
KEM의 이론적 안전성 소개

2026.02.24

Sungshin Women's University

Joohee Lee



INTRODUCTION TO PKE/KEM



BASIC GOALS OF CRYPTOGRAPHY

- 키를 갖는 암호의 종류를 크게 대칭키(Symmetric keys) 암호와 비대칭키(Asymmetric keys) 암호로 나눌 수 있고, 비대칭키 암호는 공개키 암호(public-key cryptography)라고도 함

	Message privacy	Message integrity / authentication
Symmetric keys	Symmetric encryption	Message authentication codes (MAC)
Asymmetric keys	Asymmetric encryption (a.k.a. public-key encryption), Key Encapsulation Mechanism (KEM)	Digital signatures

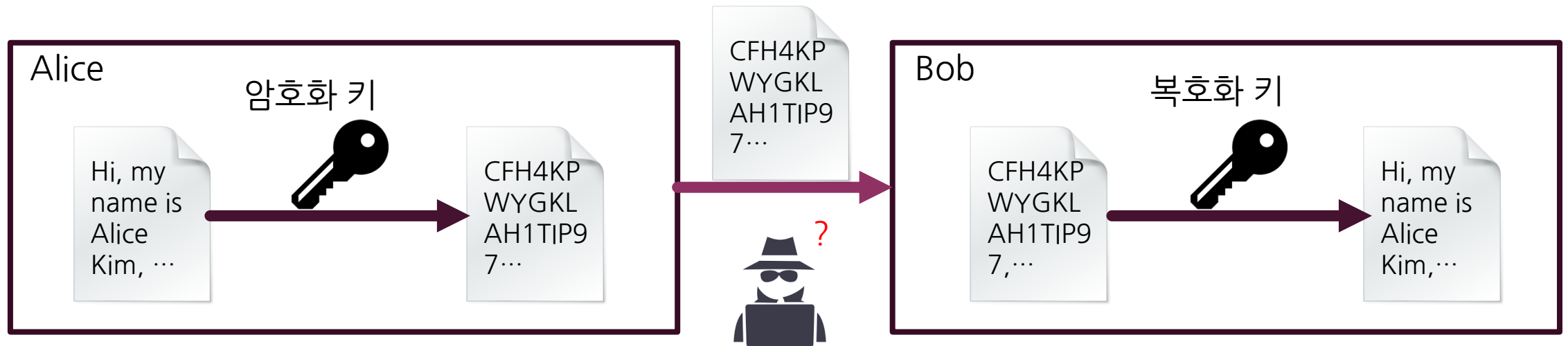
(Key exchange)



BASIC GOALS OF CRYPTOGRAPHY

- 키를 갖는 암호의 종류를 크게 대칭키(Symmetric keys) 암호와 비대칭키(Asymmetric keys) 암호로 나눌 수 있고, 비대칭키 암호는 공개키 암호(public-key cryptography)라고도 함

	Message privacy	Message integrity / authentication
Symmetric keys	Symmetric encryption	Message authentication codes (MAC)
Asymmetric keys	Asymmetric encryption (a.k.a. public-key encryption), Key Encapsulation Mechanism (KEM)	Digital signatures (Key exchange)

SYMMETRIC KEY ENCRYPTION

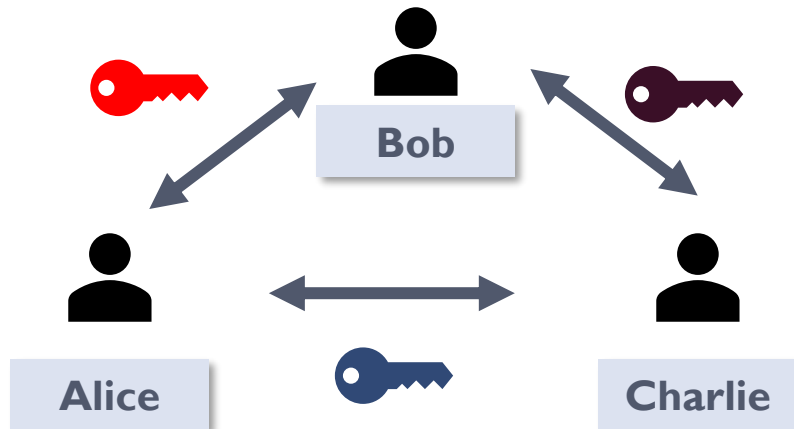


암호화 키  =  복호화 키

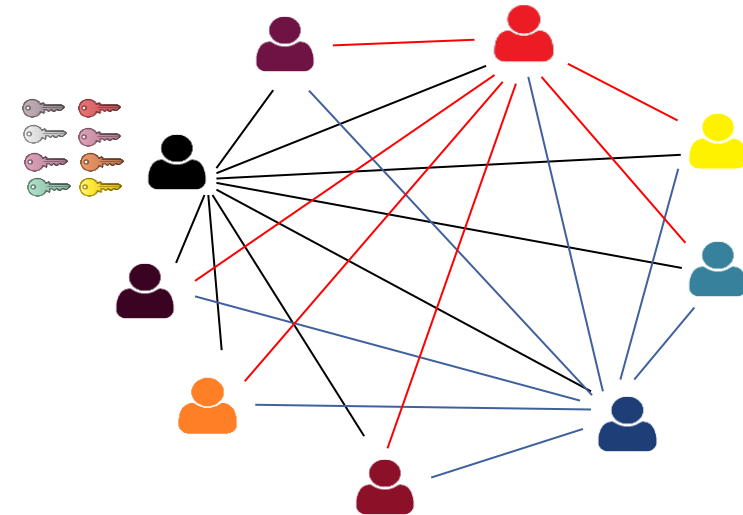
- 장점 : 암호복호화 매우 빠름, 암호문의 크기가 작음
- 예시 : DES/AES와 같은 블록암호, 스트림 암호 등

SYMMETRIC KEY ENCRYPTION

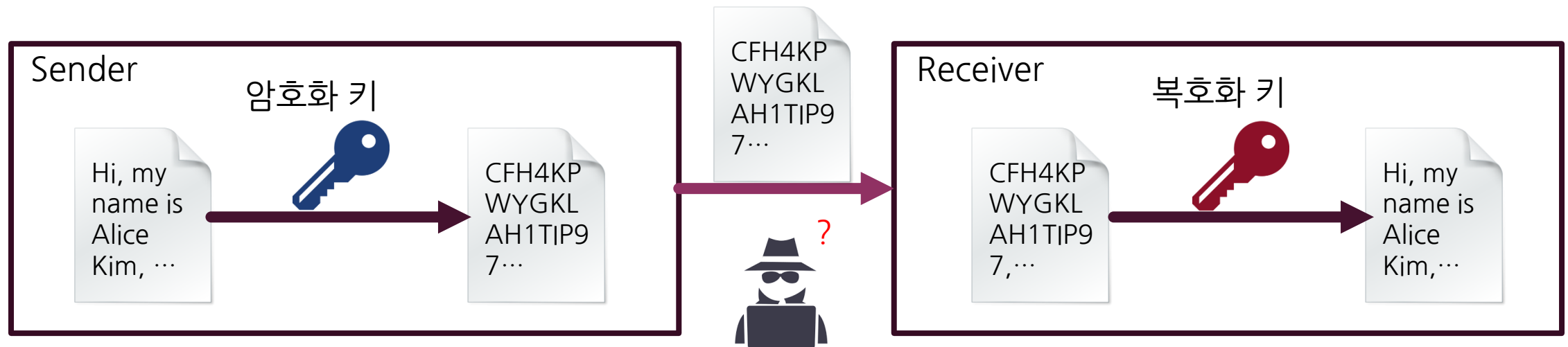
- 키 공유 문제
 - 비밀키를 어떻게 공개된 채널을 통해 **안전하게 공유**할 것인가?
- 키 개수의 문제
 - 사용자의 쌍 끼리 서로 다른 비밀키를 공유해야 한다.
 - 한 시스템에 n 명의 사용자가 있을 경우에 $\frac{n(n-1)}{2} = O(n^2)$ 개의 키가 필요하다
 - 많은 키 들을 안전하게 저장하고 관리하는 것이 어려울 수 있음





$$\Rightarrow \frac{3(3-1)}{2} = 3$$



PUBLIC KEY ENCRYPTION (A.K.A. ASYMMETRIC ENCRYPTION, 1976 ~)



암호화 키  ≠  복호화 키

- 암호화 키 = 공개키, 복호화 키 = 비밀키
- 예시 : RSA, ElGamal Encryption, 격자 기반 공개키 암호 등
- 장점 : 암호화 키를 공개할 수 있어 비밀키를 미리 공유하지 않아도 됨. 오늘날 인터넷 프로토콜의 초석이 됨
- 단점 : (대칭키 암호에 비해) 느림, 공개키 및 암호문 크기가 큼

