PAPER Special Section on Cryptography and Information Security

On the Equivalence of Several Security Notions of KEM and DEM

Waka NAGAO[†a)], ***Student Member***, **Yoshifumi MANABE**[†,††b)], ***and*** **Tatsuaki OKAMOTO**[†,††c)], ***Members***

SUMMARY KEM (Key Encapsulation Mechanism) and DEM (Data Encapsulation Mechanism) were introduced by Shoup to formalize the asymmetric encryption specified for key distribution and the symmetric encryption specified for data exchange in ISO standards on public-key encryption. Shoup defined the "semantic security (IND) against adaptive chosen ciphertext attacks (CCA2)" as a desirable security notion of KEM and DEM, that is, IND-CCA2 KEM and IND-CCA2 DEM. This paper defines "non-malleability (NM)" for KEM, which is a stronger security notion than IND. We provide three definitions of NM for KEM, and show that these three definitions are equivalent. We then show that NM-CCA2 KEM is equivalent to IND-CCA2 KEM. That is, we show that NM is equivalent to IND for KEM under CCA2 attacks, although NM is stronger than IND in the definition (or under some attacks like CCA1). In addition, this paper defines the universally composable (UC) security of KEM and DEM, and shows that IND-CCA2 KEM (or NM-CCA2 KEM) is equivalent to UC KEM and that "IND against adaptive chosen plaintext/ciphertext attacks (IND-P2-C2)" DEM is equivalent to UC DEM.

key words: *universal composability, KEM, DEM, ISO, IND-CCA2, NM-CCA2, IND-P2-C2, NM-P2-C2*

1. Introduction

The Key Encapsulation Mechanism (KEM) and the Data Encapsulation Mechanism (DEM) were proposed by Shoup as ISO standards for hybrid-public-key encryption (H-PKE) [10]. The security notion of indistinguishability (IND) (or semantically security) for KEM and DEM was also defined by Shoup. On the other hand, a definition of another stronger security notion "non-malleability (NM)" was introduced by Katz and Yung for private-key encryption (or DEM) and the relations between IND and NM were investigated [6] (their results include that IND-P2-C2 is equivalent to NM-P2-C2 for private public-key encryption).

In this paper, we investigate two stronger security notions for KEM and DEM. One is "non-malleability (NM)" for KEM and the other is "universal composability (UC)" for KEM and DEM.

NM for public-key encryption (PKE) was introduced [1], [2], [4] as a stronger security notion than IND and analogous definitions of NM for KEM were introduced in [7], [8]. As the NM of PKE have been defined with using a *message space* specified by an adversary, the existing NM definitions of KEM [7], [8] use a *key space* specified by an adversary, which corresponds to a message space of PKE. These existing NM definitions of KEM, however, are available only for a limited type of KEM schemes (e.g., a KEM scheme constructed from a PKE, where a random string plaintext to PKE is a session key output by KEM), since an adversary can specify a very small key space (e.g., $\{K_0, K_1\}$) but, in a general type of KEM scheme, it may be hard for a polynomial-time machine (an experiment in the NM definitions) to produce a ciphertext along with a key in this specified small key space as the output of the encryption function. That is, the existing NM definitions cannot be used for such a general type of KEM schemes.

A weaker security notion of non-malleability, wNM, was introduced and investigated by Herranz et al. [5]. wNM-CCA2 KEM is unlikely to imply IND-CCA2 KEM. Therefore, wNM is not considered to be a feasible definition of the NM for KEM, since a feasible definition of NM(-ATK) should imply IND(-ATK) (ATK $\in$ {CPA, CCA1, CCA2}). (In fact, the standard definition of NM(-ATK) of PKE implies IND(-ATK).)

This paper, for the first time, provides the NM definitions that satisfy the following feasible requirements: (1) the NM definitions are available for any type of KEM schemes, in which no key space is used, (2) the NM definitions are stronger than IND (i.e., NM(-ATK) implies IND(-ATK)); for more detailed description on this matter, see Sect. 3.1 and Theorem 4), and (3) the NM definitions capture the naive non-malleable property that the adversary is given challenge ciphertext C^* and he should not be able to come up with another ciphertext C such that its decapsulated key K is non-trivially related to the challenge key K^*. Here, we introduce three NM definitions of KEM, and show that the three definitions are equivalent.

It is easily obtained from one of the definitions of NM that NM-CCA2 KEM is equivalent to IND-CCA2 KEM. That is, we can now recognize that Shoup's definition, IND-CCA2, for KEM is as feasible as NM-CCA2, whereas NM itself is stronger than IND in the definition.

In addition, this paper investigates the other stronger definitions; the universally composable (UC) security for KEM and DEM. The UC framework was introduced by Canetti [3] and it guarantees very strong security, i.e., preserves stand-alone security in any type of composition with other primitives and protocols.

Although the UC security for KEM and DEM, as the

Manuscript received March 16, 2007.
Manuscript revised July 16, 2007.

[†]The authors are with the Graduate School of Informatics, Kyoto University, Kyoto-shi, 606-8501 Japan.
[††]The authors are with NTT Labs, NTT Corporation, Musashino-shi, 180-8585 Japan.
a) E-mail: w-nagao@ai.soc.i.kyoto-u.ac.jp
b) E-mail: manabe.yoshifumi@lab.ntt.co.jp
c) E-mail: okamoto.tatsuaki@lab.ntt.co.jp

DOI: 10.1093/ietfec/e91–a.1.283

ideal functionalities of KEM and KEM-DEM, has been defined and investigated in [7], [8], this paper modifies the definition, security proof and description as follows: In the previous definition of $\mathcal{F}_{\text{KEM-DEM}}$, only a single shared key was available in the DEM phase. This paper modifies $\mathcal{F}_{\text{KEM-DEM}}$ to remove this restriction so that a single copy of $\mathcal{F}_{\text{KEM-DEM}}$ accepts multiple shared keys in the DEM phase. Another problem in [7], [8] is the proof that UC KEM equals NM-CCA2 (i.e., IND-CCA2) KEM; the proof was based on a previous definition of NM which is, as mentioned above, only available for a limited type of KEM schemes. This paper corrects the proof of the equality between UC KEM and IND-CCA2 KEM, in which we directly prove it without using any NM definition, (it is equivalent to the proof through our new NM definition). In addition, this paper follows the new framework of UC that was totally revised by Canetti in 2005 [3], while [7], [8] are based on the original one in 2001. The equivalence between UC DEM and IND-P2-C2 DEM is also proven (through no NM) in this paper, while only a sketch of proof was provided (through NM) in [7], [8].

2. Preliminaries

2.1 Notations

$\mathbb{N}$ is the set of natural numbers and $\mathbb{R}$ is the set of real numbers. $\perp$ denotes a null string.

A function $f : \mathbb{N} \to \mathbb{R}$ is negligible in k, if for every constant $c > 0$, there exists integer k_c such that $f(k) < k^{-c}$ for all $k > k_c$. Hereafter, we often use $f < \epsilon(k)$ to mean that f is negligible in k. On the other hand, we use $f > \mu(k)$ to mean that f is not negligible in k. i.e., function $f : \mathbb{N} \to \mathbb{R}$ is not negligible in k, if there exists a constant $c > 0$ such that for every integer k_c, there exists $k > k_c$ such that $f(k) > k^{-c}$.

When A is a probabilistic machine or algorithm, $A(x)$ denotes the random variable of A's output on input x. $y \xleftarrow{R} A(x)$ denotes that y is randomly selected from $A(x)$ according to its distribution. When A is a set, $y \xleftarrow{U} A$ denotes that y is uniformly selected from A. When A is a value, $y \leftarrow A$ denotes that y is set as A.

We write vectors in boldface, as in $\boldsymbol{x}$, and denote the number of components in $\boldsymbol{x}$ by $|\boldsymbol{x}|$ and the i-th component by $\boldsymbol{x}[i]$, so that $\boldsymbol{x} = (\boldsymbol{x}[1], \cdots, \boldsymbol{x}[|\boldsymbol{x}|])$. We also denote a component of a vector as $\mathrm{x} \in \boldsymbol{x}$ or $\mathrm{x} \notin \boldsymbol{x}$, which means, respectively, that x is in or is not in the set $\{ \boldsymbol{x}[i] : 1 \le i \le |\boldsymbol{x}| \}$. We can simply write $\boldsymbol{x} \leftarrow \mathcal{D}(\boldsymbol{y})$ as the shorthand form of $1 \le i \le | \boldsymbol{y} | \boldsymbol{x}[i] \leftarrow \mathcal{D}(\boldsymbol{y}[i])$. We will consider a relation, *Rel*, of t variables. Rather than writing $Rel(x_1, \cdots, x_t)$, we write $Rel(x, \boldsymbol{x})$, meaning the first argument is special and the rest are bunched into vector $\boldsymbol{x}$ with $|\boldsymbol{x}| = t - 1$.

2.2 Key Encapsulation Mechanism

2.2.1 Definition of Key Encapsulation Mechanism

We recall the standard notion of key encapsulation mechanism, KEM, which was formalized by Shoup in [10]. A KEM scheme is the triple of algorithms, $\Sigma = (\mathcal{G}, \mathcal{E}, \mathcal{D})$, where

1. $\mathcal{G}$, the key generation algorithm, is a probabilistic polynomial time (PPT) algorithm that takes a security parameter $k \in \mathbb{N}$ (provided in unary) and returns a pair (pk, sk) of matching public and secret keys.
2. $\mathcal{E}$, the key encryption algorithm, is a PPT algorithm that takes as input public key pk and outputs a key/ciphertext pair (K^*, C^*).
3. $\mathcal{D}$, the decryption algorithm, is a deterministic polynomial time algorithm that takes as input secret key sk and ciphertext C^*, and outputs key K^* or $\perp$ ($\perp$ implies that the ciphertext is invalid).

We require that for all (pk, sk) output by key generation algorithm $\mathcal{G}$ and for all (K^*, C^*) output by key encryption algorithm $\mathcal{E}(pk)$, $\mathcal{D}(sk, C^*) = K^*$ holds. Here, the length of the key, $|K^*|$, is specified by $l(k)$, where k is the security parameter.

2.2.2 Basic Attack Types of KEM

From the standard notion of attack type, we consider the following three attack types of KEM: CPA, CCA1, and CCA2. CPA means "Chosen Plaintext Attacks," where an adversary is allowed to access only an encryption oracle; i.e., no decryption oracle. CCA1 means "Chosen Ciphertext Attacks," where an adversary is allowed to access both encryption and decryption oracles, but the adversary cannot access the decryption oracle after getting the target ciphertext. CCA2 means "Adaptive Chosen Ciphertext Attacks," where an adversary is allowed to access both encryption and decryption oracles even after the adversary is given the target ciphertext.

2.2.3 Definition of Indistinguishability for KEM

The indistinguishability (IND) of KEM was defined by Shoup [10]. We use "IND-ATK-KEM" to describe the security notion of indistinguishability for KEM against ATK $\in$ {CPA, CCA1, CCA2}. "IND-KEM" is used to focus on the indistinguishability of KEM without regard to attack type. If it is clear from the context that IND-ATK-KEM (and IND-KEM) is used for KEM, we will call it IND-ATK (and IND) for simplicity.

To clarify the indistinguishability of public key encryption (PKE), we may use IND-ATK-PKE and IND-PKE.

Definition 1. *Let Σ be a* KEM, *$A = (A_1, A_2)$ be an adversary, and $k \in \mathbb{N}$ be a security parameter. For ATK $\in \{CPA, CCA1, CCA2\}$, $\mathbf{Adv}^{\mathsf{IND\text{-}ATK}}_{A,\Sigma}(k)$ is defined in Fig. 1. We say that Σ is IND-ATK-KEM, if for any adversary $A \in \mathcal{P}$, $\mathbf{Adv}^{\mathsf{IND\text{-}ATK}}_{A,\Sigma}(k)$ is negligible in k, where ATK $\in \{CPA, CCA1, CCA2\}$, and $\mathcal{P}$ denotes a class of polynomial-time bounded machines.*

$$\mathtt{Adv}^{\mathsf{IND\text{-}ATK}}_{A,\Sigma}(k) \equiv \Pr[\mathtt{Expt}^{\mathsf{IND\text{-}ATK}}_{A,\Sigma}(k) = 1] - \frac{1}{2},$$

where $\mathtt{Expt}^{\mathsf{IND\text{-}ATK}}_{A,\Sigma}(k)$:

$$(pk, sk) \xleftarrow{R} \mathcal{G}(1^k);\ s \xleftarrow{R} A_1^{O_1}(pk);$$
$$(K^*, C^*) \xleftarrow{R} \mathcal{E}(pk);\ R \xleftarrow{U} \{0,1\}^{l(k)};\ b \xleftarrow{U} \{0,1\};$$
$$X \leftarrow \begin{cases} K^*, \text{ if } \ b = 0 \\ R, \text{ if } \ b = 1 \end{cases}$$
$$g \xleftarrow{R} A_2^{O_2}(s, X, C^*); \text{return 1, iff} \quad g = b$$

and
If ATK = CPA, then $O_1 = \perp$ and $O_2 = \perp$.
If ATK = CCA1, then $O_1 = \mathcal{D}(sk, \cdot)$ and $O_2 = \perp$.
If ATK = CCA2, then $O_1 = \mathcal{D}(sk, \cdot)$ and $O_2 = \mathcal{D}(sk, \cdot)$.

