{pk} : \{0,1\}^{\lambda-\lambda_0} \rightarrow \{0,1\}^{\lambda-\lambda_0}$ 이라 생각하고, OAEP 변환을 다음과 같이 재구성한다.

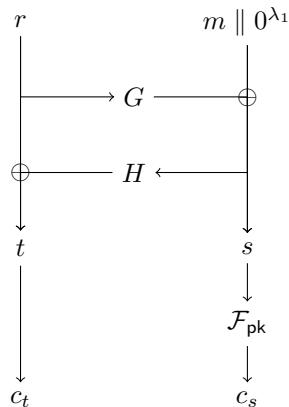


Figure 7: OAEP 변환 (F_{pk} 포함)

지금부터 할 것은, F_{pk} 가 OW-secure할 때, 위 OAEP 변환이 IND-CCA2-secure를 만족함을 보이는 것이다. 5절과 동일하게 증명을 진행한다.

7.1 평문추출기

평문추출기 \mathcal{PE} 를 다음과 같이 정의한다.

평문 추출기 \mathcal{PE} 의 입력은 다음과 같다.

- 무작위 오라클 G, H 에 대한 질의 응답 쌍을 모아 놓은 두 개의 리스트 $\mathcal{L}_G, \mathcal{L}_H$.
- 유효한 암호문 c^* .
- 후보 암호문 c . 이때, $c \neq c^*$ 이다.

추출기 \mathcal{PE} 의 동작 방식은 다음과 같다.

- 암호문 $c = t \parallel \mathcal{F}_{\text{pk}}(s)$ 가 주어지면, \mathcal{L}_G 에 있는 모든 (γ, G_γ) 와 \mathcal{L}_H 에 있는 모든 (δ, H_δ) 에 대해 다음을 계산한다.

$$\sigma = \delta, \quad \theta = \gamma \oplus H_\delta, \quad \mu = G_\gamma \oplus \delta.$$

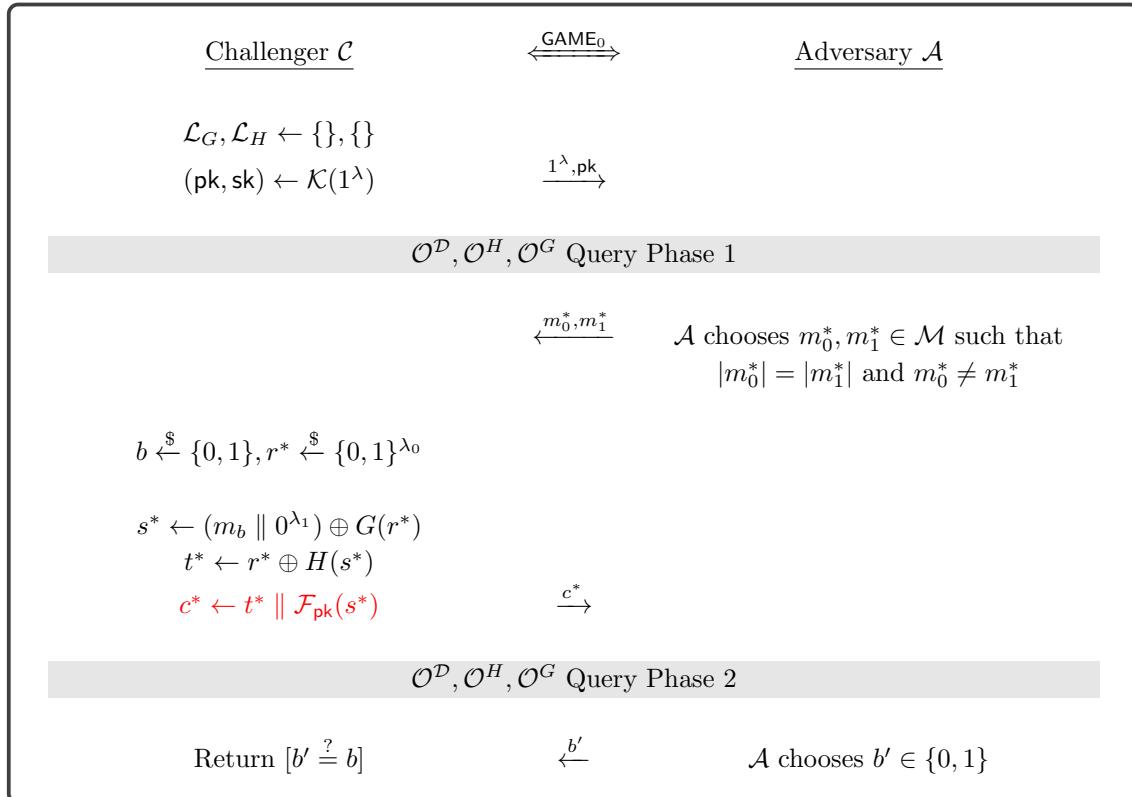
- 그리고 다음 조건을 검사한다.

$$c = \theta \parallel \mathcal{F}_{\text{pk}}(\sigma) \quad \text{and} \quad [\mu]_{\lambda_1} = 0^{\lambda_1}.$$

- 조건이 만족되면, \mathcal{PE} 는 μ 의 앞부분, 즉 $[\mu]^n$ 을 평문으로 출력하고 종료한다. 조건을 만족하는 조합이 없다면, \mathcal{PE} 는 Reject 메시지를 반환한다.