HISTORY OF PUBLIC KEY CRYPTOGRAPHY

- 1976 : Whitfield Diffie와 Martine Hellman에 의해 Diffie-Hellman 키 교환 개발
 - Public-Key Cryptography의 시작
- 1977 : Ron Rivest, Adi Shamir, Leonard Adelman에 의해 RSA 제안
- 1980 : Miller Koblitz 가 Elliptic Curve Cryptography 제안
- 현재 : Post-quantum cryptography

공개키 암호화 (SYNTAX)

공개키 암호화(public-key encryption, PKE) 스킴 Σ 는 KeyGen, Enc, Dec 세 알고리즘들의 집합

$$\Rightarrow \Sigma = (\text{KeyGen}, \text{Enc}, \text{Dec})$$

$$(sk, pk) \xleftarrow{\$} \text{KeyGen}$$

$$\text{Enc} : \mathcal{PK} \times \mathcal{M} \xrightarrow{\$} \mathcal{C}$$

$$\text{Dec} : \mathcal{SK} \times \mathcal{C} \rightarrow \mathcal{M} \cup \{\perp\}$$

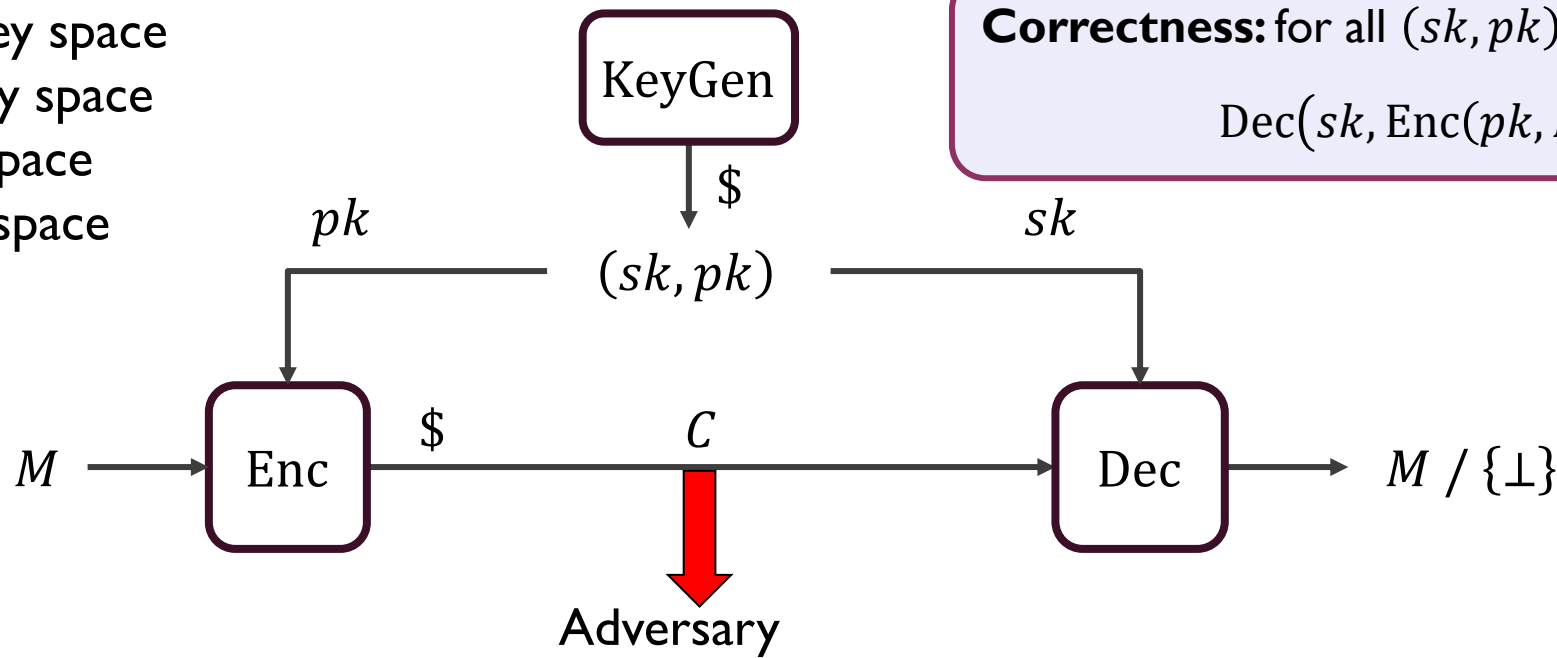
$$\text{Enc}(pk, M) = \text{Enc}_{pk}(M) = C$$

$$\text{Dec}(sk, C) = \text{Dec}_{sk}(C) = M / \perp$$

- \mathcal{SK} – private key space
- \mathcal{PK} – public key space
- \mathcal{M} – message space
- \mathcal{C} – ciphertext space

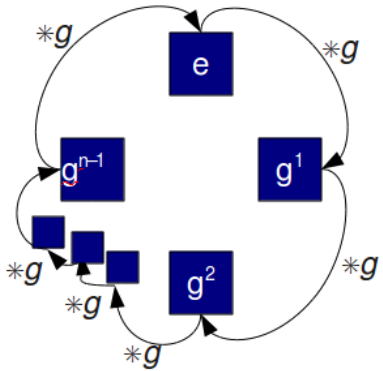
Correctness: for all $(sk, pk) \leftarrow \text{KeyGen}$:

$$\text{Dec}(sk, \text{Enc}(pk, M)) = M$$



공개키 암호화 예시 : ELGAMAL ENCRYPTION

- $|G|$: # of elements in G



$$G = \langle g \rangle = \{g, g^2, g^3, \dots, g^n\}$$

B

←

$ct = (A, C)$

→

KeyGen

1. $sk = b \xleftarrow{\$} \{1, \dots, |G|\}$
2. $pk = B \leftarrow g^b$
3. **return** (sk, pk)



Enc($pk, M \in G$)

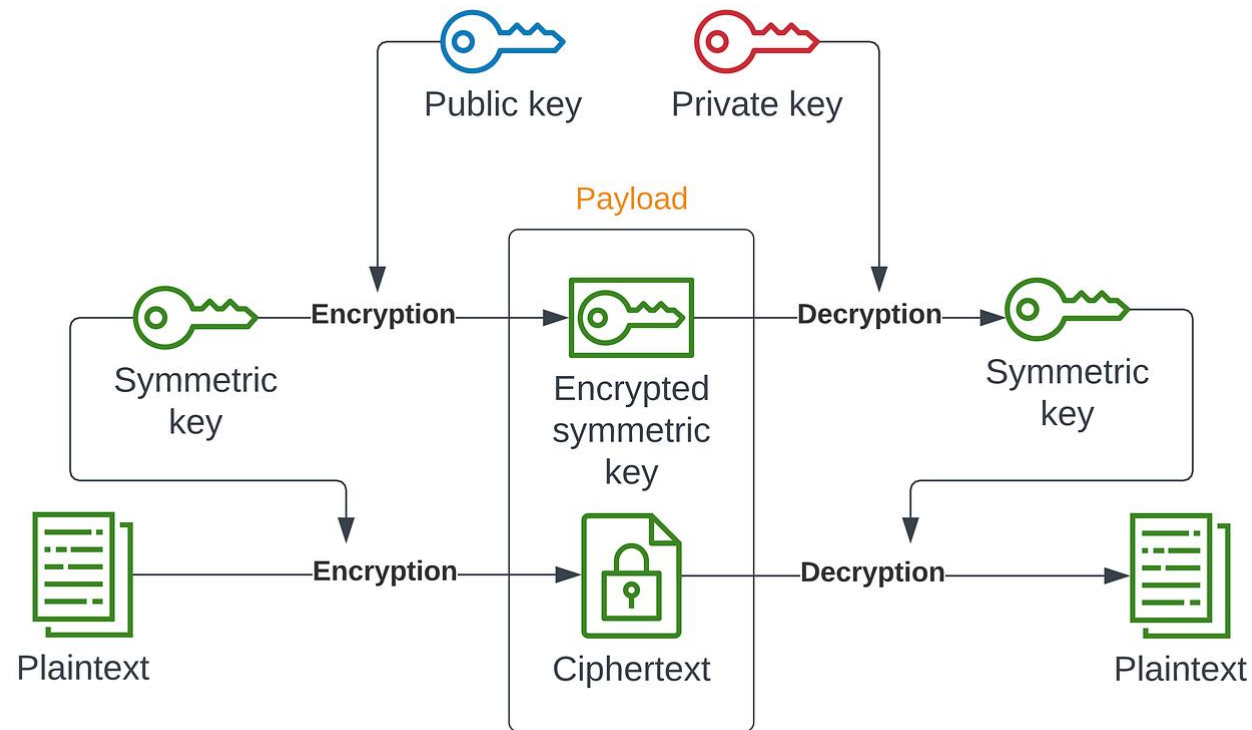
1. $a \xleftarrow{\$} \{1, \dots, |G|\}$
2. $A \leftarrow g^a$
3. $Z \leftarrow B^a = g^{ab}$
4. $C \leftarrow M \cdot Z$
5. **return** $ct = (A, C)$

