The ECCA Security of Hybrid Encryptions

Honglong Dai[1], Jinyong Chang[1,2], Zhenduo Hou[1], and Maozhi Xu[1(✉)]

[1] School of Mathematics, Peking University, Beijing, People's Republic of China
{daihonglong,changjinyong,darthvader13}@pku.edu.cn, mzxu@math.pku.edu.cn
[2] School of Information and Control Engineering, Xi'an University of Architecture and Technology, Xi'an, Shannxi, People's Republic of China

Abstract. In PKC 2014, Dana Dachman-Soled, et al. introduced enhanced chosen-ciphertext security (ECCA) for public key encryption. The enhancement refers to that the decryption oracle provided to the adversary is augmented to return not only the output of the decryption algorithm on a queried cipher-text but also of a randomness-recovery algorithm associated to the scheme. The authors have given the application of ECCA-secure encryption and we believe that ECCA security will find more application in the future. In this paper, we consider ECCA security of the well-known hybrid encryption (Tag-KEM/DEM) which was presented by Masayuki Abe, et al. in EUROCRYPT 2005. Meanwhile, we also consider ECCA security of hybrid encryption (KEM/Tag-DEM). We have proved that the hybrid encryption is secure against enhanced chosen cipher-text attack (ECCA) if both KEM part and DEM part satisfy some assumptions.

Keywords: Hybrid encryption
Enhanced chosen cipher-text attack security (ECCA)
Chosen cipher-text attack security (CCA)

1 Introduction

Secure encryption is the most basic task in cryptography, and some significant works have gone into defining and attaining it. In many commonly accepted definitions, such as chosen-plaintext attack (CPA) security and chosen-ciphertext attack (CCA) security, CCA security means that the adversary obtains no information about messages encrypted in other ciphertexts even she is allowed to query a decryption oracle on specifically chosen ciphertexts, therefore the CCA security has been accepted as the standard requirement for encryption schemes. However, in some conditions, randomness-recovering encryption is important, such as adaptive functions [8]and PKE with non-interactive opening [6]. ECCA security is motivated by the concept of randomness-recovering encryption, which was presented by Dana Dachman-Soled et al. [4]. The enhanced chosen ciphertext attack security means that the decryption oracle provided to the adversary not only outputs the decryption algorithm on a queried ciphertext but also a randomness-recovery algorithm associated to the scheme [11]. Furthermore,

J. K. Liu and P. Samarati (Eds.): ISPEC 2017, LNCS 10701, pp. 847–859, 2017.
https://doi.org/10.1007/978-3-319-72359-4_53

the authors have given many public-key encryptions satisfying ECCA security and the application of ECCA security. In this paper, our results mainly concern the case in which the randomness-recovering algorithm is efficient. ECCA security is of both practical and theoretical interest.

The first standard-model construction of CCA-secure randomness-recovering PKE was achieved by Peikert and Waters [11] but public key encryption is too slow for encrypting long messages and big data. Under such a circumstance, the hybrid encryption method, which means encrypting a key k used for symmetric encryption to encrypt the messages by asymmetric encryption, has been created. In order to obtain secure ECCA hybrid encryption, we consider the ECCA security of hybrid public key encryptions. Cramer and Shoup proved that the hybrid encryption scheme `(Tag-KEM/DEM)` satisfies CCA secure if the part of `KEM` is CCA secure and the part of `DEM` also satisfies CCA secure [13]. Masayuki Abe, et al. presented a hybrid encryption scheme `(Tag-KEM/DEM)` which provided a simple way to create threshold versions of CCA-secure hybrid encryption schemes [2]. R. Canetti, H. Krawczyk, and J. Nielsen proposed a relaxed variant of CCA security, called Replayable CCA (RCCA) security [3]. Chen and Dong considered RCCA security for the KEM+DEM paradigm. They also considered RCCA security for `(Tag-KEM/DEM)` and `KEM/Tag-DEM` paradigm [10]. Motivated by their work, we consider the ECCA security of the `Tag-KEM/DEM` paradigm and its of the `KEM/Tag-DEM` paradigm.

Organizations of the Paper. In Sect. 2, we introduce some basic notations and definitions of the building blocks. In Sect. 3, we recall the definition of well known hybrid encryptions, `KEM/Tag-DEM` and `Tag-KEM/DEM`. Then we prove its `ECCA` security in detail. Conclusions can be found in Sect. 4.

2 Preliminaries

In this section, we will review some useful notations and definitions.

Notations. Let $\mathbb{N}$ be the set of natural numbers. If M is a set, then $|M|$ denotes its size and $m \xleftarrow{R} M$ denotes the operation of picking an element m uniformly at random from M. We denote λ as the security parameter. For notational clarity we usually omit it as an explicit parameter. PPT denotes probabilistic polynomial time. Let $z \leftarrow A(x, y, \cdots)$ denote the operation of running an algorithm $\mathcal{A}$ with inputs $(x, y, \cdots)$ and output z. We say a function $\mathrm{negl}(\lambda)$ is *negligible* (in λ) if $\lambda > k_0$ and $k_0 \in \mathbb{Z}$, $\mathrm{negl}(\lambda) < \lambda^{-c}$ for any constant $c > 0$.

2.1 ECCA Security Definition

A public-key encryption scheme `PKE=(Gen,Enc,Dec)` consists of three algorithm. `Gen` is a probabilistic algorithm that on input the security parameter λ, outputs public keys and privates keys (pk, sk) and pk defines the message space M. `Enc` is a probabilistic algorithm that encrypts a message $m \in M$ into a ciphertext c. `Dec` is a deterministic algorithm that decrypts c and outputs

either $m \in M$ or a special symbol $\perp$. An adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ is a probabilistic polynomial-time oracle query machine. We now describe the attack game between a challenger and an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ used to define security against adaptive Enhanced chosen ciphertext attack.