7.2 게임 변환

7.2.1 게임 0

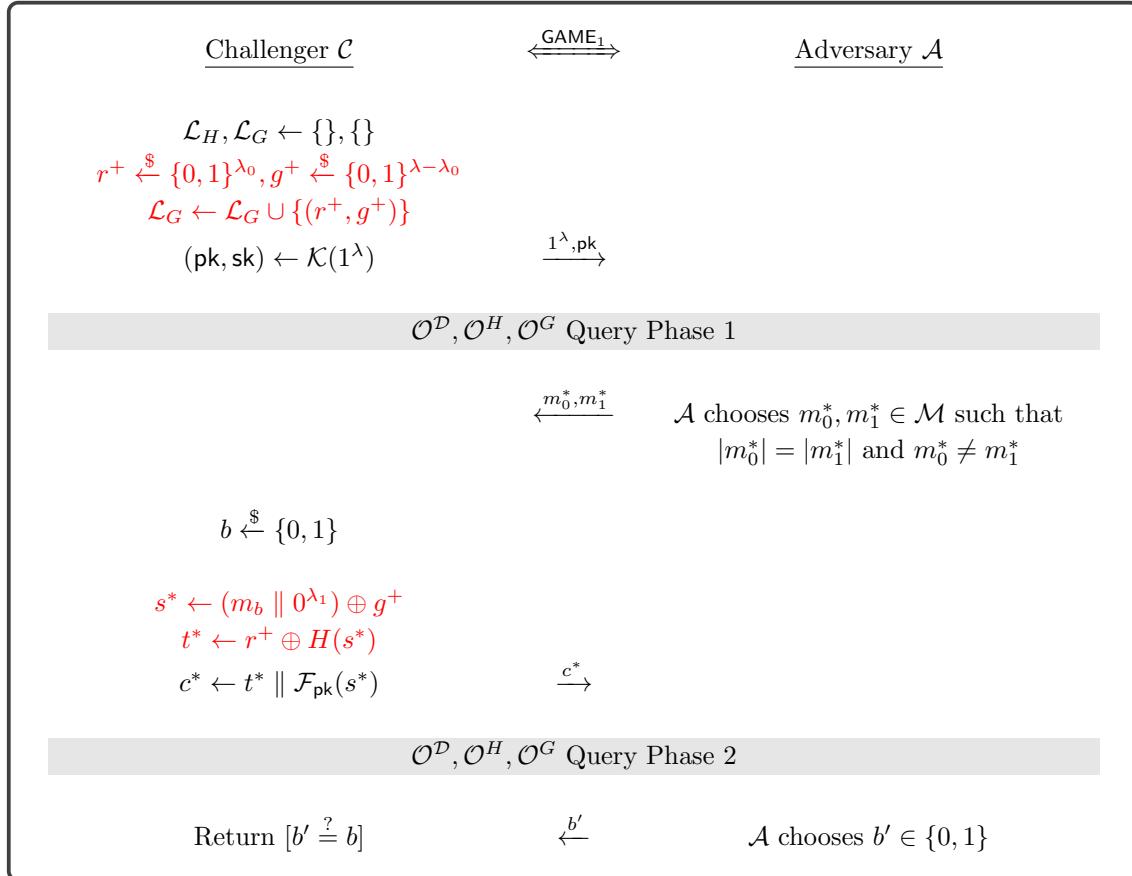


원래는 $c^* \leftarrow \mathcal{F}_{\text{pk}}(s^* \parallel t^*)$ 이지만, 바꾼 구조에서는 $c^* \leftarrow t^* \parallel \mathcal{F}_{\text{pk}}(s^*)$ 이다. GAME_0 는 IND-CCA2 게임과 동일하다. 따라서 공격자 \mathcal{A} 의 능력치는 다음과 같다.

$$\text{Adv}_{\Pi, \lambda}^{\text{IND-CCA2}}(\mathcal{A}) = 2 \cdot \left| \Pr[S_0] - \frac{1}{2} \right|.$$

S_0 는 공격자가 이 게임에서 승리할 확률을 의미한다.

7.2.2 게임 1



GAME₁에서는 r^+, g^+ 를 게임 초기에 생성하고, (r^+, g^+) 를 \mathcal{L}_G 에 저장한다. 그리고 s^*, t^* 를 다음과 같이 생성한다.

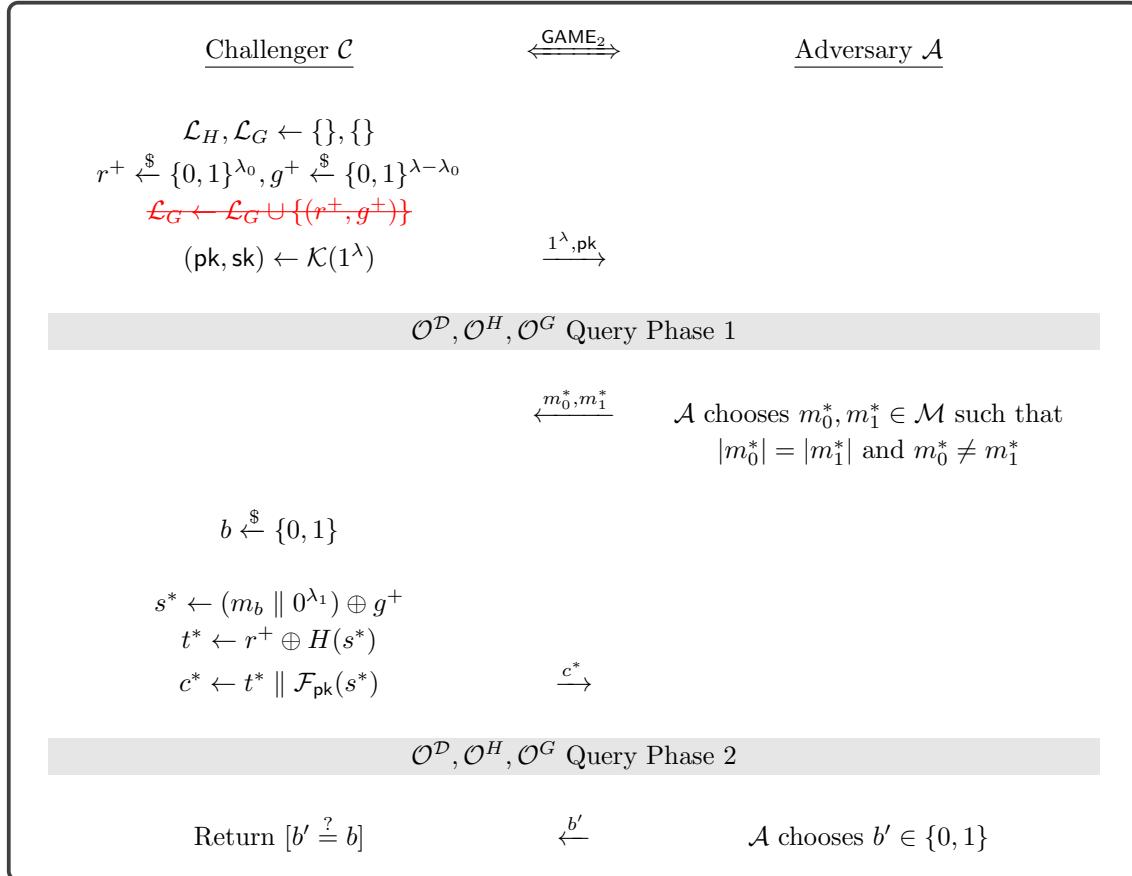
$$s^* \leftarrow (m_b \parallel 0^{\lambda_1}) \oplus g^+, \quad t^* \leftarrow r^+ \oplus H(s^*).$$

