Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 8646973, 11 pages
https://doi.org/10.1155/2018/8646973

Research Article

On the RCCA Security of Hybrid Signcryption for Internet of Things

Honglong Dai,[1] Ding Wang ,[1,2] Jinyong Chang ,[1] and Maozhi Xu[1]

[1]*Peking University, Beijing 100871, China*
[2]*State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China*

Correspondence should be addressed to Ding Wang; wangdingg@pku.edu.cn

Received 31 August 2018; Accepted 28 October 2018; Published 12 November 2018

Guest Editor: Weizhi Meng

With the rapid development of the Internet of Things (IoT), a lot of sensitive information in our daily lives are now digitalized and open to remote access. The provision of security and privacy of such data would incur comprehensive cryptographic services and has raised wide concern. Hybrid signcryption schemes could achieve various kinds of cryptographic services (e.g., confidentiality, authenticity, and integrity) with much lower cost than the combination of separate traditional cryptographic schemes with each providing a single cryptographic service. Thus, hybrid signcryption schemes are very suitable for IoT environments where resources are generally very constrained (e.g., lightweight sensors and mobile phones). To ensure that the overall hybrid signcryption scheme provides adequate cryptographic service (e.g., confidentiality, integrity, and authentication), its parts of KEM (key encryption mechanism) and DEM (data encryption mechanism) must satisfy some security requirements. Chosen-ciphertext attack (CCA) security has been widely accepted as the golden standard requirement for general encryption schemes. However, CCA security appears too strong in some conditions. Accordingly, Canetti et al. (CRYPTO 2003) proposed the notion of replayable CCA security (RCCA) for encryption schemes, which is a strictly weaker security notion than CCA security and naturally more efficient. This new security notion has proved to be sufficient for most existing applications of CCA security, e.g., encrypted password authentication. This is particularly promising for IoT environments, where security is demanding, yet resources are constrained. In this paper, we examine the RCCA security of the well-known SKEM+DEM style hybrid signcryption scheme by Dent at ISC 2005. Meanwhile, we also examine the RCCA security of the Tag-SKEM+DEM style hybrid signcryption scheme by Bjorstad and Dent at PKC 2006. We rigorously prove that a hybrid signcryption scheme can achieve RCCA security if both its SKEM part and its DEM part satisfy some security assumptions.

1. Introduction

With the booming development of wireless technology, Internet of Things (IoT) has seen its proliferation in various applications such as personal health, government work, and battle surveillance. How to ensure security and privacy of the sensitive data in these security-critical applications is a challenging issue, because it would generally incur comprehensive cryptographic services. Hybrid signcryption schemes could achieve various kinds of cryptographic services (e.g., confidentiality, authenticity, and integrity) with much lower cost than the combination of separate traditional cryptographic schemes with each providing a single cryptographic service [1]. Thus, hybrid signcryption schemes are very suitable for IoT environments where resources are generally very constrained (e.g., lightweight sensors and mobile phones).

The first signcryption scheme was proposed by Zheng [2] at CRYPTO'97. The notion of confidentiality for a signcryption scheme is analogous to an original encryption scheme, while the nonrepudiation service is analogous to a digital signature one [3]. Since then, various kinds of signcryption schemes have been suggested. At 2002, Lee [4] proposed identity-based signcryption; At AsiaCCS'08, Barbosa et al. [5] proposed certificateless signcryption. At IMACC'13, Nakano et al. [6] presented two generic constructions of signcryption in the standard model. At 2017, Li et al. [7] proposed a signcryption for cloud computing. At PQCrypto'18 Sato et

al. [8] proposed lattice-based signcryption without random oracles. At the same time, Datta et al. [9] proposed the functional signcryption.

In addition, a number of signcryption schemes have been proposed for the IoT environments (e.g., key establishment over ATM networks [10], defense against fragment duplication attack in 6LoWPAN networks [11], short signcryption scheme for IoT [12], and provably secure signcryption for IoT [13]). Belguith et al. [14] proposed privacy preserving attribute based signcryption for IoT.

However, in the traditional signcryption schemes, the keyed encapsulation encryption is generally not made full use of, and the length of messages is always related to the signcryption scheme. Further, the major weakness of asymmetric encryption schemes is that the computational efficiency is worse than these symmetric ones [15]. Accordingly, the notion of hybrid signcryption is proposed. Hybrid signcryption uses a symmetric encryption scheme to improve the overall performance and flexibility of asymmetric signcryption. Hybrid signcryption can simultaneously combine the main advantages of a public key encryption and a digital signature scheme with much lower cost when compared with traditional schemes [1, 16]. As sensor nodes in IoT are resource-constrained (e.g., limited battery power) and deployed to run for years, hybrid cryptography is particularly suitable for data storage and transmission to achieve secure and efficient communication [17]. At 2004, Dent [15] proposed a formal composition model for hybrid signcryption, and this model covers Zheng's scheme [2]. Later, Bjorstad et al. [18] proposed an improve signcryption scheme with tag-KEMs, Li et al. [19] proposed a certificateless hybrid signcryption scheme, and Zhou [20] proposed an improved certificateless hybrid signcryption scheme. Due to the usage of a symmetric encryption scheme to overcome the weakness and restricted message space of traditional asymmetric encryption schemes, these hybrid signcryption schemes can make the length of message independent of the security of the overall signcryption scheme.

Secure encryption is one of the most fundamental tasks in cryptographic schemes, while CCA security has been widely accepted as the golden standard requirement for encryption schemes [21, 22]. However, chosen-ciphertext attack security appears to be too strong in many conditions; there exist many encryption schemes that are not CCA secure but still have practical applications [23]. Here we take a CCA secure public key encryption scheme PKE as example. We change it into a public key encryption scheme PKE', which is equal to public key encryption scheme PKE except that this encryption oracle algorithm appends a bit 0 to each ciphertext and the decryption oracle algorithm of PKE' discards this bit 0 of a ciphertext. Then, one naturally obtains a different ciphertext decrypted to the same message as the original one. However, this change takes no real consequence in most situation, because the modified scheme PKE′ appears to be as secure as the scheme PKE in most situations. This example is also used in [23].

Accordingly, Canetti et al. [23] proposed the RCCA security notion at CRYPTO 2003. RCCA security is a strictly weaker security notion than CCA security, which has proved to be abundant for most cryptographic primitives, e.g., encrypted password authentication [24]. There are some studies (e.g., [23, 25]) about the RCCA security of hybrid cryptography, and there are also several studies (e.g., [2, 3, 15]) about the CCA security of hybrid signcryption. As far as we know, there is no work about examining the RCCA security of hybrid signcryption. To fill the gap, in this paper we consider the RCCA security of hybrid signcryption and show that hybrid signcryption can achieve RCCA security (rather than only CCA security) based on certain conditions.

1.1. Main Contributions. In this paper, we examine the RCCA security of the hybrid signcryption scheme Tag-SKEM+DEM [18] and the hybrid signcryption scheme SKEM+DEM [3]. We will show the following: (1) The hybrid signcryption scheme (SKEM+DEM) can be RCCA-secure if the scheme Tag-SKEM is RCCA-secure and the scheme DEM is RCCA-secure. (2) The hybrid encryption scheme (Tag-SKEM+DEM) can be RCCA-secure if the signcryption scheme Tag-SKEM is RCCA-secure and the scheme DEM is RCCA-secure. Although our results might be expected and somewhat straightforward, we concretely confirm such expectations with a formal proof. When giving our proof, we mainly use the hybrid game-based reduction technique presented in [26–28].

1.2. Related Works and Discussions. It is obvious that if the hybrid signcryption scheme is going to provide an integrity and authentication service, then its KEM part and DEM part must satisfy some kind of security criterion. Dent et al. [15] examined the CCA security of hybrid signcryption schemes (SKEM+DEM and Tag-SKEM+DEM). Chen et al. [27] examined the RCCA security for hybrid encryption scheme KEM+DEM. Cui et al. [29] gave two kinds of RKA-secure signcryption schemes. In 2017, Dai et al. [30] considered the ECCA security for hybrid encryptions Tag-KEM+DEM and KEM+Tag-DEM. Abe et al. [26] provided a hybrid encryption scheme Tag-KEM+DEM, and they presented a useful way to get CCA secure hybrid encryptions. Cramer et al. [31] have shown that the hybrid encryption scheme Tag-KEM+DEM is CCA secure if its KEM part is CCA secure and its DEM part is one-time secure.

As, for the scheme Tag-SKEM+DEM, the ciphertext of scheme DEM is a tag of the scheme Tag-SKEM, one may think that the security assumption of scheme SKEM could be weakened to chosen plaintext attack (CPA) security when considering the RCCA security of signcryption. As it is impossible to make a simulation for the decryption oracle query for an adversary when the adversary attacks the hybrid signcryption, we leave it as an open problem that the security of scheme SKEM and DEM could be relaxed to a weaker security (e.g., CPA). One may also think that, with the RCCA security of Tag-KEM and one-time security of DEM, one can get the RCCA security of hybrid signcryption scheme Tag-KEM+DEM. However, the adversary cannot generate useful challenge ciphers if the adversary does not change the tag used for the scheme Tag-DEM. In this paper, when proving our results, we make a perfect simulation for the adversary, who initiates a IND-RCCA experiment to hybrid

Table 1: The hybrid cryptology and their security notion*.