- **stage 1:** The adversary queries a key generation oracle. The key generation oracle runs $(pk, sk) \leftarrow \mathtt{Gen}(\lambda)$ and responds adversary $\mathcal{A}$ with pk.
- **stage 2:** The adversary makes a sequence of calls to a decryption oracle. For each decryption oracle query, the adversary $\mathcal{A}_1$ submits a ciphertext c to $\mathtt{Dec}^*$. The decryption oracle responds with $m \leftarrow \mathtt{Dec}(sk, c)$ and the random recovery algorithm $\mathtt{Dec}$ responds with $r \leftarrow \mathtt{Rec}(sk, c)$. We require that for all the messages $m \in M$ (M is the space of message), $(pk, sk) \leftarrow \mathtt{Gen}(1^\lambda)$,

$$\Pr[\mathtt{Enc}(pk, m; r^{'}) \neq c; r \xleftarrow{R} \{0,1\}^\lambda; c \leftarrow \mathtt{Enc}(pk, r, m_b); r^{'} \leftarrow \mathtt{Rec}(c, sk)]$$

  is negligible. Finally, if $m = \perp$, responds $\mathcal{A}$ with $\perp$, else responds $\mathcal{A}$ with (m, r).
- **stage 3:** The adversary $\mathcal{A}_1$ queries (m_0, m_1) to an encryption oracle with $|m_0| = |m_1|$. The challenger chooses $b \xleftarrow{R} \{0,1\}$, $r \xleftarrow{R} \{0,1\}^\lambda$, computes $Enc(pk, r, m_b) = c^*$, and sends c^* to adversary $\mathcal{A}_1$.
- **stage 4:** The adversary $\mathcal{A}_2$ continues to make calls c to the decryption oracle $\mathtt{Dec}$ and the random recovery algorithm $\mathtt{Rec}$, where c is subjected to the only restriction that a submitted ciphertext c is not identical to c^*. The decryption oracle responds with $m \leftarrow \mathtt{Dec}(pk, c)$ and the random recovery algorithm $\mathtt{Dec}$ responds with $r \leftarrow \mathtt{Rec}(sk, c)$. Finally, if $m = \perp$, responds $\mathcal{A}_2$ with $\perp$, else responds $\mathcal{A}_2$ with (m, r).
- **stage 5:** The adversary $\mathcal{A}$ outputs a guessing bit $b^{'} \in \{0,1\}$.

We define $\mathrm{Adv}^{\mathtt{ECCA}}_{\mathtt{PKE},\mathcal{A}}(\lambda)$ to be $|\Pr[b = b^{'}] - \frac{1}{2}|$ in the above attack game.

We say that $\mathtt{PKE} = (\mathtt{KeyGen}, \mathtt{Enc}, \mathtt{Dec})$ is secure against enhanced adaptive chosen ciphertext attack if for all probabilistic, polynomial-time adversary $\mathcal{A}$, the function $\mathrm{Adv}^{\mathtt{ECCA}}_{\mathtt{PKE},\mathcal{A}}(\lambda)$ grows negligibly in λ. $\mathtt{IND\text{-}CCA}$ security is defined all the same except that the decryption oracle does not return a randomness-recovery algorithm associated to the scheme.

2.2 Key Encapsulation Mechanism and Its ECCA Security Notions

A key encapsulation mechanism $\mathtt{KEM}$ is a public key encryption scheme, which consists of the three polynomial-time algorithms ($\mathtt{KEM.Gen}$, $\mathtt{KEM.Enc}$, $\mathtt{KEM.Dec}$) with the following interfaces:

Key Generation:	**Encapsulation:**	**Decapsulation**
$(pk, sk) \leftarrow \mathtt{KEM.Gen}(1^\lambda)$	$\psi \leftarrow \mathtt{KEM.Enc}(pk, \mathtt{K}, r)$	$\mathtt{K}$ (or $\perp$) $\leftarrow \mathtt{KEM.Dec}(sk, c)$

where $r \xleftarrow{R} \{0,1\}^\lambda$, $K \leftarrow \mathcal{K}_K$, $\mathcal{K}_K$ is the key space. $\mathtt{KDM.Dec}$ is a deterministic algorithm, (pk, sk) is a public/secret key pair and c is a ciphertext of the

encapsulated key K under pk. We now describe the attack game between the challenger and an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ used to define its security against adaptive enhanced chosen ciphertext attack.

- **stage 1:** The adversary queries a key generation oracle. The key generation oracle runs $(pk, sk) \leftarrow \mathtt{KEM.Gen}(\lambda)$ and responds adversary $\mathcal{A}$ with pk.
- **stage 2:** The adversary makes a sequence of calls to a decryption oracle. For each decryption oracle query, the adversary $\mathcal{A}_1$ submits a ciphertext ψ to Dec, the decryption oracle responds with $K \leftarrow \mathtt{Dec}(sk, \psi)$, and the random recovery algorithm Dec responds with $r \leftarrow \mathtt{Rec}(sk, \psi)$. Finally, if $K = \perp$, responds $\mathcal{A}$ with $\perp$, else responds $\mathcal{A}$ with $(\mathtt{K}, r)$.
- **stage 3:** The challenger chooses $r \overset{R}{\leftarrow} \{0,1\}^{\lambda}$ and computes $\psi^* \leftarrow \mathtt{KEM.Enc}(pk, r, K_1)$, chooses $K_0 \overset{R}{\leftarrow} \mathcal{K}_K$, $\sigma \overset{R}{\leftarrow} \{0,1\}$. Here, $\mathcal{K}_K$ is the key space, $|K_0| = |K_1|$ and sends (K_σ, ψ^*) to adversary $\mathcal{A}_1$.
- **stage 4:** The adversary $\mathcal{A}_2$ continues to make calls ψ to the decryption oracle Dec and the random recovery algorithm Rec, where ψ is subjected to the only restriction that a submitted ciphertext ψ is not identical to ψ^*. The decryption oracle responds with $K \leftarrow \mathtt{Dec}(sk, \psi)$ and the random recovery algorithm Dec responds with $r \leftarrow \mathtt{Rec}(sk, \psi)$. Finally, if $K = \perp$, responds $\mathcal{A}_2$ with $\perp$, else responds $\mathcal{A}_2$ with (K, r).
- **stage 5:** The adversary $\mathcal{A}$ outputs a guessing bit $\sigma' \in \{0,1\}$.