Dec(sk, C)

1. $Z \leftarrow A^b = g^{ab}$
2. $M \leftarrow C \cdot Z^{-1}$
3. **return** M

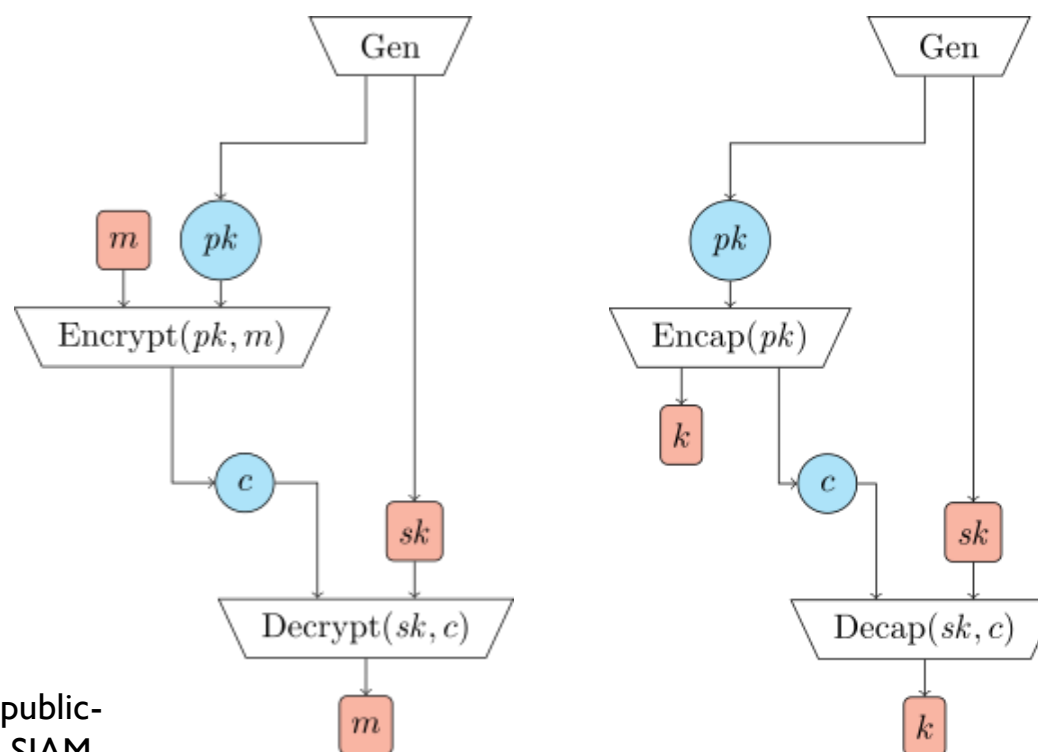
HYBRID ENCRYPTION

- Combines the convenience of a public-key encryption with the efficiency of a symmetric-key encryption



KEY ENCAPSULATION MECHANISM

- Key Encapsulation Mechanism (KEM) ; first formalized by Cramer and Shoup in 2003 [CS03], targeted for the hybrid encryption
- $\text{PKE} \leftrightarrow \text{KEM}$
 - PKE로부터 KEM을 만들 수 있다. (how?)
 - KEM으로부터 PKE를 만들 수 있다. (how?)



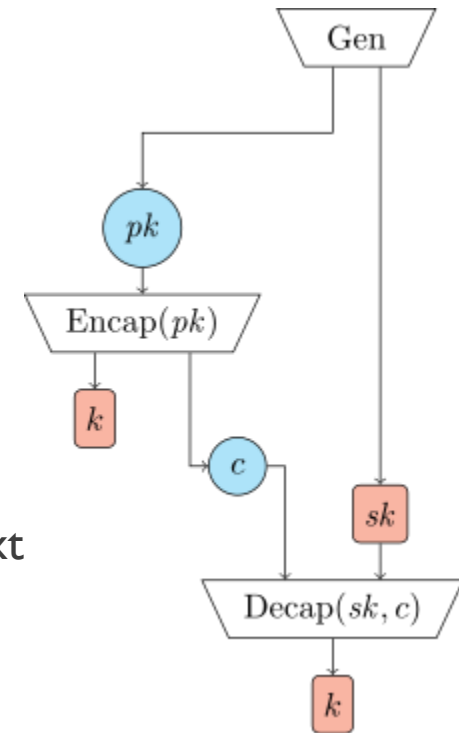
[CS03] Ronald Cramer and Victor Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. SIAM Journal on Computing, 33(1):167–226, 2003.

PKE vs. KEM

SYNTAX OF KEY ENCAPSULATION MECHANISM

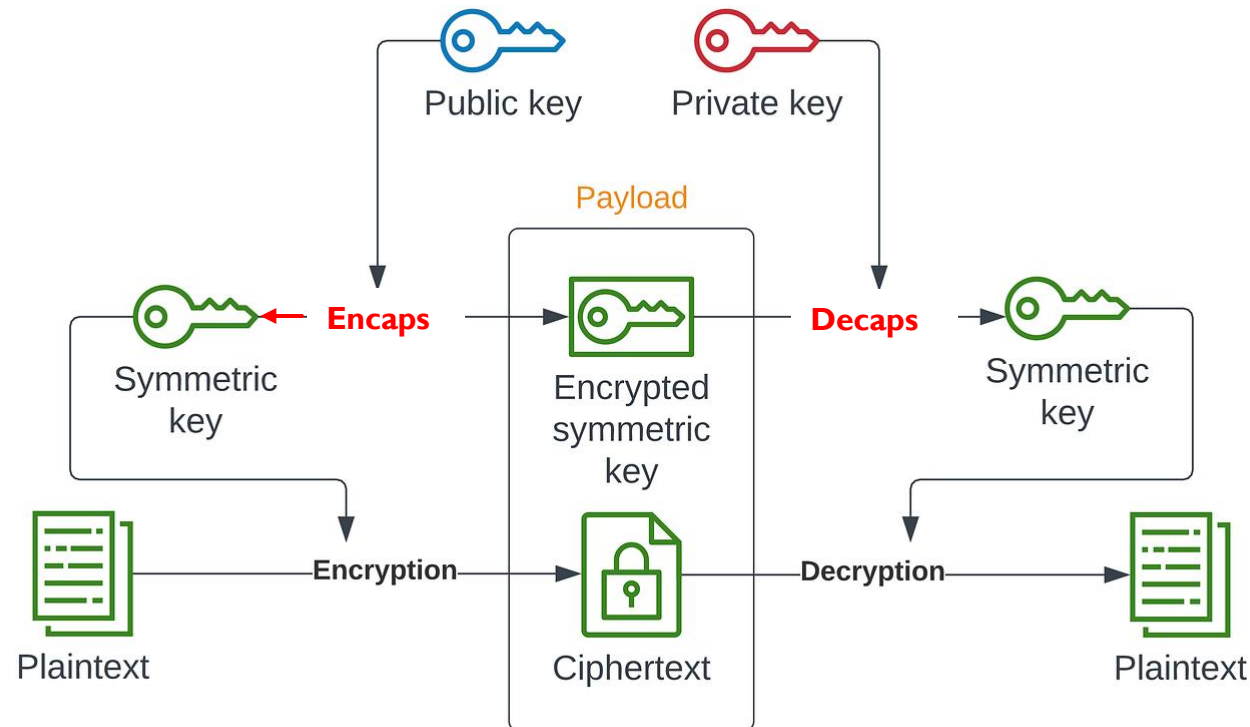
Key Encapsulation Mechanism (KEM) consists of the following three algorithms;

- Key Generation: generates an encapsulation key (public key) and a decapsulation key (private key)
- Encapsulation: takes public key (pk) as an input to generate a secret key K and a ciphertext C
- Decapsulation: takes private key (sk) and ciphertext C as inputs, and recover K from the ciphertext



HYBRID ENCRYPTION USING KEM

- Messages are encrypted using symmetric key encryption (as in the case using PKE)
- The secret key for the symmetric key encryption together with the ciphertext encapsulating it is an **output of Encapsulation** algorithm



KEM 예시

$$G = \langle g \rangle = \{g, g^2, g^3, \dots\}$$



KeyGen

1. $sk = b \xleftarrow{\$} \{1, \dots, |G|\}$
2. $pk = B \leftarrow g^b$
3. **return** (sk, pk)

B



$ct = (A, C)$



Enc(pk)

1. $K \xleftarrow{\$} G$
2. $a \xleftarrow{\$} \{1, \dots, |G|\}$
3. $A \leftarrow g^a$
4. $Z \leftarrow B^a = g^{ab}$
5. $C \leftarrow K \cdot Z$
6. **return** $(K, ct = (A, C))$

Dec(sk, C)

1. $Z \leftarrow A^b = g^{ab}$
2. $K \leftarrow C \cdot Z^{-1}$
3. **return** K



SECURITY MODELS OF PKE/KEM



SECURITY DEFINITIONS

- **Confidentiality:** an adversary cannot read plaintexts or keys (from ciphertexts).
- This statement is not clear enough, and does not reflect the prior knowledge;
 - What if an adversary can read the first half of Alice's message, but not the second half?
 - What if an adversary already knows that Alice's message starts with "Dear Bob,"?

