RCCA Security for KEM+DEM Style Hybrid Encryptions

Yuan Chen and Qingkuan Dong

State Key Laboratory of Integrated Service Networks (ISN),
Xidian University, Xi'an 710071, P.R. China
yuanchen@xidian.edu.cn, qkdong@mail.xidian.edu.cn

Abstract. RCCA security is a weaker notion than CCA security, and has been proven to be sufficient for several cryptographic tasks. This paper adapts RCCA security to the most popular hybrid paradigms, KEM+DEM and Tag-KEM/DEM.

It is open to construct an RCCA-secure scheme more efficient than CCA-secure ones. In the setting of Tag-KEM, we solve this by presenting a natural RCCA-secure RSA-based Tag-KEM scheme, named as RSA-TKEM, which is more efficient than all existing methods for constructing a CCA-secure RSA-based Tag-KEM scheme.

Unfortunately, combining our RSA-TKEM with passive secure one-time pad following Tag-KEM/DEM paradigm yields an RCCA-insecure hybrid encryption. This shows passive security of DEM is not sufficient now, and Tag-KEM/DEM looses its advantage over KEM+DEM. In spite of this and for completeness, we show RCCA secure DEMs are still sufficient to achieve RCCA-secure hybrid encryptions by following Tag-KEM/DEM.

In addition, we show RCCA-secure KEM is sufficient for achieving CCA-secure hybrid encryptions. This is done by introducing a new hybrid paradigm, named as KEM/Tag-DEM, where the ciphertext of KEM is used as a tag for Tag-DEM scheme rather than reversely in Tag-KEM/DEM, so that the security of KEM can be weakened to RCCA one. Tag-DEMs can be constructed as efficiently as DEMs, so RCCA-secure KEMs more efficient than CCA-secure ones become more appealing.

1 Introduction

The notion of Replayable CCA (RCCA) security for Public-Key Encryption (PKE) is introduced in [10]. It is a weakened variant of CCA security where the decryption oracle answers '`test`' whenever it is asked to decrypt any ciphertext that decrypts to either of the questioned messages m_0 or m_1, even if this ciphertext is different from the challenge ciphertext. Accordingly, even if the adversary can tweak the challenge ciphertext without affecting the embedded plaintext (such a feature is called benign-malleability in [20]), sending it to the decryption oracle does not help the adversary determine which of the questioned messages is hidden.

M. Kutyłowski and M. Yung (Eds.): Inscrypt 2012, LNCS 7763, pp. 102–121, 2013.

RCCA-security is proven to be sufficient for several cryptographic tasks in [10] and is believed to be sufficient for all the uses of CCA-secure encryptions in [5,20]. It also makes it possible to consider secure rerandomizable encryptions [13,17]. We can hope that a weaker definition might give rise to more efficient constructions but this has so far not been the case. There is no known instance of RCCA-secure public-key encryption that is more efficient than known CCA-secure ones as so far.

For hybrid encryptions, RCCA security was first considered in [10]. [10] shows the natural combination of an RCCA-secure PKE and an RCCA-secure Symmetric-Key Encryption (SKE) is RCCA-secure as a PKE, which is consistent with the well known CCA case. Furthermore, an RCCA-secure PKE can be made CCA-secure if combined with a CCA-secure SKE in such a way that the ciphertext of the PKE is also input to the SKE encryption. This shows the sufficiency of RCCA security of PKE for achieving CCA-secure hybrid encryptions. The two hybrid schemes named as HE1 and HE2 are shown in Table 1.

However, the work in [10] only relates to limited hybrid frameworks consisting of PKE and SKE. There are more efficient and general paradigms to realize hybrid encryptions, such as KEM+DEM [11] and Tag-KEM/DEM [3].

Table 1. RCCA-security related hybrid encryptions of PKE and SKE in [10]

	PKE	SKE	Security
HE1	Input K Output $\psi =$ $\mathsf{PKE.Enc}_{pk}(K)$	Input M Output χ= $\mathsf{SKE.Enc}_K(M)$	IND-RCCA+IND-RCCA ⇒ IND-RCCA
HE2	Input K Output ψ= $\mathsf{PKE.Enc}_{pk}(K)$	Input (M, ψ) Output χ= $\mathsf{SKE.Enc}_K(\psi \| M)$	IND-RCCA+IND-CCA ⇒ IND-CCA

KEM uses asymmetric techniques to encrypt a symmetric key, while DEM uses a symmetric cipher to encrypt the message using the key from the KEM. KEM can be built more efficiently than PKE, and DEM can be a one-time SKE, these make KEM+DEM paradigm more efficient.

For some security reasons and to capture wider variety, Masayuki Abe, et al introduced Tag-KEM/DEM framework ([3], [2], [4]). A Tag-KEM scheme takes also a *tag* as its input. The novelty of Tag-KEM/DEM lies in using the ciphertext of DEM part as the tag input to Tag-KEM. This binding way of ciphertexts makes it possible to yield CCA-secure hybrid encryptions by a DEM scheme secure simply against a passive attacker (we shortened this security for DEM as one-time security). This weakening may be the most prominent advantage of Tag-KEM/DEM.

This paper adapts RCCA security to those more general hybrid paradigms and consider whether or not similar desired properties hold as for hybrid encryptions of PKE and SKE.

1.1 Our Contributions

We first consider RCCA security for KEM+DEM paradigm. Adapting the notion of RCCA security to KEM is also done in [3] for a different purpose. It is believed in [20] that if KEM and DEM are benign malleable, then so is the hybrid encryption. We substitutes more general RCCA-security for benign malleability. That is, IND-RCCA+IND-RCCA⇒IND-RCCA. This consists with the CCA situation, as desired.

We also adapt RCCA security to Tag-KEM/DEM, but find some different status. Firstly, a natural and efficient RCCA-secure Tag-KEM construction is possible. We present such an RSA-based one in Section 3, named as RSA-TKEM. The security is proved in the random oracle model [8]. RSA-TKEM is almost as efficient as the well known RSA-KEM in [20] with only one more hash and an XOR operation. Particularly, the scheme has no ciphertext redundancy, thus no validity check of ciphertext or tag part. This is different from all existing CCA-secure schemes as much as we know of and makes the scheme more efficient. We give a short comparison with those most efficient CCA-secure RSA-based Tag-KEMs in Table 2, where f denotes the RSA permutation, MAC is a message authentication code, H is a hash function, the final ciphertext is ψ, and $|\cdot|$ denotes the length. It can be seen from the table that our RSA-TKEM has least ciphertext expansion.

Table 2. A comparison with existing efficient CCA-secure RSA-based Tag-KEMs

Schemes	In [1]	In [3] Section 4.2	In [3] Section 5.3	RSA-TKEM
Description of encapsulation	$x = r\|H(r,\tau)$ $\psi = f(x)$	$\sigma = MAC(\tau)$ $\psi = f(r)\|\sigma$	$\sigma = H(r\|\tau)$ $\psi = f(r)\|\sigma$	$s = f(r)$ $\psi = f(r) \oplus H(\tau)$
Ciphertext length	$\lvert r\rvert + \lvert H(r,\tau)\rvert$	$\lvert r\rvert + \lvert MAC(\tau)\rvert$	$\lvert r\rvert + \lvert H(r,\tau)\rvert$	$\lvert r\rvert$
Security	IND-CCA	IND-CCA	IND-CCA	IND-RCCA

For Tag-KEM/DEM, one may hope IND-RCCA+One-time security⇒IND-RCCA. Unfortunately, this is not the case. Combining the above RSA-TKEM and passive secure one-time pad yields a counterexample. An adversary when receives the challenge ciphertext (ψ^*, χ^*) can randomly choose a χ', compute $H(\chi')$ and $H(\chi^*)$, then query $(\psi = \psi^* \oplus H(\chi^*) \oplus H(\chi'), \chi')$ to its decryption oracle. Let the answer be m', with all but negligible probability, $m' \neq$ 'test'. The adversary can then obtain the K underlying (ψ^*, χ^*) by computing $K = \chi' \oplus m'$.

In fact, passive security for DEM is not sufficient now. Note that if in an RCCA-secure Tag-KEM, an adversary can only tweak (ψ^*, τ^*) to (ψ, τ) with $\tau = \tau^*$ and without affecting the encapsulated key, then passive security for DEM is sufficient. However, this is not the case, generally. Furthermore, we can prove RCCA security for DEM will be sufficient generally. The proof is given in Section 4.2. But this time Tag-KEM/DEM looses its advantage over KEM+DEM. It is interesting to notice that a little weakening of security requirement of Tag-KEM results in a total insecurity. This somewhat explains the gap between KEM+DEM and Tag-KEM/DEM paradigms.