We define $\mathrm{Adv}^{\mathtt{ECCA}}_{\mathtt{KEM},\mathcal{A}}(\lambda)$ to be $|\Pr[\sigma = \sigma'] - \frac{1}{2}|$ in the above attack game. We say that KEM = (KEM.Gen,KEM.Enc,KEM.Dec) is secure against enhanced adaptive chosen ciphertext attack if for all probabilistic polynomial-time adversary $\mathcal{A}$, the function $\mathrm{Adv}^{\mathtt{ECCA}}_{\mathtt{KEM},\mathcal{A}}(\lambda)$ grows negligibly in λ.

2.3 Date encapsution mechanism and its one time security

A DEM = (DEM.Enc,DEM.Dec) is a symmetric encryption scheme that consists of the two polynomial-time algorithms (DEM.Enc, DEM.Dec). DEM.ENC and DEM.Dec are associated to a key-space K_D and message space M.

Encapsulation:	**Decapsulation**
$\chi \leftarrow \mathtt{DEM.Enc}(K, m)$	m (or $\perp$) $\leftarrow \mathtt{DEM.Dec}(K, \chi)$

DEM.Enc is an encryption algorithm that encrypts $m \in M$ by using symmetric-key $K \in K_D$ and outputs cipher-text χ, where $K \in K_D$. DEM.Dec is a corresponding decryption algorithm that recovers message m by using the same symmetric-key when the input cipher-text χ. An adversary $\mathcal{A}$ is a probabilistic polynomial-time oracle query machine. We now describe the attack game between the challenger and an adversary $\mathcal{A}$ used to define one time security.

- **stage 1:** The adversary $\mathcal{A}$ queries (m_0, m_1) to an encryption oracle. We require that the output of $\mathcal{A}$ satisfies $|m_0| = |m_1|$. The challenger chooses $b \xleftarrow{R} \{0,1\}$, $K \xleftarrow{R} K_D$, computes $\mathtt{Enc}(K, m_b) = c^*$ and sends c^* to adversary $\mathcal{A}$. Here we stress that the ciphertext is made from a random key along with the plaintext and every key has been used only once.
- **stage 2:** The adversary $\mathcal{A}$ outputs a guessing bit $b' \in \{0,1\}$.

We define $\mathrm{Adv}^{\mathcal{OT}-\mathcal{UF}}_{\mathtt{DEM},\mathcal{A}}(\lambda)$ to be $|\Pr[b = b'] - \frac{1}{2}|$ in the above attack game.

We say that `DEM = (DEM.Enc,DEM.Dec)` is one time secure if for all probabilistic polynomial-time adversary $\mathcal{A}$, the function $\mathrm{Adv}^{\mathcal{OT}-\mathcal{UF}}_{\mathtt{DEM},\mathcal{A}_2}(\lambda)$ grows negligibly in λ.

3 ECCA Security of Hybrid Scheme

3.1 Tag-KEM/DEM

Let `Tag-KEM=(TKEM.Gen,TKEM.Enc,TKEM.Dec)` be a public key encryption scheme and `DEM = (DEM.Enc,DEM.Dec)` be a symmetric encryption scheme. Then hybrid encryption scheme

$$\mathtt{Tag\text{-}KEM/DEM} = (\mathtt{HybGen}, \mathtt{HybEnc}, \mathtt{HybDec})$$

can be constructed as follows.

- $\mathtt{HybGen}(1^\lambda)$: Run $(pk, sk) \leftarrow \mathtt{TKEM.Gen}(1^\lambda)$ and output (pk, sk).
- $\mathtt{HybEnc}(pk, m)$: Run $(\omega, K) \leftarrow \mathtt{TKEM.Key}(pk)$, $\mathtt{TKEM.Key}(\cdot)$ is a probabilistic algorithm that inputs public key pk and outputs one-time key $K \in K_D$ along with the internal state information ω. Here K_D is the key-space of `DEM`. Then choosing $r \xleftarrow{\$} \{0,1\}^\lambda$ and computing

$$\chi \leftarrow \mathtt{DEM.Enc}_K(m),$$

$$\psi \leftarrow \mathtt{TKEM.Enc}_{pk}(\omega, r, \chi),$$

  we get the result ciphertext (of m) $c := (\psi, \chi)$.
- $\mathtt{HybDec}(sk, c)$: First, parse c as $\psi || \chi$.
  Run

$$K \leftarrow \mathtt{TKEM.Dec}_{sk}(\psi, \chi), \text{ and } m \leftarrow \mathtt{DEM.Dec}_K(\chi).$$

  Then, output the message m or "reject" symbol $\perp$.