Hybrid cryptology	Security notion	Reference
KEM+DEM	RCCA+RCCA⟹RCCA	[27]
KEM+DEM	CCA+CCA⟹CCA	[31]
SKEM+DEM	RCCA+RCCA⟹RCCA	Section 3.2
SKEM+DEM	CCA+CCA⟹CCA	[15]
Tag-KEM+DEM	CCA+one time security ⟹CCA	[32]
Tag-KEM+DEM	RCCA+RCCA⟹RCCA	[27]
Tag-SKEM+DEM	CCA+CCA⟹CCA	[15]
Tag-SKEM+DEM	RCCA+RCCA⟹RCCA	Section 3.4

*KEM: key encapsulation mechanism, DEM: data encapsulation mechanism.

signcryption. We summarise the hybrid cryptology and their security in Table 1.

Organizations of the Paper. In Section 2, we review some basic notations and definitions. In Section 3 we review the definition of general hybrid signcryption scheme, $\mathtt{SKEM}+\mathtt{DEM}$ and $\mathtt{Tag}\text{–}\mathtt{SKEM}+\mathtt{DEM}$, and then we prove its $\mathtt{RCCA}$ security. In Section 4, we review our main conclusions.

2. Preliminaries

In this section, we will review some useful notations and cryptographic primitives that will be used throughout this paper.

Notations. We denote by 1^λ the security parameter and write $m \xleftarrow{R} M$ to denote the algorithm that picks an m randomly from the set M. PPT denotes probabilistic polynomial time. we write $z \longleftarrow A(x, y, \cdots)$ to denote the algorithm that runs algorithm $\mathcal{A}$ with inputs $(x, y, \cdots)$ and then outputs z. We define a function $\mathrm{negl}(\lambda)$ as *negligible*: if for any constant $c > 0$, there exits a $k_0 \in \mathbb{Z}$, such that for all $\lambda > k_0$, $\mathrm{negl}(\lambda) < \lambda^{-c}$.

2.1. RCCA Security Definition. $\mathtt{PKE} = (\mathtt{Gen}, \mathtt{Enc}, \mathtt{Dec})$ is a public key encryption (PKE) scheme that consists of three polynomial-time algorithms:

(i) $\mathtt{Gen}$ is key generation algorithm that inputs the security parameter λ and outputs a pair of public/private keys (pk, sk).

(ii) $\mathtt{Enc}$ is PPT encryption algorithm that encrypts a message m into a ciphertext c.

(iii) $\mathtt{Dec}$ is a deterministic decryption algorithm that decrypts a ciphertext c and outputs either message m or a reject symbol $\perp$.

Now, we define its RCCA security by describing the attack experiment between a challenger and an PPT adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ with the following experiment:

(i) **Setup:** The adversary $\mathcal{A}$ queries $\mathtt{Gen}$ algorithm: $(pk, sk) \longleftarrow \mathtt{Gen}(\lambda)$.

(ii) **Stage 1:** The adversary $\mathcal{A}_1$ queries a ciphertext c to $\mathtt{Dec}$: $m \longleftarrow \mathtt{Dec}(sk, c)$, and adversary $\mathcal{A}_1$ responds with m.

(iii) **Challenge stage:** The adversary $\mathcal{A}_1$ queries a pair message (m_0, m_1) to $\mathtt{Enc}$, where $|m_0| = |m_1|$, and then the challenger chooses a bit $b \xleftarrow{R} \{0, 1\}$, computes the challenge cipher $\mathtt{Enc}(pk, m_b) = c^*$, and, finally, sends the challenge c^* to $\mathcal{A}_1$.

(iv) **Stage 2:** The adversary $\mathcal{A}_2$ makes continuous queries c to $\mathtt{Dec}$; here, we require that the cipher c is not identical to the challenge cipher c^*. The decryption algorithm runs $m \longleftarrow \mathtt{Dec}(sk, c)$. Finally, if $m \in \{m_0, m_1\}$, adversary $\mathcal{A}_2$ responds with *text*, or else adversary $\mathcal{A}_2$ responds with m or reject symbol $\perp$.

(v) **Guess stage:** The adversary $\mathcal{A}$ outputs $b' \in \{0, 1\}$.

We let $\mathrm{Adv}^{\mathtt{IND-RCCA}}_{\mathtt{PKE},\mathcal{A}}(\lambda) = |\Pr[b = b'] - 1/2|$ in the above experiment.

If for any PPT adversary $\mathcal{A}$, the function $\mathrm{Adv}^{\mathtt{IND-RCCA}}_{\mathtt{PKE},\mathcal{A}}(\lambda)$ is negligible, we believe that $\mathtt{PKE} = (\mathtt{Gen}, \mathtt{Enc}, \mathtt{Dec})$ is RCCA-secure.

2.2. Signcryption Key Encryption Mechanism (SKEM) and Its RCCA Security Notions. A signcryption key encryption mechanism $\mathtt{SKEM} = (\mathtt{KEM.Gen}_S, \mathtt{KEM.Gen}_R, \mathtt{KEM.Enc}, \mathtt{KEM.Dec})$ is a asymmetric encryption scheme [3], which consists of the four algorithms with the following:

(i) $\mathtt{SKEM.Gen}_S$ is a PPT algorithm that inputs a security parameter 1^λ and outputs the sender's public/private key (sk_S, pk_s).

(ii) $\mathtt{SKEM.Gen}_R$ is a PPT algorithm that inputs a security parameter 1^λ and outputs the receiver's public/private key (sk_R, pk_R).

(iii) $\mathtt{SKEM.Enc}$ is a PPT encryption algorithm that inputs the send's private key sk_S and the receiver's public key pk_R and outputs $(\mathtt{K}, \mathtt{C})$; here, $\mathtt{K}$ is a symmetric key and $\mathtt{C}$ is the key encapsulation of $\mathtt{K}$.

(iv) $\mathtt{SKEM.Dec}$ is a deterministic, polynomial-time decryption algorithm that inputs the sender's public key pk_S, a key encapsulation $\mathtt{c}$, and the receiver's private key sk_R and outputs either a key $\mathtt{K}$ or the error symbol $\perp$.

We now define its RCCA security by describing the attack experiment; this experiment is played by an adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ and the challenger:

(i) **Setup:** The challenger queries a key generation oracle $\mathtt{SKEM.Gen}_S(\lambda)$ and $\mathtt{SKEM.Gen}_R(\lambda)$. The key generation oracle $\mathtt{SKEM.Gen}_S$ runs $(pk_S, sk_S) \longleftarrow \mathtt{SKEM.Gen}_S(\lambda)$ and the key generation oracle $\mathtt{SKEM.Gen}_S$ runs $(pk_R, sk_R) \longleftarrow \mathtt{SKEM.Gen}_R(\lambda)$. Finally, the key generation oracle $\mathtt{SKEM.Gen}_S$ and $\mathtt{SKEM.Gen}_R$ sends (pk_R, pk_S) to adversary $\mathscr{A}$.

(ii) **Stage 1:** The adversary $\mathscr{A}_1$ inputs (pk_R, pk_S) and makes queries to encapsulation oracle and decapsulation oracle. For every decapsulation oracle algorithm query, the adversary $\mathscr{A}_1$ submits a ciphertext ψ to decryption algorithm Dec: $\mathtt{K} \longleftarrow \mathtt{Dec}(pk_S, sk_R, \psi)$. Finally, responds the adversary $\mathscr{A}$ with K or $\perp$.

(iii) **Challenge stage:** The challenger computes $\psi^* \longleftarrow \mathtt{SKEM.Enc}(sk_S, pk_R, \mathtt{K}_1)$, chooses $\mathtt{K}_0 \overset{R}{\longleftarrow} \mathtt{K}_K$, $\sigma \overset{R}{\longleftarrow} \{0,1\}$, where $\mathtt{K}_K$ is the key space, $|\mathtt{K}_0| = |\mathtt{K}_1|$, and sends $(\mathtt{K}_\sigma, \psi^*)$ to adversary $\mathscr{A}_1$.

(iv) **Stage 2:** The adversary $\mathscr{A}_2$ inputs (pk_R, pk_S) and makes continuous queries ψ to SKEM.Dec. Here, we require that adversary $\mathscr{A}_2$ is not allowed to query (pk_S, c^*) to SKEM.Dec. However, we admit that adversary A_2 can query SKEM.Dec on $(pk_{S'}, C)$ for any $pk_{S'} \neq pk_{S_i}$ and on (pk_{S_i}, C) for any $C \neq C^*$. The decryption oracle algorithm responds with $\mathtt{K} \longleftarrow \mathtt{Dec}(sk_R, pk_S, \psi)$. Finally, if $\mathtt{K} \in \{\mathtt{K}_0, \mathtt{K}_1\}$, $\mathscr{A}_2$ is responded with *text*, or else $\mathscr{A}_2$ is responded with K.

(v) **Guess stage:** In the end, $\mathscr{A}$ outputs a bit $\sigma' \in \{0,1\}$.

In the attack experiment, we let $\mathrm{Adv}^{\mathtt{IND-RCCA}}_{\mathtt{SKEM},\mathscr{A}}(\lambda) = |\Pr[\sigma = \sigma'] - 1/2|$. If for any PPT adversary $\mathscr{A}$, the function $\mathrm{Adv}^{\mathtt{IND-RCCA}}_{\mathtt{SKEM},\mathscr{A}}(\lambda)$ is negligible, we say the signcryption scheme $\mathtt{SKEM} = (\mathtt{SKEM.Gen}_R, \mathtt{SKEM.Gen}_R, \mathtt{SKEM.Enc}, \mathtt{SKEM.Dec})$ is RCCA-secure.

