# Efficient Generic Constructions of Signcryption with Insider Security in the Multi-user Setting

Daiki Chiba[1], Takahiro Matsuda[2], Jacob C.N. Schuldt[2], and Kanta Matsuura[1]

[1] The University of Tokyo, Japan
{chi-ba,kanta}@iis.u-tokyo.ac.jp
[2] Research Center for Information Security, AIST, Japan
{t-matsuda,jacob.schuldt}@aist.go.jp

**Abstract.** Signcryption is a primitive which provides the combined security properties of encryption and digital signatures i.e. confidentiality and unforgeability. A number of signcryption schemes have been presented in the literature, but up until now, no scheme which simultaneously achieves the currently strongest notions of insider confidentiality and strong insider unforgeability in the multi-user setting, has been proposed, without relying on random oracles or key registration. In this paper, we propose two new generic constructions of signcryption schemes from the combination of standard primitives and simple extensions of these. From our constructions, we instantiate a number of concrete and efficient signcryption schemes which satisfy the strongest notions of insider security in the multi-user setting while still being provably secure in the standard model.

**Keywords:** signcryption, insider security, multi-user setting, generic construction.

## 1 Introduction

Public key encryption and digital signatures aim at providing very different security properties. More specifically, encryption provides confidentiality of data whereas digital signatures provides authenticity of data. However, many practical applications, such as secure e-mail, require both properties simultaneously. To address this need, Zheng [25] proposed *signcryption* which is a single primitive aiming at efficiently providing the security guarantees of both encryption and digital signatures. Although the scheme proposed in [25] was not formally proved secure, this was done in subsequent works [5,6].

Since the introduction of the primitive, many signcryption schemes have been proposed, e.g. [25,4,5,18,19,13,6,17,23,21]. However, due to the many variations of the used security models, the security level achieved by these schemes vary. The simplest security model for signcryption, which was adopted in a few of the early papers [4,13], considers the so-called two-user setting in which only a single sender and a single receiver interact. However, as pointed out by Dent [13], security in the two-user setting does not imply security in the multi-user setting in which several senders address the same receiver, and several receivers receive messages from the same sender. Hence, to ensure security in a more realistic environment, a multi-user security model must be adopted. Furthermore, another aspect of the signcryption security definition, is the concept of "insider" and "outsider" attacks. While some security models require the adversary to attack an uncompromised sender and receiver pair (i.e. the key material of both parties are unknown to the adversary), others allow an "insider" attack in which the adversary has access to the key material of one of the parties. Since security against an insider attack implies security against an outsider attack, the former is preferred. The currently strongest security definitions which capture insider confidentiality and strong insider unforgeability in the multi-user setting, were first defined and used in [18]. For a more detailed overview of the used security models, see [21].

Up until now, several practical signcryption schemes which achieve insider confidentiality and strong insider unforgeability in the multi-user setting have been proposed [19,17,21], but these are only shown secure in the random oracle model. While a number of signcryption schemes which are secure in the standard model, e.g. [4,23,21], these do *not* achieve the same level of security as the random oracle model schemes. For example, An et al. [4] showed how to strengthen the traditional Encrypt-then-Sign and Sign-then-Encrypt schemes to achieve security in the multi-user setting. However, Encrypt-then-Sign construction only achieves the so-called *generalized chosen ciphertext security* which is strictly weaker than ordinary chosen ciphertext security. Sign-then-Encrypt construction achieves ordinary chosen ciphertext security for confidentiality while it only achieves weak unforgeability for authenticity. While weak unforgeability might be sufficient in some scenarios, strong insider unforgeability guarantees that a ciphertext is non-malleable, even for a malicious receiver, and might be needed when the signcryption scheme is used as a building block in a higher level protocol.

Tan [23] proposed a scheme which achieves insider confidentiality in the strongest sense, but the strong insider unforgeability of his scheme is only achieved if *key registration* is used i.e. the adversary is required to reveal the private key corresponding to any public key he makes use of when trying to attack the scheme. In practice, this assumption requires that a traditional public-key infrastructure (PKI) is employed and that all parties engage in a zero-knowledge proof with the certificate authority (CA), which proves knowledge of their private key, before they obtain a certificate for their public key. However, executing these proofs places a heavy burden on the CA, and this type of registration is not used in most practical systems.

Matsuda et al. [21] showed generic constructions of signcryption schemes from existing basic primitives and their simple extensions such as tag-based encryption (TBE) [20,16]. Although their generic constructions were shown to achieve insider confidentiality in the strongest sense, their constructions only achieve either weak insider unforgeability or strong insider unforgeability under the assumption that key registration is used.

To the best our knowledge, there is no standard model signcryption scheme which achieves both insider confidentiality and strong insider unforgeability in the multi-user setting, without relying on key registration. The main motivation of this paper is to construct such signcryption schemes.

*Our Contribution.* In this paper, we propose two new generic constructions of simple but efficient signcryption schemes. Instantiations of our constructions are the first to achieve strong insider unforgeability and insider confidentiality without relying on random oracles or key registration.

Our first construction makes use of an IND-tag-CCA secure tag-based key encapsulation mechanism (TBKEM), an IND-CCA secure data encapsulation mechanism (DEM) which has a one-to-one property, and a sUF-CMA secure signature scheme. Note that TBKEMs are fairly easy to construct from already existing primitives, and many efficient concrete constructions are possible (see Section 2.2). Our second construction makes use of an IND-CCA secure key encapsulation mechanism (KEM), a one-time secure DEM with a one-to-one property, a one-time secure message authentication code (MAC) with a one-to-one property, and a sUF-CMA secure signature scheme. Both of our constructions are based on the ideas related to the well-known Sign-then-Encrypt approach, but they exploit the functionality of tag-based primitives and hybrid encryption (KEM/DEM approach) to overcome the limitation of the previous approaches (see Section 4 for more details).

From these two constructions, we instantiate a number of efficient concrete signcryption schemes which are insider secure in the multi-user setting (in the standard model), and compare these with the existing standard model schemes (see Section 5 for details). We emphasize that the advantage of the above constructions lies not only in the efficiency and security properties achieved by our concrete instantiations, but also in being generic constructions, which allows us to make use of any present or future instantiation of the underlying primitives and the established security results for these.

## 2 Preliminaries

In this section, we review the notation used throughout the paper and the definitions of the primitives that will be used in our first construction. The additional primitives needed for our second construction, which includes a KEM and a MAC, will be given in Appendix A.

### 2.1 Notation

In this paper, "$x \leftarrow y$" denotes that $x$ is chosen uniformly at random from $y$ if $y$ is a finite set, $x$ is output from $y$ if $y$ is a function or an algorithm, or $y$ is assigned to $x$ otherwise. "$x||y$" denotes a concatenation of $x$ and $y$. "PPT" denotes probabilistic polynomial time. "$\kappa$" always denotes the security parameter. We say that a function $f(\kappa)$ is negligible in $\kappa$ if $f(\kappa) \leq 1/p(\kappa)$ for any positive polynomial $p(\kappa)$ and all sufficiently large $\kappa$. In this paper, when we say a function is negligible then we always mean that it is negligible in the security parameter $\kappa$.

### 2.2 Tag-Based Key Encapsulation Mechanism

A tag-based key encapsulation mechanism (TBKEM) is a key encapsulation mechanism whose encapsulation and decapsulation algorithms take an arbitrary string called *tag* as an additional input. Note that a TBKEM is a KEM-analogue of tag-based encryption (TBE) [20,16], and must not be confused with a tag-KEM, which is a building block of hybrid encryption proposed by Abe et al. [2].

Formally, a TBKEM is given by the following four algorithms.