3.2 ECCA Security of Tag-KEM/DEM

Theorem 1. *If the scheme `Tag-KEM` is `IND-ECCA` secure and `DEM` is one time secure, then the hybrid scheme `(Tag-KEM/DEM)` is `IND-ECCA` secure. In particular, for every probabilistic polynomial time (`PPT`) adversary $\mathcal{A}$, there exists probabilistic adversaries $\mathcal{A}_1$ and $\mathcal{A}_2$ whose running times are essentially the same as that of $\mathcal{A}$, such that for all $\lambda \geq 0$, we have*

$$\mathrm{Adv}^{\mathit{ECCA}}_{\mathit{Tag\text{-}KEM/DEM},\mathcal{A}}(\lambda) \leq 2\mathrm{Adv}^{\mathit{ECCA}}_{\mathit{Tag\text{-}KEM},\mathcal{A}_1}(\lambda) + \mathrm{Adv}^{\mathcal{OT}-\mathcal{UF}}_{\mathit{DEM},\mathcal{A}_2}(\lambda). \quad (1)$$

Proof. Fix $\mathcal{A}$ and λ, $\mathcal{A}$ be a PPT adversary that attacks the hybrid scheme Tag-KEM/DEM. Now, the theorem can be proved via the following games. (Denote T_i if the adversary $\mathcal{A}$ wins in the i-th game).

Game$_0$: This is an ECCA experiment on the scheme Tag-KEM/DEM played between the challenger and an adversary $\mathcal{A}$. In particular, there is:

- **stage 1:** The adversary queries a key generation oracle. Then the challenger runs $(pk, sk) \leftarrow \mathtt{TKEM.Gen}(\lambda)$ and responds adversary $\mathcal{A}$ with pk.
- **stage 2:** The adversary makes a sequence of calls to a decryption oracle. For each decryption oracle query, the adversary $\mathcal{A}_1$ submits a ciphertext $c = (\psi, \chi)$ to the challenger. Then the challenger runs

$$K \leftarrow \mathtt{TKEM.Dec}_{sk}(\psi, \chi), \text{ and } m \leftarrow \mathtt{DEM.Dec}_K(\chi).$$

  and runs the random recovery algorithm $r \leftarrow \mathtt{Rec}(c, sk)$. If $m = \perp$, the challenger responds $\mathcal{A}_1$ with $\perp$, else the challenger responds $\mathcal{A}_1$ with (m, r).
- **stage 3:** The adversary $\mathcal{A}_1$ queries (m_0, m_1) to an encryption oracle, and the challenger runs $(\omega, K) \leftarrow \mathtt{TKEM.Key}(pk)$, $K \in K_D$, where K_D is the key-space of DEM. Then the challenger chooses $r \xleftarrow{R} \{0,1\}^\lambda$ and computes

$$\mathtt{DEM.Enc}_K(m_0) = \chi^*, \mathtt{TKEM.Enc}_{pk}(r, \omega, \chi^*) = \psi^*,$$

  and sends $c^* = (\psi^*, \chi^*)$ to the adversary $\mathcal{A}_1$.
- **stage 4:** The adversary $\mathcal{A}_2$ continues to make calls $c = (\psi, \chi)$ to the challenger, where c subjects to the only restriction that a submitted ciphertext c is not identical to c^*. The challenger runs

$$K \leftarrow \mathtt{TKEM.Dec}_{sk}(\psi, \chi), \text{ and } m \leftarrow \mathtt{DEM.Dec}_K(\chi)$$

  and runs the random recovery algorithm $r \leftarrow \mathtt{Rec}(c, sk)$. If $m = \perp$, the challenger responds $\mathcal{A}_2$ with $\perp$, else responds $\mathcal{A}_2$ with (m, r).
- **stage 5:** The adversary $\mathcal{A}$ outputs a guessing bit $b' \in \{0, 1\}$.

Naturally, it holds that

$$\mathrm{Adv}^{\mathtt{ECCA}}_{\mathtt{Tag\text{-}KEM/DEM},\mathcal{A}}(\lambda) = \left|\Pr[b = b'] - \frac{1}{2}\right| = \left|\Pr[T_0] - \frac{1}{2}\right|. \tag{2}$$

Game$_1$: This game is identical to the above game except we use a completely random symmetric key $K_0 \xleftarrow{R} K_D$ to encrypt m_0 in the step-4 of **Game**$_0$, so we have

Lemma 1. *There exists a probabilistic adversary $\mathcal{A}_1$ whose running time is essentially the same as that of $\mathcal{A}$, such that*

$$|\Pr[T_1] - \Pr[T_0]| \leq \mathrm{Adv}^{\mathit{ECCA}}_{\mathit{Tag\text{-}KEM},\mathcal{A}_1}(\lambda). \tag{3}$$

Proof. The claim is proven by constructing the adversary $\mathcal{A}_1$ that attacks Tag-KEM. The adversary $\mathcal{A}_1$ offers the environment for $\mathcal{A}$. We describe the interaction as follows.

- **stage 1:** The adversary $\mathcal{A}_1$ was given (pk, K_σ), and at the same time, pk was sent to adversary $\mathcal{A}$.
- **stage 2:** The adversary $\mathcal{A}$ makes a sequence of calls to a decryption oracle. For each decryption oracle query, the decryption oracle responds with $m \leftarrow \mathtt{Dec}(sk, c)$ and the random recovery algorithm responds with $r \leftarrow \mathtt{Rec}(sk, c)$. Finally, if $m = \perp$, responds $\mathcal{A}$ with $\perp$, else responds $\mathcal{A}$ with (m, r).
- **stage 3:** The adversary $\mathcal{A}$ queries (m_0, m_1) to an encryption oracle, $|m_0| = |m_1|$. The adversary $\mathcal{A}_1$ computes $\mathtt{DEM.Enc}\mathtt{S}_{K_\sigma}(m_0) = \chi^*$ and outputs χ^* as the target tag, then it receives ψ^* as a challenge cipher. Finally, the adversary $\mathcal{A}_1$ sends $c^* = (\psi^*, \chi^*)$ to adversary $\mathcal{A}$.
- **stage 4:** The adversary $\mathcal{A}$ continues to make calls $c = (\psi_i, \chi_i)$ to decryption oracle query, where c subjects to the only restriction that a submitted ciphertext c is not identical to c^*. The adversary $\mathcal{A}_1$ runs

$$K_i \leftarrow \mathtt{TKEM.Dec}_{sk}(\chi_i, \psi_i),\ m \leftarrow \mathtt{DEM.Dec}_{K_i}(\psi_i).$$

  and runs the random recovery algorithm $r \leftarrow \mathtt{Rec(c,sk)}$. If $m = \perp$, the adversary $\mathcal{A}_1$ responds $\mathcal{A}$ with $\perp$, else responds $\mathcal{A}$ with (m, r).
- **stage 5:** $\mathcal{A}$ outputs a guessing bit $b' \in \{0, 1\}$ and $\mathcal{A}_1$ outputs $\sigma' = b'$.