Fig. 1 Advantage of IND-ATK-KEM.

$$\mathtt{Adv}^{\mathsf{IND\text{-}PX\text{-}CY}}_{A,\Sigma'}(k) \equiv \Pr[\mathtt{Expt}^{\mathsf{IND\text{-}PX\text{-}CY}}_{A,\Sigma'}(k)] - \frac{1}{2},$$

where $\mathtt{Expt}^{\mathsf{IND\text{-}PX\text{-}CY}}_{A,\Sigma'}(k)$:

$$K \xleftarrow{U} \{0,1\}^{l(k)};\ (x_0, x_1, s) \xleftarrow{R} A_1^{O_1, O'_1}(1^k);$$
$$b \xleftarrow{U} \{0,1\};\ y \xleftarrow{R} \mathcal{E}'(K, x_b);$$
$$g \xleftarrow{R} A_2^{O_2, O'_2}(1^k, s, y); \text{return 1 iff} \quad g = b$$

and
If X = 0 then $O_1(\cdot) = \perp$ and $O_2(\cdot) = \perp$.
If X = 1 then $O_1(\cdot) = \mathcal{E}'(K, \cdot)$ and $O_2(\cdot) = \perp$.
If X = 2 then $O_1(\cdot) = \mathcal{E}'(K, \cdot)$ and $O_2(\cdot) = \mathcal{E}'(K, \cdot)$.
If Y = 0 then $O'_1(\cdot) = \perp$ and $O'_2(\cdot) = \perp$.
If Y = 1 then $O'_1(\cdot) = \mathcal{D}'(K, \cdot)$ and $O'_2(\cdot) = \perp$.
If Y = 2 then $O'_1(\cdot) = \mathcal{D}'(K, \cdot)$ and $O'_2(\cdot) = \mathcal{D}'(K, \cdot)$.

Fig. 2 Advantage of IND-PX-CY-DEM.

2.3 Data Encapsulation Mechanism

2.3.1 Definition of Data Encapsulation Mechanism

We recall the standard notion of data encapsulation mechanism, DEM, which was formalized by Shoup in [10]. A DEM scheme is the pair of algorithms, $\Sigma' = (\mathcal{E}', \mathcal{D}')$, where

1. $\mathcal{E}'$, the data encryption algorithm, is a PPT algorithm that takes as input secret key K (K is shared by KEM) and plaintext M, and outputs ciphertext C.
2. $\mathcal{D}'$, the data decryption algorithm, is a deterministic polynomial time algorithm that takes as input secret key K and ciphertext C, and outputs plaintext M or $\perp$ ($\perp$ implies that the ciphertext is invalid).

It is required that for all C output by data encryption algorithm $\mathcal{E}'(K, M)$, $\mathcal{D}'(K, C) = M$ holds ("soundness"). Here, the length of the key, $|K|$, is specified by $l(k)$, where k is the security parameter.

2.3.2 Basic Attack Types of DEM

From the standard notion of attack type, we consider the following nine attack types of DEM: PX-CY (X=0, 1, 2 and Y=0, 1, 2), i.e., P0-C0, P1-C0, P2-C0, P0-C1, P1-C1, P2-C1, P0-C2, P1-C2 and P2-C2.

PX (X=0, 1, 2) denotes access to the encryption oracle. P0 means no access to the encryption oracle by adversary. P1 means "Chosen Plaintext Attacks," where the adversary is allowed to access the encryption oracle, i.e., the adversary cannot access the encryption oracle after getting the target ciphertext. P2 means "Adaptive Chosen Plaintext Attacks," where the adversary is allowed to access the encryption oracle, even after it gets the target ciphertext.

CY (Y=0, 1, 2) denotes access to the decryption oracle. C0 means no access to the decryption oracle by adversary. C1 means "Chosen Ciphertext Attacks" where the adversary is allowed to access the decryption oracle, and cannot access the decryption oracle after getting the target ciphertext. C2 means "Adaptive Chosen Ciphertext Attacks," where an adversary is allowed to access the decryption oracle after it gets the target ciphertext.

2.3.3 Definition of Indistinguishability for DEM

The advantage of indistinguishability of DEM (we use "IND-DEM") following [6] is stated in Fig. 2. In this paper, we also use IND-PX-CY-DEM to describe the security notion of indistinguishability of DEM against {X, Y} ∈ {0, 1, 2}.

Definition 2. *Let Σ' be a* DEM *over message space M, $A = (A_1, A_2)$ be an adversary, and $k \in \mathbb{N}$ be the security parameter. For {X, Y} ∈ {0, 1, 2}, $\mathtt{Adv}^{\mathsf{IND\text{-}PX\text{-}CY}}_{A,\Sigma'}(k)$ is defined in Fig. 2. We say that Σ' is IND-PX-CY-DEM, if for any adversary $A \in \mathcal{P}$, $\mathtt{Adv}^{\mathsf{IND\text{-}PX\text{-}CY}}_{A,\Sigma'}(k)$ is negligible in k, where {X, Y} ∈ {0, 1, 2}, and $\mathcal{P}$ denotes a class of polynomial-time bounded machines.*

Note that, the length of x_0 equals the length of x_1, i.e., $|x_0| = |x_1|$. Furthermore, when $Y = 2$, we insist that A_2 does not ask for the decryption of challenge ciphertext y.

2.4 Notion of Universal Composability

The notion of universal composability (UC) was introduced by Canetti [3]. This notion makes it easy to introduce definitions of the real life world and the ideal process world and the framework of UC. This UC framework is a little changed in terms of the definition of security of functionality from the first version. (For more detail, see the revised version [3].) In the real life world, there is adversary A and protocol π which realizes a functionality among some parties. On the other hand, in the ideal process world, there is a simulator S that simulates the real life world, an ideal functionality $\mathcal{F}$, and dummy parties. We consider environment Z which tries to distinguish the real life world from the ideal process world.

2.4.1 The Real Life World/The Ideal Process World

- Let $\mathtt{REAL}_{\pi,A,Z}(k,z)$ denote the output of environment Z when interacting with adversary A and parties $P_1, \ldots, P_n$ running protocol π on security parameter k and input z.
- Let $\mathtt{IDEAL}_{\mathcal{F},S,Z}(k,z)$ denote the output of environment Z after interacting in the ideal process world with adversary S and ideal functionality $\mathcal{F}$, on security parameter k and input z.

2.4.2 The Security Framework of UC

Let $\mathcal{F}$ be an ideal functionality and let π be a protocol. We say that π UC-realizes $\mathcal{F}$, if for any adversary $A \in \mathcal{P}$ there exists a simulator $S \in \mathcal{P}$ such that for any environment $Z \in \mathcal{P}$,

$$\mathtt{IDEAL}_{\mathcal{F},S,Z}(k,z) \approx \mathtt{REAL}_{\pi,A,Z}(k,z),$$

where $\approx$ denotes statistically indistinguishable in k and $\mathcal{P}$ denotes a class of polynomial-time bounded machines.

3. Three Non-malleability Definitions of KEM

3.1 Definition of SNM-ATK-KEM

KEM Σ is called "SNM-ATK-KEM" in the sense that Σ is secure in *simulation based non-malleability* (SNM) for each attack type ATK $\in$ {CPA, CCA1, CCA2}.

Definition 3. *Let Σ be* KEM*, Rel be a relation, $A = (A_1, A_2)$ be an adversary, $S = (S_1, S_2)$ be an algorithm (the "simulator"), and $k \in \mathbb{N}$ be the security parameter. For ATK $\in \{CPA, CCA1, CCA2\}$, we define $\mathtt{Adv}^{\mathsf{SNM\text{-}ATK}}_{A,S,\Sigma}(Rel,k)$ in Fig. 3. We say that Σ is SNM-ATK-KEM, if for any adversary $A \in \mathcal{P}$ and all relations Rel computable in $\mathcal{P}$, there exists simulator $S \in \mathcal{P}$ such that*

$\mathtt{Adv}^{\mathsf{SNM\text{-}ATK}}_{A,S,\Sigma}(Rel,k)$ is negligible in k, where ATK $\in \{CPA, CCA1, CCA2\}$ and $\mathcal{P}$ denotes a class of polynomial-time bounded machines.

Note that adversary A_2 is not allowed to pose the challenge ciphertext C^* to its decryption oracle in the case of CCA2.

In the previous NM definitions [7], [8], the adversary can select the key space. As mentioned in Introduction, it causes a serious problem that the definitions are available only for a limited type of KEM schemes. Therefore, the revised point in this paper is to free the key space of the old version definition in $\mathtt{Expt}^{\mathsf{SNM\text{-}ATK}}_{A,\Sigma}(Rel,k)$.

In the attack scenario of SNM for public key encryption (PKE), SNM-PKE, the adversary can decide the message space [2]. Note that such a message space in the scenario is introduced to make SNM-PKE compatible with IND-PKE (i.e., to make SNM-PKE imply IND-PKE), in whose attack scenario the adversary can decide a pair of messages (a message space).

In contrast, in the attack scenario of IND-KEM, a correct key or a random value along with the target ciphertext is given to the adversary. To make SNM-KEM compatible with IND-KEM (i.e., to make SNM-KEM imply IND-KEM), our SNM-KEM's attack scenario gives the adversary a randomly-ordered pair of a correct key and a random value.

Here, if KEM Σ is not IND(-ATK) (i.e., an adversary A can distinguish (C^*, K^*) and (C^*, R^*)), Σ is not NM(-ATK). (e.g., A guesses K^* from X, sets $Rel(K^*, K')$ iff $lsb(K^*) = lsb(K')$, and randomly searches for C' such that $(K', C') \xleftarrow{R} \mathcal{E}(pk)$ and $lsb(K^*) = lsb(K')$).

Two additional minor differences between SNM-KEM and SNM-PKE are:

1. Simulator S also gets access to the decryption oracle when ATK allows it to do so.
2. Relation R utilizes state information s calculated not by A_1 or S_1 but by A_2 or S_2 in SNM-KEM.

The difference between our NM-KEM and Herrantz et al.'s wNM-KEM [5] is whether adversary A_2 can gain key information X (this includes the order of key K^* and a random string R (or another random string R^*)) or not. X in our SNM-KEM (and PNM-KEM, CNM-KEM) definition plays a similar role to the message space in the NM definitions by [1], [2] for PKE.

3.2 Definition of CNM-ATK-KEM

KEM Σ is called "CNM-ATK-KEM" in the sense that Σ is secure in *comparison based non-malleability* (CNM) for each attack type ATK $\in$ {CPA, CCA1, CCA2}.

Definition 4. *Let Σ be* KEM*, $A = (A_1, A_2)$ be an adversary, and $k \in \mathbb{N}$ be the security parameter. For ATK $\in \{CPA, CCA1, CCA2\}$, we define $\mathtt{Adv}^{\mathsf{CNM\text{-}ATK}}_{A,\Sigma}(k)$ in Fig. 4. We say that Σ is CNM-ATK-KEM, if for any adversary $A \in \mathcal{P}$, $\mathtt{Adv}^{\mathsf{CNM\text{-}ATK}}_{A,\Sigma}(k)$ is negligible in k, where ATK $\in \{CPA, CCA1, CCA2\}$, and $\mathcal{P}$ denotes a class of polynomial-time bounded machines.*

Note that adversary A_2 is not allowed to ask its oracle to decrypt the challenge ciphertext C^* in the case of CCA2.

The revised point is to free the key space of the old version definitions in $\mathtt{Expt}^{\mathsf{CNM\text{-}ATK}}_{A,\Sigma}(k)$ and $\widetilde{\mathtt{Expt}}^{\mathsf{CNM\text{-}ATK}}_{A,\Sigma}(k)$.