$\mathsf{TSetup}$: Given input a security parameter $1^\kappa$, this algorithm returns a set of public parameters $prm$ (included in $prm$ is a description of a key space $\mathcal{K}$).
$\mathsf{TKG}$: Given input $prm$, this algorithm returns a public/private key pair $(pk, sk)$.
$\mathsf{TEncap}$: Given input $prm$, a public key $pk$ and a tag $\mathsf{tag}$, this algorithm returns an encapsulation and encapsulated key pair $(c, K)$.
$\mathsf{TDecap}$: Given input $prm$, a private key $sk$, a tag $\mathsf{tag}$ and an encapsulation $c$, this algorithm returns a key $K$ or an error symbol $\perp$.

It is required for all $prm \leftarrow \mathsf{TSetup}(1^\kappa)$, all $(pk, sk) \leftarrow \mathsf{TKG}(prm)$, all tags $\mathsf{tag}$, and all $(c, K) \leftarrow \mathsf{TEncap}(prm, pk, \mathsf{tag})$, that $K = \mathsf{TDecap}(prm, sk, \mathsf{tag}, c)$

*IND-tag-CCA Security.* For a TBKEM, indistinguishability against adaptive tag and adaptive chosen ciphertext attacks (IND-tag-CCA) is defined by the following game between an adversary $\mathcal{A}$ and an IND-tag-CCA challenger $\mathcal{CH}$.

**Setup.** $\mathcal{CH}$ computes $prm \leftarrow \mathsf{TSetup}(1^\kappa)$ and $(pk, sk) \leftarrow \mathsf{TKG}(prm)$, and then forwards $(prm, pk)$ to $\mathcal{A}$ and keeps $sk$ to itself.
**Phase 1.** $\mathcal{A}$ can adaptively submit decapsulation queries $(\mathsf{tag}, c)$ to $\mathcal{CH}$. $\mathcal{CH}$ responds to each query by returning $K \leftarrow \mathsf{TDecap}(prm, sk, \mathsf{tag}, c)$.
**Challenge.** $\mathcal{A}$ chooses a challenge tag $\mathsf{tag}^*$ and sends this to $\mathcal{CH}$. $\mathcal{CH}$ computes $(c^*, K_1^*) \leftarrow \mathsf{TEncap}(pk, \mathsf{tag}^*)$, and chooses $K_0^* \in \mathcal{K}$ uniformly at random. Then $\mathcal{CH}$ flips a fair coin $b \in \{0, 1\}$, and returns the challenge encapsulation and key $(c^*, K_b^*)$ to $\mathcal{A}$.
**Phase 2.** $\mathcal{A}$ can submit decapsulation queries in the same way as in Phase 1, except that $\mathcal{A}$ is not allowed to submit the challenge pair $(\mathsf{tag}^*, c^*)$.
**Guess.** $\mathcal{A}$ outputs a bit $b'$ as its guess for $b$.

We define the IND-tag-CCA advantage of $\mathcal{A}$ attacking the TBKEM $TK$ as $\mathsf{Adv}_{TK,\mathcal{A}}^{\text{IND-tag-CCA}} = |\Pr[b' = b] - \frac{1}{2}|$.

**Definition 1.** *We say that a TBKEM $TK$ is IND-tag-CCA secure if $\mathsf{Adv}_{TK,\mathcal{A}}^{\text{IND-tag-CCA}}$ is negligible for any PPT adversary $\mathcal{A}$.*

*How to construct TBKEMs.* Although one might think that a TBKEM is not a basic primitive, any IND-CCA secure public key encryption scheme can be converted into a IND-tag-CCA secure TBKEM by encrypting a random session key together with the tag [16]. Furthermore, any IND-CCA secure tag-KEM can be used as an IND-tag-CCA secure TBKEM, which allows the many practical constructions of tag-KEMs to be used (e.g. [2,1]). Lastly, Abe et al. [2] show that IND-CCA secure tag-KEMs can be generically built from any IND-CCA secure ordinary KEM and a one-time secure MAC, which implies that this technique can be used for the construction of IND-tag-CCA secure TBKEMs as well.

## 2.3 Data Encapsulation Mechanism

A data encapsulation mechanism (DEM) is given by the following two algorithms.

$\mathsf{DEnc}$: Given input a symmetric key $K \in \mathcal{K}$ and a message $m$, this algorithm returns a ciphertext $c$. $\mathcal{K}$ is a key space.

$\mathsf{DDec}$: Given input symmetric key $K \in \mathcal{K}$ and a ciphertext $c$, this algorithm returns a message $m$ or an error symbol $\perp$.

It is required for all $K \in \mathcal{K}$ and all messages $m$ that $\mathsf{DDec}(K, \mathsf{DEnc}(K, m)) = m$.

*IND-CCA Security.* For a DEM, indistinguishability against adaptive chosen ciphertext attacks (IND-CCA) is defined by the following game between an adversary $\mathcal{A}$ and an IND-CCA challenger $\mathcal{CH}$.

**Setup.** $\mathcal{CH}$ chooses $K \in \mathcal{K}$ uniformly at random, where $\mathcal{K}$ is a key space.

**Phase 1.** $\mathcal{A}$ can adaptively submit decryption queries $c$ to $\mathcal{CH}$. $\mathcal{CH}$ responds to each query by returning $m \leftarrow \mathsf{DDec}(K, c)$.

**Challenge.** $\mathcal{A}$ chooses two plaintexts $(m_0, m_1)$ of equal length, and sends them to $\mathcal{CH}$. $\mathcal{CH}$ flips a fair coin $b \in \{0, 1\}$, and then returns the challenge ciphertext $c^* \leftarrow \mathsf{DEnc}(K, m_b)$ to $\mathcal{A}$.

**Phase 2.** $\mathcal{A}$ can submit decryption queries in the same way as in Phase 1, except that $\mathcal{A}$ is not allowed to submit the challenge ciphertext $c^*$.

**Guess.** $\mathcal{A}$ outputs a bit $b'$ as its guess for $b$.

Note that, in the above game, $\mathcal{A}$ is not allowed to make any encryption queries i.e. the above defined IND-CCA security for a DEM is strictly weaker, and therefore easier to achieve, than ordinary IND-CCA security for symmetric encryption.

We define the IND-CCA advantage of $\mathcal{A}$ attacking the DEM $D$ as $\mathsf{Adv}^{\text{IND-CCA}}_{D,\mathcal{A}} = |\Pr[b' = b] - \frac{1}{2}|$.

Furthermore, we define indistinguishability against one time attacks (IND-OT) by an IND-OT game which is defined in the same way as the IND-CCA game except that the adversary is not allowed to submit any decryption queries. The advantage $\mathsf{Adv}^{\text{IND-OT}}_{D,\mathcal{A}}$ of an adversary attacking the IND-OT security of a DEM $D$ is defined similarly to the IND-CCA advantage.

**Definition 2.** *We say that a DEM $D$ is IND-CCA (resp. IND-OT) secure if $\mathsf{Adv}^{\text{IND-CCA}}_{D,\mathcal{A}}$ (resp. $\mathsf{Adv}^{\text{IND-OT}}_{D,\mathcal{A}}$) is negligible for any PPT adversary $\mathcal{A}$.*

*One-to-One Property.* A DEM is said to be *one-to-one* if given a key $K$ and a plaintext $m$, there is at most one $c$ such that $\mathsf{DDec}(K, c) = m$. We consider this property to be quite natural for a DEM, and all IND-CCA secure DEMs based on strong pseudorandom permutations [22] satisfy this. Furthermore, the well-known Encrypt-then-MAC [7] construction of IND-CCA secure DEMs preserves the one-to-one property of the underlying (IND-OT secure) DEM and MAC i.e. if both the used DEM and MAC are one-to-one, then so is the DEM resulting from a Encrypt-then-MAC construction (the definition of the one-to-one property of a MAC is given in Section A.2). We are not aware of any natural construction of a DEM which does not have this property.