2.3. Data Encryption Mechanism and Its IND-RCCA Security. A signcryption data encryption mechanism DEM is a symmetric encryption scheme, which consists of the following two algorithms: DEM.ENC, DEM.Dec.

(i) $\mathtt{DEM.Enc} : c \longleftarrow \mathtt{DEM.Enc}(\mathtt{K}, m)$; DEM.Enc is a polynomial-time encryption algorithm; DEM.Enc encrypts m by using a key K and outputs the corresponding ciphertext c.

(ii) $\mathtt{DEM.Dnc} : m \longleftarrow \mathtt{DEM.Dec}(\mathtt{K}, c)$; DEM.Dec is a polynomial-time decryption algorithm; it inputs ciphertext χ and decrypts the cipher c by using the same key K.

We define its IND–RCCA security by describing the attack experiment; this experiment is played by an adversary $\mathscr{A} = (\mathscr{A}_1, \mathscr{A}_2)$ and the challenger:

(i) **Setup 1:** The challenger chooses a key symmetric $K \overset{R}{\longleftarrow} K_D$.

(ii) **Challenge stage:** The adversary $\mathscr{A}$ queries (m_0, m_1) to DEM.Enc, $|m_0| = |m_1|$. The challenger chooses $b \overset{R}{\longleftarrow} \{0,1\}$, computes the challenge cipher $c^* \longleftarrow \mathtt{Enc}(\mathtt{K}, m_b)$, and then sends the challenge cipher c^* to adversary $\mathscr{A}$.

(iii) **Setup 2:** The adversary $\mathscr{A}_2$ continues to make queries cipher c to Dec: $m \longleftarrow \mathtt{Dec}(c, \mathtt{K})$; here, c is not equal to the challenge cipher c^*. If $m \in \{m_0, m_1\}$, Dec responds to adversary $\mathscr{A}$ with *text*, or else Dec responds to adversary $\mathscr{A}$ with m.

(iv) **Guess stage:** In the end, the adversary $\mathscr{A}$ outputs $b' \in \{0,1\}$.

We define $\mathrm{Adv}^{\mathtt{IND-RCCA}}_{\mathtt{DEM},\mathscr{A}}(\lambda) = |\Pr[b = b'] - 1/2|$ in the above experiment.

If for any PPT adversary $\mathscr{A}$, the function $\mathrm{Adv}^{\mathtt{IND-RCCA}}_{\mathtt{DEM},\mathscr{A}_2}(\lambda)$ is negligible, the scheme DEM is IND–RCCA secure.

3. The RCCA Security of Hybrid Signcryption Schemes

In this section, we will recall the definition of hybrid signcryption which is adapted by Dent and An [15, 33]. Some definitions include the verification algorithm, whose aim is to provide nonrepudiation. However, in their view, nonrepudiation is unnecessary for most cryptography applications and hence will not be discussed further. Next, we examine the RCCA security for hybrid signcryption and consider the outsider security (the adversary is third party, neither sender nor receiver) of hybrid signcryption, which is proposed by Dent in [3].

3.1. SKEM+DEM Hybrid Signcryption Scheme and Its Relaxing Chosen Cipher Attack Security. $\mathtt{SKEM} = (\mathtt{SKEM.Gen}_S, \mathtt{SKEM.Gen}_R, \mathtt{SKEM.Enc}, \mathtt{SKEM.Dec})$ is signcryption key encapsulation mechanism, $\mathtt{DEM} = (\mathtt{DEM.Enc}, \mathtt{DEM.Dec})$ is data encapsulation mechanism, and hybrid signcryption scheme $\mathtt{SKEM+DEM} = (\mathtt{signcrypt.Gen}, \mathtt{signcrypt.Enc}, \mathtt{signcrypt.Dec})$ can be constructed from SKEM and DEM as follows:

(i) $\mathtt{signcrypt.Gen}(1^\lambda)$: It runs receiver's key generation algorithm $(pk_R, sk_R) \longleftarrow \mathtt{SKEM.Gen}_R(1^\lambda)$ and runs sender's key generation algorithm $(pk_S, sk_S) \longleftarrow \mathtt{SKEM.Gen}_S(1^\lambda)$, Finally, it outputs (pk_S, sk_S) and (pk_R, sk_R).

(ii) signcrypt.Enc(pk, m): signcrypt.Enc is a PPT algorithm that inputs the sender's private key sk_S, a message m, and the receiver's public key pk_R. It chooses $\mathtt{K} \longleftarrow \mathtt{K}_D$ and computes $\chi \longleftarrow \mathtt{SKEM.Enc}(sk_S, pk_R, \mathtt{K})$; here $\mathtt{K}_D$ is the signcryption scheme DEM's key space. Then it computes $\psi \longleftarrow \mathtt{DEM.Enc}_K(m)$, and the resulting signcryption is $c \coloneqq (\chi, \psi)$.

(iii) signcrypt.Dec(sk, c): the signcrypt.Dec algorithm inputs the sender's public key pk_S, a cipher c, and the receiver's private key sk_R. It then parses cipher c as $\psi \parallel \chi$ and runs $\mathtt{K} \longleftarrow \mathtt{TKEM.Dec}(\psi, pk_S, sk_R), m \longleftarrow \mathtt{DEM.Dec}_K(\chi)$. In the end, it outputs m or "reject" symbol $\perp$.

3.2. The RCCA Security of Hybrid Signcryption Schemes

Theorem 1. *The hybrid signcryption scheme* (SKEM + DEM) *can be constructed from a signcryption scheme* SKEM *and a scheme* DEM. *If the signcryption scheme* SKEM *is* IND–RCCA *secure and the signcryption scheme* DEM *is* IND–RCCA *secure, then hybrid signcryption scheme* (SKEM + DEM) *can achieve* IND–RCCA *security. For every given* PPT *adversary* $\mathscr{A}$, *there exist probabilistic adversary* $\mathscr{A}_1$ *and adversary* $\mathscr{A}_2$, *such that the following conclusion holds:*

$$\mathrm{Adv}^{\mathrm{RCCA}}_{\mathrm{SKEM+DEM},\mathscr{A}}(\lambda) \le 2\mathrm{Adv}^{\mathrm{RCCA}}_{\mathrm{SKEM},\mathscr{A}_1}(\lambda) + \mathrm{Adv}^{\mathrm{RCCA}}_{\mathrm{DEM},\mathscr{A}_2}(\lambda) + \frac{q_s}{|K_D|}. \tag{1}$$

Here, we assume the adversary $\mathscr{A}_1$ at most makes the q_s queries to the encryption-decryption oracle, the running times of $\mathscr{A}_1$ and $\mathscr{A}_2$ are equal to that of adversary $\mathscr{A}$, and K_D is the signcryption scheme DEM's key space.

Proof. Fix adversary $\mathscr{A}$ and λ; $\mathscr{A}$ is a PPT IND–RCCA adversary, which attacks the hybrid signcryption scheme SKEM + DEM; then we proved the theorem by the following experiments.

Experiment$_0$: This is an IND–RCCA experiment on the signcryption scheme SKEM + DEM, which is played by an adversary $\mathscr{A} = (\mathscr{A}_1, \mathscr{A}_2)$ and the challenger. (We denote by T_0 the event of adversary $\mathscr{A}$ succeeding in this experiment.)

(i) **Setup:** The adversary $\mathscr{A}$ makes queries to key generation algorithm $(pk_R, sk_R) \longleftarrow \mathtt{SKEM.Gen}_R(1^\lambda)$ and makes queries to key generation algorithm $(pk_S, sk_S) \longleftarrow \mathtt{SKEM.Gen}_S(1^\lambda)$. Finally, it sends (pk_R, pk_S) to adversary $\mathscr{A}$.

(ii) **Stage 1:** The adversary $\mathscr{A}$ inputs a public key pair (pk_R, pk_S) and makes continuous queries to decryption oracle algorithm. For adversary $\mathscr{A}$'s decryption algorithm query $c = (\psi, \chi)$, the adversary $\mathscr{A}_1$ sends a cipher $c = (\psi, \chi)$ to the challenger $\mathscr{C}$, and the challenger $\mathscr{C}$ runs decryption algorithm $\mathtt{K} \longleftarrow \mathtt{SKEM.Dec}(\psi, pk_s, sk_R)m \longleftarrow \mathtt{DEM.Dec}_K(\chi)$. In the end, the challenger responds to $\mathscr{A}_1$ with m.

(iii) **Challenge stage:** The adversary $\mathscr{A}_1$ inputs (pk_R, pk_S) and queries (m_0, m_1) to an encryption oracle, and the challenger chooses $\mathtt{K}_1 \longleftarrow \mathtt{K}_D$ and chooses $b \in \{0, 1\}$. Then the challenger computes $\chi \longleftarrow \mathtt{SKEM.Enc}(sk_S, pk_R, \mathtt{K}_1)$ and computes $\psi \longleftarrow \mathtt{DEM.Enc}_\mathtt{K}(m_b)$ the signcryption $c := (\chi, \psi)$.