This completes the description of $\mathcal{A}_1$. By construction, it is clear that decryption for $\mathcal{A}$ is perfectly simulated because the correct decryption is obtained from TKEM.Dec for every query.

- If $\sigma = 0$, we know that K_0 is a random key used for computing χ and the view of $\mathcal{A}$ is identical to that in $\mathbf{Game}_0$.
- If $\sigma = 1$, we know that K_1 is the correct key embedded in ψ and the view of $\mathcal{A}$ is identical to that in $\mathbf{Game}_1$.

we have that

$$|\Pr[T_1] - \Pr[T_0]| \leq \mathrm{Adv}^{\mathtt{ECCA}}_{\mathtt{Tag\text{-}KEM},\mathcal{A}_1}(\lambda).$$

The Lemma 1 is proved.

$\mathbf{Game}_2$: This game is identical to Game_1 except that we encrypt m_1 instead of m_0 in the step-4 of $\mathbf{Game}_1$.

Lemma 2. *There exists a probabilistic adversary $\mathcal{A}_2$ whose running time is essentially the same as that of $\mathcal{A}$, such that*

$$|\Pr[T_2] - \Pr[T_1]| \leq \mathrm{Adv}^{\mathcal{OT}-\mathcal{UF}}_{DEM,\mathcal{A}_2}(\lambda). \tag{4}$$

Proof. The claim is proven by constructing the adversary $\mathcal{A}_2$ that attacks DEM, the adversary $\mathcal{A}_2$ offers the environment for $\mathcal{A}$. We describe the interaction as follows.

- **stage 1:** The adversary $\mathcal{A}_2$ runs the key generation oracle $(pk, sk) \leftarrow \texttt{TKEM.Gen}(\lambda)$ and sends pk adversary to $\mathcal{A}$.
- **stage 2:** The adversary $\mathcal{A}$ makes a sequence of calls to a decryption oracle. For each decryption oracle query, the adversary $\mathcal{A}$ submits a ciphertext c to the decryption oracle. The decryption oracle runs $m \leftarrow \texttt{Dec}(sk, c)$ and the random recovery algorithm $r \leftarrow \texttt{Rec(sk,c)}$. If $m = \perp$, responds $\mathcal{A}$ with $\perp$, else responds $\mathcal{A}$ with (m, r).
- **stage 3:** The adversary $\mathcal{A}$ sends (m_0, m_1) to $\mathcal{A}_2$, $\mathcal{A}_2$ queries (m_0, m_1) to an encryption oracle and receives challenge ciphertext χ^*. The adversary $\mathcal{A}_2$ chooses $r \xleftarrow{R} \{0,1\}^\lambda$, runs $(\omega, K) \leftarrow \texttt{TKEM.Key}(pk)$, then computes

$$\texttt{TKEM.Enc}_{pk}(r, \omega, \chi^*) = \psi^*,$$

  and finally sends $c^* = (\psi^*, \chi^*)$ to adversary $\mathcal{A}$.
- **stage 4:** The adversary $\mathcal{A}$ continues to make calls $c = (\psi_i, \chi_i)$ to decryption oracle query, where c is subjected to the only restriction that a submitted ciphertext c is not identical to c^*. The the adversary $\mathcal{A}_2$ runs

$$K_i \leftarrow \texttt{TKEM.Dec}_{sk}(\psi_i, \chi_i),\ m \leftarrow \texttt{DEM.Dec}_{K_i}(\psi_i),$$

  and runs the random recovery algorithm $r \leftarrow \texttt{Rec}(c, sk)$. If $m = \perp$, the adversary $\mathcal{A}_2$ responds $\mathcal{A}$ with $\perp$, else the adversary $\mathcal{A}_2$ responds $\mathcal{A}$ with (m, r).
- **stage 5:** $\mathcal{A}$ outputs a guessing bit $b' \in \{0, 1\}$ and $\mathcal{A}_2$ outputs $\sigma' = b'$.

This completes the description of $\mathcal{A}_2$. By construction, the view of $\mathcal{A}$ is identical to that in $\mathbf{Game}_1$ and $\mathbf{Game}_2$, it is clear that we have

$$|\Pr[T_1] - \Pr[T_2]| \leq \mathrm{Adv}^{\mathcal{OT}-\mathcal{UF}}_{\mathtt{DEM},\mathcal{A}_2}(\lambda).$$

$\mathbf{Game}_3$: This game is identical to Game_2 except that we use the correct key K generated by TKEM.Key for DEM.Enc in the step-3 of $\mathbf{Game}_2$.

Lemma 3. *There exists a probabilistic adversary $\mathcal{A}_1$ whose running time is essentially the same as that of $\mathcal{A}$, such that*

$$|\Pr[T_2] - \Pr[T_1]| \leq \mathrm{Adv}^{\mathit{ECCA}}_{\mathit{Tag\text{-}KEM},\mathcal{A}_1}(\lambda). \tag{5}$$

Proof. The proof is similar to Lemma 1, so we omit it here.