## 2.4 Signature

A signature scheme is given by the following four algorithms.

$\mathsf{SSetup}$**:** Given input a security parameter $1^\kappa$, this algorithm returns a set of public parameters $prm$.

$\mathsf{SKG}$**:** Given input $prm$, this algorithm returns a public/private key pair $(pk, sk)$.

$\mathsf{Sign}$**:** Given input $prm$, private key $sk$ and a message $m$, this algorithm returns a signature $\sigma$.

$\mathsf{SVer}$**:** Given input $prm$, a public key $pk$, a message $m$, and a signature $\sigma$, this algorithm returns $\top$ or $\perp$.

It is required for all $prm \leftarrow \mathsf{SSetup}(1^\kappa)$, all $(pk, sk) \leftarrow \mathsf{SKG}(prm)$ and all messages $m$, that $\mathsf{SVer}(prm, pk, m, \mathsf{Sign}(prm, sk, m)) = \top$

*sUF-CMA Security.* For a signature scheme, strong unforgeability against chosen message attacks (sUF-CMA) is defined by the following game between an adversary $\mathcal{A}$ and a sUF-CMA challenger $\mathcal{CH}$.

**Setup.** $\mathcal{CH}$ computes $prm \leftarrow \mathsf{SSetup}(1^\kappa)$ and $(pk, sk) \leftarrow \mathsf{SKG}(prm)$, and generates an empty list $\mathcal{L}$ into which message/signature pairs will be stored. $\mathcal{CH}$ then forwards $(prm, pk)$ to $\mathcal{A}$ and keeps $sk$ to itself.

**Query.** $\mathcal{A}$ can adaptively submit signature queries $m$ to $\mathcal{CH}$. $\mathcal{CH}$ responds to each query by returning $\sigma \leftarrow \mathsf{Sign}(prm, sk, m)$. $\mathcal{CH}$ furthermore stores the message/signature pair $(m, \sigma)$ in $\mathcal{L}$.

**Output.** $\mathcal{A}$ outputs a message/signature pair $(m^*, \sigma^*)$.

We define the sUF-CMA advantage of $\mathcal{A}$ attacking the signature scheme $S$ as follows:

$$\mathsf{Adv}_{S,\mathcal{A}}^{\text{sUF-CMA}} = \Pr[\mathsf{SVer}(prm, pk, m^*, \sigma^*) = \top \wedge (m^*, \sigma^*) \notin \mathcal{L}]$$

**Definition 3.** *We say that a signature $S$ is sUF-CMA secure if* $\mathsf{Adv}_{S,\mathcal{A}}^{\text{sUF-CMA}}$ *is negligible for any PPT adversary $\mathcal{A}$.*

## 3 Signcryption

In this section we review the formal definition of a signcryption scheme.

*Algorithms.* A signcryption scheme is given by the following five algorithms.

$\mathsf{Setup}$: Given input a security parameter $1^\kappa$, this algorithm returns a set of public parameters $prm$.

$\mathsf{KeyGen_S}$: This is the sender's key generation algorithm which takes $prm$ as input and returns a public/private sender key pair $(pk_S, sk_S)$.

$\mathsf{KeyGen_R}$: This is the receiver's key generation algorithm which takes $prm$ as input and returns a public/private receiver key pair $(pk_R, sk_R)$.

$\mathsf{SC}$: This is the signcryption algorithm which takes $prm$, a private sender key $sk_S$, a public receiver key $pk_R$, and a message $m$ as input, and returns a ciphertext $c$.

$\mathsf{USC}$: This is the unsigncryption algorithm which takes $prm$, a public sender key $pk_S$, a private receiver key $sk_R$, and a ciphertext $c$ as input, and returns either a message $m$ or an error symbol $\perp$.

It is required for all $prm \leftarrow \mathsf{Setup}(1^\kappa)$, all $(pk_S, sk_S) \leftarrow \mathsf{KeyGen_S}(prm)$, all $(pk_R, sk_R) \leftarrow \mathsf{KeyGen_R}(prm)$, all messages $m$ and all $c \leftarrow \mathsf{SC}(prm, sk_S, pk_R, m)$, that $m = \mathsf{USC}(prm, pk_S, sk_R, c)$.

### 3.1 Security

In this subsection, we review the security model in which we will prove our signcryption constructions secure using the notations introduced in [21]. More specifically, for confidentiality, we will use the notion indistinguishability against insider chosen ciphertext attacks in the dynamic multi-user model (*dM-IND-iCCA*), and for unforgeability, we use the notion strong unforgeability against insider chosen message attacks in the dynamic multi-user model (*dM-sUF-iCMA*).

*dM-IND-iCCA Security.* For a signcryption scheme, dM-IND-iCCA security is defined by the following game between an adversary $\mathcal{A}$ and a dM-IND-iCCA challenger $\mathcal{CH}$.

**Setup.** $\mathcal{CH}$ computes $prm \leftarrow \mathsf{Setup}(1^\kappa)$ and $(pk_R, sk_R) \leftarrow \mathsf{KeyGen_R}(prm)$, forwards $(prm, pk_R)$ to $\mathcal{A}$, and keeps $sk_R$ to itself.

**Phase 1.** $\mathcal{A}$ can adaptively submit unsigncryption queries $(pk_S, c)$ to $\mathcal{CH}$. $\mathcal{CH}$ responds to each query by returning $m \leftarrow \mathsf{USC}(prm, pk_S, sk_R, c)$ to $\mathcal{A}$.

**Challenge.** $\mathcal{A}$ chooses two plaintexts $(m_0, m_1)$ of equal length and a challenge sender public/private key pair $(pk_S^*, sk_S^*)$, and sends these to $\mathcal{CH}$. It is required that $(pk_S^*, sk_S^*)$ is a valid key pair i.e. $(pk_S^*, sk_S^*)$ lies in the range of $\mathsf{KeyGen_S}(prm)$. $\mathcal{CH}$ flips a fair coin $b \in \{0, 1\}$, and then returns the challenge ciphertext $c^* \leftarrow \mathsf{SC}(prm, pk_R, sk_S^*, m_b)$ to $\mathcal{A}$.

**Phase 2.** $\mathcal{A}$ can submit unsigncryption queries in the same way as in Phase 1, except that $\mathcal{A}$ is not allowed to submit the pair $(pk_S^*, c^*)$.

**Guess.** $\mathcal{A}$ outputs a bit $b'$ as its guess for $b$.

We define the dM-IND-iCCA advantage of $\mathcal{A}$ attacking signcryption $SC$ as $\mathsf{Adv}_{SC,\mathcal{A}}^{\text{dM-IND-iCCA}} = |\Pr[b' = b] - \frac{1}{2}|$.

**Definition 4.** *We say that a signcryption scheme SC is dM-IND-iCCA secure if* $\mathsf{Adv}_{SC,\mathcal{A}}^{\text{dM-IND-iCCA}}$ *is negligible for any PPT adversary* $\mathcal{A}$.