Finally, we prove RCCA-security for KEM is sufficient for hybrid encryption as desired. To do this, we generalize HE2 in Table 1 to a new hybrid paradigm named as KEM/Tag-DEM. KEM/Tag-DEM uses a similar idea behind Tag-KEM/DEM, but rather than use the ciphertext of DEM as a tag for Tag-KEM, we use the ciphertext of KEM as a tag for Tag-DEM. Since in Tag-KEM/DEM the binding way of ciphertexts makes it possible for the CCA-security of Tag-KEM to provide integrity to the tag, the ciphertext of DEM there, so can it in our paradigm for the CCA security of Tag-DEM to compensate the malleability of KEM. This makes it possible to achieve CCA-security by RCCA-secure KEM and CCA-secure Tag-DEM. The formal treatment and possible constructions are shown in Section 5.

The CCA-security of KEM/Tag-DEM and the RCCA one of KEM+DEM actually show that RCCA-secure KEM is sufficient for constructing hybrid encryption, just as we desired: for RCCA-security, follow KEM+DEM; for CCA-security, follow KEM/Tag-DEM. We summarize all three hybrid paradigms related security results in Table 3.

Table 3. The three hybrid paradigms and their security

Hybrid paradigm	Security results
KEM+DEM	IND-CCA + IND-CCA ⇒ IND-CCA([11]) IND-RCCA + IND-RCCA ⇒ IND-RCCA (by Th2)
Tag-KEM/DEM	IND-CCA + One-time security ⇒ IND-CCA([3]) IND-RCCA + One-time security ⇏ IND-RCCA (counterexample) IND-RCCA + IND-RCCA ⇒ IND-RCCA (by Th3)
KEM/Tag-DEM	IND-RCCA + IND-CCA ⇒ IND-CCA (by Th4)

Importance of KEM/Tag-DEM. Compared to Tag-KEM/DEM, one may think KEM/Tag-DEM achieves less improvement. However, CCA-secure DEMs and Tag-DEMs can be easily and efficiently built, while CCA-secure KEMs and Tag-KEMs cost much, weakening the security requirement of KEM to RCCA-security may be beneficial. In most KEMs, the valid ciphertext of a encapsulated key is uniquely determined, this makes it difficult to build efficient and natural RCCA-secure KEM, especially when rerandomizable one is taken into consideration [13]. Nevertheless, as illustrated in [10], a CCA-secure KEM may only achieve RCCA-security in practical protocols when allowing for arbitrary padding to ciphertexts (in order to align the length) or more than one representation of ciphertexts. That time, KEM/Tag-DEM will achieve advantages over KEM+DEM paradigm.

KEM/Tag-DEM shows the diversity of hybrid encryptions and has additional practical values. In practice there are always associated data to DEM [18], thus including the ciphertext of KEM brings no significant difference in efficiency, and may "offload" as much cryptographic work from the slower KEM part onto the faster DEM part. In addition, in Tag-KEM/DEM, a receiver generally need the entire ciphertext to derive the encapsulated key, which makes it extremely unsuitable for streaming processing, while KEM/Tag-DEM suffers no such problem.

1.2 Further Discussions and Related Works

From the above, to achieve RCCA-secure hybrid encryptions, we need RCCA-secure DEMs. In [10], it has been pointed out that RCCA-secure SKE can be given by the "encrypt-then-authenticate" paradigm using a regular secure MAC (but not a strong one, which means given pairs of messages and their mac values, it is not possible to find another valid mac value for any messages), which means given pairs of messages and their mac values, it is not possible to find a valid mac value for any other messages, but may be possible to find a different valid mac value for someone of these messages. RCCA-secure DEM can be given in the same way. Thus, a regular secure but not strong MAC may helps in obtaining an RCCA-secure hybrid encryptions more efficient than CCA-secure ones. However, most MAC schemes are deterministic and the verification is done by re-computation, finding a different valid mac value for the same message is impossible. A randomized or multi-valued MAC is needed. [15] gives some examples, but just conceptual.

In [3], Abe et al. also showed how to obtain a CCA-secure hybrid encryption from an RCCA-secure KEM by making it to be a CCA-secure Tag-KEM. Their method can be explained by our new paradigm in another way. And since in Tag-DEM it is no need to provide privacy but integrity of tag τ, then using the same techniques in constructing deterministic authenticated-encryption in [19] gives more efficient schemes.

There are some other weaker security notions for KEM except RCCA-security, such as CCCA-security in [14] and LCCA-security in [3]. These security notions whose strength depend on the chosen predict are not strictly weaker than CCA-security. Since in RCCA security, a special 'test' is returned when a replay is detected, no direct relation exists between these notions.

2 Preliminaries

2.1 CCA and RCCA Security Notions for PKE

A public-key encryption scheme consists of three algorithms. Probabilistic PKE.Gen that on input the security parameter k, generates public and private-keys (pk, sk), pk defines the message space $\mathcal{M}$. Probabilistic PKE.Enc encrypts a message $m \in \mathcal{M}$ into a ciphertext c. PKE.Dec decrypts c, outputs either $m \in \mathcal{M}$ or a special symbol $\perp \notin \mathcal{M}$. An obvious soundness condition applies.

We say a PKE is IND-CCA secure if for every probablistic polynomial-time oracle machine $\mathcal{A}_E$ plays the following game, its advantage
$Adv^{\mathsf{cca}}_{\mathsf{pke},\mathcal{A}_E}(k) = |Pr[\tilde{b} = b] - \frac{1}{2}|$ is negligible in k.

[GAME.PKE]
Step 1. $(pk, sk) \leftarrow \mathsf{PKE.Gen}(1^k)$
Step 2. $(m_0, m_1, v) \leftarrow \mathcal{A}_E^{\mathcal{O}}(pk)$
Step 3. $b \leftarrow \{0, 1\}$, $c \leftarrow \mathsf{PKE.Enc}_{pk}(m_b)$.
Step 4. $\tilde{b} \leftarrow \mathcal{A}_E^{\mathcal{O}}(v, c)$

By $\mathcal{O}$, we denote $\mathsf{PKE.Dec}_{sk}(\cdot)$. In Step 4, $\mathcal{A}_E$ is restricted not to ask c to $\mathcal{O}$.

IND-RCCA security is defined all the same except that the decryption oracle returns 'test' for any ciphertext decrypts to m_0 or m_1 in step 4. Let [$\mathsf{RGAME.PKE}$] denote the game and $Adv^{\mathsf{rcca}}_{\mathsf{pke},\mathcal{A}_E}(k)$ the advantage.

2.2 KEM+DEM and Related Security Notions

A key encapsulation mechanism (KEM) consists of three algorithms. Probabilistic $\mathsf{KEM.Gen}$ that on input 1^k outputs a public/private key pair (pk, sk), pk defines the key space $\mathcal{K}_K$. Probabilistic encapsulation algorithm $\mathsf{KEM.Enc}$ that on input 1^k and a public key pk, outputs a pair (dk, ψ), where $dk \in \mathcal{K}_K$ is a key and ψ is its ciphertext. Decapsulation algorithm $\mathsf{KEM.Dec}$, on input sk and ψ, outputs either a key $dk \in \mathcal{K}_K$ or the special symbol $\perp$. An obvious soundness condition applies.

IND-CCA security for KEM is defined by the following game.

[$\mathsf{GAME.KEM}$]
Step 1. $(pk, sk) \leftarrow \mathsf{KEM.Gen}(1^k)$
Step 2. $v \leftarrow \mathcal{A}_K^{\mathcal{O}}(pk)$
Step 3. $(dk_1, \psi) \leftarrow \mathsf{KEM.Enc}_{pk}()$, $dk_0 \leftarrow \mathcal{K}_K$, $\delta \leftarrow \{0, 1\}$.
Step 4. $\tilde{\delta} \leftarrow \mathcal{A}_K^{\mathcal{O}}(v, \psi, dk_\delta)$
$\mathcal{O}$ denotes $\mathsf{KEM.Dec}_{sk}(\cdot)$. In Step 4, $\mathcal{A}_K$ is restricted not to ask ψ to $\mathcal{O}$.

For IND-RCCA security, all is the same except that the decryption oracle returns 'test' for any ciphertext decrypts to dk_0 or dk_1 in step 4, as done in [3].

A data encapsulation mechanism (DEM) is a one-time symmetric-key encryption, consists of two algorithms. Deterministic $\mathsf{DEM.Enc}$ that takes as input 1^k, a key dk and a message m, outputs a ciphertext χ. Deterministic $\mathsf{DEM.Dec}$ that takes as input a dk and a ciphertext χ, outputs a message m or the special symbol $\perp$. An obvious soundness condition applies.

The one-time security (or passive security) and IND-CCA security of DEM is defined respectively by the following game when $\mathcal{O}$ is null and $\mathsf{DEM.Dec}_{dk}(\cdot)$. In Step 3, $\mathcal{A}_D$ is restricted not to ask χ to $\mathcal{O}$.