Similar to SNM-KEM, our CNM-KEM's attack scenario gives the adversary a randomly-ordered pair of a correct key and a random value to make CNM-KEM compatible with IND-KEM.

3.3 Definition of PNM-ATK-KEM

KEM Σ is called "PNM-ATK-KEM" in the sense that Σ is secure in *parallel chosen-ciphertext attack based non-malleability* (PNM) for each attack type ATK $\in$ {CPA, CCA1, CCA2}.

$$\mathtt{Adv}^{\mathsf{SNM\text{-}ATK}}_{A,S,\Sigma}(Rel,k) \equiv \Pr[\mathtt{Expt}^{\mathsf{SNM\text{-}ATK}}_{A,\Sigma}(Rel,k)=1] - \Pr[\mathtt{Expt}^{\mathsf{SNM\text{-}ATK}}_{S,\Sigma}(Rel,k)=1],$$

where

$\mathtt{Expt}^{\mathsf{SNM\text{-}ATK}}_{A,\Sigma}(Rel,k)$:
$(pk,sk) \xleftarrow{R} \mathcal{G}(1^k)$; $s_1 \xleftarrow{R} A_1^{O_1}(pk)$
$(K^*,C^*) \xleftarrow{R} \mathcal{E}(pk)$; $R \xleftarrow{U} \{0,1\}^{l(k)}$; $b \xleftarrow{U} \{0,1\}$
$X \leftarrow (r_0,r_1)$, where $\begin{cases} r_0 \leftarrow K^* \text{ and } r_1 \leftarrow R, & \text{if } b=0 \\ r_0 \leftarrow R \text{ and } r_1 \leftarrow K^*, & \text{if } b=1 \end{cases}$
$(s_2,\boldsymbol{C}) \xleftarrow{R} A_2^{O_2}(X,s_1,C^*)$; $\boldsymbol{K} \leftarrow \mathcal{D}(sk,\boldsymbol{C})$
return 1, iff $(C^* \notin \boldsymbol{C}) \wedge Rel(K^*,\boldsymbol{K},s_2)$

$\mathtt{Expt}^{\mathsf{SNM\text{-}ATK}}_{S,\Sigma}(Rel,k)$:
$(pk,sk) \xleftarrow{R} \mathcal{G}(1^k)$; $s_1 \xleftarrow{R} S_1^{O_1}(pk)$
$R^* \xleftarrow{U} \{0,1\}^{l(k)}$; $R \xleftarrow{U} \{0,1\}^{l(k)}$; $b \xleftarrow{U} \{0,1\}$
$X \leftarrow (r_0,r_1)$, where $\begin{cases} r_0 \leftarrow R^* \text{ and } r_1 \leftarrow R, & \text{if } b=0 \\ r_0 \leftarrow R \text{ and } r_1 \leftarrow R^*, & \text{if } b=1 \end{cases}$
$(s_2,\boldsymbol{C}) \xleftarrow{R} S_2^{O_2}(X,s_1)$; $\boldsymbol{K} \leftarrow \mathcal{D}(sk,\boldsymbol{C})$
return 1, iff $Rel(R^*,\boldsymbol{K},s_2)$

and
If ATK = CPA, then $O_1 = \perp$ and $O_2 = \perp$.
If ATK = CCA1, then $O_1 = \mathcal{D}(sk,\cdot)$ and $O_2 = \perp$.
If ATK = CCA2, then $O_1 = \mathcal{D}(sk,\cdot)$ and $O_2 = \mathcal{D}(sk,\cdot)$.

Fig. 3 Advantage of SNM-ATK-KEM.

$$\mathtt{Adv}^{\mathsf{CNM\text{-}ATK}}_{A,\Sigma}(k) \equiv \Pr[\mathtt{Expt}^{\mathsf{CNM\text{-}ATK}}_{A,\Sigma}(k)=1] - \Pr[\widetilde{\mathtt{Expt}}^{\mathsf{CNM\text{-}ATK}}_{A,\Sigma}(k)=1],$$

where

$\mathtt{Expt}^{\mathsf{CNM\text{-}ATK}}_{A,\Sigma}(k)$:
$(pk,sk) \xleftarrow{R} \mathcal{G}(1^k)$; $s \xleftarrow{R} A_1^{O_1}(pk)$; $(K^*,C^*) \xleftarrow{R} \mathcal{E}(pk)$
$R \xleftarrow{U} \{0,1\}^{l(k)}$; $b \xleftarrow{U} \{0,1\}$
$X \leftarrow (r_0,r_1)$, where $\begin{cases} r_0 \leftarrow K^* \text{ and } r_1 \leftarrow R, & \text{if } b=0 \\ r_0 \leftarrow R \text{ and } r_1 \leftarrow K^*, & \text{if } b=1 \end{cases}$
$(Rel,\boldsymbol{C}) \xleftarrow{R} A_2^{O_2}(X,s,C^*)$; $\boldsymbol{K} \leftarrow \mathcal{D}(sk,\boldsymbol{C})$
return 1, iff $(C^* \notin \boldsymbol{C}) \wedge Rel(K^*,\boldsymbol{K})$

$\widetilde{\mathtt{Expt}}^{\mathsf{CNM\text{-}ATK}}_{A,\Sigma}(k)$:
$(pk,sk) \xleftarrow{R} \mathcal{G}(1^k)$; $s \xleftarrow{R} A_1^{O_1}(pk)$; $(K^*,C^*) \xleftarrow{R} \mathcal{E}(pk)$
$R^* \xleftarrow{U} \{0,1\}^{l(k)}$; $R \xleftarrow{U} \{0,1\}^{l(k)}$; $b \xleftarrow{U} \{0,1\}$
$X \leftarrow (r_0,r_1)$, where $\begin{cases} r_0 \leftarrow R^* \text{ and } r_1 \leftarrow R, & \text{if } b=0 \\ r_0 \leftarrow R \text{ and } r_1 \leftarrow R^*, & \text{if } b=1 \end{cases}$
$(Rel,\boldsymbol{C}) \xleftarrow{R} A_2^{O_2}(X,s,C^*)$; $\boldsymbol{K} \leftarrow \mathcal{D}(sk,\boldsymbol{C})$
return 1, iff $(C^* \notin \boldsymbol{C}) \wedge Rel(R^*,\boldsymbol{K})$

and
If ATK = CPA, then $O_1 = \perp$ and $O_2 = \perp$.
If ATK = CCA1, then $O_1 = \mathcal{D}(sk,\cdot)$ and $O_2 = \perp$.
If ATK = CCA2, then $O_1 = \mathcal{D}(sk,\cdot)$ and $O_2 = \mathcal{D}(sk,\cdot)$.

Fig. 4 Advantage of CNM-ATK-KEM.

$$\mathtt{Adv}^{\mathsf{PNM\text{-}ATK}}_{A,\Sigma}(k) \equiv \Pr[\mathtt{Expt}^{\mathsf{PNM\text{-}ATK}}_{A,\Sigma}(k)=1] - \frac{1}{2},$$

where $\mathtt{Expt}^{\mathsf{PNM\text{-}ATK}}_{A,\Sigma}(k)$:

$$(pk,sk) \xleftarrow{R} \mathcal{G}(1^k);\ s_1 \xleftarrow{R} A_1^{O_1}(pk);$$
$$(K^*,C^*) \xleftarrow{R} \mathcal{E}(pk);\ R \xleftarrow{U} \{0,1\}^{l(k)};\ b \xleftarrow{U} \{0,1\};$$
$$X \leftarrow \begin{cases} K^*, \text{if } b=0 \\ R, \text{if } b=1 \end{cases}$$
$$(s_2,\boldsymbol{C}) \xleftarrow{R} A_2^{O_2}(X,s_1,C^*);\ \boldsymbol{K} \leftarrow \mathcal{D}(sk,\boldsymbol{C})$$
$$g \xleftarrow{R} A_3(s_2,\boldsymbol{K});\ \text{return 1, iff } (C^* \notin \boldsymbol{C}) \ \wedge\ (g=b)$$

and
If ATK = CPA, then $O_1 = \perp$ and $O_2 = \perp$.
If ATK = CCA1, then $O_1 = \mathcal{D}(sk,\cdot)$ and $O_2 = \perp$.
If ATK = CCA2, then $O_1 = \mathcal{D}(sk,\cdot)$ and $O_2 = \mathcal{D}(sk,\cdot)$.

Fig. 5 Advantage of PNM-ATK-KEM.

Definition 5. *Let* Σ *be a* KEM, $A = (A_1,A_2,A_3)$ *be an adversary, and* $k \in \mathbb{N}$ *be the security parameter. For ATK* $\in \{CPA, CCA1, CCA2\}$, *we define* $\mathtt{Adv}^{\mathsf{PNM\text{-}ATK}}_{A,\Sigma}(k)$ *in Fig. 5. We say that* Σ *is PNM-ATK-KEM, if for any adversary* $A \in \mathcal{P}$, $\mathtt{Adv}^{\mathsf{PNM\text{-}ATK}}_{A,\Sigma}(k)$ *is negligible in k, where k is the security parameter, ATK* $\in \{CPA, CCA1, CCA2\}$, *and* $\mathcal{P}$ *denotes a class of polynomial-time bounded machines.*

Note that adversary A_2 is not allowed to ask its oracle to decrypt challenge ciphertext C^* in the case of CCA2.

The revised point is to free the key space of the old version definitions in $\mathtt{Expt}^{\mathsf{PNM\text{-}ATK}}_{A,\Sigma}(k)$.

In the PNM definition, the non-malleability property is captured by indistinguishability under parallel chosen-ciphertext attack such that A_2 outputs a vector of ciphertext $\boldsymbol{C}$ and its decryption result $\boldsymbol{K}$ is given to A_3.

4. Equivalence of the Three Non-malleability Definitions

Here, we prove the equivalence of the three non-malleability definitions.

Theorem 1. *For any ATK* $\in \{CPA, CCA1, CCA2\}$, *if* KEM Σ *is CNM-ATK-KEM, then* Σ *is SNM-ATK-KEM.*

$B_1^{O_1}(pk)$	$B_2^{O_2}(X, s, C^*)$
$t_1 \xleftarrow{R} A_1^{O_1}(pk)$	$(s_2, \boldsymbol{C}) \xleftarrow{R} A_2^{O_2}(X, s, C^*)$
$s \leftarrow t_1$	Define Rel' by $Rel'(a, \boldsymbol{b}) = 1$,
return s	iff $Rel(a, \boldsymbol{b}, s_2) = 1$
	return $(Rel', \boldsymbol{C})$

Fig. 6 CNM-ATK adversary B using SNM-ATK adversary A.

$\hat{S}_1^{O_1}(pk)$	$\hat{S}_2^{O_2}(X, s_1)$
$t_1 \xleftarrow{R} A_1^{O_1}(pk)$	$(K^*, C^*) \xleftarrow{R} \mathcal{E}(pk)$
$s_1 \leftarrow t_1$	$(s_2, \boldsymbol{C}) \xleftarrow{R} A_2^{O_2}(X, s_1, C^*)$
return s_1	If $C^* \in \boldsymbol{C}$, then return $\perp$.
	Otherwise, return $(s_2, \boldsymbol{C})$.

Fig. 7 SNM-ATK simulator $\hat{S}$ using SNM-ATK adversary A.

Theorem 2. *For any ATK* $\in \{CPA, CCA1, CCA2\}$*, if* KEM Σ *is SNM-ATK-KEM, then* Σ *is PNM-ATK-KEM.*

Theorem 3. *For any ATK* $\in \{CPA, CCA1, CCA2\}$*, if* KEM Σ *is PNM-ATK-KEM, then* Σ *is CNM-ATK-KEM.*

4.1 Proof of Theorem 1

Proof. We prove that KEM Σ is not CNM-ATK-KEM if Σ is not SNM-ATK-KEM. More precisely, we show that if adversary A and relation Rel exist such that $\mathtt{Adv}^{\mathsf{SNM\text{-}ATK}}_{A,S,\Sigma}(Rel, k)$ is not negligible in k for any simulator S, then there exists adversary B such that $\mathtt{Adv}^{\mathsf{CNM\text{-}ATK}}_{B,\Sigma}(k)$ is not negligible in k, where k is the security parameter and ATK $\in$ {CPA, CCA1, CCA2}.

Let $A = (A_1, A_2)$ be an adversary for SNM-ATK.

First, we construct a CNM-ATK adversary $B = (B_1, B_2)$ using SNM-ATK adversary A in Fig. 6.