우리는 $(r^*, G(r^*))$ 와 동일한 분포를 가지는 (r^+, g^+) 로 대체했으므로 다음을 만족한다.

$$\Pr[S_1] = \Pr[S_0].$$

증명. (r^+, g^+) 가 $(r^*, G(r^*))$ 와 동일한 분포를 가진다. 오라클 $\mathcal{O}^H, \mathcal{O}^G, \mathcal{O}^D$ 는 변하지 않고, 그리고 (s^*, t^*) 의 분포가 달라지지 않는다 따라서 공격자 \mathcal{A} 는 이전과 동일한 전략을 사용한다.

7.2.3 게임 2



GAME₂에서는 초기에 (r^+, g^+) 를 무작위로 생성하지만, \mathcal{L}_G 에 저장하지는 않는다. r^+, g^+ 는 s^*, t^* 를 계산할 때만 사용된다. 따라서 다음을 만족한다.

$$\Pr[S_2] = \frac{1}{2}.$$

메모. 공격자는 r^+, g^+ 에 대한 정보를 얻을 수 없으므로, s^*, t^* 에 대한 정보도 얻을 수 없다. 따라서 공격자는 c^* 에 대한 정보를 얻을 수 없어 (\mathcal{F}_{pk} 는 순열), 승률은 $1/2$ 이 된다.

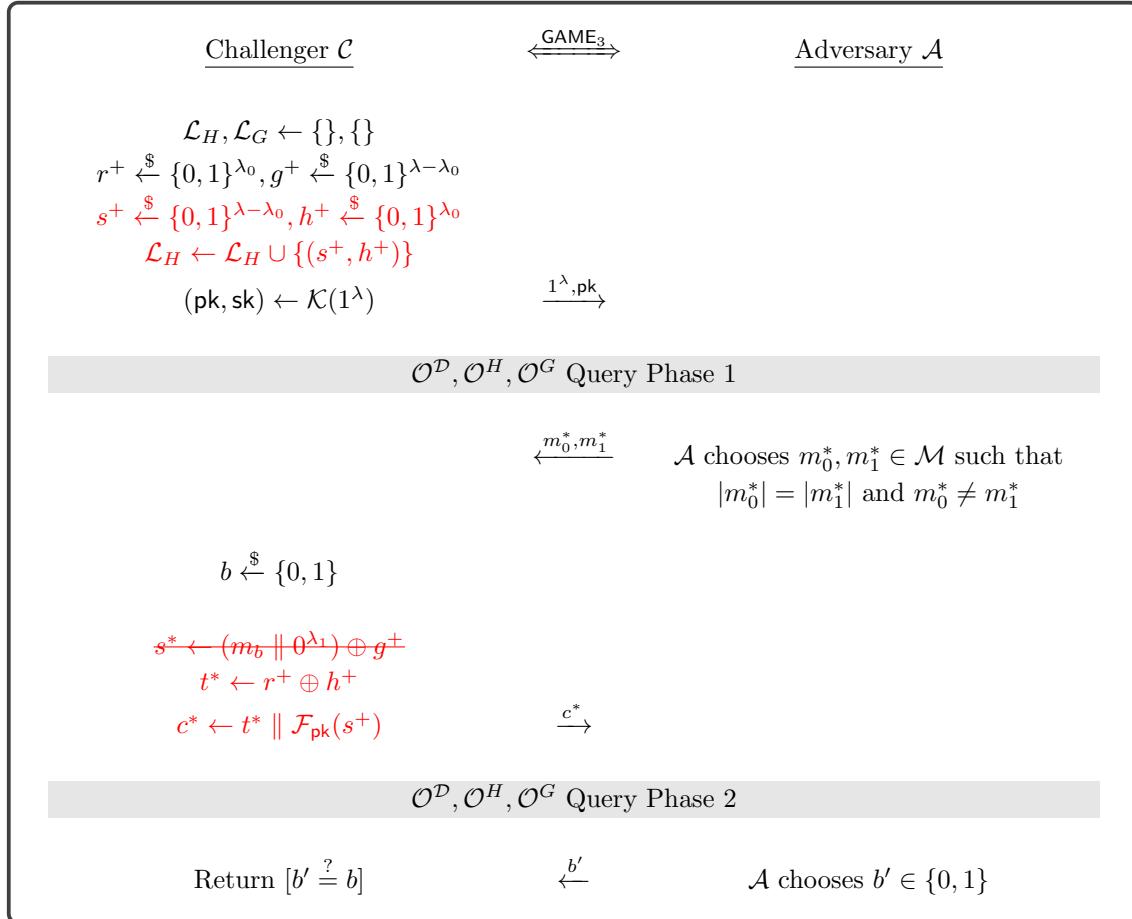
GAME₁과 GAME₂에서, r^* 가 오라클 G 에 질의되는 경우에만 공격자 \mathcal{A} 의 전략은 달라질 수 있다. Ask G_2 를 GAME₂에서 r^* 가 공격자에 의해 오라클 G 에 질의되는 사건이라 하자. (이후에서 모든 GAME_i에 대해 동일한 표기 Ask G_i 를 사용한다) 이때, 다음 부등식이 성립한다. (Appendix의 보조정리 참고)

$$|\Pr[S_2] - \Pr[S_1]| \leq \Pr[\text{Ask } G_2].$$

메모. 공격자가 r^* 를 질의하지 않는 상황에서(즉, $\neg \text{Ask } G_2$ 상황에서)

- GAME₁과 GAME₂에서 오라클 G 는 동일하게 동작.
- 공격자는 전략을 수정하지 않고, 게임의 승률은 변하지 않음.
- 즉, $\Pr[S_2 \wedge \neg \text{Ask } G_2] = \Pr[S_1 \wedge \neg \text{Ask } G_2]$.
- 따라서 보조정리에 의해, $|\Pr[S_2] - \Pr[S_1]| \leq \Pr[\text{Ask } G_2]$.