[$\mathsf{GAME.DEM}$]
Step 1. $(m_0, m_1, v) \leftarrow \mathcal{A}_D^{\mathcal{O}}(1^k)$
Step 2. $dk \leftarrow \mathcal{K}_D$, $b \leftarrow \{0, 1\}$, $\chi \leftarrow \mathsf{DEM.Enc}_{dk}(m_b)$.
Step 3. $\tilde{b} \leftarrow \mathcal{A}_D^{\mathcal{O}}(v, \chi)$

For IND-RCCA security, all is the same except that the decryption oracle returns 'test' for any ciphertext decrypts to m_0 or m_1 in step 3.

KEM+DEM hybrid paradigm works as follows, and [11] shows that if KEM and DEM are IND-CCA secure then the following HPKE is IND-CCA secure (as a public-key encryption).

```
Function: $\mathsf{HPKE.Enc}_{pk}(m)$
    $(dk, \psi) \leftarrow \mathsf{KEM.Enc}_{pk}()$
    $\chi \leftarrow \mathsf{DEM.Enc}_{dk}(m)$
    Output $c = (\psi, \chi)$
```

```
Function: $\mathsf{HPKE.Dec}_{sk}(c)$
    $(\psi, \chi) \leftarrow c$
    $dk \leftarrow \mathsf{KEM.Dec}_{sk}(\psi)$
    $m \leftarrow \mathsf{DEM.Dec}_{dk}(\chi)$
    Output $m$
```

2.3 Tag-KEM/DEM Hybrid Framework and Related Security Notions

A Tag-KEM scheme consists of the following algorithms. Probabilistic $\mathsf{TKEM.Gen}$ generates public/private-key pair (pk, sk), pk defines spaces for tags and encapsulated keys denoted by $\mathcal{T}$ and $\mathcal{K}_K$. Probabilistic $\mathsf{TKEM.Key}$ outputs one-time key $dk \in \mathcal{K}_D$ and internal state information ω, $\mathcal{K}_D$ is the key-space of DEM. Probabilistic $\mathsf{TKEM.Enc}$ encapsulates dk (embedded in ω) into ψ along with τ. $\mathsf{TKEM.Dec}$ recovers dk from ψ and τ. An obvious soundness condition applies.

Let $\mathcal{O}$ be the decapsulation oracle $\mathsf{TKEM.Dec}_{sk}(\cdot,\cdot)$, the IND-CCA security of a Tag-KEM is defined by the following game:

```
[GAME.TKEM]
Step 1. $(pk, sk) \leftarrow \mathsf{TKEM.Gen}(1^k)$
Step 2. $v_1 \leftarrow \mathcal{A}_T^{\mathcal{O}}(pk)$
Step 3. $(\omega, dk_1) \leftarrow \mathsf{TKEM.Key}(pk)$, $dk_0 \leftarrow \mathcal{K}_D$, $\delta \leftarrow \{0,1\}$.
Step 4. $(\tau, v_2) \leftarrow \mathcal{A}_T^{\mathcal{O}}(v_1, dk_\delta)$
Step 5. $\psi \leftarrow \mathsf{TKEM.Enc}(\omega, \tau)$
Step 6. $\tilde{\delta} \leftarrow \mathcal{A}_T^{\mathcal{O}}(v_2, \psi)$
```

In Step 6, A_T is restricted not to ask (ψ, τ) to decryption oracle $\mathcal{O}$. IND-RCCA security for Tag-KEM can be derived in the same way as for KEM.

The novel generic construction of Tag-KEM/DEM which uses the ciphertext output by the DEM as the tag is as follows, and [3] shows that if Tag-KEM is CCA secure and DEM is passive secure then the hybrid HPKE scheme is CCA secure.

```
Function: $\mathsf{HPKE.Enc}_{pk}(m)$
    $(\omega, dk) \leftarrow \mathsf{TKEM.Key}(pk)$
    $\chi \leftarrow \mathsf{DEM.Enc}_{dk}(m)$
    $\psi \leftarrow \mathsf{TKEM.Enc}_{pk}(\omega, \chi)$
    Output $c = (\psi, \chi)$
```

```
Function: $\mathsf{HPKE.Dec}_{sk}(c)$
    $(\psi, \chi) \leftarrow c$
    $dk \leftarrow \mathsf{TKEM.Dec}_{sk}(\psi, \chi)$
    $m \leftarrow \mathsf{DEM.Dec}_{dk}(\chi)$
    Output $m$
```

3 The Proposed RCCA-Secure Scheme: RSA-TKEM

Our RSA-TKEM is very simple and has a tight reduction, just like RSA-KEM in [20]. The description is as follows (for simplicity, we assume all elements are expressed as binary strings):

- $\mathsf{Gen}(1^\lambda)$: The same as RSA and also output a hash function H maps a bit string of arbitrary length to appropriate one (allowing the XOR). Let $((n, e), (n, d))$

denotes the public-private key pair (pk, sk). The pk also defines the output key length l.

- $\mathsf{Key}(pk)$: Generate a random number $r \in \{0, 1, ..., n-1\}$, then compute $K = G(r)$, where G is a KDF function (see [11]), it maps elements of $\mathbb{Z}_n$ to bit strings of length l. Output $(K, \omega = r)$.
- $\mathsf{Enc}_{pk}(\omega, \tau)$: Compute $s = r^e \bmod n$ and $y = s \oplus H(\tau)$. Output y.
- $\mathsf{Dec}_{sk}(y, \tau)$:Compute $s = y \oplus H(\tau)$, $r = s^d \bmod n$ and $K = G(r)$. Output K.

Note that in the scheme a hash of tag is XORed to s to yield the ciphertext rather than concatenated inside or outside the encryption operation of the underlying one-way function, so that there is no validity check operation in the decryption. In fact, a ciphertext-tag pair with different τ (except for collision) but same y will result in a different decryption, which ensures the security[1].

3.1 Security

The security of RSA-TKEM can be analyzed in the random oracle model when G is modeled as a random oracle. The formal theorem states as follows:

Theorem 1. *RSA-TKEM is IND-RCCA secure assuming the hardness of RSA problem. Particularly,*

$$Adv^{rcca}_{\mathsf{tkem},\mathcal{A}} \leq \varepsilon_{\mathsf{rsa},\mathcal{B}} + q_D \cdot \varepsilon_{ch} + \frac{q_D}{nBound}. \tag{1}$$

where

- ε_{ch} *is the advantage of finding a collision of* H*;*
- q_D *is a bound on the number of decryption oracle queries made by* $\mathcal{A}$*;*
- *nBound is an lower bound on* n*;*
- $\mathcal{B}$ *is an algorithm for solving a random instance of the RSA problem,* $\varepsilon_{\mathsf{rsa},\mathcal{B}}$ *is the success probability.* $\mathcal{B}$ *runs in time roughly the same as that of* $\mathcal{A}$*; more precisely, the running time is that of* $\mathcal{A}$*, plus the time to perform* q_G *exponentiations modulo* n*, where* q_G *is a bound on the number of random oracle queries made by* $\mathcal{A}$*, and the time to perform* q_D *hash and XOR operations, Where* q_D *is a bound on the number of decryption oracle queries.*

Proof. The proof is quite similar to that of RSA-KEM in [20], and follows the common game-modifying method. On the way of modifying the game, we use the following lemma.

Lemma 1. *(Shoup's Lemma [11]) Let* P*,* Q*, and* F *be events defined on some probability space, such that* $\Pr[P \wedge \neg F] = \Pr[Q \wedge \neg F]$*, then* $|\Pr[P] - \Pr[Q]| \leq \Pr[F]$*.*

[1] RSA-TKEM achieves publicly-detectable RCCA security [10]. Although weaker private-detectable or rerandomizable [13,17] ones are more desired, the scheme is sufficient for our purpose.

Let G_0 be the original attack game $\mathsf{RGAME.TKEM}$ played by adversary $\mathcal{A}$, and let S_0 be the event that $\mathcal{A}$ correctly guesses the hidden bit b in game G_0, and S_i be the same event in the following games. Let (y^*, τ^*) denote the target ciphertext-tag pair, and let $r^* = (y^* \oplus H(\tau^*))^{1/e} \in \mathbb{Z}_n$.

We next define a game G_1 that is the same as game G_0, except that if a ciphertext-tag pair (y, τ) with $y \oplus H(\tau) = y^* \oplus H(\tau^*) = s^*$ was submitted to the decryption oracle prior to the invocation of the encryption oracle, then the game is halted. Let F_1 be the event that game G_1 is halted as above. Since $y \oplus H(\tau)$ has the same length as n, and a smarter $\mathcal{A}$ can always choose (y, τ) satisfying $y \oplus H(\tau)$ in $\mathbb{Z}_n$, so $\Pr[F_1] \leq q_D/n \leq q_D/nBound$, and since games G_0 and G_1 proceed identically until F_1 occurs, it follows by Lemma 1 that $|\Pr[S_0] - \Pr[S_1]| \leq q_D/nBound$.