SECURITY DEFINITIONS

- **Confidentiality**: an adversary cannot read plaintexts or keys (from ciphertexts).
- It is better to define it with the following game(s)
- OW-CPA(One-Wayness under Chosen Plaintext Attack) game (for PKE);
 1. Alice generates a key pair (pk, sk) of PKE
 2. Alice chooses random message m , and encrypts it, and sends the ciphertext to Eve
 3. Eve reads the ciphertext and tries to guess which message was sent
 4. If the probability that Eve correctly guesses which message was sent is 0 (or close to 0), then the encryption scheme is confidential

SECURITY DEFINITIONS



Alice (Challenger)



Eve (Adversary)

$(sk, pk) \leftarrow \text{KeyGen}(1^\lambda)$ \xrightarrow{pk}

$M \leftarrow \mathcal{M}, C \leftarrow \text{Enc}(pk, M)$ \xrightarrow{C}

Outputs $b = 1$ if $M = M'$
and $b = 0$ otherwise $\xleftarrow{M'}$

PKE is OW-CPA secure if and only if $\text{Adv}_{\text{PKE}}^{\text{OW-CPA}} := |\Pr[\text{Alice outputs } 1]|$ is negligible

SECURITY DEFINITIONS

- **Confidentiality:** an adversary cannot read plaintexts or keys (from ciphertexts).
- It is better to define it with the following game(s)
- OW-CPA game (for KEM);
 1. Alice generates a key pair (pk, sk) of KEM
 2. Alice generates (K, C) from the encapsulation algorithm, and sends C to Eve
 3. Eve reads the ciphertext and tries to guess which key was sent (outputs K')
 4. If the probability that Eve correctly guesses which key was sent is 0 (or close to 0), then the KEM is confidential

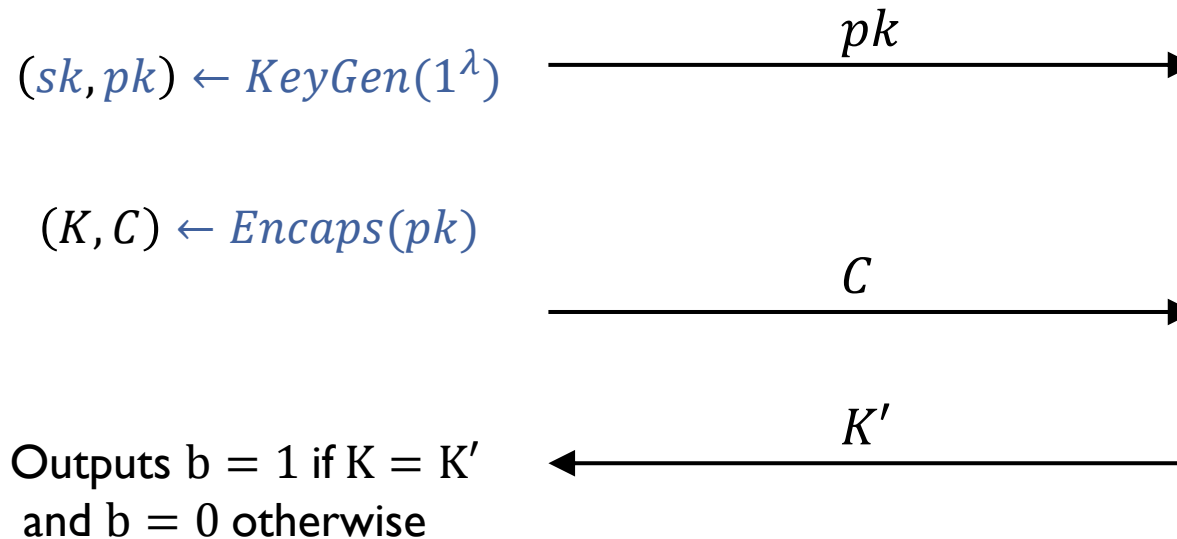
SECURITY DEFINITIONS



Alice (Challenger)



Eve (Adversary)



KEM is OW-CPA secure if and only if $\text{Adv}_{\text{KEM}}^{\text{OW-CPA}} := |\Pr[\text{Alice outputs } 1]|$ is negligible

SECURITY DEFINITIONS

- **Confidentiality**: an adversary cannot read plaintexts or keys (from ciphertexts).
- It is better to define it with the following game
- IND-CPA(INDistinguishability under Chosen Plaintext Attack) game for PKE ;
 1. Alice generates a key pair (pk, sk) of PKE
 2. Eve chooses two messages m_0, m_1 of the same length
 3. Alice chooses one message at random m_b , encrypts it, and sends the ciphertext
 4. Eve reads the ciphertext and tries to guess which message was sent
 - Eve knows either m_0 or m_1 was sent, but doesn't know which
 5. If the probability that Eve correctly guesses which message was sent is $1/2$, then the encryption scheme is confidential

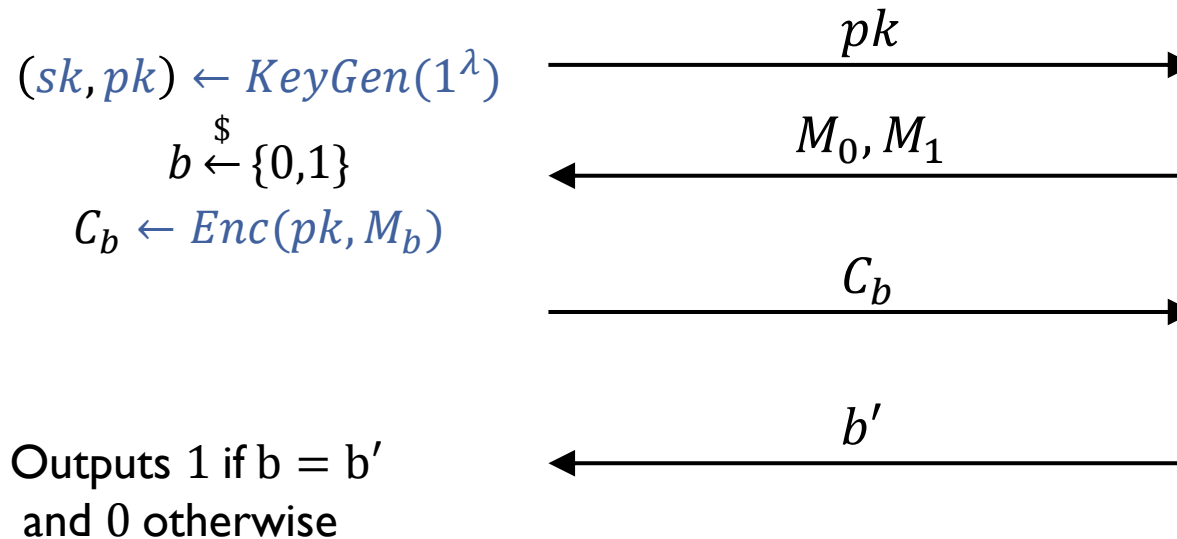
SECURITY DEFINITIONS



Alice (Challenger)



Eve (Adversary)



PKE is IND-CPA secure if and only if $\text{Adv}_{\text{PKE}}^{\text{IND-CPA}} := \left| \Pr[\text{Alice outputs 1}] - \frac{1}{2} \right|$ is negligible

SECURITY DEFINITIONS

- **Confidentiality**: an adversary cannot read plaintexts or keys (from ciphertexts).
- It is better to define it with the following game
- IND-CPA(INDistinguishability under Chosen Plaintext Attack) game for KEM ;
 1. Alice generates a key pair (pk, sk) of KEM
 2. Alice generates (K_0, C) from the encapsulation algorithm, samples $K_1 \xleftarrow{\$} \mathcal{K}$, and sends (K_b, C) to Eve
 3. Eve reads the ciphertext and tries to guess which message was sent
 4. If the probability that Eve correctly guesses which message was sent is $1/2$, then the KEM is confidential