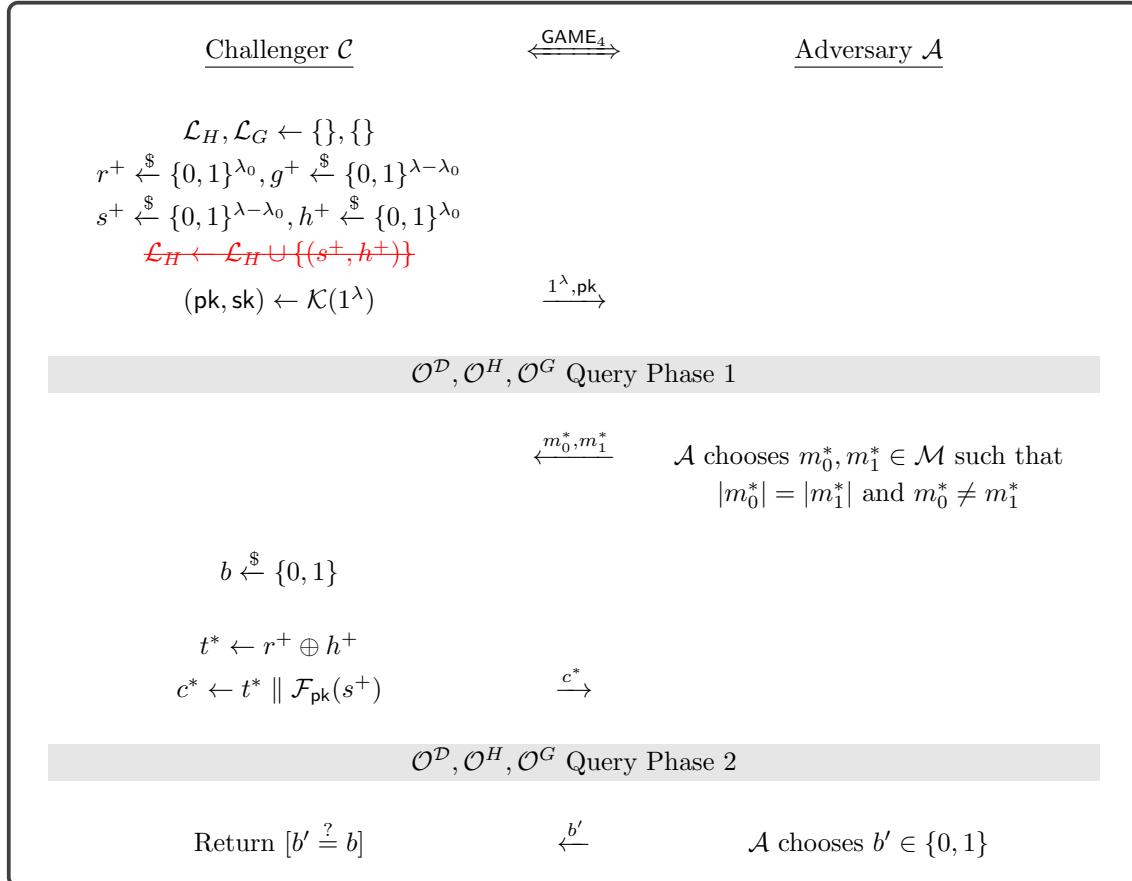
7.2.4 게임 3



GAME₃에서는 $(s^*, H(s^*))$ 대신 s^+, h^+ 를 게임 초기에 무작위로 선택하여 \mathcal{L}_H 에 저장하고, (s^+, h^+) 를 사용한다. 이때 다음을 만족한다. (GAME₀에서 GAME₁으로의 변경과 유사하므로 설명은 생략.)

$$\Pr[\mathsf{AskG}_3] = \Pr[\mathsf{AskG}_2].$$

7.2.5 게임 4



GAME₄에서는 초기에 생성한 (s^+, h^+) 를 \mathcal{L}_H 에 저장하는 과정을 생략한다. 이때 다음이 성립한다.

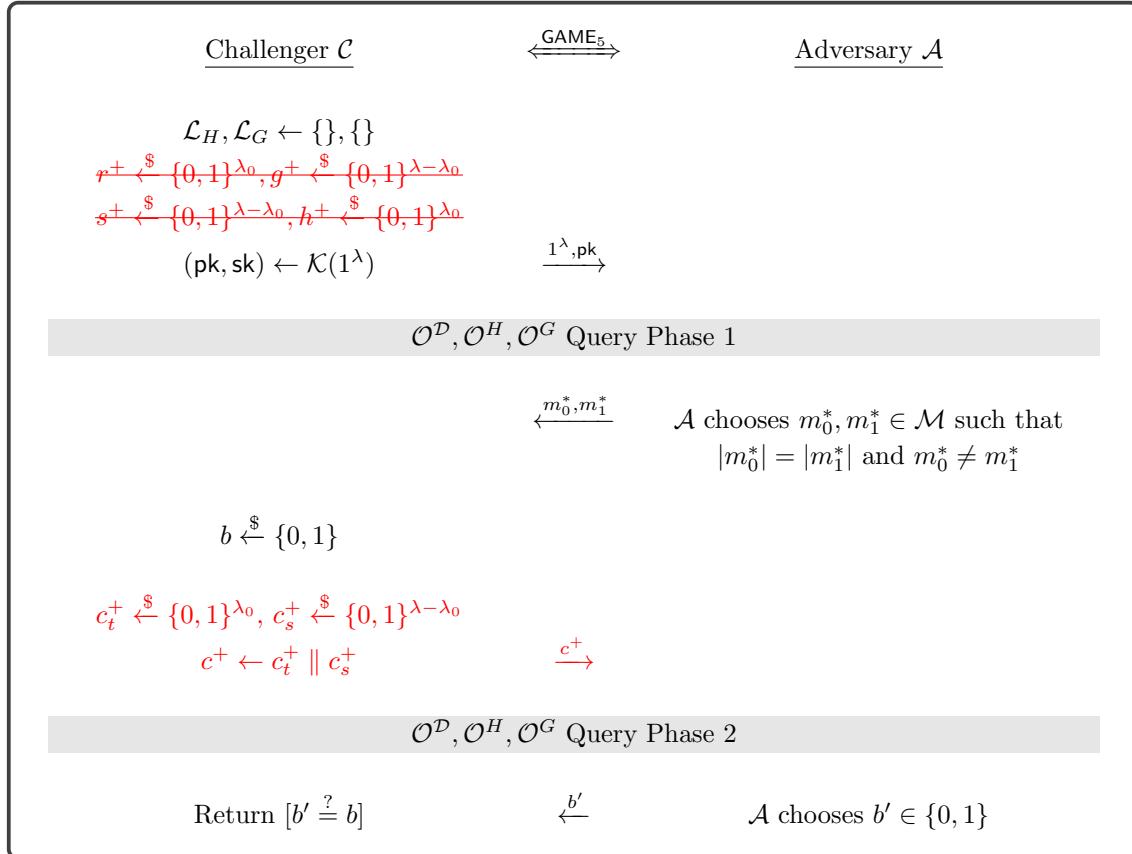
$$|\Pr[\text{AskG}_4] - \Pr[\text{AskG}_3]| \leq \Pr[\text{AskH}_4].$$