We then define a game G_2 that is the same as G_1, except that if a ciphertext (y^*, τ) with $H(\tau) = H(\tau^*)$ was queried by the adversary $\mathcal{A}$ after the invocation of the encryption oracle, we halt the game. Let F_2 be the event that game G_2 is halted as above. Since τ^* is chosen by $\mathcal{A}$ itself, if $\mathcal{A}$ can find a collision (τ_1, τ_2) for H then it can use one of them as τ^*. So, $\Pr[F_2] \leq \varepsilon_{ch}$, and since games G_1 and G_2 proceed identically until F_2 occurs, it follows that $|\Pr[S_1] - \Pr[S_2]| \leq \varepsilon_{ch}$.

Finally, we define a game G_3 that is the same as G_2, except that if (1)the target ciphertext is generated by an independent and randomly choosen s^*, and $y^* = s^* \oplus H(\tau^*)$ and (2) the adversary ever queries G at r^*, then we halt the game.

It is clear by construction that $\Pr[S_3] = 1/2$, since the key $G(r^*)$ is independent of everything else that is accessible to the adversary in game G_3, either directly or indirectly. Indeed, only the encryption oracle evaluates G at r^* in this game.

Let F_3 be the event that game G_3 is halted as above. It is clear that both games G_2 and G_3 proceed identically until F_3 occurs, and so we have $|\Pr[S_2] - \Pr[S_3]| \leq \Pr[F_3]$.

We claim that $\Pr[F_3] \leq \varepsilon_{\mathsf{rsa},\mathcal{B}}$ for an RSA inversion adversary $\mathcal{B}$ whose runtime is bounded as described in the theorem. $\mathcal{B}$ takes as input a random RSA modulus n, an RSA exponent e, and a random element $s^* \in \mathbf{Z}_n$. It creates a public key using n and e, and then lets adversary $\mathcal{A}$ run in game G_3. When $\mathcal{A}$ chooses a tag τ^* and then invokes the encryption oracle, $\mathcal{B}$ responds to $\mathcal{A}$ with $s^* \oplus H(\tau^*)$, where s^* is the above-mentioned input to $\mathcal{B}$.

$\mathcal{B}$ simulates the random oracle G as well as the decryption oracle by maintaining two lists: G&D-list and D-list. The lists are initially empty and have entries of the form (r, g, s) and (s, g) respectively, where $s = r^e$.

$\mathcal{B}$ answers $\mathcal{A}$'s random oracle queries to G and decryption oracle queries as follows:

```
Simulation for G queries
If $r \in$ G&D-list then return the corresponding $g$
Else compute $s = r^e$
    If $s \in$ D-list then return the corresponding $g$, add $(r, g, s)$ in G&D-list
    Else choose a random $g \in \mathcal{K}_D$, add $(r, g, s)$ in G&D-list, return $g$.
```

```
Simulation for Decryption queries
Compute $s = y \oplus H(\tau)$
If $s = s^*$ then return ‘test’
Else
  If $s \in$ G&D-list or $s \in$ D-list then return the corresponding $g$
  Else choose a random $g \in \mathcal{K}_D$, add $(s, g)$ in D-list, return $g$.
```

$\mathcal{B}$ perfectly simulates the view of $\mathcal{A}$, and that $\mathcal{B}$ outputs a solution to the given instance of the RSA problem with probability equal to $\Pr[F_3]$.

Collecting the probabilities, (1) follows immediately, that proves the theorem. □

4 Considering RCCA Security for the Hybrid Paradigms

In this section we discuss the security results for achieving RCCA secure hybrid encryptions from the well known paradigm, KEM+DEM and Tag-KEM/DEM.

4.1 Obtaining RCCA-Security for KEM+DEM

For KEM+DEM paradigm, similar result holds when considering both CCA security and RCCA one (benign malleability). That is, if KEM and DEM are benignly malleable then KEM+DEM paradigm is also benignly malleable [20]. The proof is quite the same as that for HE1 in [10], we omit it here.

Theorem 2. *If both KEM and DEM are IND-RCCA secure then the Hybrid PKE scheme following KEM+DEM paradigm is IND-RCCA secure (as a public-key encryption scheme).*

4.2 Obtaining RCCA-Security for Tag-KEM/DEM

However, when we consider RCCA security for Tag-KEM/DEM, things will be different. Although RCCA-secure Tag-KEMs seem simple to be constructed, they cannot be used to obtain RCCA-secure hybrid encryptions when combined with just passive secure DEMs. Take the proposed RSA-TKEM as the RCCA-secure Tag-KEM and one-time pad as the passive secure DEM, this is easily to be seen. For the hybrid encryption to be RCCA-secure, CCA security, or rigorously, RCCA security for DEM should be asked again.

Theorem 3. *If both Tag-KEM and DEM are IND-RCCA secure then the hybrid scheme in Section 2.3 is IND-RCCA secure(as a PKE). In particular, for every* $\mathcal{H}$*, there exist* $\mathcal{A}_T$ *and* $\mathcal{A}_D$ *with*

$$Adv^{rcca}_{\mathsf{pke},\mathcal{H}}(k) \leq 2Adv^{rcca}_{\mathsf{tkem},\mathcal{A}_T}(k) + Adv^{rcca}_{\mathsf{dem},\mathcal{A}_D}(k) + \frac{q_D}{|\mathcal{K}_K|}. \tag{2}$$

where q_D *is a bound on the number of decryption oracle queries made by an IND-RCCA attacker against HPKE, and* l *is the length of the key used in DEM.*

The proof for Theorem 3 can be shown in a similar way as done for CCA secure Tag-KEM and passive secure DEM in [3], except that the adversary $\mathcal{A}_D$ against DEM must use its own decryption oracle to answer a query (ψ, χ) from the adversary $\mathcal{H}$ against HPKE when $\mathsf{TKEM.Dec}_{sk}(\psi, \chi) = \mathsf{TKEM.Dec}_{sk}(\psi^*, \chi^*)$. That is why CCA security for DEM is needed here.

Proof. Let $\mathcal{H}$ be an adversary playing $\mathsf{RGAME.PKE}$. We modify the game by using a random key dk^+ in place of the legitimate one generated by $\mathsf{TKEM.Key}$ in both the encryption and decryption oracle. Call this game $\mathsf{RGAME.PKE}'$. Let T and T' be events that $\tilde{b} = b$ in $\mathsf{RGAME.PKE}$ and $\mathsf{RGAME.PKE}'$, respectively. Then we claim that $|\Pr[T] - \Pr[T']| \leq 2Adv^{rcca}_{\mathsf{tkem}}(k) + \frac{q_D}{|\mathcal{K}_K|}$, which is shown by constructing $\mathcal{A}_T$ that attacks the underlying TKEM scheme by using $\mathcal{H}$.

First $\mathcal{A}_T$ is given public-key pk and passes it to $\mathcal{H}$. Given m_0 and m_1 from $\mathcal{H}$, $\mathcal{A}_T$ requests the encryption oracle of $\mathsf{RGAME.TKEM}$ to obtain (dk_δ, ψ^*). $\mathcal{A}_T$ then selects $b \in \{0, 1\}$ and computes $\chi^* = \mathsf{DEM.Enc}_{dk_\delta}(m_b)$, and sends (ψ^*, χ^*) to $\mathcal{H}$.

$\mathcal{A}_T$ answers $\mathcal{H}$'s decryption query (ψ, χ) as follows:

- If $(\psi, \chi) \neq (\psi^*, \chi^*)$, then $\mathcal{A}_T$ just forwards (ψ, χ) to its own decryption oracle $\mathsf{TKEM.Dec}_{sk}(\cdot)$.
    - If $\perp$ is returned, then $\mathcal{A}_T$ returns $\perp$ to $\mathcal{H}$.
    - If 'test' is returned and $\chi \neq \chi^*$, then $\mathcal{A}_T$ uses dk_δ to decrypt χ.
        - If m_0 or m_1 is obtained, then $\mathcal{A}_T$ returns 'test' to $\mathcal{H}$.
        - Else $\mathcal{A}_T$ returns the result to $\mathcal{H}$.
    - If 'test' is returned and $\chi = \chi^*$, then $\mathcal{A}_T$ returns 'test' to $\mathcal{H}$.
    - If dk is returned, then $\mathcal{A}_T$ uses dk to decrypt χ.
        - If m_0 or m_1 is obtained, then $\mathcal{A}_T$ returns 'test' to $\mathcal{H}$.
        - Else $\mathcal{A}_T$ returns the result to $\mathcal{H}$.

The simulation is perfect unless ψ decrypts to dk_0 and thus 'test' is returned by $\mathsf{TKEM.Dec}_{sk}(\cdot)$ in $\mathsf{RGAME.PKE}$. However, the probability of this event is $\frac{1}{|\mathcal{K}_K|}$ for each such query since in that case dk_0 is random and independent from the view of $\mathcal{H}$.