From the construction of B, we obtain the following equivalence for all $k \in \mathbb{N}$:

$$\Pr[\mathtt{Expt}^{\mathsf{SNM\text{-}ATK}}_{A,\Sigma}(Rel, k) = 1] = \Pr[\mathtt{Expt}^{\mathsf{CNM\text{-}ATK}}_{B,\Sigma}(k) = 1]. \tag{1}$$

We then construct SNM-ATK simulator $\hat{S} = (\hat{S_1}, \hat{S_2})$ using SNM-ATK adversary A as shown in Fig. 7.

From the construction of B using A, and the construction of $\hat{S}$, we obtain the following equivalence for all $k \in \mathbb{N}$:

$$\Pr[\mathtt{Expt}^{\mathsf{SNM\text{-}ATK}}_{\hat{S},\Sigma}(Rel, k) = 1] = \Pr[\widetilde{\mathtt{Expt}}^{\mathsf{CNM\text{-}ATK}}_{B,\Sigma}(k) = 1]. \tag{2}$$

Here, note that, even if $A_2^{O_2}$ outputs $\boldsymbol{C}$ with $C^* \in \boldsymbol{C}$, $B_2^{O_2}$ outputs the ciphertext vector $\boldsymbol{C}$, and $\widetilde{\mathtt{Expt}}^{\mathsf{CNM\text{-}ATK}}_{B,\Sigma}(k)$ returns 0 because of $C^* \in \boldsymbol{C}$. $\hat{S_2}^{O_2}$ returns $\perp$ and $\mathtt{Expt}^{\mathsf{SNM\text{-}ATK}}_{\hat{S},\Sigma}(Rel, k)$ returns 0. (A problem regarding this note was investigated in [9]).

The assumption (for contradiction) is that, for any S, $\mathtt{Adv}^{\mathsf{SNM\text{-}ATK}}_{A,S,\Sigma}(Rel, k) > \mu(k)$ implies $\mathtt{Adv}^{\mathsf{SNM\text{-}ATK}}_{A,\hat{S},\Sigma}(Rel, k) > \mu(k)$ (for specific $\hat{S}$). From this inequality and Eqs. (1) and (2), we obtain

$$\begin{aligned}
&\mathtt{Adv}^{\mathsf{CNM\text{-}ATK}}_{B,\Sigma}(k) \\
&= \Pr[\mathtt{Expt}^{\mathsf{CNM\text{-}ATK}}_{B,\Sigma}(k) = 1] - \Pr[\widetilde{\mathtt{Expt}}^{\mathsf{CNM\text{-}ATK}}_{B,\Sigma}(k) = 1] \\
&= \Pr[\mathtt{Expt}^{\mathsf{SNM\text{-}ATK}}_{A,\Sigma}(Rel, k) = 1] \\
&\quad - \Pr[\mathtt{Expt}^{\mathsf{SNM\text{-}ATK}}_{\hat{S},\Sigma}(Rel, k) = 1] \\
&= \mathtt{Adv}^{\mathsf{SNM\text{-}ATK}}_{A,\hat{S},\Sigma}(Rel, k) > \mu(k).
\end{aligned}$$

□

$B_1^{O_1}(pk)$
$t_1 \xleftarrow{R} A_1^{O_1}(pk)$; $s_1 \leftarrow t_1$; return s_1
$B_2^{O_2}(X, s_1, C^*)$, where $s_1 = t_1$ and $X = (r_0, r_1)$
$(t_2, \boldsymbol{C}) \xleftarrow{R} A_2^{O_2}(r_0, t_1, C^*)$
Choose random coins σ for A_3.
$s_2 \leftarrow (t_2, \sigma, X)$; return $(s_2, \boldsymbol{C})$
$Rel(Y, \boldsymbol{K}, s_2)$, where $s_2 = (t_2, \sigma, X)$
If Y is not an element of X, return 0.
If $Y = r_0$, then $b = 0$. Otherwise, $b = 1$.
$g \leftarrow A_3(t_2, \boldsymbol{K}; \sigma)$; return 1, iff $b = g$

Fig. 8 SNM-ATK adversary B and relation Rel using PNM-ATK adversary A.

4.2 Proof of Theorem 2

Proof. We prove that KEM Σ is not SNM-ATK-KEM if Σ is not PNM-ATK-KEM. More precisely, we show that if there exists adversary A such that $\mathtt{Adv}^{\mathsf{PNM\text{-}ATK}}_{A,\Sigma}(k)$ is not negligible in k, then adversary B and relation Rel exist for any simulator S such that $\mathtt{Adv}^{\mathsf{SNM\text{-}ATK}}_{B,S,\Sigma}(Rel, k)$ is not negligible in k, where k is a security parameter and ATK $\in$ {CPA, CCA1, CCA2}.

Let $A = (A_1, A_2, A_3)$ be an adversary for PNM-ATK. First, we construct SNM-ATK adversary $B = (B_1, B_2)$ and relation Rel using PNM-ATK adversary A as shown in Fig. 8. Here, we say event Bad occurs iff Y is not an element of X. From the construction of B, we obtain the following equivalence for all $k \in \mathbb{N}$:

$$\Pr[\mathtt{Expt}^{\mathsf{PNM\text{-}ATK}}_{A,\Sigma}(k) = 1] = \Pr[\mathtt{Expt}^{\mathsf{SNM\text{-}ATK}}_{B,\Sigma}(Rel, k) = 1] \tag{3}$$

By Eq. (4), we show that, given relation Rel, for any simulator S, the success probability of $\mathtt{Expt}^{\mathsf{SNM\text{-}ATK}}_{S,\Sigma}(Rel, k)$ is at

most $\frac{1}{2}$.

$$\begin{aligned}
&\Pr[\mathtt{Expt}^{\mathsf{SNM\text{-}ATK}}_{S,\Sigma}(Rel,k)=1]\\
&\quad=\Pr[g=b\wedge\neg\mathtt{Bad}]\\
&\quad=\Pr[b=0\wedge g=0\wedge\neg\mathtt{Bad}]+\Pr[b=1\wedge g=1\wedge\neg\mathtt{Bad}]\\
&\quad=\Pr[b=0\wedge\neg\mathtt{Bad}]\times\Pr[g=0|b=0\wedge\neg\mathtt{Bad}]\\
&\qquad+\Pr[b=1\wedge\neg\mathtt{Bad}]\times\Pr[g=1|b=1\wedge\neg\mathtt{Bad}]\\
&\quad\le\frac{1}{2}\times\Pr[g=0|b=0\wedge\neg\mathtt{Bad}]+\frac{1}{2}\times\Pr[g=1|b=1\wedge\neg\mathtt{Bad}]\\
&\qquad\text{(here } b \text{ and } \mathtt{Bad} \text{ are independent of } g)\\
&\quad=\frac{1}{2}\times(\Pr[g=0]+\Pr[g=1])=\frac{1}{2} \qquad (4)
\end{aligned}$$

By applying Eqs. (3), (4) and the above-mentioned assumption that $\mathtt{Adv}^{\mathsf{PNM\text{-}ATK}}_{A,\Sigma}(k)>\mu(k)$, we obtain:

$$\begin{aligned}
&\mathtt{Adv}^{\mathsf{SNM\text{-}ATK}}_{B,S,\Sigma}(Rel,k)\\
&\quad=\Pr[\mathtt{Expt}^{\mathsf{SNM\text{-}ATK}}_{B,\Sigma}(Rel,k)=1]\\
&\qquad-\Pr[\mathtt{Expt}^{\mathsf{SNM\text{-}ATK}}_{S,\Sigma}(Rel,k)=1]\\
&\quad\ge\Pr[\mathtt{Expt}^{\mathsf{PNM\text{-}ATK}}_{A,\Sigma}(k)=1]-\frac{1}{2}\\
&\quad=\mathtt{Adv}^{\mathsf{PNM\text{-}ATK}}_{A,\Sigma}(k)>\mu(k).
\end{aligned}$$

□

4.3 Proof of Theorem 3

Proof. We prove that KEM Σ is not PNM-ATK-KEM if Σ is not CNM-ATK-KEM. More precisely, we show that if there exists adversary A such that $\mathtt{Adv}^{\mathsf{CNM\text{-}ATK}}_{A,\Sigma}(k)$ is not negligible in k, then there exists adversary B such that $\mathtt{Adv}^{\mathsf{PNM\text{-}ATK}}_{B,\Sigma}(k)$ is not negligible in k, where k is the security parameter and ATK $\in$ {CPA, CCA1, CCA2}.

Let $A=(A_1,A_2)$ be an adversary for CNM-ATK. We construct PNM-ATK adversary $B=(B_1,B_2,B_3)$ using CNM-ATK adversary A as shown in Fig. 9. From the construction of B, we obtain

$$\begin{aligned}
&\Pr[\mathtt{Expt}^{\mathsf{PNM\text{-}ATK}}_{B,\Sigma}(k)=1]\\
&\quad=\Pr[b=g]\\
&\quad=\Pr[b=0\wedge g=0]+\Pr[b=1\wedge g=1]\\
&\quad=\Pr[b=0]\times\Pr[g=0|b=0]+\Pr[b=1]\times\Pr[g=1|b=1]\\
&\quad=\frac{1}{2}\Pr[\mathtt{Expt}^{\mathsf{CNM\text{-}ATK}}_{A,\Sigma}(k)=1]\\
&\qquad+\frac{1}{2}(1-\Pr[\widetilde{\mathtt{Expt}}^{\mathsf{CNM\text{-}ATK}}_{A,\Sigma}(k)=1])\\
&\quad=\frac{1}{2}(\Pr[\mathtt{Expt}^{\mathsf{CNM\text{-}ATK}}_{A,\Sigma}(k)=1]\\
&\qquad-\Pr[\widetilde{\mathtt{Expt}}^{\mathsf{CNM\text{-}ATK}}_{A,\Sigma}(k)=1])+\frac{1}{2}.
\end{aligned}$$

That is,

$$\begin{aligned}
&\Pr[\mathtt{Expt}^{\mathsf{PNM\text{-}ATK}}_{B,\Sigma}(k)=1]-\frac{1}{2}\\
&\quad=\frac{1}{2}(\Pr[\mathtt{Expt}^{\mathsf{CNM\text{-}ATK}}_{A,\Sigma}(k)=1]-\Pr[\widetilde{\mathtt{Expt}}^{\mathsf{CNM\text{-}ATK}}_{A,\Sigma}(k)=1])\\
&\quad=\frac{1}{2}\mathtt{Adv}^{\mathsf{CNM\text{-}ATK}}_{A,\Sigma}(k). \qquad (5)
\end{aligned}$$

$\underline{B_1^{O_1}(pk)}$

$t \xleftarrow{R} A_1^{O_1}(pk)$; $s_1 \leftarrow t$; return s_1

$\underline{B_2^{O_2}(X, s_1, C^*)}$, where $s_1 = t$ and $X = K^*$ or R

$R' \xleftarrow{U} \{0,1\}^{l(k)}$; $c \xleftarrow{U} \{0,1\}$

$X' \leftarrow \begin{cases}(R', X), \text{ if } c=0\\(X, R'), \text{ if } c=1\end{cases}$

$(Rel, \boldsymbol{C}) \xleftarrow{R} A_2^{O_2}(X', s_1, C^*)$

$s_2 \leftarrow (Rel, X)$; return $(s_2, \boldsymbol{C})$

$\underline{B_3(s_2, \boldsymbol{K})}$, where $s_2 = (Rel, X)$

If $Rel(X, \boldsymbol{K})$, then $g \leftarrow 0$,
otherwise $g \leftarrow 1$; return g

Fig. 9 PNM-ATK adversary B using CNM-ATK adversary A.

By applying Eq. (5) and the above-mentioned assumption that $\mathtt{Adv}^{\mathsf{CNM\text{-}ATK}}_{A,\Sigma}(k)>\mu(k)$, we obtain

$$\mathtt{Adv}^{\mathsf{PNM\text{-}ATK}}_{B,\Sigma}(k)=\frac{1}{2}\mathtt{Adv}^{\mathsf{CNM\text{-}ATK}}_{A,\Sigma}(k)>\mu(k)/2.$$

□

4.4 Equivalence of the Three Non-malleability Definitions

From Theorems 1, 2 and 3, we immediately obtain the equivalence of the three non-malleable definitions, SNM-ATK-KEM, CNM-ATK-KEM and PNM-ATK-KEM. Hereafter, we use NM-ATK-KEM to refer to the three non-malleable definitions. If it is clear that NM-ATK-KEM is used for KEM, we will call it just NM-ATK.