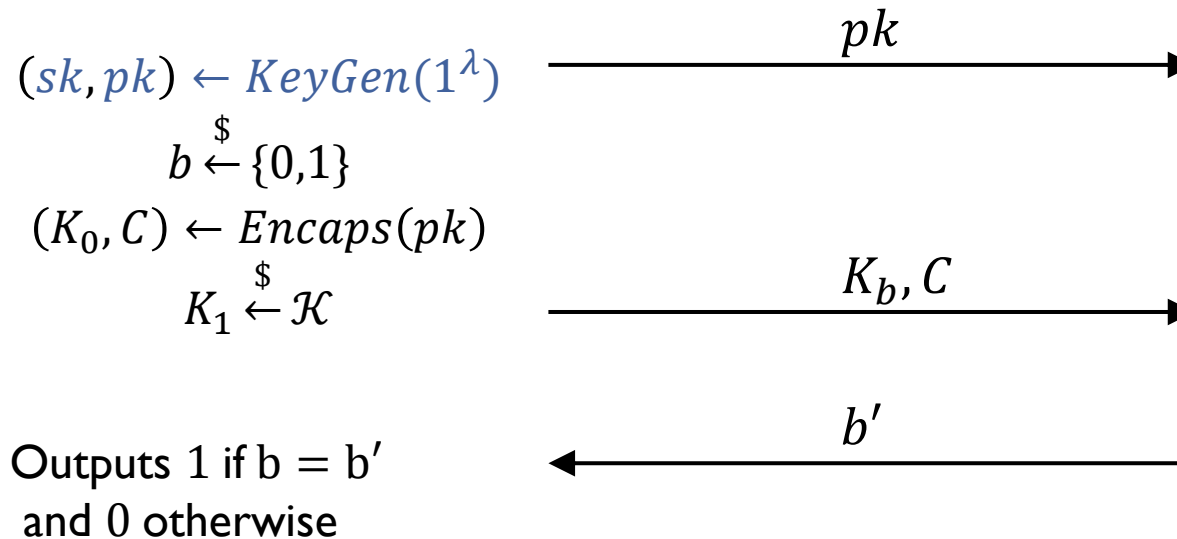
SECURITY DEFINITIONS



Alice (Challenger)



Eve (Adversary)



KEM is IND-CPA secure if and only if $\text{Adv}_{\text{KEM}}^{\text{IND-CPA}} := \left| \Pr[\text{Alice outputs 1}] - \frac{1}{2} \right|$ is negligible

SECURITY DEFINITIONS

- **Confidentiality:** an adversary cannot read plaintexts or keys (from ciphertexts).
- It is better to define it with the following game
- IND-CCA(INDistinguishability under Chosen Ciphertext Attack) game for PKE ;
 1. Alice generates a key pair (pk, sk) of PKE
 2. Eve sends ciphertexts to Decryption oracle, and gets answers from it
 3. Eve chooses two messages m_0, m_1 of the same length
 4. Alice chooses one message at random m_b , encrypts it, and sends the ciphertext
 5. Eve reads the ciphertext and sends ciphertexts (other than the challenge ciphertext) to Decryption oracle, and gets answers from it
 6. Eve tries to guess which message was sent
 - Eve knows either m_0 or m_1 was sent, but doesn't know which
 7. If the probability that Eve correctly guesses which message was sent is $1/2$, then the encryption scheme is confidential

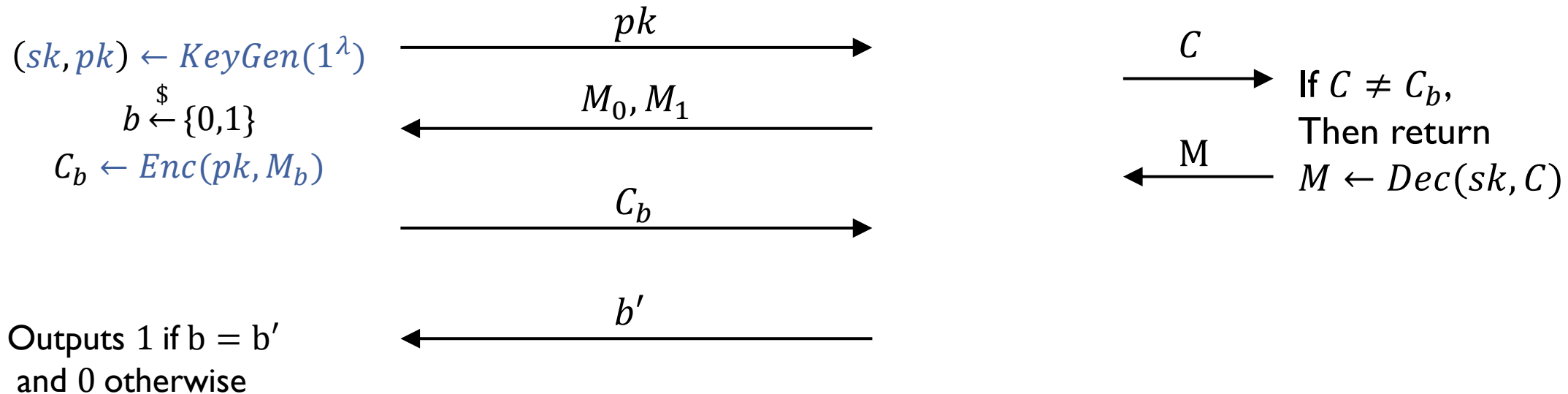
SECURITY DEFINITIONS



Alice (Challenger)



Eve (Adversary)



PKE is IND-CCA secure if and only if $\text{Adv}_{PKE}^{\text{IND-CCA}} := \left| \Pr[\text{Alice outputs 1}] - \frac{1}{2} \right|$ is negligible

SECURITY DEFINITIONS

- **Confidentiality**: an adversary cannot read plaintexts or keys (from ciphertexts).
- It is better to define it with the following game
- IND-CPA(INDistinguishability under Chosen Plaintext Attack) game for KEM ;
 1. Alice generates a key pair (pk, sk) of KEM
 2. Eve sends ciphertexts to Decapsulation oracle, and gets answers from it
 3. Eve generates (K_0, C) from the encapsulation algorithm, samples $K_1 \xleftarrow{\$} \mathcal{K}$, and sends (K_b, C) to Eve
 4. Eve reads the ciphertext and sends ciphertexts (other than the challenge ciphertext) to Decapsulation oracle, and gets answers from it
 5. Eve tries to guess which message was sent
 6. If the probability that Eve correctly guesses which message was sent is $1/2$, then the KEM is confidential

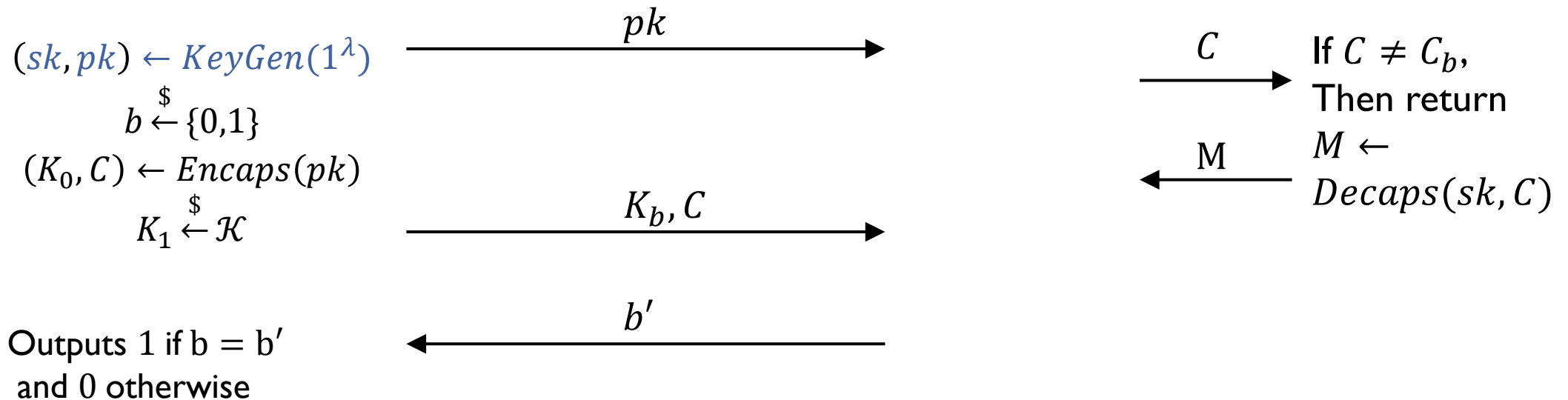
SECURITY DEFINITIONS



Alice (Challenger)