여기서 AskH₄는 GAME₄에서 공격자 또는 복호화 오라클에 의해 s^* 가 H 오라클에 질의되는 사건을 나타낸다.

예제. 공격자가 s^* 를 질의하지 않는 상황에서(즉, $\neg \text{AskH}_4$ 상황에서)

- GAME₃과 GAME₄에서 오라클 H 는 동일하게 동작.
- 공격자는 전략을 수정하지 않음.
- 즉, $\Pr[\text{AskG}_4 \wedge \neg \text{AskH}_4] = \Pr[\text{AskG}_3 \wedge \neg \text{AskH}_4]$.
- 따라서 보조정리에 의해, $|\Pr[\text{AskG}_4] - \Pr[\text{AskG}_3]| \leq \Pr[\text{AskH}_4]$.

7.2.6 게임 5



GAME₅에서는 도전 암호문 c_t^+ 와 c_s^+ 을 무작위로 선택하고, $c^+ = c_t^+ \parallel c_s^+$ 를 전달한다. 이때 다음을 만족한다.

$$\Pr[\mathsf{AskH}_5] = \Pr[\mathsf{AskH}_4].$$

메모. GAME₄에서 s^* 과 t^* 은 균등분포이고, \mathcal{F}_{pk} 는 순열이므로, c^* 은 균등분포를 따른다. c^+ 와 c^* 은 동일한 분포를 따르므로, 공격자는 동일한 전략을 사용한다.

7.2.7 게임 6

GAME₅에서 복호화 오라클 \mathcal{D} 는 질의한 암호문 $c = c_t \parallel c_s$ 에 대해 다음과 같이 동작한다.

- $s = \mathcal{I}_{\text{sk}}(c_s)$ 을 계산한 후, $r = c_t \oplus H(s)$, $M = s \oplus G(r)$ 을 계산한다.
- 만약 $[M]_{\lambda_1} = 0^{\lambda_1}$ 이면 $[M]^n$ 을 출력하고, 아니라면 “Reject”를 출력한다.

GAME₆에서는 복호화 오라클이 c 에 대응하는 r 값이 공격자에 의해 이전에 G 오라클에 질의되지 않았을 때, Reject를 반환하게 만든다. 이렇게 바꾼 복호화 오라클은 다음과 같이 동작한다.

- $s = \mathcal{I}_{\text{sk}}(c_s)$ 을 계산한 후, $r = c_t \oplus H(s)$ 을 계산한다.
- $r \circ | G$ 에 질의되지 않았다면, “Reject”를 출력한다.
- $M = s \oplus G_r$ 을 계산한다.
- 만약 $[M]_{\lambda_1} = 0^{\lambda_1}$ 이면 $[M]^n$ 을 출력하고, 아니라면 “Reject”를 출력한다.

메모. $r \circ | G$ 에 질의되지 않았다는 것은 $(r, G_r) \notin \mathcal{L}_G$ 를 의미한다. 따라서 두 번째 단계를 넘어가면, $(r, G_r) \in \mathcal{L}_G$ 이므로, $M = s \oplus G_r$ 를 계산할 수 있다. 이때, r 을 G 에 질의하지 않고, G_r 을 꺼내와서 사용한다.

이는 c 가 유효한 암호문인데도 $r \circ | G$ 에 질의되지 않은 경우에만 차이를 발생시킨다. $G(r)$ 이 균등 분포를 따르므로, 다음과 같은 등식이 성립할 확률은 다음과 같다.

$$\Pr[[s \oplus G(r)]_{\lambda_1} = 0^{\lambda_1}] = \frac{1}{2^{\lambda_1}}.$$

모든 복호화 질의를 고려하면, 다음 부등식을 얻는다.

$$|\Pr[\text{AskH}_6] - \Pr[\text{AskH}_5]| \leq \frac{q_D}{2^{\lambda_1}}.$$

메모. 복호화 오라클이 유효한 암호문 c 를 입력받을 때,

- GAME₅에서
 - $r \circ | G$ 에 질의되었다. $\implies [M]_{\lambda_1} \neq 0^{\lambda_1}$ 이면 Reject.
 - $r \circ | G$ 에 질의되지 않았다. $\implies [M]_{\lambda_1} \neq 0^{\lambda_1}$ 이면 Reject.
- GAME₆에서
 - $r \circ | G$ 에 질의되었다. $\implies [M]_{\lambda_1} \neq 0^{\lambda_1}$ 이면 Reject.
 - $r \circ | G$ 에 질의되지 않았다. \implies Reject.

즉, $r \circ | G$ 에 질의되지 않은 상황에서, $[M]_{\lambda_1} = 0^{\lambda_1}$ 인 사건이 발생하면 GAME₅와 GAME₆의 복호화 오라클 응답이 달라질 수 있다. 즉, Bad 사건은 “ q_D 번의 복호화 오라클 질의동안 $[M]_{\lambda_1} = 0^{\lambda_1}$ 인 경우가 발생하는 사건”이다.