When $\mathcal{H}$ finally outputs $\tilde{b}$, if $\tilde{b} = b$ then $\mathcal{A}_T$ outputs $\tilde{\delta} = 1$, meaning dk_δ is the real key. Otherwise, $\mathcal{A}_T$ outputs $\tilde{\delta} = 0$. Accordingly, $|\Pr[\tilde{b} = b|\delta = 1] - \Pr[T]| \leq \frac{q_D}{|\mathcal{K}_K|}$, and $\Pr[\tilde{b} = b|\delta = 0] = \Pr[T']$. Therefore,

$$\begin{aligned} Adv^{rcca}_{\mathsf{tkem},\mathcal{A}_T}(k) &= |\Pr[\tilde{\delta} = \delta] - \frac{1}{2}| \\ &= \frac{1}{2}|\Pr[\tilde{\delta} = 1|\delta = 1] - \Pr[\tilde{\delta} = 1|\delta = 0]| \\ &= \frac{1}{2}|\Pr[\tilde{b} = b|\delta = 1] - \Pr[\tilde{b} = b|\delta = 0]| \\ &\geq \frac{1}{2}|\Pr[T] - \Pr[T'] - \frac{q_D}{|\mathcal{K}_K|}|. \end{aligned}$$

Hence, we have $|\Pr[T] - \Pr[T']| \leq 2Adv^{rcca}_{\mathsf{tkem},\mathcal{A}_T}(k) + \frac{q_D}{|\mathcal{K}_K|}$.

Lastly, we show that $\mathcal{H}$ playing $\mathsf{RGAME.PKE}'$ essentially conducts an IND-RCCA attack to DEM, i.e. we claim $|\Pr[T'] - \frac{1}{2}| \leq Adv^{rcca}_{\mathsf{dem}}(k)$. This is shown by constructing an RCCA attacker $\mathcal{A}_D$ to DEM. $\mathcal{A}_D$ first generate (pk, sk) by using $\mathsf{PKE.Gen}$ and gives pk to $\mathcal{H}$. When (m_0, m_1) has been chosen by $\mathcal{H}$, $\mathcal{A}_D$ forwards them to encryption oracle of $\mathsf{RGAME.DEM}$ and receives ciphertext χ^*. It then generates (dk^*, ψ^*) by using χ^* as a tag and following $\mathsf{TKEM.Key}$ and $\mathsf{TKEM.Enc}$, and sends (ψ^*, χ^*) to $\mathcal{H}$. Note that the key dk^+ chosen by encryption oracle of $\mathsf{RGAME.DEM}$ and the one embedded in ψ^* are independent and randomly chosen. $\mathcal{A}_D$ answers $\mathcal{H}$'s decryption query (ψ, χ) as follows:

- If $(\psi, \chi) \neq (\psi^*, \chi^*)$, then $\mathcal{A}_D$ uses sk to decrypt (ψ, χ).
  - If the result is $\perp$, then $\mathcal{A}_D$ returns $\perp$ to $\mathcal{H}$.
  - If the result is dk^* and $\chi = \chi^*$ then $\mathcal{A}_D$ returns 'test' to $\mathcal{H}$.
  - If the result is dk^* and $\chi \neq \chi^*$ then $\mathcal{A}_D$ forwards χ to its own decryption oracle $\mathsf{DEM.Dec}_{dk^+}(\cdot)$, and returns the result to $\mathcal{H}$.
  - If the result is $dk \neq dk^*$ then $\mathcal{A}_D$ uses dk to decrypt χ.
    - If m_0 or m_1 is obtained, then $\mathcal{A}_D$ returns 'test' to $\mathcal{H}$.
    - Else $\mathcal{A}_D$ returns the result to $\mathcal{H}$.

When $\mathcal{H}$ outputs $\tilde{b}$, $\mathcal{A}_D$ outputs $\tilde{b}$, too. $\mathcal{A}_D$ perfectly simulates

$\mathsf{RGAME.PKE}'$, and whenever $\mathcal{H}$ wins, so does $\mathcal{A}_D$. Hence $|\Pr[T'] - \frac{1}{2}| = Adv^{rcca}_{\mathsf{dem},\mathcal{A}_D}(k)$.

In summary, we have:

$$\begin{aligned} |(\Pr[T] - \frac{1}{2}) - (\Pr[T'] - \frac{1}{2})| &\leq 2Adv^{rcca}_{\mathsf{tkem},\mathcal{A}_T}(k) + \frac{q_D}{|\mathcal{K}_K|} \\ Adv^{rcca}_{\mathsf{pke},\mathcal{H}}(k) &\leq 2Adv^{rcca}_{\mathsf{tkem},\mathcal{A}_T}(k) + Adv^{rcca}_{\mathsf{dem},\mathcal{A}_D}(k) + \frac{q_D}{|\mathcal{K}_K|}. \quad \square \end{aligned}$$

5 KEM/Tag-DEM: From RCCA Security to CCA One

Consider the method in [10] that makes an RCCA-secure PKE to be CCA-secure one, we generalize it to be based on an RCCA-secure KEM scheme.

By using the ciphertext of KEM as a tag of DEM, we will let the security of DEM provide non-malleability for KEM.

5.1 Tag-DEM and the Hybrid Paradigm

A Tag-DEM is a one-time symmetric-key encryption scheme with a tag as an additional input. It consists of two algorithm. Deterministic $\mathsf{TDEM.Enc}$ takes an input 1^k, a key dk, a message m and a tag $\tau \in \mathcal{T}$, and outputs a ciphertext χ, $\mathcal{T}$ is the tag space. Deterministic $\mathsf{TDEM.Dec}$ takes as input 1^k, a key dk, a ciphertext χ and a tag τ, and outputs a message m or the special symbol $\perp$. An obvious soundness condition applies.

We define IND-CCA security for Tag-DEMs by the following game. $\mathcal{O}$ denotes $\mathsf{TDEM.Dec}_{dk}(\cdot,\cdot)$. In Step 3, $\mathcal{A}_D$ is restricted not to ask (χ,τ) to $\mathcal{O}$.

```
[GAME.TDEM]
Step 1. $(m_0, m_1, \tau, v) \leftarrow \mathcal{A}_D(1^k)$
Step 2. $dk \leftarrow \mathcal{K}_D$, $b \leftarrow \{0,1\}$, $\chi \leftarrow \mathsf{TDEM.Enc}_{dk}(m_b, \tau)$.
Step 3. $\tilde{b} \leftarrow \mathcal{A}_D^{\mathcal{O}}(v, \chi)$
```

Note that Tag-DEMs are deterministic symmetric-key encryptions with tag, and we only ask one-time CCA-security but not authenticity, this makes them weaker than authenticated encryptions with associated data [18], even a deterministic one.

Our KEM/Tag-DEM works as follows:

```
Function: $\mathsf{HPKE.Enc}_{pk}(m)$
    $(dk, \psi) \leftarrow \mathsf{KEM.Enc}_{pk}()$
    $\chi \leftarrow \mathsf{TDEM.Enc}_{dk}(m, \psi)$
    Output $c = (\psi, \chi)$
```

```
Function: $\mathsf{HPKE.Dec}_{sk}(c)$
    $(\psi, \chi) \leftarrow c$
    $dk \leftarrow \mathsf{KEM.Dec}_{sk}(\psi)$
    $m \leftarrow \mathsf{TDEM.Dec}_{dk}(\chi, \psi)$
    Output $m$
```

Theorem 4. *If KEM is IND-RCCA secure and TDEM is IND-CCA secure then the Hybrid PKE scheme above is IND-CCA secure(as a public-key encryption scheme). In particular, for every $\mathcal{H}$, there exist $\mathcal{A}_K$ and $\mathcal{A}_D$ with*

$$Adv^{\mathsf{cca}}_{\mathsf{pke},\mathcal{H}} \le 2Adv^{\mathsf{rcca}}_{\mathsf{kem},\mathcal{A}_K} + Adv^{\mathsf{cca}}_{\mathsf{tdem},\mathcal{A}_D} + \frac{q_D}{|\mathcal{K}_K|}. \tag{3}$$

Proof. Let $\mathcal{H}$ be an adversary playing $\mathsf{GAME.PKE}$. We modify the game by using a random key dk^{+} in place of the legitimate one generated by $\mathsf{KEM.Enc}$ in both the encryption and decryption oracle. Call this game $\mathsf{GAME.PKE}'$. Let T and T' be events that $\tilde{b} = b$ in $\mathsf{GAME.PKE}$ and $\mathsf{GAME.PKE}'$, respectively. Then we claim that $|\Pr[T] - \Pr[T']| \le 2Adv^{rcca}_{\mathsf{kem},\mathcal{A}_K}(k)$, which is shown by constructing $\mathcal{A}_K$ that attacks the underlying KEM scheme by using $\mathcal{H}$.