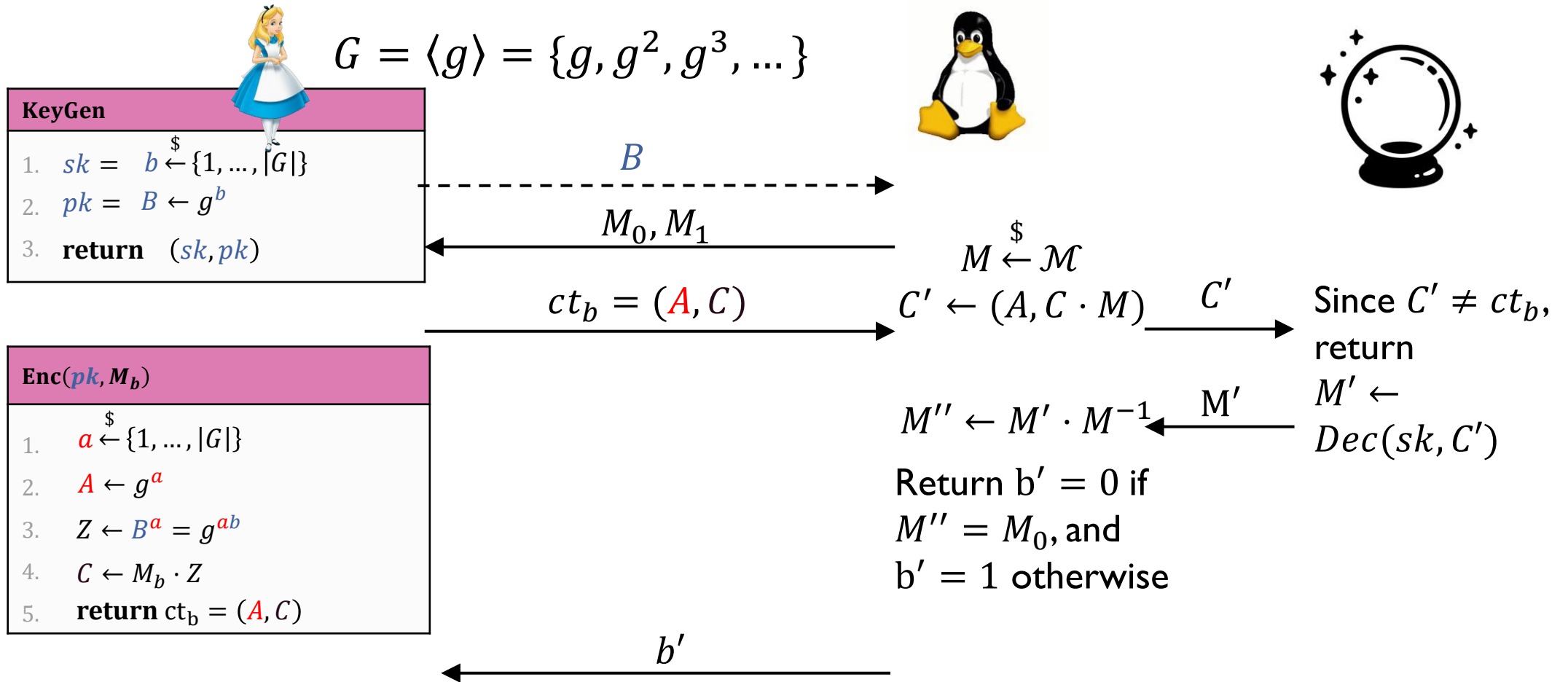


Eve (Adversary)

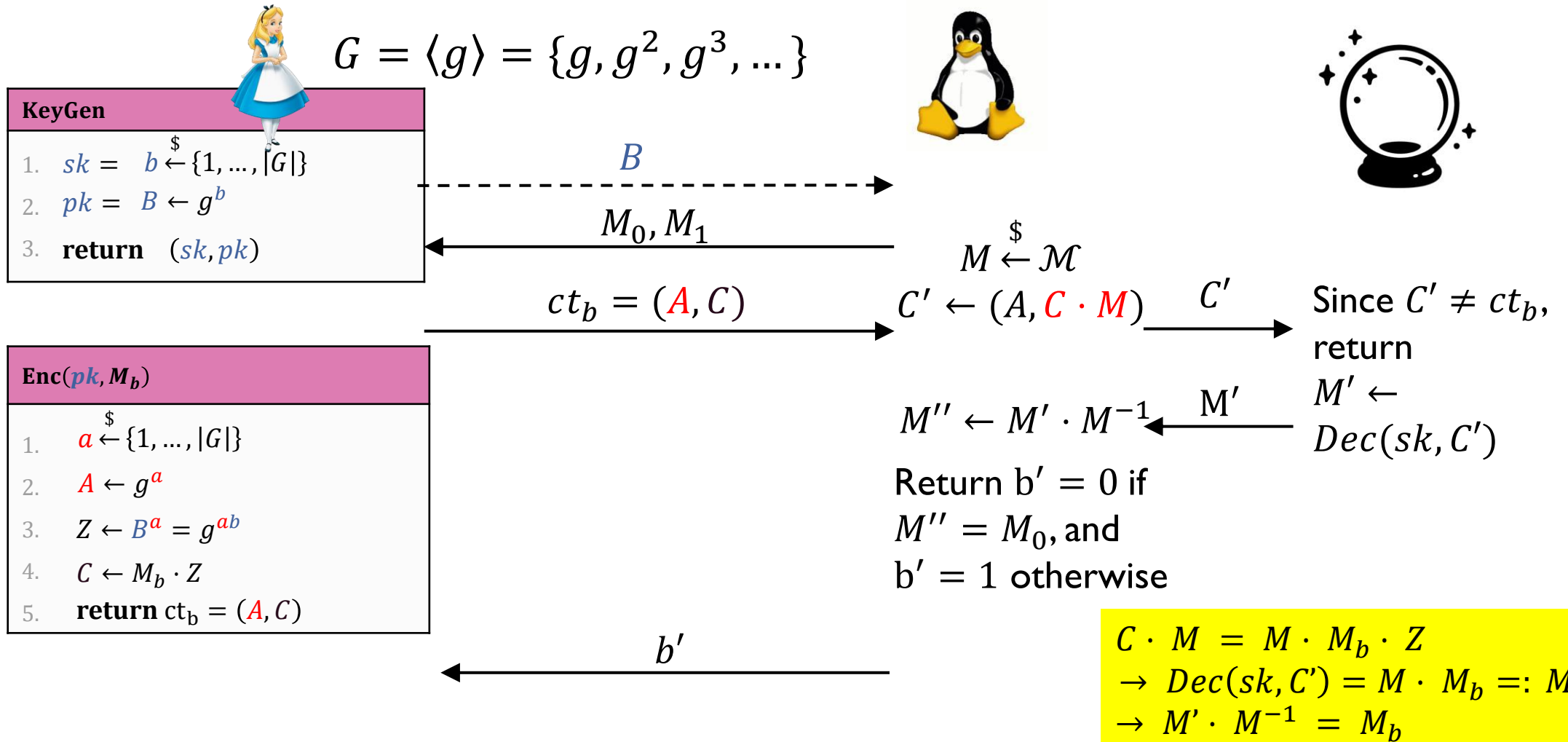


KEM is IND-CCA secure if and only if $\text{Adv}_{\text{KEM}}^{\text{IND-CCA}} := \left| \Pr[\text{Alice outputs 1}] - \frac{1}{2} \right|$ is negligible

공개키 암호화 예시 : ELGAMAL ENCRYPTION (WHY IT IS NOT IND-CCA SECURE)



공개키 암호화 예시 : ELGAMAL ENCRYPTION (WHY IT IS NOT IND-CCA SECURE)



SECURITY DEFINITIONS (REMARKS)

- Remark I. $\text{OW-CPA} < \text{IND-CPA} < \text{IND-CCA}$
 - If a PKE/KEM is IND-CPA secure, then it is OW-CPA secure
 - If a PKE/KEM is IND-CCA secure, then it is IND-CPA secure
- Remark II. Hybrid encryption in the previous slides requires IND-CCA secure schemes
 - IND-CCA security is required to **protect encrypted data against active attackers** who can **intercept, modify, and submit manipulated ciphertexts for decryption** to gain information
 - While IND-CPA secure systems only protect against eavesdropping, IND-CCA guarantees privacy even when attackers can actively interact with the decryption device/server
 - In systems like SSL/TLS where the hybrid encryption is applied, an adversary can send manipulated ciphertexts to a server. If the system is only IND-CPA secure, the server's error responses to these manipulated ciphertexts might allow the adversary to decrypt the original messages



FUJISAKI-OKAMOTO TRANSFORM



FUJISAKI-OKAMOTO TRANSFORM (HIGH LEVEL)

- To construct INDistinguishable under Chosen Ciphertext Attack (IND-CCA) KEM from OW-CPA/IND-CPA PKE, many schemes use the well-known Fujisaki-Okamoto Transform.
 - Encaps(pk);
 - Generate a random message M
 - Generate a randomness r , output key K from the hash value of M
 - Compute an output ciphertext C as an encryption of M with randomness r
 - Decapsulation involves decryption (to achieve M) and re-encryption from M to check if correct encryption randomness is used
- Intuition : It makes transforming ciphertexts difficult (some limitation for ciphertext generation)
- The resulting IND-CCA KEM is proven to be secure in the quantum random oracle model.
 - That is, it is IND-CCA secure against a quantum adversary who can make both classical and quantum queries to the hash functions.
 - [BHH+'19] Tighter Proofs of CCA Security in the Quantum Random Oracle Model, TCC 2019
 - **[HHM22] Failing gracefully: Decryption failures and the Fujisaki-Okamoto transform, Asiacrypt 2022**
 - [HHMS25] (Un)breakable Curses - Re-encryption in the Fujisaki-Okamoto Transform, Eurocrypt 2025