5. IND-CCA2 KEM is Equivalent to NM-CCA2 KEM

This section shows that non-malleability is equivalent to indistinguishability for KEM against adaptive chosen ciphertext attacks (CCA2). For public-key encryption (PKE), it has been already proven that non-malleability is equivalent to indistinguishability against CCA2 [1].

Theorem 4. KEM Σ *is NM-CCA2-KEM, if and only if* Σ *is IND-CCA2-KEM.*

Proof. To prove this theorem, it is enough to show that PNM-CCA2-KEM is equivalent to IND-CCA2-KEM. It is trivial from the definition that KEM Σ is not IND-CCA2-KEM if Σ is not PNM-CCA2-KEM. The opposite direction, that Σ is not PNM-CCA2-KEM if Σ is not IND-CCA2-

KEM, is also easy as follows: Let $A = (A_1, A_2)$ be an attacker for IND-CCA2-KEM. We then construct an attacker $B = (B_1, B_2, B_3)$ for PNM-CCA2-KEM using A such that B_1 executes A_1, and B_2 executes A_2 which outputs g and outputs $(s_2, \boldsymbol{C})$ such that $s_2 \leftarrow g$ and $\boldsymbol{C}$ is an arbitrary ciphertext. B_3 outputs $s_2 (= g)$ regardless of the value of $\boldsymbol{K}$. Clearly, B is an attacker for PNM-CCA2-KEM with the same advantage as A for IND-CCA2-KEM. □

6. UC KEM

Let $\Sigma = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ be a key encapsulation mechanism (KEM). We define the key encapsulation mechanism functionality $\mathcal{F}_{\mathrm{KEM}}$ and protocol π_Σ that is constructed from KEM Σ and has the same interface with the environment as $\mathcal{F}_{\mathrm{KEM}}$.

Definition 6. *Let $\mathcal{F}_{\mathrm{KEM}}$ be the key encapsulation mechanism functionality shown in Fig. 10, and let π_Σ be the key encapsulation mechanism protocol in Fig. 11.*

Here, note that there is no functionality of data transmission between parties in $\mathcal{F}_{\mathrm{KEM}}$.

7. UC KEM Is Equivalent to IND-CCA2 KEM

This section shows that KEM Σ is UC secure if and only if Σ is IND-CCA2 (or NM-CCA2).

Theorem 5. *Let Σ be a KEM scheme, and $\mathcal{F}_{\mathrm{KEM}}$ and π_Σ be as described in Definition 6. Protocol π_Σ UC-realizes $\mathcal{F}_{\mathrm{KEM}}$ with respect to non-adaptive adversaries, if and only if Σ is IND-CCA2-KEM.*

Proof.
("Only if" part) Let $\Sigma = (\mathcal{G}, \mathcal{E}, \mathcal{D})$ be a KEM scheme. We prove that if Σ is not IND-CCA2-KEM, then π_Σ does not UC-realize $\mathcal{F}_{\mathrm{KEM}}$. In more detail, we can construct environment Z such that, for any ideal process world adversary (simulator) S, Z can tell whether it is interacting with A and π_Σ or with S and the ideal protocol for $\mathcal{F}_{\mathrm{KEM}}$, by using adversary G that breaks Σ in the sense of IND-CCA2-KEM with not-negligible advantage (i.e., $\mathrm{Adv}^{\mathrm{IND-CCA2}}_{G,\Sigma}(k) > \mu(k)$).

Z activates parties E and D, and uses adversary G as follows:

1. Activates key decryptor D with (KEM.KeyGen, *sid*) for *sid*=$(D, 0)$, obtains encryption algorithm e, and hands e to G.
2. Activates E with (KEM.Encrypt, *sid*, e), and obtains (key, cip). Z chooses $b \xleftarrow{\cup} \{0,1\}$ and $R \xleftarrow{\cup} \{0,1\}^{l(k)}$. If $b = 0$, then $key' \leftarrow key$. If $b = 1$, then $key' \leftarrow R$. Z hands (key', cip) to G as a target pair of key and ciphertext in the IND-CCA2 game shown in Fig. 1.
3. When G asks its decryption oracle to decrypt ciphertext $C^\dagger \neq cip$, Z activates D with input (KEM.Decrypt, *sid*, $C^\dagger$), obtains key $K^\dagger$, and hands $K^\dagger$ to G.

Functionality $\mathcal{F}_{\mathrm{KEM}}$

$\mathcal{F}_{\mathrm{KEM}}$ which runs with adversary S proceeds as follows:

Key Generation: Upon receiving (KEM.KeyGen, *sid*) from some party D, verify that *sid*=(D, sid') for some *sid'*. If not, then ignore the request. Else, hand (KEM.KeyGen, *sid*) to adversary S. Upon receiving (Algorithms, *sid*, e, d) from S, where e, d are descriptions of PPT ITMs, output (Encryption Algorithm, *sid*, e) to D.

Encryption: Upon receiving (KEM.Encrypt, *sid*, e') from any party E, do: If $e' \neq e$, or decryptor D is corrupted, then execute e' and obtain (K^*, C^*). Let $(key, cip) \leftarrow (K^*, C^*)$. Else, obtain (K^*, C^*) by e' and $R \xleftarrow{\cup} \{0,1\}^{l(k)}$, then let $(key, cip) \leftarrow (R, C^*)$ and record (key, cip). Output (Key and Ciphertext, *sid*, *key*, *cip*) to E.

Decryption: Upon receiving a value (KEM.Decrypt, *sid*, C^*) from D (and D only), do: If there is a recorded entry (K, C^*) for some K then return (Shared Key, *sid*, K) to D. Else, return (Shared Key, *sid*, $d(C^*)$) to D. (If there are more than one K recorded for C^*, then output an error message.)

Fig. 10 Key encapsulation mechanism functionality $\mathcal{F}_{\mathrm{KEM}}$.

Protocol π_Σ

π_Σ proceeds with parties E and D as follows:

Key Generation: Upon input (KEM.KeyGen, *sid*), party D verifies that *sid*=(D, sid') for some *sid'*. If not, then ignore the request. Else, D obtains public key *pk* and secret key *sk* by running the algorithm $\mathcal{G}$, and generates $e \leftarrow \mathcal{E}(pk, \cdot)$ and $d \leftarrow \mathcal{D}(sk, \cdot)$, then outputs (Encryption Algorithm, *sid*, e).

Encryption: Upon input (KEM.Encrypt, *sid*, e), party E obtains pair $(key, cip) \leftarrow (K^*, C^*)$ of a key and a ciphertext by running algorithm e and outputs (Key and Ciphertext, *sid*, *key*, *cip*).

Decryption: Upon input (KEM.Decrypt, *sid*, C^*), party D (that has d) obtains $K^* \leftarrow d(C^*)$ and outputs (Shared Key, *sid*, K^*).

Fig. 11 Key encapsulation mechanism protocol π_Σ.

4. When G outputs $g \in \{0, 1\}$, Z outputs $g \oplus b$ and halts.

Here note that Z corrupts no party and interacts with no adversary.

When Z interacts with π_Σ, the view of G interacting with Z is exactly the same as that behaving in the real IND-CCA2 game in Fig. 1. Therefore, in this case (say `Real`), $g = b$ with probability $> \frac{1}{2} + \mu(k)$.

In contrast, when Z interacts with the ideal process world for $\mathcal{F}_{\mathrm{KEM}}$, the view of G interacting with Z is independent of b, since b is independent of (key', cip) generated by Z in step 2 and is independent of the decryption result $K^\dagger$ in step 3 (as key' and $K^\dagger$ are random strings independent of b). Hence, in this case (say `Ideal`), $g = b$ with probability of exactly $\frac{1}{2}$.

Thus, $|\Pr[Z \to 0 \mid \texttt{Real}] - |\Pr[Z \to 0 \mid \texttt{Ideal}]| > \mu(k)$.

("If" part) We show that if π_Σ does not UC-realize $\mathcal{F}_{\mathrm{KEM}}$, then Σ is not IND-CCA2-KEM. To do so, we first assume that for any simulator S there exists a real world adversary A and an environment Z that distinguishes with probability $> \frac{1}{2} + \mu(k)$ whether it is interacting with S and the ideal process for $\mathcal{F}_{\mathrm{KEM}}$ or with A and π_Σ. We then show that there exists an IND-CCA2 attacker G against Σ using Z.

First we show that Z can distinguish (A, π_Σ) and $(S, \mathcal{F}_{\mathrm{KEM}})$ only when no party is corrupted. Since we are dealing with non-adaptive adversaries, there are three cases; Case 1: Sender E is corrupted (throughout the protocol), Case 2: Decryptor D is corrupted (throughout the protocol), Case 3: E and D are uncorrupted.

In Case 1, we can construct simulator S such that no Z can distinguish (A, π_Σ) and $(S, \mathcal{F}_{\mathrm{KEM}})$ as follows:

1. When Z sends (`KEM.KeyGen`, *sid*) to D, D forwards it to $\mathcal{F}_{\mathrm{KEM}}$. $\mathcal{F}_{\mathrm{KEM}}$ sends (`KEM.KeyGen`, *sid*) to S, S computes (pk,sk) by running algorithm $\mathcal{G}$, and generates e and d, where $e \leftarrow \mathcal{E}(pk, \cdot)$ and $d \leftarrow \mathcal{D}(sk, \cdot)$. S returns (`Algorithms`, *sid*, e, d) to $\mathcal{F}_{\mathrm{KEM}}$.
2. When Z sends (`KEM.Encrypt`, *sid*, e) to the corrupted party E (i.e., S), S receives the message and sends it to the simulated copy of A, which replies to S. S then returns A's reply (which may be $\perp$) to Z.
3. When Z sends (`KEM.Decrypt`, *sid*, C^*) to D, D forwards it to $\mathcal{F}_{\mathrm{KEM}}$. $\mathcal{F}_{\mathrm{KEM}}$ then returns (`Shared Key`, *sid*, $d(C^*)$), since E (i.e., S) sends no (`KEM.Encrypt`, *sid*, e) to $\mathcal{F}_{\mathrm{KEM}}$, which records nothing as (key, cip). Note that, S does not receive any message in this step.

In this case, Z cannot distinguish (A, π_Σ) from $(S, \mathcal{F}_{\mathrm{KEM}})$, because the message returned by S (using A) as E in the ideal world is the same as that returned by A as E in the real world, and (`Shared Key`, *sid*, $d(C^*)$) returned by $\mathcal{F}_{\mathrm{KEM}}$ is exactly the same as that returned by D in the real world.

In Case 2, we can also construct simulator S such that no Z can distinguish (A, π_Σ) and $(S, \mathcal{F}_{\mathrm{KEM}})$ as follows:

1. When Z sends (`KEM.KeyGen`, *sid*) to the corrupted party D (i.e., S), S receives the message and sends it to the simulated copy of A, which returns a reply message (which may be $\perp$) to S. S sends it to Z.
2. When Z sends (`KEM.Encrypt`, *sid*, e) to E, E forwards it to $\mathcal{F}_{\mathrm{KEM}}$. $\mathcal{F}_{\mathrm{KEM}}$ generates a corresponding pair (K^*, C^*) by executing e, sets $(key, cip) \leftarrow (K^*, C^*)$ and returns (`Key and Ciphertext`, *sid*, *key*, *cip*) to E, since D (i.e., S) sends no (`KEM.KeyGen`, *sid*) to $\mathcal{F}_{\mathrm{KEM}}$, which records nothing as encryption algorithm e.
3. When Z sends (`KEM.Decrypt`, *sid*, C^*) to D (i.e., S), S sends (`KEM.Decrypt`, *sid*, C^*) to A. A returns a reply (which may be $\perp$) to S, which forwards A's reply to Z.

In this case, Z cannot distinguish (A, π_Σ) and $(S, \mathcal{F}_{\mathrm{KEM}})$, because the message returned by S (using A) as D in the ideal world is the same as that returned by A as D in the real world, and (`Key and Ciphertext`, *sid*, *key*, *cip*) returned by $\mathcal{F}_{\mathrm{KEM}}$ is exactly the same as that returned by E in the real world.

Thus, Z cannot distinguish the real/ideal worlds in Cases 1 and 2. Hereafter, we consider only Case 3: E and D are uncorrupted.

Referring to the UC framework, three types of messages are sent from Z to A. The first message type is to corrupt either party, the second message type is to report on message sending, and the third message type is to deliver some message. In our protocol π_Σ, parties don't send messages to each other over the network. In addition, we consider the case that no party is corrupted. Therefore, there are no messages from Z to A (and S).