7.2.8 게임 7

GAME_7 에서는 복호화 오라클이 $c = c_t \parallel c_s$ 에 대응하는 s 값이 공격자에 의해 이전에 오라클 H 에 질의되지 않은 경우 Reject를 반환하게 만든다.

- $s = \mathcal{I}_{\text{sk}}(c_s)$ 을 계산한다.
- s 가 H 에 질의되지 않았다면, Reject를 반환한다.
- $r = c_t \oplus H_s$ 을 계산한다.
- $r \circ | G$ 에 질의되지 않았다면, “Reject”를 출력한다.
- $M = s \oplus G_r$ 을 계산한다.
- 만약 $[M]_{\lambda_1} = 0^{\lambda_1}$ 이면 $[M]^n$ 을 출력하고, 아니라면 “Reject”를 출력한다.

이 변경은 c 가 유효한 암호문이고, r 은 오라클 G 에 이미 질의된 반면, s 는 여전히 H 에 질의되지 않았을 경우에만 차이를 발생시킨다. $r = c_t \oplus H_s$ 는 균등 분포를 따르므로, $r \circ | G$ 에 질의되었을 확률은 $q_G / 2^{\lambda_0}$ 보다 작다. 이전 게임에서는 복호화 오라클이 G 에 추가 질의를 하지 않는다는 점에 유의한다. 모든 복호화 질의를 고려하면 다음 부등식을 얻는다.

$$|\Pr[\text{AskH}_7] - \Pr[\text{AskH}_6]| \leq \frac{q_D q_G}{2^{\lambda_0}}.$$

메모. 복호화 오라클이 유효한 암호문 c 를 입력받을 때,

- GAME_6 에서,
 - $r \circ | G$ 에 질의되지 않았다. $\implies \text{Reject}$.
 - $r \circ | G$ 에 질의되었고, s 가 H 에 질의되었다. $\implies [M]_{\lambda_1} \neq 0^{\lambda_1} \circ |$ 면 Reject.
 - $r \circ | G$ 에 질의되었고, s 가 H 에 질의되지 않았다. $\implies [M]_{\lambda_1} \neq 0^{\lambda_1} \circ |$ 면 Reject.
- GAME_7 에서,
 - $r \circ | G$ 에 질의되지 않았다. $\implies \text{Reject}$.
 - $r \circ | G$ 에 질의되었고, s 가 H 에 질의되었다. $\implies [M]_{\lambda_1} \neq 0^{\lambda_1} \circ |$ 면 Reject.
 - $r \circ | G$ 에 질의되었고, s 가 H 에 질의되지 않았다. $\implies \text{Reject}$.

즉, $r \circ | G$ 에 질의되었고, $s \circ | H$ 에 질의되지 않은 상황에서, $[M]_{\lambda_1} = 0^{\lambda_1}$ 인 사건이 발생하면 GAME_6 와 GAME_7 의 복호화 오라클 응답이 달라질 수 있다. 즉, Bad 사건은 “ q_D 번의 복호화 오라클 질의하는 동안, $r \circ | G$ 에 질의되지 않으면서 $[M]_{\lambda_1} = 0^{\lambda_1}$ 인 경우가 발생하는 사건”이다.

7.2.9 게임 8

GAME_8 에서는 복호화 오라클을 평문 추출기로 완전히 대체한다. 평문 추출기는 GAME_7 의 복호화 오라클을 완벽히 시뮬레이션할 수 있다. 따라서, 다음을 만족한다.

$$\Pr[\text{AskH}_8] = \Pr[\text{AskH}_7].$$

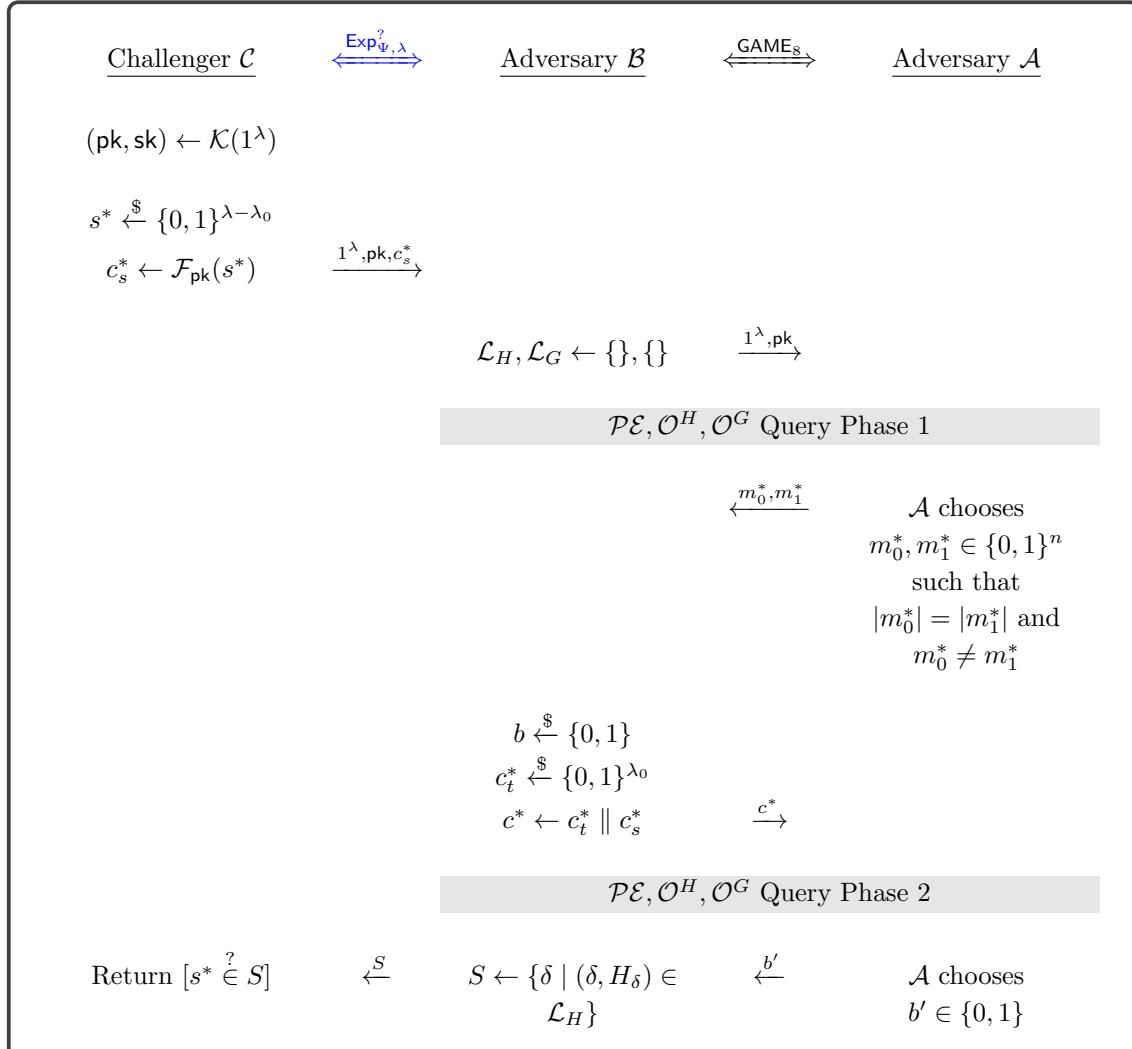
메모. 평문 추출기 \mathcal{PE} 가 유효한 암호문 c 를 입력받을 때,