FUJISAKI-OKAMOTO TRANSFORM (ALGORITHMS)

- G, H : hash functions
- Step I. Construct a Deterministic Encryption PKE^G from OW-CPA/IND-CPA PKE PKE
 - $Enc^G(pk, m)$:
 - $c \leftarrow Enc(pk, m; G(m))$
 - Return c
 - $Dec^G(sk, c)$:
 - $m' := Dec(sk, c)$
 - If $m' = \perp$ or $c \neq Enc(pk, m'; G(m'))$
Return \perp
 - Else
Return m'
- Step II. Construct an IND-CCA KEM KEM from PKE^G and H
 - $Encaps(pk)$:
 - $m \leftarrow \mathcal{M}$
 - $c \leftarrow Enc(pk, m; G(m))$
 - $K \leftarrow H(m)$
 - Return (K, c)
 - $Decaps(sk, c)$:
 - $m' := Dec(sk, c)$
 - If $m' = \perp$ or $c \neq Enc(pk, m'; G(m'))$
Return \perp
 - Else
Return $K := H(m')$

FUJISAKI-OKAMOTO TRANSFORM (ALGORITHMS)

- G, H : hash functions
- Step I. Construct a Deterministic Encryption PKE^G from OW-CPA/IND-CPA PKE PKE

$Enc^G(pk, m)$:

- $c \leftarrow Enc(pk, m; G(m))$
- Return c

$Dec^G(sk, c)$:

- $m' := Dec(sk, c)$
- If $m' = \perp$ or $c \neq Enc(pk, m'; G(m'))$
Return \perp
- Else
Return m'

- Step II. Construct an IND-CCA KEM KEM from PKE^G and H

$Encaps(pk)$:

- $m \leftarrow \mathcal{M}$
- $c \leftarrow Enc(pk, m; G(m))$
- $K \leftarrow H(m)$
- Return (K, c)

$Decaps(sk, c)$:

- $m' := Dec(sk, c)$
- If $m' = \perp$ or $c \neq Enc(pk, m'; G(m'))$
Return \perp
- Else
Return $K := H(m')$

Note. If an adversary slightly modify a ciphertext c , then it would be no longer valid with high probability (so the example attack could not be launched successfully)



LEARNING WITH ERRORS



BASE RINGS

- Z_q
 - It is defined as $\left(-\frac{q}{2}, \frac{q}{2}\right] \cap \mathbb{Z}$
 - E.g. $Z_5 = \{-2, -1, 0, 1, 2\}$; $2 + 2 = 4 = -1 \bmod 5$
- $R_q = \mathbb{Z}[x]/(x^n + 1, q)$
 - $\{\sum_{i=0}^{n-1} c_i \cdot x^i : c_i \in \mathbb{Z}_q\}$
 - E.g. $n = 3$; $R_5 = \mathbb{Z}[x]/(x^3 + 1, 5) = \{\sum_{i=0}^2 c_i \cdot x^i : c_i \in \mathbb{Z}_5\}$;
 $(x + 1)(x^2 + 1) = x^3 + x^2 + x + 1 = x^2 + x \bmod (x^3 + 1, 5)$

SOLVING A LINEAR EQUATION SYSTEM

• Q.

1	3	-3
4	5	-3
-4	-4	-1
2	-3	3
3	-2	-3
5	4	2
1	0	5
4	5	3

\cap
 $\mathbb{Z}_{10}^{8 \times 3}$

x_1
x_2
x_3

=

-3
-1
2
-1
-4
-2
2
7

(mod 10)



Find

x_1
x_2
x_3

!

; Easy!

(We can solve it by using
Gaussian elimination)

LEARNING WITH ERRORS (LWE) PROBLEM

• Q.

1	3	-3
4	5	-3
-4	-4	-1
2	-3	3
3	-2	-3
5	4	2
1	0	5
4	5	3

$\mathbb{Z}_{10}^{8 \times 3}$

x_1
 x_2
 x_3

$+$

0
2
-1
1
0
1
0
-2

Small Error (unknown)

$=$

-3
-1
2
-1
-4
-2
2
7

(mod 10)

\rightarrow Find

x_1
x_2
x_3

 !

; Hard!

DECISION-LWE PROBLEM

- Q. Distinguish

1	3	7
4	5	7
6	6	9
2	7	3
3	8	7
5	4	2
1	0	5
4	5	3

,

7
1
1
0
6
0
2
5

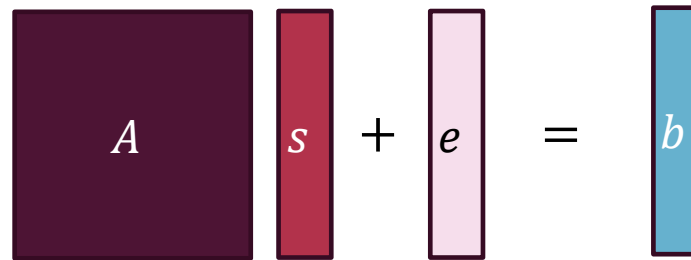
(mod 10)

From a uniform
random sample in $\mathbb{Z}_{10}^{8 \times 4}$

; Hard!

MODULE-LWE (MLWE)

- $A \in R_q^{m \times n}$, $s \in R_q^n$, $e \in R_q^m$;


$$A \cdot s + e = b$$



Find $s \in R_q^n$!

- The above problem is known as the (search-)Module LWE (MLWE) problem.
- The decision-MLWE is defined analogously
- E.g., $q = 7, n = 2$;

$$A = \begin{bmatrix} 6x+3 & x+1 \\ 3x+4 & 2x+5 \end{bmatrix}, s = \begin{bmatrix} x+3 \\ 4x+2 \end{bmatrix}, e = \begin{bmatrix} x+1 \\ 6x \end{bmatrix};$$

$$b = A \cdot s + e \pmod{7, x^2 + 1} = \begin{bmatrix} 10x^2 + 28x + 12 \\ 11x^2 + 43x + 22 \end{bmatrix} \pmod{7, x^2 + 1} = \begin{bmatrix} 2 \\ x-3 \end{bmatrix}$$



LWE-BASED ENCRYPTION



ROUNDING

- q ; prime, $x \in \mathbb{Z}_q$
- $x' = x \bmod^+ q$

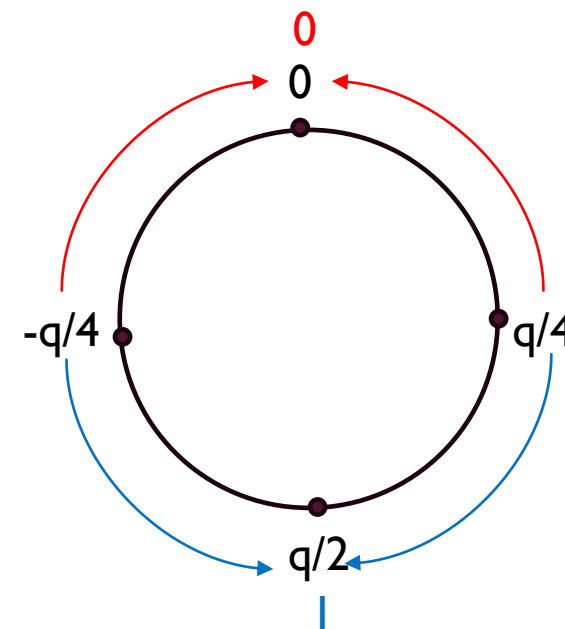
$$\text{Round}_q(x) = \begin{cases} 0, & \text{if } -\frac{q}{4} < x' < \frac{q}{4} \\ 1, & \text{otherwise} \end{cases}$$

- E.g., $q = 11$;

$$\text{Round}_q(x) = \begin{cases} 0, & \text{if } -2 \leq x' \leq 2 \\ 1, & \text{otherwise} \end{cases}$$

- E.g., $q = 3329$;

$$\text{Round}_q(x) = \begin{cases} 0, & \text{if } -832 \leq x' \leq 832 \\ 1, & \text{otherwise} \end{cases}$$



LWE + LWE [LPI I]

$KeyGen(1^\lambda);$ $pk: \begin{matrix} m \\ \left\{ \begin{matrix} \boxed{A} \end{matrix} \right\} \\ n \end{matrix}, \quad \begin{matrix} \boxed{b} \end{matrix} = \begin{matrix} \boxed{A} \end{matrix} \begin{matrix} \boxed{s} \end{matrix} + \begin{matrix} \boxed{e} \end{matrix} \quad sk: \begin{matrix} \boxed{s} \end{matrix}$