First $\mathcal{A}_K$ is given pk and passes it to $\mathcal{H}$. Given m_0 and m_1 from $\mathcal{H}$, $\mathcal{A}_K$ requests the encryption oracle of $\mathsf{RGAME.KEM}$ to obtain (dk_δ, ψ^*). $\mathcal{A}_K$ then selects $b \in \{0,1\}$ and computes $\chi^* = \mathsf{DEM.Enc}_{dk_\delta}(m_b, \psi^*)$, and sends (ψ^*, χ^*) to $\mathcal{H}$.

$\mathcal{A}_K$ answers $\mathcal{H}$'s decryption query (ψ, χ) as follows:

- If $\psi = \psi^*$ and so that $\chi \ne \chi^*$, then $\mathcal{A}_K$ uses dk_δ to decrypt (χ, ψ).
- If $\psi \ne \psi^*$, then $\mathcal{A}_K$ just forwards ψ to its own decryption oracle $\mathsf{KEM.Dec}_{sk}(\cdot)$.
    - If $\perp$ is returned, then $\mathcal{A}_K$ returns $\perp$ to $\mathcal{H}$.
    - If 'test' is returned, then $\mathcal{A}_K$ uses dk_δ to decrypt (χ, ψ) by applying $\mathsf{TDEM.Dec}_{dk_\delta}(\cdot,\cdot)$, returns the result to $\mathcal{H}$.
    - If dk is returned, then $\mathcal{A}_K$ uses this dk to decrypt (χ, ψ) by applying $\mathsf{TDEM.Dec}_{dk}(\cdot,\cdot)$, and returns the result to $\mathcal{H}$.

When $\mathcal{H}$ finally outputs $\tilde{b}$, if $\tilde{b} = b$ then $\mathcal{A}_K$ outputs $\tilde{\delta} = 1$, meaning dk_δ is the real key. Otherwise, $\mathcal{A}_K$ outputs $\tilde{\delta} = 0$. Accordingly, $|\Pr[\tilde{b} = b|\delta = 1] - \Pr[T]| \leq \frac{q_D}{|\mathcal{K}_K|}$, and $\Pr[\tilde{b} = b|\delta = 0] = \Pr[T']$. Therefore, similar as in the proof of Theorem 3, we have $|\Pr[T] - \Pr[T']| \leq 2Adv^{rcca}_{\mathsf{kem},\mathcal{A}_K}(k) + \frac{q_D}{|\mathcal{K}_K|}$.

Lastly, we show that $\mathcal{H}$ playing $\mathsf{GAME.PKE}'$ essentially conducts an IND-CCA attack to TDEM, i.e. we claim $|\Pr[T'] - \frac{1}{2}| \leq Adv^{cca}_{\mathsf{tdem},\mathcal{A}_D}(k)$. This is shown by constructing a CCA attacker $\mathcal{A}_D$ to DEM. $\mathcal{A}_D$ first generate (pk, sk) by using $\mathsf{PKE.Gen}$ and gives pk to $\mathcal{H}$. When (m_0, m_1) has been chosen by $\mathcal{H}$, $\mathcal{A}_D$ first generates (dk^*, ψ^*) by following $\mathsf{KEM.Enc}$, then forwards (m_0, m_1, ψ^*) to encryption oracle of $\mathsf{GAME.TDEM}$. When $\mathcal{A}_D$ receives ciphertext χ^*, it sends (ψ^*, χ^*) to $\mathcal{H}$. Note that the key chosen by encryption oracle of $\mathsf{GAME.TDEM}$ dk^+ and the one embedded in ψ^* are independent and randomly chosen.

$\mathcal{A}_D$ answers $\mathcal{H}$'s decryption query (ψ, χ) as follows:

- If $\psi = \psi^*$ and so that $\chi \neq \chi^*$, then $\mathcal{A}_D$ forwards (χ, ψ) to its own decryption oracle $\mathsf{DEM.Dec}_{dk^+}(\cdot,\cdot)$, and returns the result to $\mathcal{H}$.
- If $\psi \neq \psi^*$, then $\mathcal{A}_D$ uses sk to decrypt ψ.
    - If the result is $\perp$, then $\mathcal{A}_D$ returns $\perp$ to $\mathcal{H}$.
    - If the result is dk^*, then $\mathcal{A}_D$ forwards (χ, ψ) to its own decryption oracle $\mathsf{DEM.Dec}_{dk^+}(\cdot,\cdot)$, and returns the result to $\mathcal{H}$.
    - If the result is $dk \neq dk^*$, then $\mathcal{A}_D$ uses this dk to decrypt χ.

When $\mathcal{H}$ outputs $\tilde{b}$, $\mathcal{A}_D$ outputs $\tilde{b}$, too. $\mathcal{A}_D$ perfectly simulates $\mathsf{GAME.PKE}'$, and whenever $\mathcal{H}$ wins, so does $\mathcal{A}_D$. Hence $|\Pr[T'] - \frac{1}{2}| = Adv^{cca}_{\mathsf{tdem},\mathcal{A}_D}(k)$.

In summary, we have (3) immediately. □

We can illustrate the security result in anther view. In Tag-KEM/DEM paradigm, the ciphertext of DEM is used as the tag input to the encryption of Tag-KEM, this can weaken the security requirement for DEM. By the same reason, in our KEM/Tag-DEM paradigm, the use of ciphertext of KEM as the tag of DEM should weaken the security requirement of KEM. Now we proved the security is weakened to be IND-RCCA security. Whether it can be further weakened remains open.

5.2 Two Direct Constructions of CCA Secure Tag-DEM Schemes

In this section we present some methods for constructing IND-CCA secure Tag-DEM schemes from One-time secure DEMs. The first one is based on the double-encryption structure used to make RCCA-secure PKE CCA-secure in [10]:

```
Function: $\mathsf{TDEM.Enc}(dk, mk, m, \tau)$
    $c_1 \leftarrow \mathsf{DEM.Enc}_{dk}(m\|\tau)$
    $c_2 \leftarrow \mathsf{MAC.Sign}_{mk}(c_1)$
    Output $\chi = (c_1\|c_2)$
```

```
Function: $\mathsf{TDEM.Dec}(dk, mk, \chi, \tau)$
    parse $\chi$ as $c_1\|c_2$
    If $\mathsf{MAC.Ver}_{mk}(c_2, c_1) = 1$ then
    $m\|\bar{\tau} \leftarrow \mathsf{DEM.Dec}(dk, c_1)$
    Else output $\perp$ EndIf
    If $\bar{\tau} = \tau$ then output $m$ Else output $\perp$
```

Theorem 5. *If DEM is One-time secure and* MAC *is one-time secure, then the above TDEM is IND-CCA secure. In particular, for every* $\mathcal{A}_D$*, there exists* $\mathcal{B}$ *with*

$$Adv^{\mathsf{cca}}_{\mathsf{tdem},\mathcal{A}_D} \leq Adv^{\mathsf{ot}}_{\mathsf{dem},\mathcal{B}} + q_D \cdot Adv^{\mathsf{forge}}_{\mathsf{mac}}. \tag{4}$$

where q_D *is a bound on the number of decryption oracle queries made by* $\mathcal{A}_D$*.*

Proof. Let $\mathcal{A}_D$ be an adversary playing GAME.TDEM, we construct a passive adversary $\mathcal{B}$ against DEM by using $\mathcal{A}_D$ as follows:

$\mathcal{B}$ forwards 1^k to $\mathcal{A}_D$. Given (m_0, m_1, τ^*) from $\mathcal{A}_D$, $\mathcal{B}$ computes $x_0 = m_0 \| \tau^*$ and $x_1 = m_1 \| \tau^*$, and requests (x_0, x_1) to the encryption oracle of GAME.DEM to obtain c^*. Then $\mathcal{B}$ let $c_1^* = c^*$, and randomly chooses mk from $\mathcal{K}_M$, computes $c_2^* = MAC.Sign_{mk}(c_1^*)$, sends $\chi^* = (c_1^*, c_2^*)$ to $\mathcal{A}_D$. For all decryption queries $(\chi = (c_1, c_2), \tau)$ from $\mathcal{A}_D$, $\mathcal{B}$ just returns $\perp$. Finally, when $\mathcal{A}_D$ outputs $\tilde{b}$, $\mathcal{B}$ outputs $\tilde{b}$, too.

For each decryption query, the simulation is correct unless $\mathsf{MAC.Ver}(c_2, c_1) = 1$. Let Forge denote this event, we have $\Pr[\mathsf{Forge}] \leq q_D \cdot Adv^{\mathsf{forge}}_{\mathsf{mac}}$, and (4) follows immediately. □

The second method avoids double encryption and just makes the tag τ as an input part of MAC:

```
Function: TDEM.Enc($dk, mk, m, \tau$)
  $c_1 \leftarrow$ DEM.Enc$_{dk}(m)$
  $c_2 \leftarrow$ MAC.Sign$_{mk}(c_1 \| \tau)$
  Output $\chi = (c_1 \| c_2)$
```

```
Function: TDEM.Dec($dk, mk, \chi, \tau$)
  parse $\chi$ as $c_1 \| c_2$
  If MAC.Ver$_{mk}(c_2, c_1 \| \tau) = 1$ then
  $m \leftarrow$ DEM.Dec$(dk, c_1)$
  Else output $\perp$ EndIf
  Output $m$.
```

Combined with an RCCA secure KEM, the above scheme coincides with the one in [3] to construct a CCA secure hybrid encryption from an RCCA secure KEM.