- $r \circ | G$ 에 질의되지 않았거나, s 가 H 에 질의되지 않았다.
 $\implies c = \theta \parallel \mathcal{F}_{pk}(\sigma)$ 가 성립하지 않으므로, Reject를 반환한다.
- $r \circ | G$ 에 질의되었고, s 가 H 에 질의되었다.
 $\implies [M]_{\lambda_1} \neq 0^{\lambda_1}$ 이면 Reject를 반환한다.

이는 평문 추출기가 GAME_7 의 복호화 오라클과 동일한 출력을 반환함을 의미하고, 평문 추출기가 복호화 오라클을 완벽히 시뮬레이션 할 수 있음을 의미한다.

7.3 리덕션

다음과 같이 실험을 구성한다.



이 실험은 다음을 만족하는 공격자 \mathcal{B} 가 존재함을 보장한다.

$$\Pr[\text{AskH}_8] \leq \text{Adv}_{\Psi, \lambda}^?(\mathcal{B}; \tau', q_H).$$

따라서, 다음이 성립한다. (유도과정은 이전과 비슷하므로 생략한다)

$$\frac{1}{2} \cdot \text{Adv}_{\Pi, \lambda}^{\text{IND-CCA2}}(\mathcal{A}; \tau, q_D, q_H, q_G) \leq \text{Adv}_{\Psi, \lambda}^?(\mathcal{B}; \tau', q_H) + \frac{q_G + q_D + q_D q_G}{2^{\lambda_0}} + \frac{q_D}{2^{\lambda_1}}.$$

8 ??

논문 5절 Application to RSA-OAEP에는 다음과 같은 내용이 있다.

thanks to the random self-reducibility of RSA, the partial-domain one-wayness of RSA is equivalent to that of the whole RSA problem, as soon as a constant fraction of the most significant bits (or the least significant bits) of the pre-image can be recovered.

이는 RSA의 무작위 자기환원성 덕분인데, RSA의 부분 영역 일방향성은 전체 RSA 문제의 일방향성과 동등하며, 이는 전상 이미지(pre-image)의 가장 상위 비트(또는 가장 하위 비트)의 일정 비율을 복구할 수 있을 때 성립합니다.

본 절에서는 RSA의 부분 영역 일방향성이 전체 RSA 문제의 일방향성과 동등함을 보인다.

메모. 본 논문의 제목은 ‘RSA-OAEP is Secure under the RSA Assumption’이다. 이는 RSA가 OW일 때, RSA-OAEP가 IND-CCA2 안전함을 말하는 제목이다. 그러나 논문의 4절에서는 RSA가 PD-OW일 때를 가정하고 IND-CCA2 안전함을 보인다.

만약 RSA가 OW일 때, RSA는 PD-OW도 만족한다면, OW, PD-OW, IND-CCA2로 연결될 수 있다. 그러나 RSA가 OW일 때, PD-OW를 만족함은 자명하지 않다. (그 역은 자명하다) 논문의 5절에서는 이에 대해 증명하여, OW와 PD-OW가 동등함을 보인다. 그리고 증명하는데 사용하는 기법이 Random Self-Reducibility이다.

8.1 Random Self-Reducibility

여기서는 ‘Rajeev Motwani and Prabhakar Raghavan. Randomized Algorithms. Cambridge University Press, 1995’의 Section 14.4의 내용을 다룬다.

임의의 알고리듬 \mathcal{A} 에 대해, 집합 $C(\mathcal{A})$ 를 다음과 같이 정의한다.

$$C(\mathcal{A}) := \{c \in \mathbb{Z}_n^* : \mathcal{A} \text{ can compute } c^d \bmod n, \text{ given that } \mathcal{A} \text{ knows only } n \text{ and } e\}.$$

정리 1. (아마도 무작위성을 가지는) 다향 시간 알고리듬 \mathcal{A} 이 존재한다고 가정하자. 이때 $|C(\mathcal{A})| \geq \epsilon |\mathbb{Z}_n^*|$ 를 만족하며, 여기서 $\epsilon > 0$ 이다. 그러면 $C(\mathcal{B}) = \mathbb{Z}_n^*$ 을 만족하는 LasVegas 알고리듬 \mathcal{B} 가 존재하며, \mathcal{B} 의 기대 실행 시간은 $\log n$ 과 $1/\epsilon$ 에 대해 다향식이다.

Proof. 임의의 $c \in \mathbb{Z}_n^*$ 를 고정하고, 알고리듬 \mathcal{B} 가 알고리듬 \mathcal{A} 을 블랙박스로 사용하여 $c^d \bmod n$ 을 계산할 수 있음을 보이겠다. 알고리듬 \mathcal{B} 는 다음과 같이 동작한다.