(iv) **Stage 2:** The adversary $\mathscr{A}_2$ inputs (pk_{R_i}, pk_{S_i}) and makes continuous queries $c = (\psi, \chi)$ to the challenger. Here, the adversary $\mathscr{A}_2$ is not admitted to query (pk_S, c^*) to the decryption oracle algorithm. But we admit that adversary A_2 can make a query to the decryption oracle algorithm on $(pk_{S'}, C)$ for any $pk_{S'} \neq pk_{S_i}$ and on (pk_{S_i}, C) for any cipher $C \neq C^*$. The challenger runs decryption oracle $\mathtt{K} \longleftarrow \mathtt{SKEM.Dec}(pk_{S_i}, sk_{R_i}, \psi, \chi)$ and $m \longleftarrow \mathtt{DEM.Dec}_{K_i}(\chi)$. If $m \in \{m_0, m_1\}$, the challenger responds to $\mathscr{A}_2$ with *text* or else responds to $\mathscr{A}_2$ with m.

(v) **Guess stage:** In the end, the adversary $\mathscr{A}$ outputs a guessing bit $b' \in \{0, 1\}$.

The following conclusion holds:

$$\mathrm{Adv}^{\mathrm{RCCA}}_{\mathrm{SKEM+DEM},\mathscr{A}}(\lambda) = \left|\Pr\left[b = b'\right] - \frac{1}{2}\right| = \left|\Pr\left[T_0\right] - \frac{1}{2}\right|. \tag{2}$$

Experiment$_1$: We now modify experiment G_0 to obtain a new experiment G_1. These two experiments are identical except that we use a uniformly random key $\mathtt{K}_0 \overset{R}{\longleftarrow} \mathtt{K}_D$ to compute the challenge cipher $C^* = (\chi^*, \psi^*)$ in step 3 of $\mathtt{Game}_0$; the challenge cipher $C^* = (\chi^*, \psi^*)$ is computed by the encryption algorithm $\mathtt{SKEM.Enc}(sk_S, pk_R, \mathtt{K}_0)$ and $\psi \longleftarrow \mathtt{DEM.Enc}_K(m)$. To maintain consistency, the challenger should use the symmetric key $\mathtt{K}_0$ to answer the decryption oracle algorithm query $(pk_S, \chi, .)$. Hence, the distinction between experiment G_0 and experiment G_1 mainly lies in how the scheme SKEM runs. (Denote by T_1 the sign of the adversary $\mathscr{A}$ succeeding in this experiment.) We have the following conclusion.

Lemma 2. *There is an adversary* $\mathscr{A}_1$, *and its running time is equal to the running time of adversary* $\mathscr{A}$; *the following conclusion holds:*

$$\left|\Pr\left[T_1\right] - \Pr\left[T_0\right]\right| \le 2\mathrm{Adv}^{\mathrm{RCCA}}_{\mathrm{SKEM},\mathscr{A}_1}(\lambda) + \frac{q_s}{|\mathtt{K}_D|}. \tag{3}$$

Proof. We prove the lemma by constructing an adversary $\mathscr{A}_1$ who attacks the signcryption scheme SKEM. The adversary $\mathscr{A}_1$ simulates the environment for $\mathscr{A}$, their interactions can be described as follows:

(i) **Setup:** The adversary $\mathscr{A}_1$ was given $(pk_S, pk_R, \mathtt{K}_\sigma)$, and the adversary $\mathscr{A}_1$ sent (pk_S, pk_R) to $\mathscr{A}$.

(ii) **Stage 1:** The adversary $\mathscr{A}$ inputs (pk_S, pk_R) and makes some queries c to the decryption oracle: $m \longleftarrow \mathtt{Dec}(sk_R, pk_S, c)$. Finally, $\mathscr{A}$ is reponded with m or reject symbol $\perp$.

(iii) **Challenge stage:** The adversary $\mathscr{A}$ inputs (pk_S, pk_R) and queries (m_0, m_1) to an encryption oracle, $|m_0| = |m_1|$. The adversary $\mathscr{A}_1$ computes $\chi^* \longleftarrow \mathtt{SKEM.Enc}(sk_S, pk_R, \mathtt{K}_1)$ and computes $\psi^* \longleftarrow \mathtt{DEM.Enc}_\mathtt{K}(m_b)$, and finally the adversary $\mathscr{A}_1$ sends the challenge cipher $c^* = (\psi^*, \chi^*)$ to the adversary $\mathscr{A}$.

(iv) **Stage 2:** The adversary $\mathscr{A}$ inputs (pk_{S_i}, pk_{R_i}) and makes queries $c = (\psi_i, \chi_i)$ to decryption oracle algorithm. Here, we require that $\mathscr{A}_2$ cannot query (pk_S, c^*) to the decryption oracle algorithm. However, we admit that adversary A_2 can query the decryption oracle algorithm on $(pk_{S'}, C)$ for any $pk_{S'} \neq pk_{S_i}$ and on (pk_{S_i}, C) for $C \neq C^*$. The adversary $\mathscr{A}_1$ runs

$$\begin{aligned} &\mathtt{K}_i \longleftarrow \mathtt{SKEM.Dec}\left(pk_{S_i}, sk_{R_i}, \chi_i, \psi_i\right), \\ &m \longleftarrow \mathtt{DEM.Dec}_{K_i}\left(\psi_i\right). \end{aligned} \tag{4}$$

Finally, if $m \in \{m_0, m_1\}$, $\mathscr{A}_2$ responds with *text*, or else $\mathscr{A}_2$ is responded with m.

(v) **Guess stage:** The adversary $\mathscr{A}$ outputs a guessing bit $b' \in \{0, 1\}$ and $\mathscr{A}_1$ outputs $\sigma' = b'$ in the end.

This has completed the construction of $\mathscr{A}_1$. By description, we can see that the adversary $\mathscr{A}_1$ played a perfectly simulated decryption for adversary $\mathscr{A}$ unless the cipher ψ^* is decrypted to $\mathtt{K}_1$ and test is returned by the correct answer from the decryption oracle SKEM.Dec for every query. However, the probability of this event is $1/|\mathtt{K}_D|$ since in that case the key $\mathtt{K}_1$ is uniformly random and independent of the opinion of the adversary $\mathscr{A}_1$ for each such query.

(i) If $\sigma = 0$, we can obtain that cipher χ is computed by a random key $\mathtt{K}_0$; meanwhile, the opinion of the adversary $\mathscr{A}$ is equal to that in $\mathtt{Experiment}_0$.

(ii) If $\sigma = 1$, we can obtain that $\mathtt{K}_1$ is corresponding correct key embedded in the cipher ψ; meanwhile, the opinion of the adversary $\mathscr{A}$ is equal to that in $\mathtt{Experiment}_1$.

Thus,

$$\begin{aligned}
\mathrm{Adv}^{\mathrm{RCCA}}_{\mathtt{SKEM},\mathscr{A}_1}(\lambda) &= \left|\Pr\left[b = b'\right] - \frac{1}{2}\right| = \left|\Pr\left[T_0\right] - \frac{1}{2}\right| \\
&= \frac{1}{2}\left|\Pr\left[\sigma' = 1 \mid \sigma = 1\right] - \Pr\left[\sigma' = 1 \mid \sigma = 0\right]\right| \\
&= \frac{1}{2}\left|\Pr\left[b = b' \mid \sigma = 1\right] - \Pr\left[b = b' \mid \sigma = 0\right]\right| \\
&\geq \frac{1}{2}\left|\Pr\left[T_0\right] - \Pr\left[T_1\right] - \frac{q_s}{|\mathtt{K}_D|}\right|.
\end{aligned} \tag{5}$$

We can get the following conclusion:

$$\left|\Pr\left[T_1\right] - \Pr\left[T_0\right]\right| \leq 2\mathrm{Adv}^{\mathrm{RCCA}}_{\mathtt{SKEM},\mathscr{A}_1}(\lambda) + \frac{q_s}{|\mathtt{K}_D|}. \tag{6}$$

Lemma 2 is proved. □

In the stage of experiment G_1's encryption and decryption oracle algorithms, we use a uniformly random key $\mathtt{K}_0$, so the challenger cipher ψ^* is not be decrypted. From this point, we notice that the challenge cipher χ^* is generated by using a random symmetric key $\mathtt{K}_0$ in experiment G_1. Meanwhile, the other cipher $\chi = \chi^*$ is decrypted by using random key $\mathtt{K}_0$, which has no other role in experiment G_1. Hence, in experiment G_1, the adversary $\mathscr{A}$ plays an adaptive replayable chosen ciphertext attack against (RCCA) the signcryption scheme DEM in substance, so the following conclusion holds.

Lemma 3. *There is a probabilistic adversary $\mathscr{A}_2$, and its running time is equal to the running time of the adversary $\mathscr{A}$, such that the following conclusion holds:*

$$\left|\Pr\left[T_1\right] - \frac{1}{2}\right| \leq \mathrm{Adv}^{\mathrm{RCCA}}_{\mathtt{DEM},\mathscr{A}_2}(\lambda). \tag{7}$$

Proof. The symmetric key $\mathtt{K}_0$ was chosen uniformly, randomly, and independently, so the challenge cipher χ does not reveal related information about which message was encrypted. Hence, to gain success in experiment 2, the adversary must learn some information from the challenger cipher χ. We prove Lemma 3 by constructing a probabilistic adversary $\mathscr{A}_2$, who attacks the signcryption scheme DEM, and $\mathscr{A}_2$ provides an environment for the adversary $\mathscr{A}$. Now, we describe their interactions:

(i) **Setup:** The adversary $\mathscr{A}_2$ runs receiver key generation algorithm $(pk_R, sk_R) \longleftarrow \mathtt{SKEM.Gen}_R(1^\lambda)$, runs sender key generation algorithm $(pk_S, sk_S) \longleftarrow \mathtt{SKEM.Gen}_S(1^\lambda)$, and sends (pk_R, pk_S) to $\mathscr{A}$.