Theorem 6. *If DEM is One-time secure and* MAC *is one-time secure, then the above TDEM is IND-CCA secure. In particular, for every* $\mathcal{A}_D$*, there exists* $\mathcal{B}$ *with*

$$Adv^{\mathsf{cca}}_{\mathsf{tdem},\mathcal{A}_D} \leq Adv^{\mathsf{ot}}_{\mathsf{dem},\mathcal{B}} + q_D \cdot Adv^{\mathsf{forge}}_{\mathsf{mac}}. \tag{5}$$

where q_D *is a bound on the number of decryption oracle queries made by* $\mathcal{A}_D$*.*

Proof. Let $\mathcal{A}_D$ be an adversary playing GAME.TDEM, we construct a passive adversary $\mathcal{B}$ against DEM by using $\mathcal{A}_D$ as follows:

$\mathcal{B}$ forwards 1^k to $\mathcal{A}_D$. Given (m_0, m_1, τ^*) from $\mathcal{A}_D$, $\mathcal{B}$ lets $x_0 = m_0$, $x_1 = m_1$, and requests (x_0, x_1) to the encryption oracle of GAME.DEM to obtain c^*. Then $\mathcal{B}$ lets $c_1^* = c^*$, and randomly chooses mk from $\mathcal{K}_M$, computes $c_2^* = MAC.Sign_{mk}(c_1^* \| \tau^*)$, sends $\chi^* = (c_1^*, c_2^*)$ to $\mathcal{A}_D$.

For all decryption queries ($\chi = (c_1, c_2), \tau$) from $\mathcal{A}_D$, $\mathcal{B}$ just returns $\perp$. Finally, when $\mathcal{A}_D$ outputs $\tilde{b}$, $\mathcal{B}$ outputs $\tilde{b}$, too.

The simulation is correct unless $\mathsf{MAC.Ver}(c_2, c_1 \| \tau) = 1$. Let Forge denote this event, we have $\Pr[\mathsf{Forge}] \le q_D \cdot Adv_{\mathsf{mac}}^{\mathsf{forge}}$, and equation (5) follows immediately. □

Additionally, Tag-DEMs can be built more efficiently from conventional IV-based encryption schemes as in deterministic authenticated encryptions [19]. Taking the tag τ as a header, in the encryption it is no need to require the privacy of τ. But there must be a way to bind the tag τ to the encrypted message and provide authenticity for both of them. For example, the SIV construction in [19] provides a method for constructing Tag-DEM with shorter ciphertext. Details can be found in Appendix.

Acknowledgment. This work is supported by the National Natural Science Foundations of China (No. 60970120) and the Fundamental Research Funds for the Central Universities (Nos. K50510010024 and K50511010009.).

References

1. Abe, M., Cui, Y., Imai, H., Kurosawa, K.: Tag-KEM from Set Partial Domain One-Way Permutations. In: Batten, L., Safavi-Naini, R. (eds.) ACISP 2006. LNCS, vol. 4058, pp. 360–370. Springer, Heidelberg (2006)
2. Abe, M., Gennaro, R., Kurosawa, K.: Tag-KEM/DEM: A new framework for hybrid encryption. Cryptology ePrint Archive: Report 2005/027 (2005)
3. 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)
4. Abe, M., Gennaro, R., Kurosawa, K., Shoup, V.: Tag-KEM/DEM: A new framework for hybrid encryption. J. Cryptology 21(1), 97–130 (2008)
5. An, J.H., Dodis, Y., Rabin, T.: On the security of joint signature and encryption. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 83–107. Springer, Heidelberg (2002)
6. Bellare, M., Desai, A., Pointcheval, D., Rogaway, P.: Relations among notions of security for public-key encryption schemes. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 26–45. Springer, Heidelberg (1998)
7. Bellare, M., Desai, A., Jokipii, E., Rogaway, P.: A concrete security treatment of symmetric encryption: analysis of the DES modes of operation. In: Proceedings of the 38th Symposium on Foundations of Computer Science, pp. 394–403. IEEE Press (1997)
8. Bellare, M., Rogaway, P.: Random oracles are practicala paradigm for designing efficient protocols. In: Proceedings of the First Annual Conference on Computer and Communications Security, pp. 62–73. ACM, New York (1993)
9. Bellare, M., Namprempre, C.: Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 531–545. Springer, Heidelberg (2000)

10. Canetti, R., Krawczyk, H., Nielsen, J.: Relaxing chosen-ciphertext security. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 565–582. Springer, Heidelberg (2003)
11. Cramer, R., Shoup, V.: Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack. Manuscript (December 17, 2001); SIAM Journal of Computing 33(1), 167–226 (2003)
12. Dent, A.W.: A designer's guide to KEMs. In: Paterson, K.G. (ed.) Cryptography and Coding 2003. LNCS, vol. 2898, pp. 133–151. Springer, Heidelberg (2003)
13. Groth, J.: Rerandomizable and replayable adaptive chosen ciphertext attack secure cryptosystems. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 152–170. Springer, Heidelberg (2004)
14. Hofheinz, D., Kiltz, E.: Secure Hybrid Encryption from Weakened Key Encapsulation. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 553–571. Springer, Heidelberg (2007)
15. Krawczyk, H.: The order of encryption and authentication for protecting communications (or: How secure is SSL?). In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 310–331. Springer, Heidelberg (2001)
16. Kurosawa, K., Desmedt, Y.: A new paradigm of hybrid encryption scheme. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 426–442. Springer, Heidelberg (2004)
17. Prabhakaran, M.M., Rosulek, M.: Rerandomizable RCCA Encryption. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 517–534. Springer, Heidelberg (2007)
18. Rogaway, P.: Authenticated-Encryption with Associated-Data. In: Proceedings of the 9th Annual Conference on Computer and Communications Security (CCS-9), pp. 98–107. ACM, New York (2002)
19. Rogaway, P., Shrimpton, T.: Deterministic Authenticated-Encryption: A Provable-Security Treatment of the Key-Wrap Problem. Full version of: A Provable-Security Treatment of the Key-Wrap Problem. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 373–390. Springer, Heidelberg (2006)
20. Shoup, V.: ISO 18033-2: An emerging standard for public-key encryption, committee draft (2004), http://shoup.net/iso/

A Message Authentication Code

MAC is a pair of algorithms ($\mathsf{MAC.Sign}$, $\mathsf{MAC.Ver}$). A key space $\mathcal{K}_M$ is defined by security parameter k. $\mathsf{MAC.Sign}$ takes a key $mk \in \mathcal{K}_M$ and a message $m \in 0, 1^*$ as inputs, and outputs a string σ. We say (σ, m) is valid with regard to mk if $\sigma = \mathsf{MAC.Sign}_{mk}(m)$. $\mathsf{MAC.Ver}$ takes a triple (mk, σ, m) as input and outputs 1 if (σ, m) is valid with respect to mk, or outputs 0, otherwise.

We define a one-time chosen message attacking game, $\mathsf{GAME.MAC}$. An adversary chooses an arbitrary message m and is given m's MAC σ created with mk randomly chosen from $\mathcal{K}_M$, it outputs (σ', m') which is different from (σ, m). The adversary wins if (σ', m') is valid with respect to the same mk. We say MAC is secure against one-time chosen message attack, or shorten as one-time secure, if any PPT adversary wins $\mathsf{GAME.MAC}$ with at most negligible probability in k, say $Adv_{\mathsf{mac}}^{\mathsf{forge}}(k)$.

B The SIV Construction of Tag-DEM

B.1 Conventional IV-Based Encryption Scheme

Encryption modes like CBC and CTR are the so called conventional IV-based encryption schemes. Such a scheme is a tuple $\Pi = (\mathsf{K},\mathsf{E},\mathsf{D})$, where K is a probabilistic algorithm, which on input 1^k outputs a key $K \in \mathcal{K}$, $\mathcal{K}$ be the *key space*. E is a deterministic encryption algorithm that takes as input an IV from *IV space* $\mathcal{IV}$, a key dk and a message m, and outputs a ciphertext $C = \mathsf{E}^{\mathcal{IV}}_{dk}(m)$. D is a deterministic decryption algorithm that takes as input an IV, a key dk and a ciphertext C, and outputs a message m or the special symbol $\perp$. Obvious soundness condition applies.