- 알고리듬 \mathcal{B} 는 무작위 원소 $r \in \mathbb{Z}_n^*$ 를 선택하고 $z = cr^e$ 를 계산한다.
- 그런 다음 입력 z 에 대해 알고리듬 \mathcal{A} 를 실행한다.
- \mathcal{A} 로부터 받은 값에 r 의 곱셈 역원을 곱한다.

$z^d = c^d r^{ed} \equiv c^d r \pmod{n}$ 임을 주목하자. \mathcal{B} 는 z^d 로부터 c^d 의 값을 쉽게 유추할 수 있다. 따라서, \mathcal{A} 가 z 에 대해 성공하면 알고리듬 \mathcal{B} 는 성공한다.

우리는 z 가 \mathbb{Z}_n^* 위에서 균등하게 분포한다고 주장하며, 따라서 $z \in C(\mathcal{A})$ 일 확률은 최소한 ϵ 이상이다. 이 주장은 곱셈과 e 제곱 연산이 군 \mathbb{Z}_n^* 에서 일대일 대응이면서 전사 함수(즉, 순열)라는 관찰에서 따라온다. 따라서 무작위 r 에 대해 $z = cr^e$ 역시 \mathbb{Z}_n^* 에서 균등하게 분포한다.

\mathcal{B} 가 ϵ 의 확률로 성공하므로, 독립 반복을 통해 성공 확률을 원하는 수준까지 끌어올릴 수 있다는 것은 자명하다. 또한, 기대 실행 시간이 $\log n$ 및 $1/\epsilon$ 에 대해 다향식인 LasVegas 공격자으로 변환하는 것도 가능하다. \square

메모. 알고리듬 \mathcal{A} 는 암호문의 일부만 복호화 가능하다. 예를 들어, $\epsilon = 1/10$ 이라면, \mathbb{Z}_n^* 의 $1/10$ 영역만 복호화 가능하고, 나머지 $9/10$ 영역은 복호화 불가능하다. 임의의 암호문 $c \in \mathbb{Z}_n^*$ 을 고정하고, 알고리듬 \mathcal{A} 에게 주어진다면, \mathcal{A} 는 복호화를 못할 수 있다.

반면에, 알고리듬 \mathcal{B} 는 어떤 암호문이라도 상관없다. \mathcal{B} 는 어떤 암호문이라도 r 을 무작위로 뽑아, 균등분포를 가지는 새로운 암호문 z 를 만들고, 이를 \mathcal{A} 에게 전달한다. \mathcal{A} 가 복호화할 수 있는 암호문 집합에 z 가 속한다면 (z 는 균등분포를 따르므로, 이 집합에 속할 확률은 ϵ 이다) \mathcal{B} 는 z^d 에 해당하는 값을 받을 것이고, \mathcal{B} 는 z^d 에 r^{-1} 을 곱해 c^d 를 얻을 수 있다.

메모. 암호학에서 *Random Self-Reducibility*는 우리가 생각하는 *Reduction*의 일종으로 보인다.

- *Random*: \mathcal{B} 가 c 를 새로 무작위화 하여 z 를 만든다.
- *Self*: \mathcal{A}, \mathcal{B} 둘 다 RSA 문제를 푸는 공격자이다.
- *Reducibility*: \mathcal{B} 가 \mathcal{A} 를 이용하여 문제를 푼다.

A 보조정리

보조정리 3. E_1, E_2, F_1, F_2 를 하나의 확률 공간 상에 정의된 사건들이라고 하자. 만약

$$\Pr[E_1 \wedge \neg F_1] = \Pr[E_2 \wedge \neg F_2], \quad \Pr[F_1] = \Pr[F_2] = \varepsilon$$

가 성립한다면, 다음을 만족한다.

$$|\Pr[E_1] - \Pr[E_2]| \leq \varepsilon$$

Proof. $|\Pr[E_1] - \Pr[E_2]|$ 은 다음과 같이 표현 가능하다.

$$|\Pr[E_1 \wedge \neg F_1] + \Pr[E_1 \wedge F_1] - \Pr[E_2 \wedge \neg F_2] - \Pr[E_2 \wedge F_2]|.$$

가정에 의해 $\Pr[E_1 \wedge \neg F_1] = \Pr[E_2 \wedge \neg F_2]$ 이므로, 다음과 같이 식을 줄일 수 있다.

$$|\Pr[E_1 \wedge F_1] - \Pr[E_2 \wedge F_2]|.$$

위 식을 조건부 확률로 표현하면 다음과 같다.

$$|\Pr[E_1 | F_1] \cdot \Pr[F_1] - \Pr[E_2 | F_2] \cdot \Pr[F_2]|.$$

가정에 의해 $\Pr[F_1] = \Pr[F_2] = \varepsilon$ 이므로, 위 식은 다음과 같이 표현할 수 있다.

$$\varepsilon \cdot |\Pr[E_1 | F_1] - \Pr[E_2 | F_2]|.$$

어떤 사건의 조건부 확률은 언제나 0 이상 1 이하이므로, $|\Pr[E_1 | F_1] - \Pr[E_2 | F_2]| \leq 1$ 을 만족한다. 따라서 다음을 만족한다.

$$|\Pr[E_1] - \Pr[E_2]| = \varepsilon \cdot |\Pr[E_1 | F_1] - \Pr[E_2 | F_2]| \leq \varepsilon \cdot 1 = \varepsilon.$$

□

메모. 두 사건 E_1, E_2 의 확률이 어떤 Bad 사건 F 가 발생할 때만 차이가 발생한다면, 두 사건의 확률 차이는 Bad 사건의 확률 이하로 조절된다는 의미이다. 암호학에서는, 두 게임 간 차이를 분석할 때 사용할 수 있다. 예를 들어,

- S_1 : GAME₁에서의 성공 사건
- S_2 : GAME₂에서의 성공 사건
- ε : 각각의 게임에서 발생할 수 있는 Bad 사건의 확률

이라고 할 때, 두 게임이 Bad 사건 외에서는 동일하게 동작하면, 두 게임의 성공 확률 차이는 Bad 사건의 확률 이하로 상한된다. 즉, $|\Pr[S_1] - \Pr[S_2]| \leq \varepsilon$ 이다.