(ii) **Stage 1:** The adversary $\mathscr{A}$ inputs (pk_R, pk_S) and makes queries ciphertext c to a decryption oracle algorithm: $\mathtt{K} \longleftarrow \mathtt{SKEM.Dec}(pk_S, sk_R, \chi)$, $m \longleftarrow \mathtt{DEM.Dec}(\mathtt{K}, \chi)$. If $m = \perp$, the decryption oracle algorithm responds to adversary $\mathscr{A}$ with m or reject symbol $\perp$.

(iii) **Challenge Stage:** The adversary $\mathscr{A}$ inputs a public key pair (pk_R, pk_S) and sends (m_0, m_1) to the adversary $\mathscr{A}_2$. the adversary $\mathscr{A}_2$ chooses $\mathtt{K}_1 \overset{R}{\longleftarrow} \mathtt{K}_D$, runs $\psi^* \longleftarrow \mathtt{SKEM.Enc}(pk_S, sk_R, \mathtt{K}_1)$, and sends the challenge $c^* = (\psi^*, \chi^*)$ to $\mathscr{A}$. We notice that the symmetric key $\mathtt{K}_1$ was chosen as the encryption key of scheme DEM and embedded in cipher ψ^*, which is uniformly random and independent of each other.

(iv) **Stage 2:** The adversary inputs (pk_R, pk_S) and makes continuous queries $c = (\psi, \chi)$ to decryption oracle algorithm. Here, we require that adversary $\mathscr{A}_2$ cannot query (pk_S, c^*) to the decryption oracle algorithm. However, we admit adversary A_2 can make a query to the decryption oracle algorithm on $(pk_{S'}, C)$ for any cipher $pk_{S'} \neq pk_{S_i}$ and on (pk_{S_i}, C) for any $C \neq C^*$. The adversary $\mathscr{A}_2$ uses the secret key sk_R to run the decryption oracle algorithm and answer the decryption query $c = (\psi, \chi)$ of adversary $\mathscr{A}$ with the following:

(a) If $\psi_i = \psi^*$, hence $\chi_i \neq \chi^*$. Then The adversary $\mathscr{A}_2$ runs the decryption oracle $\mathtt{K} \longleftarrow \mathtt{SKEM.Dec}(pk_S, sk_R, \chi)$. If $\mathtt{K} = \perp$, the adversary $\mathscr{A}_2$ responds to $\mathscr{A}$ with $\perp$ or else $m \longleftarrow \mathtt{DEM.Dec}(\mathtt{K}, \chi)$. If $m \in \{m_0, m_1\}$, the adversary $\mathscr{A}_2$ responds to $\mathscr{A}$ with *text*, or else $\mathscr{A}_2$ responds to $\mathscr{A}$ with m.

(v) **Guess Stage:** Finally, adversary $\mathscr{A}$ outputs a bit $b' \in \{0, 1\}$ and $\mathscr{A}_2$ also outputs a bit b'.

This has completed the description of the adversary $\mathscr{A}_2$. By our construction, it is obvious that the adversary $\mathscr{A}_2$ plays a perfectly simulated decryption for $\mathscr{A}$, and whenever $\mathscr{A}$ gets success, so does $\mathscr{A}_2$. We have the following conclusion:

$$\left|\Pr\left[T_1\right] - \frac{1}{2}\right| \leq \mathrm{Adv}^{\mathrm{RCCA}}_{\mathtt{DEM},\mathscr{A}_2}(\lambda). \tag{8}$$

□

We can know that the advantage of $\mathscr{A}$ in experiment$_0$ is

$$\begin{aligned}\mathrm{Adv}^{\mathrm{RCCA}}_{\mathrm{SKEM+DEM},\mathscr{A}}(\lambda) &= \left|\Pr[T_0] - \frac{1}{2}\right| \\ &\leq 2\mathrm{Adv}^{\mathrm{RCCA}}_{\mathrm{SKEM},\mathscr{A}_1}(\lambda) + \mathrm{Adv}^{\mathrm{RCCA}}_{\mathrm{DEM},\mathscr{A}_2}(\lambda),\end{aligned} \tag{9}$$

which is negligible; we have proved Theorem 1.

3.3. The Hybrid Signcryption Scheme Tag-SKEM+DEM and Its RCCA Security

Definition 4 (signcryption scheme Tag-SKEM). A signcryption scheme $\mathtt{Tag\text{–}KEM} = (\mathtt{Gen}_R, \mathtt{Gen}_S, \mathtt{Tag\text{–}KEM.Enc}, \mathtt{Tag\text{–}KEM.Dec})$ consists of the following three algorithms:

(i) $\mathtt{Tag\text{–}KEM.Gen}_S(1^\lambda)$:$\mathtt{Tag\text{–}KEM.Gen}_S(1^\lambda)$ is a PPT algorithm that inputs a security parameter 1^λ and outputs a pair of public/private keys (sk_S, pk_S).

(ii) $\mathtt{Tag\text{–}KEM.Gen}_R(1^\lambda)$:$\mathtt{TKEM.Gen}_S(1^\lambda)$ is a PPT algorithm that inputs a security parameter 1^λ and outputs a pair of public/private keys (sk_R, pk_R).

(iii) An encryption algorithm $\mathtt{Tag\text{–}KEM.Enc}$: It runs $(\omega, \mathtt{K}) \longleftarrow \mathtt{Tag\text{–}KEM.Key}(pk_R, sk_S)$. $\mathtt{Tag\text{–}KEM.Key}(\cdot)$ is a PPT algorithm that inputs the private key of sender sk_S and public key of receiver pk_R and outputs one-time key $\mathtt{K}$ and Intermediate state information ω. Choose $r \overset{R}{\leftarrow} \{0,1\}^\lambda$ and compute $\psi \longleftarrow \mathtt{TKEM.Enc}(\omega, r, pk_R, sk_S, \tau)$. $\mathtt{Tag\text{–}KEM.Enc}$ is a PPT algorithm that encrypts the key $\mathtt{K}$ (embedded in ω) into cipher ψ along with a tag $\tau \in T$ and returns a cipher ψ; here, τ is called a tag.

(iv) An decryption algorithm $\mathtt{Tag\text{–}KEM.Dec} : \mathtt{K} \longleftarrow \mathtt{Tag\text{–}KEM.Dec}(pk_S, sk_R, \psi, \tau)$. $\mathtt{TKEM.Dec}$ is a deterministic decryption verification algorithm for a signcryption cipher, which inputs the receiver's private key sk_R, the cipher c, the sender's public key pk_S, and a tag τ; the decryption oracle $\mathtt{Tag\text{–}KEM.Dec}$ returns a key $\mathtt{K}$ or reject symbol $\perp$.

Definition 5 (hybrid signcryption scheme Tag-SKEM+DEM). The signcryption scheme

$$\mathtt{Tag\text{–}SKEM} = \left(\mathtt{Gen}_R, \mathtt{Gen}_S, \mathtt{TKEM.Enc}, \mathtt{TKEM.Dec}\right) \tag{10}$$

is a asymmetric encryption scheme and the signcryption scheme $\mathtt{DEM} = (\mathtt{DEM.Enc}, \mathtt{DEM.Dec})$ is a corresponding symmetric encryption scheme [18].

Then the hybrid signcryption scheme

$$\begin{aligned}&\mathtt{Tag\text{–}SKEM + DEM} \\ &\quad = (\mathtt{Gen}, \mathtt{signcrypt}, \mathtt{unsigncrypt})\end{aligned} \tag{11}$$

can be constructed as follows:

(i) Key generation algorithm $\mathtt{Gen}(1^\lambda)$: $\mathtt{Gen}_R(1^\lambda)$ is a probabilistic receiver's key generation algorithm that inputs a 1^λ and outputs the receiver's public/private key pair (pk_R, sk_R); we write this as $(pk_R, sk_R) \longleftarrow \mathtt{Gen}_R(1^\lambda)$. $\mathtt{Gen}_S(1^\lambda)$ is a probabilistic receiver key generation algorithm, which takes a as input a security parameter 1^λ and as output a receiver's public/private key pair (pk_S, sk_S); we write this as $(pk_S, sk_S) \longleftarrow \mathtt{Gen}_S(1^\lambda)$.

(ii) An encryption algorithm $\mathtt{signcrypt} : \mathtt{Tag\text{–}SKEM.Key}(\cdot)$ is a probabilistic algorithm that inputs the receiver's public key pk_S and outputs a symmetric key $\mathtt{K} \in \mathtt{K}_D$ and the internal state information ω, $(\omega, \mathtt{K}) \longleftarrow \mathtt{Tag\text{–}SKEM.Key}(pk_S)$. Here $\mathtt{K}_D$ is the scheme $\mathtt{DEM}$'s key space. Then choose $r \overset{R}{\leftarrow} \{0,1\}^\lambda$ and compute $\chi \longleftarrow \mathtt{DEM.Enc}_K(m)$, $\psi \longleftarrow \mathtt{Tag\text{–}SKEM.Enc}(pk_S, sk_R, \omega, r, \chi)$. Finally, output the signcrypt cipher $c = (\psi, \chi)$.