Note that, in the dM-IND-iCCA security game, $\mathcal{A}$ can freely choose the sender keys submitted in the unsigncryption queries, and is not required to prove knowledge of or reveal the private keys corresponding to these (i.e. $\mathcal{A}$ is allowed to employ an attack strategy in which these private keys are unknown to $\mathcal{A}$). This will ensure security in the multi-user scenario in which a receiver decrypts ciphertexts from multiple senders. Furthermore, we would like to emphasize that the challenge sender key pair $(pk_S^*, sk_S^*)$ is also freely chosen by $\mathcal{A}$, and in particular, $\mathcal{A}$ knows $sk_S^*$. This implies that confidentiality is maintained, even if the private sender key is compromised i.e. insider confidentially is guaranteed. This is currently the strongest security notion used in the signcryption literature, and is strictly stronger than the security notion used in a number of previous papers [5,4,13,6]. See [21] for a more detailed di scussion of the security models for signcryption.

*dM-sUF-iCMA Security.* For a signcryption scheme, dM-sUF-iCMA security is defined by the following game between an adversary $\mathcal{A}$ and a dM-sUF-iCMA challenger $\mathcal{CH}$.

**Setup.** $\mathcal{CH}$ computes $prm \leftarrow \mathsf{Setup}(1^\kappa)$ and $(pk_S, sk_S) \leftarrow \mathsf{KeyGen_S}(prm)$, and generates an empty list $\mathcal{L}$ into which the adversary's queries and answers will be stored. $\mathcal{CH}$ forwards $(prm, pk_S)$ to $\mathcal{A}$ and keeps $sk_S$ to itself.

**Query.** $\mathcal{A}$ can adaptively submit signcryption queries $(pk_R, m)$ to $\mathcal{CH}$. $\mathcal{CH}$ responds to each query by returning $c \leftarrow \mathsf{SC}(prm, sk_S, pk_R, m)$. Furthermore, $\mathcal{CH}$ stores the tuple $(pk_R, c, m)$ in $\mathcal{L}$.

**Output.** $\mathcal{A}$ outputs a tuple $(pk_R^*, sk_R^*, c^*)$. It is required that $(pk_R^*, sk_R^*)$ is a valid key pair.

We define the dM-sUF-iCMA advantage of $\mathcal{A}$ attacking signcryption $SC$ as follows:

$$\mathsf{Adv}_{SC,\mathcal{A}}^{\text{dM-sUF-iCMA}} = \Pr[\mathsf{USC}(prm, pk_S^*, sk_R^*, c^*) = m^* \neq \perp \wedge (pk_R^*, c^*, m^*) \notin \mathcal{L}]$$

**Definition 5.** *We say that a signcryption scheme SC is dM-sUF-iCMA secure if* $\mathsf{Adv}_{SC,\mathcal{A}}^{\text{dM-sUF-iCMA}}$ *is negligible for any PPT adversary* $\mathcal{A}$.

Similar to the confidentiality definition, $\mathcal{A}$ can freely choose public receiver keys in the signcryption queries, and is not required to prove knowledge of or reveal the private keys corresponding to these i.e. $\mathcal{A}$ can observe signcryptions under multiple receiver keys as required for multi-user security. Furthermore, $\mathcal{A}$ can freely choose the challenge receiver key pair $(pk_R^*, sk_R^*)$, and proving security in this model ensures insider unforgeability. Like dM-IND-iCCA, this is the strongest unforgeability notion used in the signcryption literature.

## 4 Proposed Generic Constructions

In this section, we show our generic constructions of signcryption schemes which achieve both insider confidentiality and strong insider unforgeability in the multi-user setting. The first construction is based on a TBKEM, a signature scheme

```
$\mathsf{Setup}(1^\kappa)$ :
    $prm_{tk} \leftarrow \mathsf{TSetup}(1^\kappa)$
    $prm_{sig} \leftarrow \mathsf{SSetup}(1^\kappa)$
    Output $prm \leftarrow (prm_{tk}, prm_{sig})$.

$\mathsf{KeyGen_S}(prm)$ :
    Output $(pk_S, sk_S) \leftarrow \mathsf{SKG}(prm_{sig})$.

$\mathsf{KeyGen_R}(prm)$ :
    Output $(pk_R, sk_R) \leftarrow \mathsf{TKG}(prm_{tk})$.

$\mathsf{SC}(prm, sk_S, pk_R, m)$ :
    $\mathsf{tag} \leftarrow pk_S$
    $(c_1, K) \leftarrow \mathsf{TEncap}(prm_{tk}, pk_R, \mathsf{tag})$
    $\sigma \leftarrow \mathsf{Sign}(prm_{sig}, sk_S, (m||c_1||pk_R))$
    $c_2 \leftarrow \mathsf{DEnc}(K, (m||\sigma))$
    Output $c \leftarrow (c_1, c_2)$

$\mathsf{USC}(prm, pk_S, sk_R, c)$ :
    Parse $c$ as $(c_1, c_2)$
    $\mathsf{tag} \leftarrow pk_S$
    $K \leftarrow \mathsf{TDecap}(prm_{tk}, sk_R, \mathsf{tag}, c_1)$
        (if $K = \perp$, then return $\perp$)
    $(m||\sigma) \leftarrow \mathsf{DDec}(K, c_2)$
        (if $\mathsf{DDec}$ outputs $\perp$, then return $\perp$)
    If $\mathsf{SVer}(prm_{sig}, pk_S, (m||c_1||pk_R), \sigma) = \perp$
        then return $\perp$
    Output $m$
```

**Fig. 1.** Construction using TBKEM: $SC_{tk}$

and a DEM, and is given in Section 4.1. The second construction is based on a KEM, a signature scheme, a MAC and a DEM, and is given in Section 4.2.

## 4.1 Composition Using TBKEM

Let $TK = (\mathsf{TSetup}, \mathsf{TKG}, \mathsf{TEncap}, \mathsf{TDecap})$ be an IND-tag-CCA secure TBKEM, let $D = (\mathsf{DEnc}, \mathsf{DDec})$ be an IND-CCA secure DEM, and let $S = (\mathsf{SSetup}, \mathsf{SKG}, \mathsf{Sign}, \mathsf{SVer})$ be a sUF-CMA secure signature scheme. Then, we construct a signcryption scheme $SC_{tk}$ as shown in Fig.1. For simplicity, we assume the symmetric key space of $TK$ and that of $D$ are $\{0,1\}^\kappa$.

*Construction Idea.* The proposed construction can be regarded as a variant of the Sign-then-Encrypt construction. However, as shown by the previous results [4,21], the ordinary Sign-then-Encrypt constructions cannot achieve strong insider unforgeability in the multi-user setting (dM-sUF-iCMA security), because an insider adversary who has a receiver's private key and receives a ciphertext from a signcryption query can decrypt the ciphertext and then re-encrypt it, which can be a successful forgery in the sense of strong unforgeability. However, in the proposed construction, we adopt the KEM/DEM approach and thus can sign the KEM ciphertext together with the message. Because of this, an insider adversary can no longer mount the above attack, and thus to break the strong unforgeability he has to modify the DEM ciphertext so that the whole ciphertext remains valid. However, such modification of the DEM ciphertext is impossible if the underlying DEM is one-to-one, which guarantees the dM-sUF-iCMA security of our construction. To achieve security in the multi-user setting, we make use of a similar idea to [21] and employ a tag-based KEM. More specifically, we sign the receiver's public key with a signature scheme, and use the sender's public key as a tag of the TBKEM. This will ensure that the ciphertext is only valid for a specific sender and receiver key pairs.

*Security.* The security of $SC_{tk}$ is formally guaranteed by the following theorems.

**Theorem 1.** *If the TBKEM $TK$ is IND-tag-CCA secure and the DEM $D$ is IND-CCA secure, then the proposed signcryption scheme $SC_{tk}$ is dM-IND-iCCA secure.*

*Proof.* Suppose there exists an adversary $\mathcal{A}$ that breaks the dM-IND-iCCA security of the signcryption scheme $SC_{tk}$. We then consider the following games. (Values used in the challenge phase are marked with an asterisk (*))

$\mathsf{Game}_0$**:** This is the ordinary dM-IND-iCCA game regarding $SC_{tk}$.