We know that $\mathcal{A}$'s advantage in Game_0

$$\mathrm{Adv}^{\mathtt{ECCA}}_{\mathtt{Tag\text{-}KEM/DEM},\mathcal{A}}(\lambda) = \left|\Pr[T_0] - \frac{1}{2}\right| \leq 2\mathrm{Adv}^{\mathtt{ECCA}}_{\mathtt{Tag\text{-}KEM},\mathcal{A}_1}(\lambda) + \mathrm{Adv}^{\mathcal{OT}-\mathcal{UF}}_{\mathtt{DEM},\mathcal{A}_2}(\lambda)$$

is negligible.

Putting all the facts together, the Theorem 1 is proved.

3.3 KEM/Tag-DEM

Let $\mathtt{KEM} = (\mathtt{Gen}, \mathtt{KEM.Enc}, \mathtt{KEM.Dec})$ be a public key encryption scheme and $\mathtt{Tag\text{-}DEM} = (\mathtt{TDEM.Enc}, \mathtt{TDEM.Dec})$ be a symmetric key encryption scheme. Then hybrid cryptosystem scheme

$$\mathtt{KEM/Tag\text{-}DEM} = (\mathtt{HybGen}, \mathtt{HybEnc}, \mathtt{HybDec})$$

can be constructed as follows.

- $\mathtt{HybGen}(1^\lambda)$: Run $(pk, sk) \leftarrow \mathtt{Gen}(1^\lambda)$ and output (pk, sk).
- $\mathtt{HybEnc}(pk, m)$: Choose $r \xleftarrow{R} \{0,1\}^\lambda$, $K \in K_D$. Here K_D is the key-space of $\mathtt{DEM}$.
  Then compute
  $$\psi \leftarrow \mathtt{KEM.Enc}_{pk}(r, K),$$
  $$\chi \leftarrow \mathtt{TDEM.Enc}_K(m, \psi),$$
  and output the ciphertext (of m) $c := (\psi, \chi)$.
- $\mathtt{HybDec}(sk, c)$: First, parse c as $\psi || \chi$.
  Run
  $$K \leftarrow \mathtt{KEM.Dec}_{sk}(\psi), \text{ and } m \leftarrow \mathtt{TDEM.Dec}_K(\chi, \psi).$$
  Then, output the message m or "reject" symbol $\perp$.

3.4 ECCA Security of KEM/Tag-DEM

Theorem 2. *If the public key encryption scheme* $\mathtt{KEM} = (\mathtt{Gen}, \mathtt{KEM.Enc}, \mathtt{KEM.Dec})$ *is IND-ECCA secure and symmetric key encryption* $\mathtt{Tag\text{-}DEM} = (\mathtt{TDEM.Enc}, \mathtt{TDEM.Dec})$ *is IND-CCA secure, the hybrid encryption scheme* $\mathtt{KEM/Tag\text{-}DEM}$ *is IND-ECCA secure. In particular, for every probabilistic polynomial time (PPT) adversary* $\mathcal{A}$*, there exists probabilistic adversary* $\mathcal{A}_1$ *and* $\mathcal{A}_2$ *whose running times are essentially the same as that of A, such that for all* $\lambda \geq 0$*, we have*

$$\mathrm{Adv}^{ECCA}_{\mathtt{KEM/Tag\text{-}DEM},\mathcal{A}}(\lambda) \leq \mathrm{Adv}^{ECCA}_{\mathtt{KEM},\mathcal{A}_1}(\lambda) + \mathrm{Adv}^{CCA}_{\mathtt{Tag\text{-}DEM},\mathcal{A}_2}(\lambda).$$

Proof. Fix $\mathcal{A}$ and λ. Let $\mathcal{A}$ be a PPT adversary who attacks on the hybrid scheme $\mathtt{KEM/Tag\text{-}DEM}$. Now, the theorem can be proved via the following games. (Denote by T_i the adversary $\mathcal{A}$ wins in the i-th game).

Game$_0$: This is an original ECCA experiment on the hybrid scheme $\mathtt{KEM/Tag\text{-}DEM}$ played between the challenger and the adversary $\mathcal{A}$. In particular,

- **stage 1:** The adversary queries a key generation oracle. The challenger runs $(pk, sk) \leftarrow \mathtt{Gen}(\lambda)$ and responds the adversary $\mathcal{A}$ with pk.
- **stage 2:** The adversary makes a sequence of calls to a decryption oracle. For each decryption oracle query, the adversary $\mathcal{A}_1$ submits a ciphertext c to the challenger. The challenger then runs the decryption oracle $m \leftarrow \mathtt{Dec}(sk, c)$ and the random recovery algorithm $r \leftarrow \mathtt{Rec}(sk, c)$. If $m = \perp$, the challenger responds with $\perp$, else the challenger responds with (m, r).

– **stage 3:** The adversary $\mathcal{A}_1$ queries (m_0, m_1) to an encryption oracle. The challenger chooses $b \xleftarrow{R} \{0,1\}$, $r \xleftarrow{R} \{0,1\}^{\lambda}, K \xleftarrow{R} K_D$, computes

$$\texttt{KEM.EncP}_{pk}(r, K) = \psi^*, \texttt{TDEM.EncS}_K(m_b, \psi) = \chi^*$$

and sends $c^* = (\psi^*, \chi^*)$ to adversary $\mathcal{A}_1$.
– **stage 4:** The adversary $\mathcal{A}_2$ continues to make calls $c = (\psi, \chi)$ to the challenger, where c is subjected to the only restriction that a submitted ciphertext c is not identical to c^*. The challenger runs

$$K \leftarrow \texttt{KEM.Dec}_{sk}(\psi),\ m \leftarrow \texttt{TDEM.Dec}_K(\chi, \psi).$$

and the random recovery algorithm $r \leftarrow \texttt{Rec}(c, sk)$. If $m = \perp$, the challenger responds $\mathcal{A}_2$ with $\perp$, else the challenger responds $\mathcal{A}_2$ with (m, r).
– **stage 5:** The adversary outputs a guessing bit $b' \in \{0,1\}$.