Fix $\mathcal{IV} = \{0,1\}^n$. For simplicity, we assume Π is length-preserving. Let $\mathsf{E}^{\$}_{dk}$ be the probabilistic algorithm defined from E, which on input dk and m, randomly chooses IV from $\mathcal{IV}$, then computes C as $\mathsf{E}^{\mathcal{IV}}_{dk}(m)$.

For consistency, we only require find-then-guess security against passive attacker, and demand a random IV. This makes the security notion rather weak, but sufficient for our purposes. Let $\mathcal{A}_E$ be a polynomial-time oracle machine that plays the following game.

[$\mathsf{GAME}.\Pi$]
Step 1. $(m_0, m_1, v) \leftarrow \mathcal{A}_E(1^k)$
Step 2. $dk \leftarrow \mathcal{K}$, $b \leftarrow \{0,1\}$, $(C, IV) \leftarrow \mathsf{E}^{\$}_{dk}(m_b)$.
Step 3. $\tilde{b} \leftarrow \mathcal{A}_E(v, C, IV)$
We define

$$Adv^{\mathsf{ot}}_{\Pi,\mathcal{A}_E}(k) = |Pr[\tilde{b} = b] - \frac{1}{2}|$$

and

$$Adv^{\mathsf{ot}}_{\Pi}(k) = max_{\mathcal{A}_E}(Adv^{\mathsf{ot}}_{\Pi,\mathcal{A}_E}(k)).$$

We say that Π is one-time secure if $Adv^{\mathsf{ot}}_{\Pi}(k)$ is negligible in k.

B.2 Arbitrary-Input PRFs

A pseudorandom function(PRF) is a map $F : \mathcal{K} \times \mathcal{X} \rightarrow \{0,1\}^n$ for some $n \geq 1$, $\mathcal{K}$ and $\mathcal{X}$ are fixed nonempty sets. F is pseudorandom if its input-output behavior is indistinguishable from that of a random function of the same domain and range.

We write $F_K(X)$ for $F(K, X)$. Let $Func(\mathcal{X}, \mathcal{Y})$ be the set of all functions from $\mathcal{X}$ to $\mathcal{Y}$ and let $Func(\mathcal{X}, n) = Func(\mathcal{X}, \{0,1\}^n)$. Regarding a function in $Func(\mathcal{X}, n)$ as the key, which associates a random string in $\{0,1\}^n$ to each $X \in \mathcal{X}$. The pseudorandomness of F is defined by the following game:

[GAME.PRF]
Step 1. $\rho \rightarrow Func(\mathcal{X}, n)$, $\mathcal{O}_0 \leftarrow \rho$, $K \leftarrow \mathcal{K}$, $\mathcal{O}_1 \leftarrow F_K$
Step 2. $\tilde{d} \leftarrow \mathcal{B}^{\mathcal{O}_d}$.
We define $Adv^{\mathsf{prf}}_{F,\mathcal{B}} = |Pr[\tilde{d} = 1|d = 1] - Pr[\tilde{d} = 1|d = 0]|$.

B.3 The Construction

The security notion used in [19] for DAE is stronger than our CCA one for Tag-DEM. For the consistency, we still use the CCA security defined in this paper.

```
Function: $\mathsf{TDEM.Enc}(dk1, dk2, m, \tau)$
    $IV \leftarrow \mathsf{F}_{dk1}(m, \tau)$
    $C \leftarrow \mathsf{E}^{IV}_{dk2}(m)$
    Output $\chi = IV \| C$
```

```
Function: $\mathsf{TDEM.Dec}(dk1, dk2, \chi, \tau)$
    parse $\chi$ as $IV \| C$
    $m \leftarrow \mathsf{D}^{IV}_{dk2}(C)$
    $IV' \leftarrow \mathsf{F}_{dk1}(m, \tau)$
    If $IV = IV'$ then output $m$
    Else output $\perp$
```

Theorem 7. *If $\Pi = (E, D)$ is a One-time secure IV-based encryption scheme with Iv-length n and F is a PRF, then the above TDEM is IND-CCA secure. In particular,*

$$Adv^{\mathsf{cca}}_{\mathsf{tdem},\mathcal{A}_D} \leq Adv^{\mathsf{ot}}_{\Pi,\mathcal{A}_E} + Adv^{\mathsf{prf}}_{\mathsf{F},B} + q_D/2^n. \tag{6}$$

where q_D is a bound on the number of decryption oracle queries made by an IND-CCA attacker against TDEM.

Proof. Let G_0 be the original attack game $\mathsf{GAME.TDEM}$ played by adversary $\mathcal{A}_D$, and let S_0 be the event that $\mathcal{A}$ correctly guesses the hidden bit b in game G_0. Let (ψ^*, τ^*) denote the target ciphertext-tag pair, and $IV^* = F_{dk1}(m_b, \tau^*)$, $C^* = E^{IV^*}_{dk2}(m_b)$ be the underlying IV and C. Thus, $\psi^* = IV^* \| C^*$.

We next define a game G_1 that is the same as game G_0, except that F_{dk1} is replaced by a random function $\rho \in Func(\mathcal{X}, n)$ in both the encryption and decryption oracle, where $\mathcal{X} = \mathcal{T} \times \{0,1\}^*$, $\mathcal{T}$ is the tag space. And ρ is hidden from the view of $\mathcal{A}_D$. Let S_1 be the event in game G_1 corresponding to the event S_0.

We claim that there is a adversary $\mathcal{B}$ against the pseudorandomness of F_K, such that

$$|\Pr[S_1] - \Pr[S_0]| = Adv^{\mathsf{prf}}_{\mathsf{F},\mathcal{B}}$$

$\mathcal{B}$ runs $\mathcal{A}_D$, forwards 1^k to $\mathcal{A}_D$. Given (m_0, m_1, τ^*) from $\mathcal{A}_D$, $\mathcal{B}$ chooses randomly $dk2$ and $b \leftarrow \{0,1\}$, and asks (τ^*, m_b) to its $F_{dk1}(\cdot)$ or $\rho(\cdot)$ oracle, when obtains IV^*, it computes $C^* = \mathsf{E}^{IV^*}_{dk2}(m_b)$, then returns $\psi^* = IV^* \| C^*$ to $\mathcal{A}_D$. For all decryption queries (χ, τ) from $\mathcal{A}_D$, $\mathcal{B}$ parses χ as $IV \| C$, since $\mathcal{B}$ knows $dk2$, it decrypts C and computes $m = \mathsf{D}^{IV}_{dk2}(C)$, asks (m, τ) to its own oracle, check whether or not the returned IV' is equal to IV, if yes then return m else return $\perp$.

$\mathcal{B}$ perfectly simulates the oracles of $\mathcal{A}_D$. If $\mathcal{A}_D$ outputs $\tilde{b} = b$, then $\mathcal{B}$ outputs $\tilde{d} = 1$, else outputs $\tilde{d} = 0$.

Finally, we define a game G_2 that is the same as G_1, except that $\perp$ is returned for all decryption queries. Let S_2 be the event in the game G_2 corresponding to the event S_0.

Let F_2 be the event that a valid ciphertext $\psi = IV \| C$ has been asked by $\mathcal{A}_D$, then it follows by Lemma 1 that $|\Pr[S_2] - \Pr[S_1]| \leq \Pr[F_2]$. A ciphertext is valid only when $\rho(\tau, m) = IV$, where $m = \mathsf{D}_{dk2}^{IV}(C)$. Since ρ is a random function, this happens only with probability of $\frac{1}{2^n}$. Thus, $|\Pr[S_1] - \Pr[S_2]| \leq \Pr[F_2] \leq \frac{q_D}{2^n}$.

Furthermore, we claim that there is an adversary $\mathcal{A}_E$ under the sense of one-time security against $\mathit{\Pi}$, such that $|\Pr[S_2] - \frac{1}{2}| = Adv_{\mathit{\Pi},\mathcal{A}_E}^{\mathsf{ot}}$.

$\mathcal{A}_E$ runs $\mathcal{A}_D$, forwards 1^k to $\mathcal{A}_D$. Given (m_0, m_1, τ^*) from $\mathcal{A}_D$, $\mathcal{A}_E$ asks (m_0, m_1) to its own encryption oracle $E_{dk}^{\$}(\cdot)$, when obtains the target ciphertext (C^*, IV^*), it forwards $IV^*\|C^*$ to $\mathcal{A}_D$. For all decryption queries (χ, τ) from $\mathcal{A}_D$, $\mathcal{A}_E$ returns $\perp$.

Since all ciphertexts decrypt to $\perp$, ρ loses its role in decryption, and a randomly chosen IV properly substitutes an IV computed by the random ρ. $\mathcal{A}_E$ perfectly simulates the oracles of $\mathcal{A}_D$. When $\mathcal{A}_D$ outputs $\tilde{b}$, $\mathcal{A}_E$ outputs $\tilde{d} = \tilde{b}$.

Collecting the probabilities, (6) will follow immediately, that proves the theorem. □