$\mathsf{Game}_1$**:** In this game, a uniformly random key $K' \in \mathcal{K}$ is chosen at the beginning of the game, and the second component $c_2^*$ of the challenge ciphertext is computed by using this $K'$ ($c_2^* \leftarrow \mathsf{DEnc}(K', (m||\sigma))$). Moreover, for an unsigncryption query of the form $(pk_S^*, c_1^*, c_2)$ with $c_2 \neq c_2^*$, $c_2$ is decrypted using $K'$. The rest is unchanged from $\mathsf{Game}_0$.

We define $\mathsf{Succ}_i$ as the event that $\mathcal{A}$, in Game $i$, succeeds in guessing the bit $b$ used for generating the challenge ciphertext. Then, we have

$$\mathsf{Adv}^{\text{dM-IND-iCCA}}_{SC_{tk},\mathcal{A}} = |\Pr[\mathsf{Succ}_0] - \frac{1}{2}| \leq |\Pr[\mathsf{Succ}_0] - \Pr[\mathsf{Succ}_1]| + |\Pr[\mathsf{Succ}_1] - \frac{1}{2}| \quad (1)$$

To complete the proof, we prove the following lemmas.

**Lemma 1.** $|\Pr[\mathsf{Succ}_0] - \Pr[\mathsf{Succ}_1]|$ *is negligible.*

*Proof.* (of Lemma 1) Towards a contradiction, assume $|\Pr[\mathsf{Succ}_0] - \Pr[\mathsf{Succ}_1]|$ is non-negligible. Using $\mathcal{A}$ as a building block, we then show how to construct a PPT adversary $\mathcal{S}$ which breaks the IND-tag-CCA security of the TBKEM scheme $TK$, thereby proving the lemma by contradiction. $\mathcal{S}$ plays its own IND-tag-CCA game regarding the TBKEM scheme $TK$ as follows.

**Setup.** Given $(prm_{tk}, pk_R)$, $\mathcal{S}$ runs $prm_{sig} \leftarrow \mathsf{SSetup}(1^\kappa)$ and sets $prm \leftarrow (prm_{tk}, prm_{sig})$. Then $\mathcal{S}$ runs $\mathcal{A}$ with input $(prm, pk_R)$.

**Phase 1.** $\mathcal{S}$ responds to $\mathcal{A}$'s unsigncryption queries $(pk_S, (c_1, c_2))$ as follow. $\mathcal{S}$ first sets $\mathsf{tag} \leftarrow pk_S$, issues $(\mathsf{tag}, c_1)$ to $\mathcal{CH}$ as his decapsulation query and then obtains $K$. Then $\mathcal{S}$ computes $(m||\sigma) \leftarrow \mathsf{DDec}(K, c_2)$ and $\mathsf{SVer}(prm_{sig}, pk_S, (m||c_1||pk_R), \sigma)$. Finally, if the value returned from $\mathsf{SVer}$ is $\top$, $\mathcal{S}$ returns $m$ to $\mathcal{A}$, otherwise returns $\perp$.

**Challenge.** When $\mathcal{A}$ submits $(m_0, m_1, pk_S^*, sk_S^*)$ to $\mathcal{S}$, $\mathcal{S}$ regards $pk_S^*$ as its challenge tag $\mathsf{tag}^* = pk_S^*$ and submits $\mathsf{tag}^*$ to $\mathcal{CH}$ and obtains $\mathcal{S}$'s challenge ciphertext/key pair $(c_1^*, K_\beta^*)$, where $\beta$ is the challenge bit of $\mathcal{S}$ in the IND-tag-CCA game. Then $\mathcal{S}$ flips a coin $b \in \{0,1\}$ uniformly at random, and computes $\sigma^* \leftarrow \mathsf{Sign}(prm_{sig}, sk_S^*, (m_b||c_1^*||pk_R))$ and $c_2^* \leftarrow \mathsf{DEnc}(K_\beta^*, (m_b||\sigma^*))$. $\mathcal{S}$ finally returns $c^* = (c_1^*, c_2^*)$ to $\mathcal{A}$ as $\mathcal{A}$'s challenge ciphertext.

**Phase 2.** $\mathcal{S}$ responds to $\mathcal{A}$'s unsigncryption queries $(pk_S, (c_1, c_2))$ as follows.

1. **If** $(pk_S, c_1) = (pk_S^*, c_1^*)$**:** $\mathcal{S}$ computes $(m||\sigma) \leftarrow \mathsf{DDec}(K_\beta^*, c_2)$ and $\mathsf{SVer}(prm_{sig}, pk_S^*, (m||c_1^*||pk_R), \sigma)$. If the value returned from $\mathsf{SVer}$ is $\top$, $\mathcal{S}$ returns $m$ to $\mathcal{A}$, otherwise returns $\perp$.

2. **Otherwise:** $\mathcal{S}$ returns $m/\bot$ in the same way as the unsigncryption queries in Phase 1.

**Guess.** $\mathcal{A}$ outputs a bit $b'$. If $b' = b$, $\mathcal{S}$ outputs $\beta' = 1$ and otherwise, outputs $\beta' = 0$ as its guess.

Note that $\mathcal{S}$ never issues the prohibited decapsulation query $(\mathsf{tag}^*, c_1^*) = (pk_S^*, c_1^*)$ to $\mathcal{CH}$.

Then, the IND-tag-CCA advantage of the adversary $\mathcal{S}$ is estimated as:

$$\begin{aligned}\mathsf{Adv}_{TK,\mathcal{S}}^{\text{IND-tag-CCA}} &= |\Pr[\beta' = \beta] - \frac{1}{2}| \\ &= \frac{1}{2}|\Pr[\beta' = 0|\beta = 1] - \Pr[\beta' = 0|\beta = 0]| \\ &= \frac{1}{2}|\Pr[b' = b|\beta = 1] - \Pr[b' = b|\beta = 0]|\end{aligned}$$

Now, consider the case when $\beta = 1$, i.e. $K_\beta^* = K_1^*$ is a real symmetric key corresponding to $c_1^*$ under the tag $\mathsf{tag} = pk_S^*$. Then, it is easy to see that $\mathcal{S}$ perfectly simulates $\mathsf{Game}_0$ for $\mathcal{A}$ in which the challenge bit for $\mathcal{A}$ is $b$. Specifically, $\mathcal{A}$'s responses to decryption queries are perfectly answered as in $\mathsf{Game}_0$, and the challenge ciphertext $c^*$ for $\mathcal{A}$ is generated as in $\mathsf{Game}_0$ ($c_2^*$ is computed with a correct symmetric key $K_1^*$ as in the proposed signcryption scheme $SC_{tk}$). Under this situation, the event $b' = b$ corresponds to the event $\mathsf{Succ}_0$, i.e. $\Pr[b' = b|\beta = 1] = \Pr[\mathsf{Succ}_0]$.

When $\beta = 0$, i.e. $K_\beta^* = K_0^*$ is a uniformly random value in $\{0,1\}^\kappa$, on the other hand, $\mathcal{S}$ perfectly simulates $\mathsf{Game}_1$ for $\mathcal{A}$ in which the challenge bit for $\mathcal{A}$ is $b$. Specifically, $c_2^*$ in the challenge ciphertext is an encryption of $m_b$ under the random key $K_0^*$, and an unsigncryption query of the form $(pk_S^*, c_1^*, c_2)$ where $c_2 \neq c_2^*$ is answered using $K_0^*$. Under this situation, the event $b' = b$ corresponds to the event $\mathsf{Succ}_1$, i.e. $\Pr[b' = b|\beta = 0] = \Pr[\mathsf{Succ}_1]$.