Naturally, it holds that

$$\mathrm{Adv}^{\texttt{ECCA}}_{\texttt{KEM/Tag-DEM},\mathcal{A}}(\lambda) = \left|\Pr[b = b'] - \frac{1}{2}\right| = \left|\Pr[T_0] - \frac{1}{2}\right|. \tag{6}$$

Game$_1$: This game is identical to Game$_1$ except that we use a completely random symmetric key K_0 in place of the key K_1 in both the encryption and decryption oracles. We have

Lemma 4. *There exists a probabilistic adversary $\mathcal{A}_1$ whose running time is essentially the same as that of $\mathcal{A}$, such that*

$$|\Pr[T_0] - \Pr[T_1]| \leq \mathrm{Adv}^{\textit{ECCA}}_{\textit{KEM},\mathcal{A}_1}(\lambda). \tag{7}$$

Proof. The claim is proven by constructing a probabilistic adversary $\mathcal{A}_1$ that attacks KEM: $\mathcal{A}_1$ offers the environment for $\mathcal{A}$. We describe the interaction as follows.

– First, the adversary $\mathcal{A}_1$ receives pk and sends it to $\mathcal{A}$.
– $\mathcal{A}_1$ chooses (m_0, m_1) and sends them to $\mathcal{A}_1$. Meanwhile, the adversary $\mathcal{A}_1$ runs the encryption of KEM.Enc, and receives (K_δ, ψ^*). Then the adversary$\mathcal{A}_1$ chooses $b \in \{0,1\}$ and computes $\texttt{TDEM.Enc}(m_b, \psi^*) = \chi^*$. Finally, $\mathcal{A}_1$ sends (ψ^*, χ^*) to $\mathcal{A}$.
– $\mathcal{A}$ continues to submit a cipher-text $c = (\psi, \chi)$ to the decryption oracle, where c is subjected to the only restriction that a submitted ciphertext c is not identical to c^*.
  • If $\psi \neq \psi^*$, $\mathcal{A}_1$ sends ψ to its own decryption oracle $K \leftarrow \texttt{KEM.Dec}_{sk}(\psi), m \leftarrow \texttt{TDEM.Dec}_K(\psi, \chi), r \leftarrow \texttt{Rec}(c, sk)$. If $m = \perp$, the $\mathcal{A}_1$ responds $\mathcal{A}$ with $\perp$, else responds with (m, r).
  • If $\psi = \psi^*$, $\mathcal{A}_1$ uses K_σ to decrypt (χ, ψ): $m \leftarrow \texttt{TDEM.Dec}_K(\psi, \chi), r \leftarrow \texttt{Rec}(c, sk)$. If $m = \perp$, the $\mathcal{A}_1$ responds $\mathcal{A}$ with $\perp$, else responds with (m, r).
– Finally, $\mathcal{A}$ outputs a guessing bit $b' \in \{0,1\}$,

$\mathcal{A}_1$ outputs 1 if $b = b^{'}$ and 0 if $b \neq b^{'}$. This completes the description of $\mathcal{A}_1$ and it is clear that we have

$$|\Pr[T_0] - \Pr[T_1]| \leq \mathrm{Adv}^{\mathtt{ECCA}}_{\mathtt{KEM},\mathcal{A}_1}(\lambda). \tag{8}$$

In game G_1, we use a random symmetric key in both the encryption and decryption oracles so the cipher-text ψ^* cannot be decrypted. To see this, it is noticed that in game G_1 the cipher-text χ^* is produced by using the random symmetric encryption key K_0. Meanwhile, some other cipher-texts $\chi = \chi^*$ are being decrypted by using K_0 which plays no other role in game G_1. Thus, in game G_1, the adversary $\mathcal{A}$ essentially just carries out an adaptive chosen cipher-text attack against `Tag-DEM`. So we have

Lemma 5. *There exists a probabilistic adversary $\mathcal{A}_2$ whose running time is essentially the same as that of $\mathcal{A}$, such that*

$$|\Pr[T_1] - \frac{1}{2}| \leq \mathrm{Adv}^{CCA}_{Tag\text{-}DEM,\mathcal{A}_2}(\lambda). \tag{9}$$

Proof. We construct a probabilistic adversary $\mathcal{A}_2$ that attacks `Tag-DEM` and $\mathcal{A}_2$ offers the environment for $\mathcal{A}$. We describe the interaction as follows.

- The adversary $\mathcal{A}_2$ runs the key generation oracle $(pk, sk) \leftarrow \mathtt{TKEM.Gen}(\lambda)$ and sends pk adversary to $\mathcal{A}$.
- The adversary $\mathcal{A}$ makes a sequence of calls to a decryption oracle. For each decryption oracle query, the adversary $\mathcal{A}$ submits a ciphertext c to the decryption oracle and the decryption oracle runs $m \leftarrow \mathtt{Dec}(sk, c)$ and the random recovery algorithm $r \leftarrow \mathtt{Rec(sk,c)}$. If $m = \perp$, responds $\mathcal{A}$ with $\perp$, else responds $\mathcal{A}$ with (m, r).
- The adversary $\mathcal{A}$ sends (m_0, m_1) to $\mathcal{A}_2$. $\mathcal{A}_2$ chooses $K \xleftarrow{R} K_D$, $r \xleftarrow{R} \{0,1\}^{\lambda}$, runs $\psi^* \leftarrow \mathtt{KEM.Enc}_{pk}(r, K)$ and then sends (m_0, m_1, ψ^*) to encryption oracle `Tag-DEM`. The $\mathcal{A}_2$ receives ciphertext χ^*, and sends $c^* = (\psi^*, \chi^*)$ to $\mathcal{A}$. We note that the key K^* chosen as the encryption key of `Tag-DEM` as well as embedded in ψ^* is completely random and mutually independent with each other.
- $\mathcal{A}$ continues to submit a ciphertext $c = (\psi, \chi)$ to the decryption oracle, where c is subjected to the only restriction that a submitted ciphertext c is not identical to c^*. $\mathcal{A}_2$ runs the decryption oracle by using the secret key sk.