(iii) A decryption algorithm $\mathtt{unsigncrypt}$: First, it parses the cipher c to obtain $\psi \parallel \chi$. Next, it computes $\mathtt{K} \longleftarrow \mathtt{Tag\text{–}SKEM.Dec}(sk_R, pk_R, \psi, \chi)$ to obtain a symmetric key $\mathtt{K}$ and computes $m \longleftarrow \mathtt{DEM.Dec}_K(\chi)$. Finally, it outputs the message m or "reject" symbol $\perp$.

3.4. The RCCA Security of Hybrid Signcryption Scheme Tag-SKEM+DEM

Theorem 6. *The hybrid signcryption scheme* ($\mathtt{Tag\text{–}SKEM}$ + $\mathtt{DEM}$) *is constructed from a scheme* $\mathtt{Tag\text{–}SKEM}$ *and a scheme* $\mathtt{DEM}$. *If the signcryption scheme* $\mathtt{Tag\text{–}SKEM}$ *is* $\mathtt{IND\text{–}RCCA}$ *secure and the signcryption scheme* $\mathtt{DEM}$ *is* $\mathtt{IND\text{–}RCCA}$ *secure, then the hybrid signcryption scheme* ($\mathtt{Tag\text{–}SKEM}$ + $\mathtt{DEM}$) *is also* $\mathtt{IND\text{–}RCCA}$ *secure. For every* PPT *adversary* $\mathscr{A}$, *there are probabilistic adversary* $\mathscr{A}_1$ *and adversary* $\mathscr{A}_2$, *whose running times are essentially equal to that of adversary* $\mathscr{A}$, *such that for all* $\lambda \geq 0$, *the following holds.*

$$\begin{aligned}\mathrm{Adv}^{\mathrm{RCCA}}_{\mathtt{Tag\text{–}SKEM+DEM},\mathscr{A}}(\lambda) &\leq 2\mathrm{Adv}^{\mathrm{RCCA}}_{\mathtt{Tag\text{–}KEM},\mathscr{A}_1}(\lambda) \\ &\quad + \mathrm{Adv}^{\mathrm{RCCA}}_{\mathrm{DEM},\mathscr{A}_2}(\lambda) + \frac{q_s}{|\mathtt{K}_D|}.\end{aligned} \tag{12}$$

Here, we assume the adversary at most makes the q_s queries to the encryption-decryption oracle algorithm and $\mathtt{K}_D$ is the scheme $\mathtt{DEM}$'s key space.

Proof. We prove the theorem by constructing a PPT adversary $\mathscr{A}$ who attacks the hybrid signcryption scheme $\mathtt{Tag\text{–}SKEM}$ + $\mathtt{DEM}$ with the following experiments. (We denote by T_i the event of the adversary $\mathscr{A}$ succeeding in the i-th game.)

Experiment$_0$: This is the $\mathtt{IND\text{–}RCCA}$ experiment on the signcryption scheme $\mathtt{Tag\text{-}SKEM+DEM}$, and this experiment is played between an adversary $\mathscr{A}$ and the challenger as follows:

(i) **Setup:** The adversary queries a key generation oracle. The challenger runs receiver key generation algorithm $(pk_R, sk_R) \longleftarrow \mathtt{Tag\text{–}SKEM.Gen}_R(1^\lambda)$, runs sender key generation algorithm $(pk_S, sk_S) \longleftarrow \mathtt{Tag\text{–}SKEM.Gen}_S(1^\lambda)$, and responds to the adversary $\mathscr{A}$ with (pk_R, pk_S).

(ii) **Stage 1:** The adversary $\mathcal{A}$ inputs (pk_R, pk_S) and makes continuous queries to decryption oracle algorithm. The adversary $\mathcal{A}$ sends a cipher c to the decryption oracle algorithm, and the decryption oracle algorithm runs $\mathtt{K} \longleftarrow \mathtt{Tag\text{-}SKEM.Dec}(pk_S, sk_R, \chi, \psi)$, $m \longleftarrow \mathtt{DEM.Dec}(\mathtt{K}, \chi)$. If $m = \perp$, the decryption oracle algorithm responds to $\mathcal{A}_1$ with $\perp$ or else responds to $\mathcal{A}_1$ with m.

(iii) **Challenge stage:** The adversary $\mathcal{A}_1$ inputs a public key pair (pk_R, pk_S) and queries (m_0, m_1) to an encryption oracle algorithm, and then the challenger runs $(\omega, \mathtt{K}) \longleftarrow \mathtt{Tag\text{-}SKEM.Key}(pk_R sk_S)$, $\mathtt{K} \in \mathtt{K}_D$. Then the challenger computes $\mathtt{DEM.Enc}_K(m_0) = \chi^*$, $\mathtt{Tag\text{-}SKEM.Enc}(pk_R, sk_S, \omega, \chi^*) = \psi^*$ and sends the challenge cipher $c^* = (\psi^*, \chi^*)$ to the adversary $\mathcal{A}_1$.

(iv) **Stage 2:** The adversary $\mathcal{A}_2$ inputs a public key pair (pk_{R_i}, pk_{S_i}) and makes continuous queries $c = (\psi, \chi)$ to the challenger. Here, we require that adversary $\mathcal{A}_2$ is not admitted to query (pk_S, c^*) to the decryption oracle. However, we admit that adversary A_2 can make a query to the decryption oracle on $(pk_{S'}, C)$ for any public key $pk_{S'} \neq pk_{S_i}$ and on (pk_{S_i}, C) for any cipher $C \neq C^*$. The challenger runs decryption oracle.

$$\begin{aligned} &\mathtt{K} \longleftarrow \mathtt{Tag\text{-}SKEM.Dec}\left(sk_{R_i}, pk_{S_i}, \psi, \chi\right), \\ &m \longleftarrow \mathtt{DEM.Dec}_K\left(\chi\right) \end{aligned} \tag{13}$$

Finally, if $m \in \{m_0, m_1\}$, $\mathcal{A}_2$ responds with *text*, or else $\mathcal{A}_2$ responds with m.

(v) **Guess stage:** In the end, the adversary $\mathcal{A}$ outputs a guess bit $b' \in \{0, 1\}$.

Naturally, the following holds:

$$\begin{aligned} \mathrm{Adv}^{\mathtt{RCCA}}_{\mathtt{Tag-SKEM+DEM},\mathcal{A}}(\lambda) &= \left|\Pr\left[T_0\right] - \frac{1}{2}\right| \\ &= \left|\Pr\left[b = b'\right] - \frac{1}{2}\right|. \end{aligned} \tag{14}$$

Experiment$_1$: We now modify $\mathtt{Experiment}_0$ to obtain a new $\mathtt{Experiment}_1$; this experiment is equal to the above experiment except that we just use a random key $\mathtt{K}_0 \overset{R}{\longleftarrow} \mathtt{K}_D$ to encrypt the message m_0 in step 3 of $\mathtt{experiment}_0$; hence, we get the following conclusion.

Lemma 7. *There exists a probabilistic adversary $\mathcal{A}_1$, and its running time is equal to that of adversary $\mathcal{A}$, such that the following conclusion holds:*

$$\left|\Pr\left[T_1\right] - \Pr\left[T_0\right]\right| \leq \mathrm{Adv}^{\mathtt{RCCA}}_{\mathtt{Tag-SKEM},\mathcal{A}_1}(\lambda) + \frac{q_s}{\left|\mathtt{K}_D\right|}. \tag{15}$$

Here, we assume the adversary at most makes the q_s queries to the encryption-decryption oracle algorithm.

Proof. We prove the lemma by constructing an adversary $\mathcal{A}_1$ who attacks signcryption scheme Tag-SKEM. The adversary $\mathcal{A}_1$ simulates an environment for adversary $\mathcal{A}$; their interactions can be described as follows:

(i) **Stage 1:** The adversary $\mathcal{A}_2$ was given $(pk_R, pk_S, \mathtt{K}_\sigma)$, and at the same time, (pk_R, pk_S) was sent to adversary $\mathcal{A}$.

(ii) **Stage 2:** The adversary $\mathcal{A}$ inputs a public key pair (pk_R, pk_S) and makes continuous queries c to a decryption oracle algorithm Dec. The decryption oracle algorithm runs $m \longleftarrow \mathtt{Dec}(pk_S, sk_R, c)$. Finally, if $m = \perp$, $\mathcal{A}$ responds with m or reject symbol $\perp$.

(iii) **Stage 3:** The adversary $\mathcal{A}$ inputs a pair public key (pk_R, pk_S) and queries (m_0, m_1) to the encryption oracle, $|m_0| = |m_1|$. The adversary $\mathcal{A}_1$ requires the encryptions oracle of scheme Tag-SKEM to obtain $(\mathtt{K}_\sigma, \psi^*)$. The adversary $\mathcal{A}_1$ chooses $b \in \{0, 1\}$ and computes $\mathtt{DEM.EncS}(K_\sigma, m_b) = \chi^*$. Finally, the adversary $\mathcal{A}_1$ sends challenge cipher $c^* = (\psi^*, \chi^*)$ to the adversary $\mathcal{A}$.