$Enc(pk, M \in \{0,1\});$ $\begin{matrix} \boxed{r^T} \end{matrix} \begin{matrix} \boxed{A} \end{matrix} + \begin{matrix} \boxed{e_0} \end{matrix}, \quad \begin{matrix} \boxed{r^T} \end{matrix} \begin{matrix} \boxed{b} \end{matrix} + \begin{matrix} \boxed{M \cdot \lfloor q/2 \rfloor} \end{matrix} + \begin{matrix} \boxed{e_1} \end{matrix}$

$Dec(sk, c = (c_0, c_1));$ i) Compute $d := c_1 - \langle c_0, s \rangle \bmod q$
 ii) Output $Round_q(d)$

- each coefficient of e, e_0, e_1 is drawn from discrete Gaussian distribution



SECURITY PROOF (SKETCH)

- The encryption can be rewritten as $\begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} A^T \\ b^T \end{bmatrix} \cdot r + \begin{bmatrix} e_0 \\ e_1 \end{bmatrix} + \begin{bmatrix} 0 \\ \left\lfloor \frac{q}{2} \right\rfloor \cdot M \end{bmatrix}$
- By the (decision-)LWE assumption, $\begin{bmatrix} A^T \\ b^T \end{bmatrix}$ is indistinguishable from uniform random.
- Also, by the (decision-)LWE assumption, $\left(\begin{bmatrix} A^T \\ b^T \end{bmatrix}, \begin{bmatrix} A^T \\ b^T \end{bmatrix} \cdot r + \begin{bmatrix} e_0 \\ e_1 \end{bmatrix} \right)$ is indistinguishable from uniform random.
- Hence, from the adversary's point of view, $\begin{bmatrix} c_0 \\ c_1 \end{bmatrix}$ is the sum of the random element in Z_q^{n+1} and $\begin{bmatrix} 0 \\ \left\lfloor \frac{q}{2} \right\rfloor \cdot M \end{bmatrix}$.
- This means that the adversary cannot learn any information about the message M
 - the scheme is IND-CPA secure, and not IND-CCA secure



CRYSTALS-KYBER (ML-KEM)



KYBER PUBLIC-KEY ENCRYPTION (SIMPLIFIED)

- $n = 256, q = 3329, k = 3;$

- $\text{KeyGen}(1^\lambda);$

$$A \in R_q^{k \times k} \quad A \cdot s + e = b$$

$$pk = (A, b) \in R_q^{k \times k} \times R_q^k,$$

$$sk = s \in R_q^k$$

- $\text{Enc}(pk, M \in \{0,1\}^n);$
-
- $$r^T A + e_0^T + r^T b + e_1^T + M \cdot [q/2]$$

- $\text{Dec}(sk, c = (c_0, c_1)); M \leftarrow \text{Round}_q(c_1 - \langle c_0, s \rangle \bmod q)$

EXAMPLE

- E.g., $q = 13, n = 2, k = 2; \frac{q}{4} = 3.25$

- $\text{KeyGen}(1^\lambda);$

$$A = \begin{bmatrix} 6x + 4 & x + 12 \\ 3x + 7 & 2x + 5 \end{bmatrix}, s = \begin{bmatrix} x + 2 \\ -x + 1 \end{bmatrix}, e = \begin{bmatrix} x + 1 \\ -x \end{bmatrix};$$

$$b = A \cdot s + e \pmod{13, x^2 + 1} = \begin{bmatrix} 5x^2 + 6x + 21 \\ x^2 + 9x + 19 \end{bmatrix} \pmod{13, x^2 + 1} = \begin{bmatrix} 6x + 3 \\ 9x + 5 \end{bmatrix}$$

- $\text{Enc}(pk, M = 11_{(2)});$ convert $M = 11_{(2)} \rightarrow x + 1;$

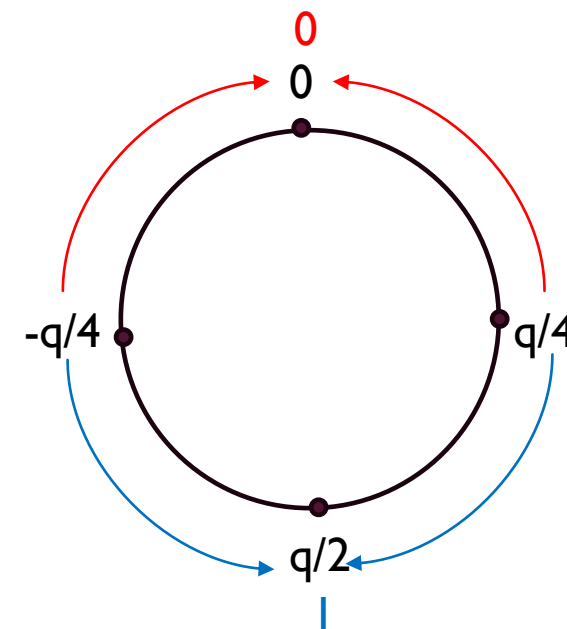
$$r = \begin{bmatrix} 2x + 1 \\ -x + 2 \end{bmatrix}, e_0 = \begin{bmatrix} x \\ -x + 1 \end{bmatrix}, e_1 = 2x - 1;$$

$$c_0^T = r^T A + e_0^T \pmod{13, x^2 + 1} = [x - 4, 10]$$

$$c_1 = r^T b + e_1 + M \cdot \left\lfloor \frac{q}{2} \right\rfloor \pmod{13, x^2 + 1} = x - 1 + 6 \cdot (x + 1) = 7x + 5$$

- $\text{Dec}(sk, c = (c_0, c_1));$

$$\text{Round}_q(c_1 - \langle c_0, s \rangle \pmod{q}) = \text{Round}_q(7x + 5 - (x + 1)) = \text{Round}_q(6x - 4) = x + 1$$



KYBER KEY ENCAPSULATION MECHANISM - SCHEME

- $n = 256, q = 3329, k = 3$, H, G, J are hash functions;
- $KeyGen(1^\lambda)$;
 - Use the Kyber PKE KeyGen algorithm to generate $(A, b) \in R_q^{k \times k} \times R_q^k, s \in R_q^k$
 - Randomly select $z \in \{0,1\}^{256}$
 - Return a key pair $(pk = (A, b), sk = (s, pk, H(pk), z))$

KYBER KEY ENCAPSULATION MECHANISM - SCHEME

- $n = 256, q = 3329, k = 3, H, G, J$ are hash functions;
- $(pk = (A, b), sk = (s, pk, H(pk), z))$
- $Encaps(pk);$
 - Randomly select $m \in \{0,1\}^{256}$
 - Compute $h = H(pk),$
 - $(K, R) \leftarrow G(m, h)$
 - $c \leftarrow KyberPKE.Enc(pk, m; R)$
 - Output (K, c)
- $Decaps(sk = (s, pk, H(pk), z), c);$
 - $m' \leftarrow KyberPKE.Dec(s, c)$
 - $(K', R') \leftarrow G(m', H(pk))$
 - Compute $\bar{K} \leftarrow J(z, c)$
 - Compute $c' \leftarrow KyberPKE.Enc(pk, m'; R')$ and check if $c = c'$
 - If $c \neq c'$, then return \bar{K}
 - Otherwise, return K'

KYBER KEY ENCAPSULATION MECHANISM – PARAMETER SETS

	n	q	k	η_1	η_2	d_{c_0}	d_{c_1}	Pk size	Ct size	Dec.fail. rate	Security category
ML-KEM-512	256	3329	2	3	2	10	4	800	768	$< 2^{-139}$	1
ML-KEM-768	256	3329	3	2	2	10	4	1184	1088	$< 2^{-164}$	3
ML-KEM-1024	256	3329	4	2	2	11	5	1568	1568	$< 2^{-174}$	5

- Security category 1,3,5 ; the best known attacks require as much resources as needed for exhaustive key search for AES-128,AES-192,AES-256, resp.

KYBER KEY ENCAPSULATION MECHANISM – PERFORMANCE

- <http://pq-crystals.org/kyber/>
- Measured on one core of Intel Core-i7 4770K (Haswell) CPU

Kyber-512

Sizes (in bytes)		Haswell cycles (ref)		Haswell cycles (avx2)	
sk:	1632	gen:	122684	gen:	33856
pk:	800	enc:	154524	enc:	45200
ct:	768	dec:	187960	dec:	34572

Kyber-768

Sizes (in bytes)		Haswell cycles (ref)		Haswell cycles (avx2)	
sk:	2400	gen:	199408	gen:	52732
pk:	1184	enc:	235260	enc:	67624
ct:	1088	dec:	274900	dec:	53156

Kyber-1024

Sizes (in bytes)		Haswell cycles (ref)		Haswell cycles (avx2)	
sk:	3168	gen:	307148	gen:	73544
pk:	1568	enc:	346648	enc:	97324
ct:	1568	dec:	396584	dec:	79128

Thank you for your attention!



Any questions?

jooheele@sungshin.ac.kr