  $$\mathtt{K} \leftarrow \mathtt{KEM.Dec}_{sk}(\psi), m \leftarrow \mathtt{TDEM.Dec}_K(\psi, \chi),$$

  and runs the random recovery algorithm $r \leftarrow \mathtt{Rec}(c, sk)$, If $m = \perp$, $\mathcal{A}_2$ responds $\mathcal{A}$ with $\perp$, else $\mathcal{A}_2$ responds $\mathcal{A}$ with (m, r).
- Finally, $\mathcal{A}$ outputs a guessing bit $b^{'} \in \{0, 1\}$ and $\mathcal{A}_2$ also outputs $b^{'}$.

This completes the description of $\mathcal{A}_2$. By construction, it is clear that the decryption for $\mathcal{A}$ is perfectly simulated, and whenever $\mathcal{A}$ wins, so does $\mathcal{A}_2$. We have that

$$|\Pr[T_1] - \frac{1}{2}| \leq \mathrm{Adv}^{\mathtt{CCA}}_{\mathtt{Tag\text{-}DEM},\mathcal{A}_2}(\lambda). \tag{10}$$

we know that the $\mathcal{A}$'s advantage in Game_0

$$\text{Adv}^{\texttt{ECCA}}_{\texttt{KEM/Tag-DEM},\mathcal{A}}(\lambda) = \left|\Pr[T_0] - \frac{1}{2}\right| \leq \text{Adv}^{\texttt{ECCA}}_{\texttt{KEM},\mathcal{A}_1}(\lambda) + \text{Adv}^{\texttt{CCA}}_{\texttt{Tag-DEM},\mathcal{A}_2}(\lambda),$$

which is negligible.

Putting all the facts together, the Theorem 2 is proved.

4 Conclusion

In this paper, we discuss the security results for achieving ECCA secure hybrid encryptions from the well-known hybrid paradigms, KEM/Tag-DEM and Tag-KEM/DEM. We have proven that the hybrid encryption scheme (KEM/Tag-DEM) can beECCA secure if the KEM part is ECCA secure and the DEM part is CCA secure. Meanwhile, we have also proven that the hybrid encryption scheme (Tag-KEM/DEM) can beECCA secure if the KEM part is ECCA secure and the DEM part is one-time secure.

Acknowledgements. We are grateful to the anonymous reviewers for their helpful comments and suggestions. This research is supported by the National Natural Science Foundation of China (No. 61602061; No. 61672059; No. 61272499; No. 61472016; No. 61472414; No. 61402471) and China Postdoctoral Science Foundation (Grant No. 2017M610021).

References

1. Abe, M., Gennaro, R., Kurosawa, K., Shoup, V.: Tag-KEM/DEM: a new framework for hybrid encryption and a new analysis of kurosawa-desmedt KEM. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 128–146. Springer, Heidelberg (2005). https://doi.org/10.1007/11426639_8
2. Abe, M., Gennaro, R., Kurosawa, K., Shoup, V.: Tag-KEM/DEM: a new framework for hybrid encryption. J. Cryptol. **21**(1), 97–130 (2008)
3. Canetti, R., Krawczyk, H., Nielsen, J.B.: Relaxing chosen-ciphertext security. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 565–582. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45146-4_33
4. Dachman-Soled, D., Fuchsbauer, G., Mohassel, P., O'Neill, A.: Enhanced chosen-ciphertext security and applications. In: Krawczyk, H. (ed.) PKC 2014. LNCS, vol. 8383, pp. 329–344. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54631-0_19
5. Dachman-Soled, D., Fuchsbauer, G., Mohassel, P., O'Neill, A.: Enhanced chosen-ciphertext security and applications. Cryptology ePrint Archive, Report 2012/543 (2012)
6. Damgård, I., Thorbek, R.: Non-interactive proofs for integer multiplication. In: Naor, M. (ed.) EUROCRYPT 2007. LNCS, vol. 4515, pp. 412–429. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72540-4_24
7. Damgård, I., Hofheinz, D., Kiltz, E., Thorbek, R.: Public-key encryption with non-interactive opening. In: Malkin, T. (ed.) CT-RSA 2008. LNCS, vol. 4964, pp. 239–255. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-79263-5_15

8. Kiltz, E., Mohassel, P., O'Neill, A.: Adaptive trapdoor functions and chosen-ciphertext security. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 673–692. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_34
9. Naor, M., Yung, M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. In: STOC 1990, pp. 427–437. ACM, New York (1990)
10. Chen, Y., Dong, Q.: RCCA security for KEM+DEM style hybrid encryptions. In: Kutyłowski, M., Yung, M. (eds.) Inscrypt 2012. LNCS, vol. 7763, pp. 102–121. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38519-3_8
11. Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: STOC 2008, pp. 187–196. ACM, New York (2008)
12. Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. Full version of [11]. http://www.cc.gatech.edu/~cpeikert/pubs/lossy_tdf.pdf
13. Cramer, R., Shoup, V.: Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. SIAM J. Comput. **33**(1), 167–226 (2003)
14. Canetti, R., Krawczyk, H., Nielsen, J.: Relaxing chosen ciphertext security (2003). http://eprint.iacr.org