Since there exists at least one environment Z that can distinguish the real life world from the ideal process world for any simulator S, we consider the following special simulator S:

When S receives message (`KEM.KeyGen`, *sid*) from $\mathcal{F}_{\mathrm{KEM}}$, S runs key generation algorithm $\mathcal{G}$ and obtains public key pk and secret key sk. S sets $e \leftarrow \mathcal{E}(pk, \cdot)$ and $d \leftarrow \mathcal{D}(sk, \cdot)$, and returns (`Algorithms`, *sid*, e, d) to $\mathcal{F}_{\mathrm{KEM}}$.

We now show that we can construct adversary G that breaks IND-CCA2-KEM by using the simulated copy of Z which distinguishes real/ideal worlds. To do so, we assume that there is an environment Z such that

$$|\texttt{IDEAL}_{\mathcal{F}_{\mathrm{KEM}},S,Z}(k, z) - \texttt{REAL}_{\pi_\Sigma,A,Z}(k, z)| > \mu(k).$$

We then show that G using Z can correctly guess b in the IND-CCA2 game in Fig. 1 with probability of at least $\frac{1}{2} + \mu(k)/2\ell$, where ℓ is the total number of times the encryption oracle is invoked.

In the IND-CCA2 game, G, given a target public-key (encryption algorithm) e and a target pair (key, cip) from the encryption oracle with private random bit b, is allowed to query the decryption oracle, and finally outputs g, which is G's guess of b. G runs Z with the following simulated interaction as protocol $\pi_\Sigma/\mathcal{F}_{\mathrm{KEM}}$.

G acts as follows, where K_i^*, C_i^* and R_i denote the i-th key, ciphertext and random value of the length $l(k)$, respectively:

1. When Z activates some party D with (`KEM.KeyGen`, *sid*), G lets D output (`Encryption Algorithms`, *sid*,

e), where e is the target public-key (encryption algorithm) for G in the IND-CCA2 game.
2. For the first h times that Z asks some party E to generate (key, cip) with sid, G lets E return $(key, cip) \leftarrow (K_i^*, C_i^*)$ by using algorithm e.
3. The h-th time that Z asks to generate (key, cip) with sid, G queries its encryption oracle in the IND-CCA2 game, and obtains corresponding pair $(key, cip) \leftarrow (K_h^*, C_h^*)$ (when $b = 0$) or non-corresponding pair $(key, cip) \leftarrow (R_h, C_h^*)$ (when $b = 1$), where $R_h \xleftarrow{\cup} \{0,1\}^{l(k)}$. Accordingly, G hands the pair of (key, cip) to Z.
4. For the remaining $\ell - h$ times that Z asks E to generate (key, cip) with sid, G lets E return $(key, cip) \leftarrow (R_i, C_i^*)$, where $R_i \xleftarrow{\cup} \{0,1\}^{l(k)}$.
5. Whenever Z activates decryptor D with ($\mathtt{KEM.Decrypt}$, sid, C^*), where $C^* = C_i^*$ for some i, G lets D return the corresponding key K_i^* or R_i^* for any i. If C^* is different from all C_i^*'s, G then poses C^* to its decryption oracle, obtains value v, and lets D return v to Z.
6. When Z halts, G outputs whatever Z outputs and halts.

We use a standard hybrid argument to analyze the success probability of G in the IND-CCA2 game.

For $h \in \{0, \ldots, \ell\}$, let $\mathtt{Env}_h$ be an event that for the first h times that Z asks some party E to generate (key, cip) with sid, E returns $(key, cip) \leftarrow (K_i^*, C_i^*)$ by using algorithm e; the h-th time that Z asks E to generate (key, cip) with sid, E returns $(key, cip) \leftarrow (K_i^*, C_i^*)$ or $(key, cip) \leftarrow (R_i, C_i^*)$, where $R_i \xleftarrow{\cup} \{0,1\}^{l(k)}$. For the remaining $\ell - h$ times that Z asks E to generate (key, cip) with sid, E returns $(key, cip) \leftarrow (R_i, C_i^*)$, where $R_i \xleftarrow{\cup} \{0,1\}^{l(k)}$. The replies to Z from decryptor D are the same as those shown in step 5 above.

Let H_h be $\Pr[Z \rightarrow 1|\mathtt{Env}_h]$.We then obtain the following inequality.

$$\sum_{h=1}^{\ell} |H_h - H_{h-1}| \geq |H_\ell - H_0|. \tag{6}$$

Here, from the construction of H_h it is clear that

$$H_0 = \mathtt{IDEAL}_{\mathcal{F}_{\mathrm{KEM}},S,Z}(k, z), \tag{7}$$

$$H_\ell = \mathtt{REAL}_{\pi_\Sigma,A,Z}(k, z). \tag{8}$$

Therefore,

$$\begin{aligned}\sum_{h=1}^{\ell} |H_h - H_{h-1}| &\geq |H_\ell - H_0| \\ &= |\mathtt{REAL}_{\pi_\Sigma,A,Z}(k, z) - \mathtt{IDEAL}_{\mathcal{F}_{\mathrm{KEM}},S,Z}(k, z)| > \mu(k).\end{aligned}$$

Then there exists some $h \in \{1, \cdots \ell\}$ that satisfies

$$|H_h - H_{h-1}| > \mu(k)/\ell. \tag{9}$$

Here, w.l.o.g., let $H_{h-1} - H_h > \mu(k)/\ell$,since if $H_h - H_{h-1} > \mu(k)/\ell$for Z, we can obtain $H_{h-1} - H_h > \mu(k)/\ell$for Z^*, where Z^* outputs the opposite of Z's output bit.

In step 3 of G's construction, if G gets the corresponding pair of (K_h^*, C_h^*) (when $b = 0$), then the probability that Z outputs 1 is identical to H_h. If, on the other hand, G gets the non-corresponding pair of (R_h, C_h^*) (when $b = 1$), then the probability that Z outputs 1 is identical to H_{h-1}.

Since G's output follows Z's output,

$$H_h = \Pr[g = 1|b = 0], \tag{10}$$

$$H_{h-1} = \Pr[g = 1|b = 1], \tag{11}$$

where b is the private random bit of the encryption oracle in the IND-CCA2 game and g is G's output (G's guess of b).

Since $\Pr[g = 1|b = 0] + \Pr[g = 0|b = 0] = 1$,we obtain $\Pr[g = 0|b = 0] = 1 - \Pr[g = 1|b = 0]$.

Therefore, we obtain G's success probability, $\Pr[\mathtt{Expt}_{G,\Sigma}^{\mathrm{IND-CCA2}}(k) = 1]$, as follows:

$$\begin{aligned}&\Pr[\mathtt{Expt}_{G,\Sigma}^{\mathrm{IND-CCA2}}(k) = 1] \\ &\quad = \Pr[b = g] \\ &\quad = \Pr[b = 0] \times \Pr[g = 0|b = 0] \\ &\qquad + \Pr[b = 1] \times \Pr[g = 1|b = 1] \\ &\quad = \frac{1}{2} \times (\Pr[g = 0|b = 0] + \Pr[g = 1|b = 1]) \\ &\quad = \frac{1}{2} \times (1 - \Pr[g = 1|b = 0] + \Pr[g = 1|b = 1]) \\ &\quad = \frac{1}{2} \times (1 - H_h + H_{h-1}) > \frac{1}{2} + \mu(k)/2\ell.\end{aligned}$$

That is, $\mathtt{Adv}_{G,\Sigma}^{\mathrm{IND-CCA2}}(k) > \mu(k)/2\ell$, which is not negligible in k since ℓ is polynomially bounded in k. □

8. UC DEM

Let $\Sigma' = (\mathcal{E}', \mathcal{D}')$ be a DEM scheme and let Σ'' be a $\mathcal{F}_{\mathrm{KEM}}$-hybrid DEM scheme. We define the key encapsulation mechanism and data encapsulation mechanism functionality $\mathcal{F}_{\text{KEM-DEM}}$ and protocol $\pi_{\Sigma''}$ that is constructed from DEM Σ' in the $\mathcal{F}_{\mathrm{KEM}}$ hybrid model and has the same interfaces which the environment Z uses to communicate with $\mathcal{F}_{\text{KEM-DEM}}$.

Definition 7. *Let $\mathcal{F}_{\text{KEM-DEM}}$ be the key encapsulation mechanism and data encapsulation mechanism (KEM-DEM) functionality shown in Fig. 12, and let $\pi_{\Sigma''}$ be the KEM-DEM protocol in Fig. 13.*

Here, note that there is no functionality for the data transmission between parties in $\mathcal{F}_{\text{KEM-DEM}}$, and we consider that the algorithm e in KEM.KeyGen of $\mathcal{F}_{\text{KEM-DEM}}$ outputs different key ciphertext C^*.

The revised point from the previous definition [7], [8] is to remove the restriction that the previous $\mathcal{F}_{\text{KEM-DEM}}$ can have only one key in the DEM phase. To solve this problem, we made current functionality accept the multiple key ciphertexts generated by (DEM.Decrypt, sid, c, C') in DEM.Decrypt of $\mathcal{F}_{\text{KEM-DEM}}$, where c is the ciphertext of a message and C' is the encryption of some key.

Functionality $\mathcal{F}_{\text{KEM-DEM}}$

$\mathcal{F}_{\text{KEM-DEM}}$ proceeds as follows, running with party $P \in \{E_1, \ldots, E_n, D\}$ and adversary S.

KEM.KeyGen: Upon receiving (KEM.KeyGen, *sid*) from key decryptor D, verify that $sid=(D, sid')$ for some sid'. If not, then ignore the request. Else, hand (KEM.KeyGen, *sid*) to adversary S. Upon receiving (Algorithms, *sid*, e, d, e_{DEM}, d_{DEM}) from S, where e, d, e_{DEM} and d_{DEM} are descriptions of PPT TMs, output (KEM Encryption Algorithm, *sid*, e) to D.

KEM.Encrypt: Upon receiving (KEM.Encrypt, *sid*, e') from key encryptor $E_i (i \in \{1, \ldots, n\})$, do:

- If $e' \neq e$, or key decryptor D is corrupted, then obtain K and C^* by e', record $(E_i, K, C^*, 0)$ and send (KEM.Ciphertext, *sid*, C^*) to E_i.
- Else, obtain C^* by e' and $K \xleftarrow{\cup} \{0,1\}^{l(k)}$, record $(E_i, K, C^*, 1)$ and send (KEM.Ciphertext, *sid*, C^*) to E_i.

KEM.Decrypt: Upon receiving (KEM.Decrypt, *sid*, C') from key decryptor D (and D only), do:

- If C' is in the memory $(E_i, K, C', 1)$ for some E_i and K, record $(D, K, C', 1)$ and send *ok* to D.
- Else, record $(D, d(C'), C', 0)$ and send *ok* to D.

DEM.Encrypt: Upon receiving (DEM.Encrypt, *sid*, m, C') from party P, proceed as follows:

- If $(P, K, C', 1)$ is recorded in the memory for some K and $\tilde{P}$ is uncorrupted ($\tilde{P}$ denotes D if P is E_i, $\tilde{P}$ denotes $E_i^{C'}$ if P is D, where $E_i^{C'}$ denotes the party such that $(E_i, *, C', 1)$ is recorded), then do as follows:
  1. Generate c by $e_{\text{DEM}}(K, \mu)$, where μ is a fixed message, and record (m, c, C') in the memory.
  2. Send (DEM.Ciphertext, *sid*, c) to P.
- Else if ($(P, K, C', 1)$ is recorded and $\tilde{P}$ is corrupted) or $(P, K, C', 0)$ is recorded in the memory for some K, then do as follows:
  1. Generate c by $e_{\text{DEM}}(K, m)$.
  2. Send (DEM.Ciphertext, *sid*, c) to P.
- Else, do nothing.

DEM.Decrypt: Upon receiving (DEM.Decrypt, *sid*, c, C') from party P, proceed as follows:

- If $(P, K, C', 1)$ is recorded in the memory for some K and (m, c, C') is recorded, then send (DEM.Plaintext, *sid*, m) to P.
- Else if $(P, K, C', *)$ is recorded in the memory for some K, send (DEM.Plaintext, *sid*, $d_{\text{DEM}}(K, c)$) to P.
- Else, do nothing.

Fig. 12 The KEM-DEM functionality.

Protocol $\pi_{\Sigma''}$

$\pi_{\Sigma''}$ proceeds as follows, running with party $P \in \{E_1, \ldots, E_n, D\}$ and an ideal functionality $\mathcal{F}_{\text{KEM}}$.