(iv) **Stage 4:** The adversary $\mathcal{A}$ inputs (pk_{R_i}, pk_{S_i}) and makes continuous calls $c = (\psi_i, \chi_i)$ to decryption oracle query. Here, we require that adversary $\mathcal{A}_2$ is not admitted to query (pk_S, c^*) to the decryption oracle algorithm. However, we admit that adversary A_2 can make a query to the decryption oracle on $(pk_{S'}, C)$ for any $pk_{S'} \neq pk_{S_i}$ and on (pk_{S_i}, C) for any $C \neq C^*$. The adversary $\mathcal{A}_1$ runs its own decryption oracle $\mathtt{Tag\text{-}SKEM.Dec}(pk_S, sk_R, .)$ to answer the adversary $\mathcal{A}$'s decryption query as follows:

(a) If $\perp$ is returned, then the adversary $\mathcal{A}_1$ responds to $\mathcal{A}$ with $\perp$.

(b) If $\perp$ is returned and $\chi_i \neq \chi^*$, then $\mathcal{A}_1$ uses K_σ to decrypt the cipher χ.

(1) If m_0 or m_0 is returned, then the adversary $\mathcal{A}_1$ responds to $\mathcal{A}$ with *text*.
(2) Otherwise, $\mathcal{A}_1$ responds to $\mathcal{A}$ with the result.

(c) If *test* is returned and cipher $\chi_i = \chi^*$, then the adversary $\mathcal{A}_1$ responds to $\mathcal{A}$ with *text*.

(d) If $\mathtt{K}_1$ is returned, then the adversary uses $\mathtt{K}_1$ to decrypt the cipher χ.

(1) If m_0 or m_0 is returned, then the adversary $\mathcal{A}_1$ responds to adversary $\mathcal{A}$ with *text*.
(2) Otherwise, $\mathcal{A}_1$ responds to adversary $\mathcal{A}$ with the result.

(v) **Stage 5:** In the end, the adversary $\mathcal{A}$ outputs a guess bit $b' \in \{0, 1\}$, and $\mathcal{A}_1$ outputs a bit $\sigma' = b'$.

This has completed the description of $\mathcal{A}_1$; it is clear that the adversary $\mathcal{A}_1$ plays a perfectly simulated decryption for $\mathcal{A}$ unless the cipher ψ^* is decrypted to $\mathtt{K}_1$ and test is returned by the correct answer from the decryption oracle Tag-SKEM.Dec for every query. However, the probability of this event is $1/|\mathtt{K}_D|$ since in that case the key $\mathtt{K}_1$ is random and independent of the opinion of the adversary $\mathcal{A}_1$ for each such query.

(i) If $\sigma = 0$, we can know that random key $\mathtt{K}_0$ is used for computing the cipher χ and the view of $\mathscr{A}$ is identical to that in $\mathtt{Experiment}_0$. Accordingly, $\Pr[b' = b \mid \sigma = 0] = \Pr[T_2]$.

(ii) If $\sigma = 1$, we can know that the key $\mathtt{K}_1$ is the correct key embedded in the cipher ψ and the view of $\mathscr{A}$ is equal to that in $\mathtt{Experiment}_1$. Accordingly, $|\Pr[b' = b \mid \sigma = 1] - \Pr[T_1]| \leq q_D/|\mathtt{K}_D|$.

Thus,

$$\begin{aligned}
\mathrm{Adv}^{\mathtt{RCCA}}_{\mathtt{Tag-SKEM},\mathscr{A}_1}(\lambda) &= \left|\Pr[T_0] - \frac{1}{2}\right| \\
&= \left|\Pr[b = b'] - \frac{1}{2}\right| \\
&= \frac{1}{2}\left|\Pr[\sigma' = 1 \mid \sigma = 1] - \Pr[\sigma' = 1 \mid \sigma = 0]\right| \\
&= \frac{1}{2}\left|\Pr[b = b' \mid \sigma = 1] - \Pr[b = b' \mid \sigma = 0]\right| \\
&\geq \frac{1}{2}\left|\Pr[T_0] - \Pr[T_1] - \frac{q_d}{|\mathtt{K}_D|}\right|.
\end{aligned} \tag{16}$$

Hence,

$$\left|\Pr[T_1] - \Pr[T_0]\right| \leq 2\mathrm{Adv}^{\mathtt{RCCA}}_{\mathtt{Tag-SKEM},\mathscr{A}_1}(\lambda) + \frac{q_s}{|\mathtt{K}_D|}. \tag{17}$$

□

Lemma 2 is proved. Next, we show that the adversary $\mathscr{A}$ playing $\mathtt{Experiment}_1$ essentially conducts an IND-RCCA attack on the signcryption scheme DEM; we claim the following.

Lemma 8. *There is a probabilistic adversary $\mathscr{A}_2$, and its running time is equal to that of $\mathscr{A}$, and the following conclusion holds:*

$$\left|\Pr[T_2] - \frac{1}{2}\right| \leq \mathrm{Adv}^{\mathtt{RCCA}}_{\mathtt{DEM},\mathscr{A}_2}(\lambda). \tag{18}$$

Proof. This can be shown by constructing an adversary $\mathscr{A}_2$ who attacks the signcryption scheme DEM. The adversary $\mathscr{A}_2$ simulates the environment for adversary $\mathscr{A}$; their interactions can be described as follows:

(i) **Stage 1:** The adversary $\mathscr{A}_2$ queries receiver's key generation algorithm $(pk_R, sk_R) \longleftarrow \mathtt{Tag\text{-}SKEM.Gen}_R(1^\lambda)$, queries sender's key generation algorithm $(pk_S, sk_S) \longleftarrow \mathtt{Tag\text{-}SKEM.Gen}_S(1^\lambda)$, and sends a public key pair (pk_R, pk_S) to $\mathscr{A}$.

(ii) **Stage 2:** The adversary $\mathscr{A}$ inputs a public key pair (pk_R, pk_S) and makes continuous queries c to a decryption oracle algorithm: $m \longleftarrow \mathtt{Dec}(sk_R, pk_S, c)$. In the end, adversary $\mathscr{A}$ is responded with m or reject symbol $\perp$.

(iii) **Stage 3:** The adversary $\mathscr{A}$ inputs a pair public key (pk_R, pk_S) and sends (m_0, m_1) to the adversary $\mathscr{A}_2$; $\mathscr{A}_2$ queries (m_0, m_1) to the encryption oracle algorithm and then receives a challenge ciphertext χ^*. $\mathscr{A}_2$ runs $(\omega, K_1) \longleftarrow \mathtt{Tag\text{-}SKEM.Key}(pk_R, sk_S)$ and then computes the following.

$$\mathtt{Tag\text{-}SKEM.Enc}_{pk}(\omega, \chi^*) = \psi^* \tag{19}$$

In the end, the adversary $\mathscr{A}_2$ submits the challenger cipher $c^* = (\psi^*, \chi^*)$ to adversary $\mathscr{A}$.

(iv) **Stage 4:** The adversary $\mathscr{A}$ inputs a public key pair (pk_{R_i}, pk_{S_i}) and makes continuous queries $c = (\psi_i, \chi_i)$ to decryption oracle algorithm. Here, we require that adversary $\mathscr{A}_2$ is not admitted to query (pk_S, c^*) to the decryption oracle. However, we admit that the adversary A_2 is admitted to query the decryption oracle algorithm on $(pk_{S'}, C)$ for any $pk_{S'} \neq pk_{S_i}$ and on (pk_{S_i}, C) for any cipher $C \neq C^*$. The adversary $\mathscr{A}_2$ uses (pk_{S_i}, sk_{R_i}) to decrypt the cipher $c = (\psi_i, \chi_i)$. The adversary $\mathscr{A}_2$ runs the decryption oracle.

$$\begin{aligned}
&\mathtt{K}_i \longleftarrow \mathtt{Tag\text{-}SKEM.Dec}(pk_S, sk_R, \psi_i, \chi_i), \\
&m \longleftarrow \mathtt{DEM.Dec}_{K_i}(\chi_i)
\end{aligned} \tag{20}$$

The adversary $\mathscr{A}_2$ answers $\mathscr{A}$'s decryption query $c = (\psi_i, \chi_i)$ with the following:

(a) If $\mathtt{K}_i = \perp$, then $\mathscr{A}_2$ responds to $\mathscr{A}$ with $\perp$.

(b) If $\mathtt{K}_i = \mathtt{K}_1$ and $\chi = \chi^*$, then $\mathscr{A}_2$ responds to $\mathscr{A}$ with $'text'$.

(c) If $\mathtt{K}_i = \mathtt{K}_1$ and $\chi \neq \chi^*$, then $\mathscr{A}_2$ responds to $\mathscr{A}$ with m.

(d) If $\mathtt{K}_i \neq \mathtt{K}_1$, then $\mathscr{A}_2$ uses $\mathtt{K}_i$ to decrypt the cipher χ.

(1) If $m \in \{m_0, m_1\}$, then adversary $\mathscr{A}_2$ sends $'text'$ to adversary $\mathscr{A}$.

(2) Otherwise, adversary $\mathscr{A}_2$ sends m to adversary $\mathscr{A}$.