In summary, we have $\mathsf{Adv}_{TK,\mathcal{S}}^{\text{IND-tag-CCA}} = \frac{1}{2}|\Pr[\mathsf{Succ}_0] - \Pr[\mathsf{Succ}_1]|$ which is not negligible by the assumption we made at the beginning of the proof of this lemma. Since this contradicts IND-tag-CCA security of the TBKEM $TK$, $|\Pr[\mathsf{Succ}_0] - \Pr[\mathsf{Succ}_1]|$ must be negligible. This completes the proof of Lemma 1. □

**Lemma 2.** $|\Pr[\mathsf{Succ}_1] - \frac{1}{2}|$ *is negligible.*

*Proof.* (of Lemma 2) Assume towards a contradiction that $|\Pr[\mathsf{Succ}_1] - \frac{1}{2}|$ is not negligible. Then we show that we can construct another PPT adversary $\mathcal{S}$ that uses $\mathcal{A}$ as a subroutine and has non-negligible IND-CCA advantage against the DEM $D$, thereby proving the lemma by contradiction. The description of the IND-CCA adversary $\mathcal{S}$ is as follows.

**Setup.** $\mathcal{S}$ first computes $prm_{tk} \leftarrow \mathsf{TSetup}(1^\kappa)$ and $prm_{sig} \leftarrow \mathsf{SSetup}(1^\kappa)$, then sets $prm \leftarrow (prm_{tk}, prm_{sig})$. Then $\mathcal{S}$ computes $(pk_R, sk_R) \leftarrow \mathsf{TKG}(prm_{tk})$ and runs $\mathcal{A}$ with input $(prm, pk_R)$.

**Phase 1.** $\mathcal{S}$ responds to $\mathcal{A}$'s unsigncryption queries $(pk_S, (c_1, c_2))$ by returning $m \leftarrow \mathsf{USC}(prm, pk_S, sk_R, (c_1, c_2))$. This is possible because $\mathcal{S}$ owns $sk_R$.

**Challenge.** When $\mathcal{A}$ submits $(m_0, m_1, pk_S^*, sk_S^*)$ to $\mathcal{S}$, $\mathcal{S}$ first sets $\mathsf{tag} \leftarrow pk_S^*$, computes $(K', c_1^*) \leftarrow \mathsf{TEncap}(prm_{tk}, pk_R, \mathsf{tag})$. Next $\mathcal{CH}$ computes $\sigma_0 \leftarrow \mathsf{Sign}(prm_{sig}, sk_S^*, (m_0||c_1^*||pk_R^*))$ and $\sigma_1 \leftarrow \mathsf{Sign}(prm_{sig}, sk_S^*, (m_1||c_1^*||pk_R^*))$. Then $\mathcal{S}$ submits two plaintexts $M_0 = (m_0||\sigma_0)$ and $M_1 = (m_1||\sigma_1)$ to $\mathcal{CH}$ as $\mathcal{S}$'s challenge and obtains $c_2^*$. $\mathcal{S}$ finally returns $c^* = (c_1^*, c_2^*)$ to $\mathcal{A}$ as $\mathcal{A}$'s challenge ciphertext.

**Phase 2.** $\mathcal{S}$ responds to $\mathcal{A}$'s unsigncryption queries $(pk_S, (c_1, c_2))$ as follows.

1. **If** $(pk_S, c_1) = (pk_S^*, c_1^*)$**:** $\mathcal{S}$ issues $c_2$ as a decryption query and obtains $(m||\sigma)$. Then $\mathcal{S}$ computes $\mathsf{SVer}(prm_{sig}, pk_S, (m||c_1||pk_R), \sigma)$. Finally, if the returned value of $\mathsf{SVer}$ is $\top$, $\mathcal{S}$ returns $m$ to $\mathcal{A}$, otherwise returns $\perp$.
2. **Otherwise:** $\mathcal{S}$ returns $m/\perp$ in the same way as the unsigncryption queries in Phase 1.

**Guess.** $\mathcal{A}$ outputs a bit $b'$. $\mathcal{S}$ outputs $b'$ as its guess.

Note that $\mathcal{S}$ never issues the prohibited decryption query $c_2^*$ to $\mathcal{CH}$. Moreover it is easy to see that $\mathcal{S}$ perfectly simulates $\mathsf{Game}_1$ for $\mathcal{A}$ in which $\mathcal{A}$'s challenge bit is that of $\mathcal{S}$'s. Therefore, $\mathcal{S}$'s IND-CCA advantage can be caluculated as $\mathsf{Adv}_{D,\mathcal{S}}^{\text{IND-CCA}} = |\Pr[\mathsf{Succ}_1] - \frac{1}{2}|$ which is not negligible according to the assumption we made at the beginning of the proof of this lemma. Since this contradicts the IND-CCA security of the DEM $D$, $|\Pr[\mathsf{Succ}_1] - \frac{1}{2}|$ must be negligible. This completes the proof of Lemma 2. □

According to the inequality (1) and Lemmas 1 and 2, $\mathsf{Adv}_{SC_{tk},\mathcal{A}}^{\text{dM-IND-iCCA}}$ is upper-bounded to be negligible for any PPT adversary $\mathcal{A}$. This completes the proof of Theorem 1. □

**Theorem 2.** *If the signature scheme $S$ is sUF-CMA secure and the DEM $D$ is one-to-one, then the proposed signcryption scheme $SC_{tk}$ is dM-sUF-iCMA secure.*

*Proof.* Suppose there exists an adversary $\mathcal{A}$ that breaks the dM-sUF-iCMA security of the signcryption scheme $SC_{tk}$. Using $\mathcal{A}$ as a building block, we then show how to construct an algorithm $\mathcal{S}$ which breaks sUF-CMA security of the signature scheme $S$, thereby proving the theorem by contradiction. $\mathcal{S}$ plays its own sUF-CMA game regarding the signature scheme $S$ and is defined as follows.

**Setup.** Given $(prm_{sig}, pk_S)$, $\mathcal{S}$ runs $prm_{tk} \leftarrow \mathsf{TSetup}(1^\kappa)$, sets $prm \leftarrow (prm_{tk}, prm_{sig})$ and sets $\mathsf{tag} \leftarrow pk_S$. Then $\mathcal{S}$ runs $\mathcal{A}$ with input $(prm, pk_S)$.

**Query.** When $\mathcal{A}$ issues a signcryption query of the form $(pk_R, m)$, $\mathcal{S}$ first computes $(c_1, K) \leftarrow \mathsf{TEncap}(prm_{tk}, pk_R, \mathsf{tag})$. $\mathcal{S}$ then issues $(m||c_1||pk_R)$ to $\mathcal{CH}$ as a signing query and obtains $\sigma$. Then $\mathcal{S}$ computes $c_2 \leftarrow \mathsf{DEnc}(K, (m||\sigma))$, and finally returns $(c_1, c_2)$ to $\mathcal{A}$ as an answer to the signcryption query.

**Output.** When $\mathcal{A}$ terminates with output a receiver key pair $(pk_R^*, sk_R^*)$ and a ciphertext $c^* = (c_1^*, c_2^*)$, $\mathcal{S}$ firstly computes $K^* \leftarrow \mathsf{TDecap}(prm, sk_R^*, \mathsf{tag}, c_1^*)$ and $(m^*||\sigma^*) \leftarrow \mathsf{DDec}(K^*, \sigma^*)$, and then outputs $((m^*||c_1^*||pk_R^*), \sigma^*)$ as its own forgery and terminates. (If either $\mathsf{TDecap}$ or $\mathsf{DDec}$ returns $\perp$, then $\mathcal{A}$ simply gives up and aborts.)