KEM.KeyGen: Upon input (KEM.KeyGen, *sid*) within key decryptor D,

1. D sends (KEM.KeyGen, sid') to $\mathcal{F}_{\text{KEM}}$.
2. Upon receiving (KEM Key, sid', e) from $\mathcal{F}_{\text{KEM}}$, D outputs (KEM Encryption Algorithm, *sid*, e).

KEM.Encrypt: Upon input (KEM.Encrypt, *sid*, e') within key encryptor E_i,

1. E_i sends (KEM.Encrypt, sid', e') to $\mathcal{F}_{\text{KEM}}$.
2. Upon receiving (Key and Ciphertext, sid', K, C^*) from $\mathcal{F}_{\text{KEM}}$, stores (K, C^*) in E_i's memory.
3. E_i outputs (KEM.Ciphertext, *sid*, C^*).

KEM.Decrypt: Upon input (KEM.Decrypt, *sid*, C') within D,

1. D sends (KEM.Decrypt, sid', C^*) to $\mathcal{F}_{\text{KEM}}$.
2. Upon receiving (Shared Key, sid', K), D stores (K, C') in D's memory.
3. D outputs *ok*.

DEM.Encrypt: Upon input (DEM.Encrypt, *sid*, m, C') within party P, proceed as follows:

- If (K, C') exists in P's memory, P obtains ciphertext $c = \mathcal{E}'(K, m)$ and outputs (DEM.Ciphertext, *sid*, c).
- Else, do nothing.

DEM.Decrypt: Upon input (DEM.Decrypt, *sid*, c, C') within party P, proceed as follows:

- If (K, C') exists in P's memory, P obtains message $m = \mathcal{D}'(K, c)$ and outputs (DEM.Plaintext, *sid*, m).
- Else, do nothing.

Fig. 13 The KEM-DEM protocol.

9. UC DEM Is Equivalent to IND-P2-C2 DEM

This section shows that DEM Σ' is UC secure if and only if Σ' is IND-P2-C2 by using the UC hybrid model.

The following theorem implies that UC DEM is equivalent to IND-P2-C2 DEM.

Theorem 6. *Let $\Sigma' = (\mathcal{E}', \mathcal{D}')$ be a DEM scheme, let Σ'' be a $\mathcal{F}_{\text{KEM}}$-hybrid DEM scheme. Let $\mathcal{F}_{\text{KEM-DEM}}$ and $\pi_{\Sigma''}$ be as described in Definition 7. Protocol $\pi_{\Sigma''}$ UC-realizes $\mathcal{F}_{\text{KEM-DEM}}$ with respect to non-adaptive adversaries in the $\mathcal{F}_{\text{KEM}}$-hybrid model, if and only if Σ' is IND-P2-C2-DEM.*

Proof.

("only if" part) We prove that if $\pi_{\Sigma''}$ is not IND-P2-C2-DEM secure in the $\mathcal{F}_{\text{KEM}}$ - hybrid model, then $\pi_{\Sigma''}$ does not UC-realize $\mathcal{F}_{\text{KEM-DEM}}$. In more detail, we can construct an environment Z such that, for any ideal process world adversary (simulator) S, Z can tell whether it is interacting with A and $\pi_{\Sigma''}$ in the $\mathcal{F}_{\text{KEM}}$ hybrid model or with S and the ideal protocol for $\mathcal{F}_{\text{KEM-DEM}}$ by using adversary F that breaks IND-P2-C2-DEM with non-negligible advantage (i.e., $\mathbf{Adv}^{\text{IND-P2-C2}}_{F,\Sigma''}(k) > \mu(k)$)).

Z activates party E_i and D, and uses adversary F as follows:

1. Activates key receiver D with (KEM.KeyGen, *sid*) for $sid = (D, 0)$, obtains encryption algorithm e.
2. Activates key encryptor E_i with (KEM.Encrypt, *sid*, e'), obtains C^* from the output (KEM.Ciphertext, *sid*, C^*).
3. Activates D with (KEM.Decrypt, *sid*, C^*), obtains *ok*.
4. When F generates the two plaintext (m_0, m_1), Z chooses $b \xleftarrow{\cup} \{0, 1\}$, activates E_i with (DEM.Encrypt, *sid*, m_b, C^*), then obtains c from the output (DEM.Ciphertext, *sid*, c). Z hands c to F in the IND-P2-C2-DEM game shown in Fig. 2.
5. When F asks its encryption oracle to encrypt message $m^\flat$ (which may be m_0 or m_1), Z activates E_i with input (DEM.Encrypt, *sid*, $m^\flat$, C^*), obtains ciphertext $c^\flat$, and hands $c^\flat$ to F.
6. When F asks its decryption oracle to decrypt ciphertext $c^\dagger \neq c$, Z activates D with input (DEM.Decrypt, *sid*, $c^\dagger$, C^*), obtains message $m^\dagger$, and hands $m^\dagger$ to F.
7. When F outputs $g \in \{0, 1\}$, Z outputs $g \oplus b$ and halts.

Here note that Z corrupts no party and interacts with no adversary.

When Z interacts with $\pi_{\Sigma''}$, the view of F interacting with Z is exactly the same as that behaving in the real IND-P2-C2 game in Fig. 2. Therefore, in this case (say Real), $g = b$ with probability $> \frac{1}{2} + \mu(k)$.

In contrast, when Z interacts with the ideal process world for $\mathcal{F}_{\text{KEM-DEM}}$, the view of F interacting with Z is independent of b, since b is independent of (m_0, m_1, c, μ) in step 4, and is independent of the encryption and decryption result $c^\flat$ and $m^\dagger$ in steps 5 and 6 (as $c^\flat$, m_0, m_1 and $m^\dagger$ are random strings independent of b). Hence, in this case (say Ideal), $g = b$ with probability of exactly $\frac{1}{2}$.

Thus, $|\Pr[Z \to 0 \,|\, \text{Real}] - |\Pr[Z \to 0 \,|\, \text{Ideal}]| > \mu(k)$.

("if" part) We show that if $\pi_{\Sigma''}$ does not UC-realize $\mathcal{F}_{\text{KEM-DEM}}$ in the $\mathcal{F}_{\text{KEM}}$-hybrid model, then $\pi_{\Sigma''}$ is not IND-P2-C2-DEM. To do so, we first assume that for any simulator S there is an adversary A and an environment Z that distinguishes with probability $> \frac{1}{2} + \mu(k)$ whether it interacts with S and $\mathcal{F}_{\text{KEM-DEM}}$ or with A and $\pi_{\Sigma''}$. We then show that there exists an IND-P2-C2-DEM attacker F against Σ'' using Z in the $\mathcal{F}_{\text{KEM}}$-hybrid model.

First we show that Z can distinguish $(A, \pi_{\Sigma''})$ in the $\mathcal{F}_{\text{KEM}}$-hybrid model and $(S, \mathcal{F}_{\text{KEM-DEM}})$ only when no party is corrupted. Since we are dealing with non-adaptive adversaries, there are three cases; Case 1: Sender E_i is corrupted (throughout the protocol), Case 2: Decryptor D is corrupted (throughout the protocol), Case 3: E_i and D are uncorrupted.

These cases are done in the $\mathcal{F}_{\text{KEM}}$-hybrid model, so Z can't tell whether Z interacts with the protocol $\pi_{\Sigma''}$ or the ideal $\mathcal{F}_{\text{KEM-DEM}}$ in the KEM= $(\mathcal{G}, \mathcal{E}, \mathcal{D})$ phase. The KEM phase in all cases is done as follows:

1. When Z sends (KEM.KeyGen, *sid*) to D, $\mathcal{F}_{\text{KEM-DEM}}$ sends (KEM.KeyGen, *sid*) to S, S computes (pk,sk) by running algorithm $\mathcal{G}$, and generates e, d, e_{DEM} and d_{DEM} where $e \leftarrow \mathcal{E}(pk, \cdot)$, $d \leftarrow \mathcal{D}(sk, \cdot)$, $e_{\text{DEM}} \leftarrow \mathcal{E}'$ and $d_{\text{DEM}} \leftarrow \mathcal{D}'$. S returns (Algorithms, *sid*, e, d, e_{DEM}, d_{DEM}) to $\mathcal{F}_{\text{KEM-DEM}}$ and $\mathcal{F}_{\text{KEM-DEM}}$ forwards (KEM Encryption Algorithm, *sid*, e) to D.
2. When Z sends (KEM.Encrypt, *sid*, e) to the corrupted party E_i, E_i receives the output (KEM.Ciphertext, *sid*, C^*).
3. When Z sends (KEM.Decrypt, *sid*, C^*) to D, D receives the output *ok*.

We assume that Z can't distinguish the ideal/real world in the KEM phase of all cases. (hereafter, we discuss all cases after the KEM phase is finished.)

In Case 1, we can construct simulator S such that no Z can distinguish $(A, \pi_{\Sigma''})$ in the $\mathcal{F}_{\text{KEM}}$-hybrid model and $(S, \mathcal{F}_{\text{KEM-DEM}})$ as follows:

1. When Z sends (DEM.Encrypt, *sid*, m, C^*) to the corrupted party E_i (i.e., S), S receives the message and sends it to the simulated copy of A, which replies to S. S then returns A's reply (which may be $\perp$) to Z.
2. When Z sends (DEM.Decrypt, *sid*, c, C^*) to D, D forwards it to $\mathcal{F}_{\text{KEM-DEM}}$. $\mathcal{F}_{\text{KEM-DEM}}$ then returns (DEM.Plaintext, *sid*, $d_{\text{DEM}}(K, c)$) since E (i.e., S) sends no (DEM.Encrypt, *sid*, m, C') to $\mathcal{F}_{\text{KEM-DEM}}$, which records nothing as (m, c, C'). Note that, S does not receive any message in this step.

In this case, Z cannot distinguish $(A, \pi_{\Sigma''})$ and $(S, \mathcal{F}_{\text{KEM}})$, because the message returned by S as E_i in the ideal world is the same as that returned by A as E_i in the real world, and (DEM.Plaintext, *sid*, $d_{\text{DEM}}(K, c)$) returned by $\mathcal{F}_{\text{KEM-DEM}}$ is exactly the same as that returned by D in the real world.

In Case 2, we can also construct simulator S such that

no Z can distinguish $(A, \pi_{\Sigma''})$ and $(S, \mathcal{F}_{\text{KEM-DEM}})$ as follows:

1. When Z sends (DEM.Encrypt, *sid*, m, C^*) to E_i, E_i forwards it to $\mathcal{F}_{\text{KEM-DEM}}$. $\mathcal{F}_{\text{KEM-DEM}}$ generates c by $e_{\text{DEM}}(K, m)$ and returns (DEM.Ciphertext, *sid*, c) to P to E_i, since D (i.e., S) is corrupted by A, which records nothing as ciphertext c.
2. When Z sends (DEM.Decrypt, *sid*, c, C^*) to D (i.e., S), S sends it to A. A returns a reply (which may be $\perp$) to S, which forwards A's reply to Z.

In this case, Z cannot distinguish $(A, \pi_{\Sigma''})$ from $(S, \mathcal{F}_{\text{KEM-DEM}})$, because the message returned by S (using A) as D in the ideal world is the same as that returned by A as D in the real world, and (DEM.Decrypt, *sid*, c, C^*) returned by $\mathcal{F}_{\text{KEM-DEM}}$ is exactly the same as that returned by E_i in the real world.

Thus, Z cannot distinguish the real/ideal worlds in Cases 1 and 2. Hereafter, we consider only Case 3: E_i and D are uncorrupted.

Referring to the UC framework, three types of messages are sent from Z to A. The first message type is to corrupt either party, the second message type is to report on message sending, and the third message type is to deliver some message. In our protocol $\pi_{\Sigma''}$, parties don't send messages to each other over the network. In addition, we consider the case that no party is corrupted. Therefore, there are no messages from Z to A (and S).

Since there exists at least one environment Z that can distinguish the real life world from the ideal process world for any simulator S, we consider the following special simulator S:

- Receiving (KEM.KeyGen, *sid*) from $\mathcal{F}_{\text{KEM-DEM}}$, S obtains (pk, sk) by running F and sets KEM encryption algorithm $e \leftarrow \mathcal{E}(pk)$ and KEM decryption algorithm and $d \leftarrow \mathcal{D}(sk, \cdot)$. S then chooses DEM encryption algorithm $e_{\text{DEM}} \leftarrow \mathcal{E}'$ and DEM decryption algorithm $d_{\text{DEM}} \leftarrow \mathcal{D}'$ and sends (Algorithms, *sid*, e, d, e_{DEM}, d_{DEM}) to $\mathcal{F}_{\text{KEM-DEM}}$.