Here, we notice that the key $\mathtt{K}_0$ chosen by the signcryption scheme DEM's encryption oracle and embedded in the cipher χ^* is randomly chosen and independent.

(v) **Stage 5:** In the end, the adversary $\mathscr{A}$ outputs $b' \in \{0, 1\}$, and the adversary $\mathscr{A}$ also outputs $\sigma' = b'$.

We have described the construction of the adversary $\mathscr{A}_2$. $\mathscr{A}_2$ plays a perfect simulation $\mathtt{Experiment}$ for $\mathscr{A}$; the view of $\mathscr{A}$ is equal to that in $\mathtt{Experiment}_0$ and $\mathtt{Experiment}_1$; hence, we have

$$\left|\Pr[T_1] - \frac{1}{2}\right| \leq \mathrm{Adv}^{\mathtt{RCCA}}_{\mathtt{DEM},\mathscr{A}_2}(\lambda). \tag{21}$$

Putting all the facts together, we have the following conclusion:

$$\begin{aligned}
&\left|\left(\Pr[T_0] - \frac{1}{2}\right) - \left(\Pr[T_1] - \frac{1}{2}\right)\right| \\
&\leq \mathrm{Adv}^{\mathtt{RCCA}}_{\mathtt{Tag-SKEM+DEM},\mathscr{A}}(\lambda) + \frac{q_s}{|\mathtt{K}_D|}
\end{aligned}$$

$$\mathrm{Adv}^{\mathrm{RCCA}}_{\mathtt{Tag-SKEM+DEM},\mathscr{A}}(\lambda) \leq 2\mathrm{Adv}^{\mathrm{RCCA}}_{\mathtt{Tag-SKEM},\mathscr{A}_1}(\lambda) + \frac{q_s}{|K_D|} + \mathrm{Adv}^{\mathrm{CCA}}_{\mathtt{Tag-DEM},\mathscr{A}_2}(\lambda). \quad (22)$$

□

We have proved Theorem 6.

4. Conclusion

We have examined the RCCA security of two representative hybrid signcryption schemes, i.e., SKEM + DEM [3] and Tag−SKEM+DEM [18], in this paper. We proved that the hybrid signcryption scheme SKEM+DEM is RCCA-secure if the signcryption scheme SKEM is RCCA-secure and the signcryption scheme DEM is RCCA-secure. Meanwhile, we showed that the hybrid encryption scheme Tag−SKEM + DEM can be RCCA-secure if the signcryption scheme Tag−SKEM is RCCA-secure and the scheme DEM is RCCA-secure.

Data Availability

Data sharing is not applicable to this article as no new data was created or analyzed in this study.

Conflicts of Interest

The authors declare that no conflicts of interest exist.

Acknowledgments

This paper is supported by the National Key Research and Development Plan of China under Grant No. 2016YFB0800600 and the National Natural Science Foundation of China (No. 61802006; No. 61602061; No. 61672059; No. 61472016).

References

[1] A. W. Dent and Y. Zheng, *Practical Signcryption, a volume in Information Security and Cryptography*, Springer, 2010.

[2] Y. Zheng, "Digital signcryption or how to achieve cost(signature ∝ encryption) << cost(signature) + cost(encryption)," in *Proceedings of the CRYPTO 1997*, pp. 165–179, 1997.

[3] A. W. Dent, "Hybrid signcryption schemes with outsider security (extended abstract)," in *Proceedings of the ISC*, vol. 3650, pp. 203–217.

[4] J. Lee, "Identity-Based Signcryption, IACR Cryptology ePrint Archive, 2002/098," availabel online https://eprint.iacr.org/2002/098.pdf.

[5] M. Barbosa and P. Farshim, "Certificateless signcryption," in *Proceedings of the ACM Symposium on Information, Computer and Communications Security (ASIACCS '08)*, pp. 369–372, ACM, March 2008.

[6] R. Nakano and J. Shikata, "Constructions of Signcryption in the Multi-user Setting from Identity-Based Encryption," in *Proceedings of the IMACC 2013: Cryptography and Coding*, pp. 324–343, 2013.

[7] F. Li, B. Liu, and J. Hong, "An efficient signcryption for data access control in cloud computing," *Computing*, vol. 99, no. 5, pp. 465–479, 2017.

[8] S. Sato and J. Shikata, "Lattice-Based Signcryption Without Random Oracles," in *Proceedings of the PQCrypto 2018*, pp. 331–351, 2018.

[9] P. Datta, R. Dutta, and S. Mukhopadhyay, "Functional Signcryption," *Journal of Information Security and Applications*, vol. 42, pp. 118–134, 2018.

[10] Y. Zheng and H. Imai, "Compact and unforgeable key establishment over an ATM network," in *Proceedings of the 1998 17th Annual IEEE Conference on Computer Communications, INFOCOM. Part 1 (of 3)*, pp. 411–418, April 1998.

[11] M. Nikravan, A. Movaghar, and M. Hosseinzadeh, "Correction to: A lightweight signcryption scheme for defense against fragment duplication attack in the 6LoWPAN networks," *Peer-to-Peer Networking and Applications*, pp. 1–18, 2018.

[12] X. Zhou, Z. Jin, Y. Fu, H. Zhou, and L. Qin, "Short signcryption scheme for the Internet of Things," *Informatica. An International Journal of Computing and Informatics*, vol. 35, no. 4, pp. 521–530, 2011.

[13] K. T. Nguyen, N. Oualha, and M. Laurent, "Lightweight certificateless and provably-secure signcryptosystem for the internet of things," in *Proceedings of the 14th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2015*, pp. 467–474, Finland, August 2015.

[14] S. Belguith, N. Kaaniche, M. Mohamed, and G. Russello, "C-ABSC: Cooperative Attribute Based SignCryption Scheme for Internet of Things Applications," in *Proceedings of the 2018 IEEE International Conference on Services Computing (SCC)*, pp. 245–248, San Francisco, CA, USA, July 2018.

[15] A. Dent, *Hybrid cryptography*, 2004, Available from http://eprint.iacr.org/2004/210/.

[16] Y. Cui and G. Hanaoka, *Applications of Signcryption*, Springer, Germany, 2010.

[17] S. Prakash and A. Rajput, "Hybrid cryptography for secure data communication in wireless sensor networks," *Advances in Intelligent Systems and Computing*, vol. 696, pp. 589–599, 2018.

[18] T. Bjorstad and A. Dent, "Building better signcryption schemes with tag-KEMs," in *Proceedings of the PKC 2006*, pp. 491–507, 2006.

[19] F. Li, M. Shirase, and T. Takagi, "Certificateless Hybrid Signcryption," in *Proceedings of the ISPEC 2009*, pp. 112–123, 2009.

[20] C. Zhou, "Improved certificateless hybrid signcryption scheme," *Application Research of Computers*, vol. 30, no. 1, pp. 273-272, 2013.

[21] R. Cramer and V. Shoup, "A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack," in *Proceedings of the CRYPTO 1998*, pp. 13–25, 1998.

[22] R. Canetti, S. Halevi, and J. Katz, "Chosen-ciphertext security from identity-based encryption," in *Proceedings of the Eurocrypt 2008*, pp. 207–222, 2008.

[23] R. Canetti, H. Krawczyk, and J. B. Nielsen, "Relaxing chosen-ciphertext security," in *Proceedings of the CRYPTO*, pp. 565–582, 2003.

[24] R. Canetti, H. Krawczyk, and J. Nielsen, *Relaxing Chosen Ciphertext Security*, 2003, available online at http://eprint.iacr.org.

[25] M. Yoshida and T. Fujiwara, "On the Security of Tag-KEM for Signcryption," *Electronic Notes in Theoretical Computer Science*, vol. 171, pp. 83–91, 2007.

[26] M. Abe, R. Gennaro, and K. Kurosawa, "Tag-KEM/DEM: a new framework for hybrid encryption," *Journal of Cryptology. The Journal of the International Association for Cryptologic Research*, vol. 21, no. 1, pp. 97–130, 2008.

[27] Y. Chen and Q. Dong, "RCCA Security for KEM+DEM Style Hybrid Encryptions," in *Proceedings of the Inscrypt*, pp. 102–121, 2012.

[28] V. Shoup, "On formal models for secure key exchange, IACR Cryptology ePrint Archive,Report 1999/012," Available online http://eprint.iacr.org/1999/012.pdf.

[29] H. Cui, Y. Mu, and M. H. Au, "Signcryption secure against linear related-key attacks," *The Computer Journal*, vol. 57, no. 10, pp. 1472–1483, 2014.

[30] H. Dai, J. Chang, Z. Hou, and M. Xu, "The ECCA security of hybrid encryptions," in *Proceedings of the ISPEC*, pp. 847–859, 2017.

[31] R. Cramer and V. Shoup, "Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack," *SIAM Journal on Computing*, vol. 33, no. 1, pp. 167–226, 2003.

[32] M. Abe, R. Gennaro, K. Kurosawa, and V. Shoup, "Tag-KEM/DEM: A new frame-work for hybrid encryption and a new analysis of Kurosawa-Desmedt KEM," in *Proceedings of the EUROCRYPT*, pp. 128–146, 2005.

[33] J. H. An, *Authenticated encryption in the public-key setting: Security notions and analyses*, 2001, Available from http://eprint.iacr.org/2001/079.