It is straightforward to see that $\mathcal{S}$'s simulation for $\mathcal{A}$ is perfect. Specifically, the parameters $prm = (prm_{tk}, prm_{sig})$ and the key $pk_S$ given to $\mathcal{A}$ are distributed identically to those given to $\mathcal{A}$ in the dM-sUF-iCMA game. Furthermore, $\mathcal{A}$'s signcryption queries are properly answered by using signing queries.

Hence, in order to complete the proof, all we need to show is that $\mathcal{S}$'s forgery is valid whenever $\mathcal{A}$'s forgery is valid. Let $q$ be the number of $\mathcal{A}$'s signcryption queries. For $i \in \{1, ..., q\}$, let $(pk_R^{(i)}, m^{(i)})$ be $\mathcal{A}$'s $i$-th signcryption query, let $(c_1^{(i)}, c_2^{(i)})$ be $\mathcal{S}$'s response to $\mathcal{A}$'s $i$-th query, and let $K^{(i)}$ be the symmetric key which is the output value of $\mathsf{TEncap}(prm_{tk}, pk_R^{(i)}, \mathsf{tag})$ used to compute $c_2^{(i)}$, where $\mathsf{tag} = pk_S$. If $\mathcal{A}$ succeeds in the dM-sUF-CMA game regarding $SC_{tk}$, it must hold that

$$\begin{aligned}
\mathsf{TDecap}(prm_{tk}, sk_R^*, \mathsf{tag}, c_1^*) &= K^* \neq \bot \\
\mathsf{DDec}(K^*, c_2^*) &= (m^* || \sigma^*) \neq \bot \\
\mathsf{SVer}(prm_{sig}, pk_S, (m^* || c_1^* || pk_R^*), \sigma^*) &= \top \\
{}^{\forall} i \in \{1, ..., q\} : \qquad (pk_R^*, m^*, c_1^*, c_2^*) &\neq (pk_R^{(i)}, m^{(i)}, c_1^{(i)}, c_2^{(i)})
\end{aligned}$$

where $\mathsf{tag} = pk_S$. Under the above conditions, in order for $\mathcal{S}$ to succeed in the sUF-CMA game regarding $S$, it must hold that there exists no $i \in \{1, ..., q\}$ such that $((m^* || c_1^* || pk_R^*), \sigma^*) = ((m^{(i)} || c_1^{(i)} || pk_R^{(i)}), \sigma^{(i)})$.

Now assume towards a contradiction that such $i$ exists. Then, due to the correctness of $TK$ and the fact that in this simulation $\mathcal{S}$ always uses the same tag $\mathsf{tag} = pk_S$, $pk_R^* = pk_R^{(i)}$ and $c_1^* = c_1^{(i)}$ implies $K^* = K^{(i)}$. Next, due to the one-to-one property of $D$, $(K^*, (m^* || \sigma^*)) = (K^{(i)}, (m^{(i)} || \sigma^{(i)}))$ implies $c_2^* = c_2^{(i)}$. Putting everything together, for this $i$ we have $(pk_R^*, m^*, c_1^*, c_2^*) = (pk_R^{(i)}, m^{(i)}, c_1^{(i)}, c_2^{(i)})$. But this contradicts the fourth succeeding condition of $\mathcal{A}$ above, and thus such $i$ never exists.

Therefore, $\mathcal{S}$ succeeds in the sUF-CMA game regarding $S$ whenever $\mathcal{A}$ succeeds, and we have $\mathsf{Adv}_{S,\mathcal{S}}^{\text{sUF-CMA}} = \mathsf{Adv}_{SC_{tk},\mathcal{A}}^{\text{dM-sUF-iCMA}}$ is non-neglibible, which contradicts that $S$ is sUF-CMA secure. Hence, $\mathsf{Adv}_{SC_{tk},\mathcal{A}}^{\text{dM-sUF-iCMA}}$ must be negligible for any PPT adversary $\mathcal{A}$. This completes the proof of Theorem 2. □

## 4.2 Composition Using KEM

Let $KM$ = (KSetup, KKG, Encap, Decap) be an IND-CCA secure KEM, let $D$ = (DEnc, DDec) be an IND-CCA secure DEM, let $M$ = (Mac, MVer) be a sUF-OT secure MAC, and let $S$ = (SSetup, SKG, Sign, SVer) be a sUF-CMA secure signature scheme (the definitions of a KEM and a MAC are given in Appendix A). Then, we construct a signcryption scheme $SC_{kem}$ as shown in Fig.2. We assume symmetric key space of $KM$ is $\{0,1\}^{2\kappa}$ and that of $D$ and $M$ are $\{0,1\}^{\kappa}$. (We can always achieve this by using an appropriate key derivation function and/or a pseudorandom generator.)

*Construction Idea.* The basic idea of $SC_{kem}$ is almost the same as that of $SC_{tk}$. In $SC_{kem}$, we use a KEM and a MAC as building blocks, instead of a TBKEM. As

```
$\mathsf{Setup}(1^\kappa)$ :
    $prm_{kem} \leftarrow \mathsf{KSetup}(1^\kappa)$
    $prm_{sig} \leftarrow \mathsf{SSetup}(1^\kappa)$
    Output $prm \leftarrow (prm_{kem}, prm_{sig})$.

$\mathsf{KeyGen_S}(prm)$ :
    Output $(pk_S, sk_S) \leftarrow \mathsf{SKG}(prm_{sig})$.

$\mathsf{KeyGen_R}(prm)$ :
    Output $(pk_R, sk_R) \leftarrow \mathsf{KKG}(prm_{kem})$.

$\mathsf{SC}(prm, sk_S, pk_R, m)$ :
    $(c_1, K) \leftarrow \mathsf{Encap}(prm_{kem}, pk_R)$
    $(K_m || K_a) \leftarrow K$
    $\sigma \leftarrow \mathsf{Sign}(prm_{sig}, sk_S, (m||c_1||pk_R))$
    $c_2 \leftarrow \mathsf{DEnc}(K_m, (m||\sigma))$
    $\tau \leftarrow \mathsf{Mac}(K_a, (pk_S||c_1||c_2))$
    Output $c \leftarrow (c_1, c_2, \tau)$

$\mathsf{USC}(prm, pk_S, sk_R, c)$ :
    Parse $c$ as $(c_1, c_2, \tau)$
    $K \leftarrow \mathsf{Decap}(prm_{kem}, sk_R, c_1)$
        (if $K = \perp$, then return $\perp$)
    $(K_m || K_a) \leftarrow K$
    If $\mathsf{MVer}(K_a, (pk_S||c_1||c_2), \tau) = \perp$
        then return $\perp$
    $(m||\sigma) \leftarrow \mathsf{DDec}(K_m, c_2)$
        (if $\mathsf{DDec}$ outputs $\perp$, then return $\perp$)
    If $\mathsf{SVer}(prm_{sig}, pk_S, (m||c_1||pk_R), \sigma) = \perp$
        then return $\perp$
    Output $m$
```

**Fig. 2.** Construction using KEM: $SC_{kem}$

mentioned earlier, we can construct an IND-tag-CCA secure TBKEM using an IND-CCA secure KEM and a one-time secure MAC [2], and we can also construct an IND-CCA secure DEM using a one-time secure DEM and a sUF-OT secure MAC [7]. However, we find that if we use an IND-tag-CCA secure TBKEM and an IND-CCA secure DEM constructed as above in the first construction $SC_{tk}$, the one-time secure MAC can be shared and thus can be slightly more efficient. $SC_{kem}$ is constructed based on this observation.