We now show that we can construct adversary F that breaks IND-P2-C2-DEM by using the simulated copy of Z which distinguishes real/ideal worlds in the $\mathcal{F}_{\text{KEM}}$-hybrid model. To do so, we assume that there is an environment Z such that

$$|\text{IDEAL}_{\mathcal{F}_{\text{KEM-DEM}},S,Z}(k, z) - \text{REAL}_{\pi_{\Sigma''},A,Z}(k, z)| > \mu(k),$$

when Z communicates with the message sending party $E_i \in \{E_1, \cdots, E_n\}$ and the message receiving party D.

We then show that F using Z can correctly guess b in the IND-P2-C2 game in Fig. 2 with probability of at least $\frac{1}{2} + \mu(k)/2n\ell$, where ℓ is the total number of times the encryption oracle is invoked and n is the number of all message sending parties E_i ($i \in \{1, \cdots, n\}$).

In the IND-P2-C2 game, F, chooses a target message pair (x_0, x_1) with $|x_0| = |x_1|$, given ciphertext y with private random bit $b \xleftarrow{\cup} \{0, 1\}$ selected by the encryption oracle, is allowed to query the encryption and decryption oracles, and finally outputs g, which is F's guess of b. F runs Z with the following simulated interaction as protocol $\pi_{\Sigma''}/\mathcal{F}_{\text{KEM-DEM}}$ in the $\mathcal{F}_{\text{KEM}}$-hybrid model.

F acts as follows, where k, ℓ, m_j, c_j, K_i, C_i, K_{atk} and C_{atk} denote the security parameter, the total number of encrypting messages that Z activates some party E_i with DEM.Encrypt, the j-th message, the j-th ciphertext, the key of F' choosing for message sending party E_i, the ciphertext of key for E_i, the shared key gained by using $\mathcal{F}_{\text{KEM}}$ between the message sending party E_{atk} and the message receiving party D, and the key ciphertext of K_{atk}, respectively: For some $h \in \{0, \cdots, \ell\}$,

1. F randomly selects one party E_{atk}.
2. For the first h times that Z activates some party E_i with (DEM.Encrypt, *sid*, m_j, C_i) to encrypt some message m_j, if $E_i \neq E_{atk}$, F lets E_i return $c_j \xleftarrow{\text{R}} e_{\text{DEM}}(K_i, m_j)$, where $K_i \xleftarrow{\cup} \{0, 1\}^{l(k)}$ is F's chosen key for party E_i. Else (i.e., $E_i = E_{atk}$), and F lets E_{atk} return c_j after asking F's encryption oracle on m_j.
3. The h-th time that Z activates E_i with (DEM.Encrypt, *sid*, m_h, C_{atk}), if $E_i \neq E_{atk}$, F halts. Else (i.e., $E_i = E_{atk}$), then F queries $(x_0, x_1) \leftarrow (m_h, \mu)$ to its encryption oracle in the IND-P2-C2 game, and obtains corresponding ciphertext $c_h \leftarrow e_{\text{DEM}}(K_{atk}, m_h)$ (when $b = 0$) or non-corresponding ciphertext $c_h \leftarrow e_{\text{DEM}}(K_{atk}, \mu)$ (when $b = 1$). F lets E_{atk} return c_h to Z.
4. For the remaining $\ell - h$ times that Z activates some party E_i with (DEM.Encrypt, *sid*, m_j, C_i) to encrypt some message m_j, if $E_i \neq E_{atk}$, F lets E_i return $c_j \xleftarrow{\text{R}} e_{\text{DEM}}(K_i, \mu)$, where μ is the fixed message. Else (i.e., $E_i = E_{atk}$), then F lets E_{atk} return c_j after asking the F's encryption oracle on μ.
5. Whenever Z activates D with (DEM.Decrypt, *sid*, c, C_i), where $c = c_j$ for some j, F lets D return the corresponding message m_j. Here, if c is not all c_j then F makes the decryption message of c_j with the key K_i for C_i and lets D return it to Z. Here, if $C_i = C_{atk}$ then F asks to its decryption oracle with c_j, obtains value v, and lets D return v to Z.
6. When Z halts, F outputs whatever Z outputs and halts.

Here, we also use a standard hybrid argument to analyze F's success probability in the IND-P2-C2 game.

For $h \in \{0, \ldots, \ell\}$, let Env_h be an event that for the first h times that Z asks some party E_i (which may be E_{atk}) to generate ciphertext c_j with *sid*, E_i returns m_j's encryption c_j according to the above mentioned ways, for the h-th time that Z asks E_i (which may be E_{atk}) to generate ciphertext c_j with *sid*, E_i returns m_j's encryption or μ's encryption and for the remaining $\ell - h$ times that Z asks E_i (which may be E_{atk}) to generate c_j with *sid*, E_i returns μ's encryption c_j. The replies to Z from decryptor D are the same as those shown in step 5 above.

Let H_h be $\Pr[Z \to 1|\text{Env}_h]$.We then obtain the following inequality.

$$\sum_{h=1}^{\ell} |H_h - H_{h-1}| \geq |H_\ell - H_0|. \quad (12)$$

Here, from the construction of H_h it is clear that

$$H_0 = \mathtt{IDEAL}_{\mathcal{F}_{\text{KEM-DEM}},S,Z}(k,z), \quad (13)$$

$$H_\ell = \mathtt{REAL}_{\pi_{\Sigma''},A,Z}(k,z). \quad (14)$$

Therefore,

$$\begin{aligned}\sum_{h=1}^{\ell} |H_h - H_{h-1}| &\geq |H_\ell - H_0| \\ &= |\mathtt{REAL}_{\pi_{\Sigma''},A,Z}(k,z) - \mathtt{IDEAL}_{\mathcal{F}_{\text{KEM-DEM}},S,Z}(k,z)| \\ &> \mu(k).\end{aligned} \quad (15)$$

Then there exists some $h \in \{1, \cdots \ell\}$ that satisfies

$$|H_h - H_{h-1}| > \mu(k)/\ell. \quad (16)$$

Here, w.l.o.g., let $H_{h-1} - H_h > \mu(k)/\ell$,since if $H_h - H_{h-1} > \mu(k)/\ell$for Z, we can obtain $H_{h-1} - H_h > \mu(k)/\ell$for Z^*, where Z^* outputs the opposite of Z's output bit.

In step 3 of F's construction, F can continue the IND-P2-C2-DEM game, when the h-th time activation occurs on just E_{atk}. The probability that Z activates E_{atk} from all parties $E_i \in \{E_0, \cdots, E_n\}$ is $1/n$. If F gets the corresponding pair of (m_h, c_h) (when $b = 0$), then the probability that Z outputs 1 is identical to H_h/n. If, on the other hand, F gets the non-corresponding ciphertext of (μ, c_j) (when $b = 1$), then the probability that Z outputs 1 is identical to H_{h-1}/n.

Since F's output follows Z's output,

$$\Pr[g = 1|b = 0] = H_h/n, \quad (17)$$

$$\Pr[g = 1|b = 1] = H_{h-1}/n, \quad (18)$$

where b is the private random bit of the encryption oracle in the IND-P2-C2 game and g is F's output (F's guess of b).

Since $\Pr[g = 1|b = 0] + \Pr[g = 0|b = 0] = 1$, we obtain $\Pr[g = 0|b = 0] = 1 - \Pr[g = 1|b = 0]$.

Therefore, from the above equalities, we obtain F's success probability,
$\Pr[\mathtt{Expt}^{\text{IND-P2-C2}}_{F,\Sigma''}(k) = 1]$,as follows:

$$\begin{aligned}&\Pr[\mathtt{Expt}^{\text{IND-P2-C2}}_{F,\Sigma''}(k) = 1] \\ &= \Pr[b = g] \\ &= \Pr[b = 0] \times \Pr[g = 0|b = 0] \\ &\quad + \Pr[b = 1] \times \Pr[g = 1|b = 1] \\ &= \frac{1}{2} \times (\Pr[g = 0|b = 0] + \Pr[g = 1|b = 1]) \\ &= \frac{1}{2} \times (1 - \Pr[g = 1|b = 0] + \Pr[g = 1|b = 1]) \\ &= \frac{1}{2} \times \left(1 - \frac{H_h}{n} + \frac{H_{h-1}}{n}\right) > \frac{1}{2} + \mu(k)/2n\ell.\end{aligned}$$

That is, $\mathtt{Adv}^{\text{IND-P2-C2}}_{F,\Sigma''}(k) > \mu(k)/2n\ell$,which is not negligible in k since n and ℓ are polynomially bounded in k.

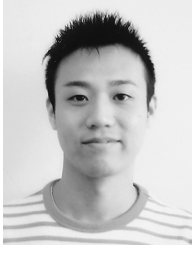
Finally, we conclude that if $\pi_{\Sigma''}$ does not UC-realize $\mathcal{F}_{\text{KEM-DEM}}$ in the $\mathcal{F}_{\text{KEM-DEM}}$-hybrid model, then $\pi_{\Sigma''}$ is not IND-P2-C2 DEM. □

Acknowledgements

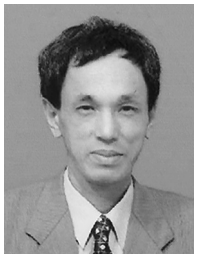
The authors would like to thank Masayuki Abe for his comment on a problem with our non-malleability formulation in [8].

References

[1] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway, "Relations among notions of security for public-key encryption schemes," CRYPTO'98, LNCS 1462, pp.26–45, Springer Verlag, 1998.
[2] M. Bellare and A. Sahai, "Non-malleable encryption: Equivalence between two notions, and an indistinguishability-based characterisation," CRYPTO'99, LNCS 1666, pp.519–536, Springer Verlag, 1999.
[3] R. Canetti, "Universally composable security: A new paradigm for cryptographic protocols," 42nd FOCS, 2001. IACR ePrint Archive 2000/067, http://eprint.iacr.org
[4] D. Dolev, C. Dwork, and M. Naor, "Non-malleable cryptography," 23rd ACM Annual ACM Symposium on Theory of Computing, pp.542–552, ACM, 1991.
[5] J. Herranz, D. Hofheinz, and E. Kiltz, "KEM/DEM: Necessary and sufficient conditions for secure hybrid encryption," IACR ePrint Archive 2006/265, http://eprint.iacr.org
[6] J. Katz and M. Yung, "Characterization of security notions for probabilistic private-key encryption," J. Cryptol., vol.19, no.1, pp.67–96, 2006.
[7] W. Nagao, Y. Manabe, and T. Okamoto, "A universally composable secure channel based on the KEM-DEM framework," TCC'05, LNCS 3378, pp.426–444, Springer Verlag, 2005.
[8] W. Nagao, Y. Manabe, and T. Okamoto, "A universally composable secure channel based on the KEM-DEM framework," IEICE Trans. Fundamentals, vol.E89-A, no.1, pp.28–38, Jan. 2006.
[9] S. Miyagawa, K. Yoneyama, K. Ohta, and N. Kunihiro, "Non-malleability for KEM and Tag-KEM reconsidered," SCIS2007, 4C1-1, Jan. 2007.
[10] V. Shoup, "A proposal for an ISO standard for public key encryption (version 2.1)," ISO/IEC JTC1/SC27, N2563, Dec. 2001. http://shoup.net/papers/



Waka Nagao received the B.E. and M.E. degrees from Osaka Prefecture University and Kyoto University, Osaka and Kyoto, Japan, in 2003 and 2005, respectively. Currently, he is a doctor course student of Kyoto University. His research interests are cryptography and information security.



Yoshifumi Manabe received the B.E., M.E., and Dr.E. degrees from Osaka University, Osaka, Japan, in 1983, 1985, and 1993, respectively. In 1985, he joined Nippon Telegraph and Telephone Corporation. Currently, he is a senior research scientist, supervisor of NTT Communication Science Laboratories. His research interests include distributed algorithms, cryptography, and operating systems. He has been a guest associate professor of Kyoto University since 2001. He is a member of ACM, IPSJ, and IEEE.

Tatsuaki Okamoto received the B.E., M.E., and Dr.E. degrees from the University of Tokyo, Tokyo, Japan, in 1976, 1978, and 1988, respectively. He is a Fellow of NTT Information Sharing Platform Laboratories. He is presently engaged in research on cryptography and information security. Dr. Okamoto is a director of the Japan Society for Industrial and Applied Mathematics, and a guest professor of Kyoto University.