*Security.* The security of $SC_{kem}$ is formally guaranteed by the following theorems. The proofs of these theorems are similar to the proofs of Theorems 1 and 2, and will not be given here.

**Theorem 3.** *If the KEM KM is IND-CCA secure, the DEM D is IND-OT secure and MAC M is sUF-OT secure, then the proposed signcryption scheme $SC_{kem}$ is dM-IND-iCCA secure.*

**Theorem 4.** *If the signature scheme S is sUF-CMA secure, the DEM D is one-to-one and MAC M is one-to-one, then the proposed signcryption scheme $SC_{kem}$ is dM-sUF-iCMA secure.*

## 5 Comparison

In Fig. 3, we compare concrete instantiations of standard model signcryption schemes. In the figure, tBMW1 denotes the TBKEM (and the TBE scheme) which is obtained from the PKE scheme by Boyen, Mei, and Waters [11] based on the observation in [21, Sect. 7.2], and BMW2 denotes the KEM by Boyen, Mei, and Waters in [12, Sect. 4]. MMS-StTE(X, Y) denotes "Sign-then-Tag-based-Encrypt" schemes [21, Sect. 5] where the TBE scheme X and the signature

| Scheme | Confidentiality /Assumption | Unforgeability /Assumption | Comp. Cost SC / USC | Ciphertext OH Elements/Bits |
|---|---|---|---|---|
| Tan [23] | dM-IND-iCCA /DBDH | dM-sUF-iCMA(KR) /$q$-SDH | [3,2;0] / [3,1;4] | $3\|\mathbb{G}_p\| + 2\|\mathbb{Z}_p\|$ / 800 |
| MMS-StTE (tBMW1,BB[8]) | dM-IND-iCCA /DBDH | dM-wUF-iCMA /$q$-SDH | [4,0;0] + 1W / [1,1;2] | $3\|\mathbb{G}_p\| + \|\mathbb{Z}_p\|$ / 640 |
| MMS-SC (tBMW1,Waters[24]) | dM-IND-iCCA /DBDH | dM-wUF-iCMA(KR) /co-CDH | [4,0;0] / [1,0;3] + 1W | $3\|\mathbb{G}_p\|$ / 480 |
| MMS-SC (tBMW1,BSW[10]) | dM-IND-iCCA /DBDH | dM-sUF-iCMA(KR) /co-CDH | [4,1;0] / [1,1;3] + 1W | $3\|\mathbb{G}_p\| + \|\mathbb{Z}_p\|$ / 640 |
| **Ours:** $SC_{tk}$ (tBMW1,BB[8]) | dM-IND-iCCA /DBDH | dM-sUF-iCMA /$q$-SDH | [4,0;0] + 1W / [1,1;2] | $3\|\mathbb{G}_p\| + \|\mathbb{Z}_p\|$ / 640 |
| **Ours:** $SC_{kem}$ (BMW2,BB[8]) | dM-IND-iCCA /DBDH | dM-sUF-iCMA /$q$-SDH | [3,1;0] / [1,1;2] | $3\|\mathbb{G}_p\| + \|\mathbb{Z}_p\|$ $+\|\mathtt{MAC}\|$ / 720 |

**Fig. 3.** Comparison of existing and proposed signcryption schemes with insider security in the dynamic multi-user model. Columns "Confidentiality" and "Unforgeability" list the achieved security notions as well as the underlying security assumptions. In these columns, the security notions followed with (KR) are security in the key registered model. Column "Comp. Cost" lists the computational overhead for signcryption (SC) and unsigncryption (USC), where $[a, b; c]$ denotes $a$ exponentiations, $b$ multi-exponentiations and $c$ pairing computations, and W denotes computation of the so called Waters hash [24] (other multiplications, computation costs of hash functions and symmetric key primitives are ignored). In the overhead column, $|\mathbb{G}|$ denotes the size of a group element of an elliptic curve equipped with an asymmetric pairing, and $|\mathbb{Z}_p|$ denotes the size of an exponent (i.e. the order of the group), and $|\mathrm{MAC}|$ denotes the size of a (sUF-OT secure) MAC tag. When instantiated to achieve 80 bits of security and minimize ciphertext overhead, listed in the last column, we set $|\mathbb{G}| = |\mathbb{Z}_p| = 160$ bits, and $|\mathtt{MAC}| = 80$ bits. We assume that a DEM has no ciphertext overhead [22] (i.e. length preserving), regardless of whether it is IND-OT or IND-CCA secure. "Ciphertext OH" means the difference between the ciphertext size and the plaintext size.

scheme Y are used as concrete building blocks. MMS-SC(X, Y) denotes a signcryption scheme constructed from "signcryption composable" TBKEM X and signature scheme Y [21, Sect. 6]. $SC_{tk}$(X, Y) (resp. $SC_{kem}$(X, Y)) denotes the signcryption scheme constructed from $SC_{tk}$ (resp. $SC_{kem}$) in Section 4 where the TBKEM (resp. the KEM) X and a signature scheme Y are used as concrete building blocks.

As shown in Fig. 3, our schemes are the only ones which achieve dM-IND-iCCA security and dM-sUF-iCMA security simultaneously without random oracles or key registration. $SC_{tk}$(tBMW1, BB) has better efficiency in all aspects than Tan and MMS-StTE(tBMW1,BB). Furthermore, we see that there exists a tradeoff in the achieved security/assumption, computational costs, and ciphertext size (overhead) among $SC_{tk}$(tBMW1, BB), MMS-SC(tBMW1, Waters), and MMS-CS(tBMW1, BSW). However, we would like to stress that if we admit the $q$-SDH assumption and if we need strong unforgeability, then $SC_{tk}$(tBMW1, BB) is the best choice over other schemes. We remark that $SC_{kem}$ and Tan have constant size user (i.e. sender and receiver) public keys, and $SC_{kem}$ is better in all aspects than Tan.

Although we have chosen tBMW1, BMW2, and BB as concrete building blocks for comparison in Fig. 3 we would like to stress that our proposed constructions $SC_{tk}$ and $SC_{kem}$ are generic constructions (which is also true in the constructions in [21]), and thus a number of different combinations of concrete instantiations are possible. For example, if one wants to construct an insider secure signcryption scheme from factoring-based assumptions rather than discrete-logarithm-based assumptions, one can use the recent KEM (and its TBKEM-analogue that can be obtained through the observation in [21]) by Hofheinz and Kiltz [14] whose security is proved from the factoring assumption and the recent signature scheme by Hohenberger and Waters [15] whose security is proved from the RSA assumption. Alternatively, one can also instantiate insider secure signcryption schemes based on lattice-based assumptions from recent progress regarding identity-based encryption and signature schemes in this area, such as the schemes from [3]. (Note that we can always construct CCA secure KEMs from identity-based encryption schemes due to [9])

**Acknowledgement**

The authors would like to thank the reviewers for their useful comments. Takahiro Matsuda and Jacob Schuldt are supported by a JSPS Fellowship for Young Scientists.

## References

1. Abe, M., Cui, Y., Imai, H., Kiltz, E.: Efficient hybrid encryption from ID-based encryption. Designs, Codes and Cryptography 54(3), 205–240 (2010)
2. Abe, M., Gennaro, R., Kurosawa, K.: Tag-KEM/DEM: A new framework for hybrid encryption. J. of Cryptology 21(1), 97–130 (2008)
3. Agrawal, S., Boneh, D., Boyen, X.: Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE. In: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6223, pp. 98–115. Springer, Heidelberg (2010)
4. An, J., 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)
5. Baek, J., Steinfeld, R., Zheng, Y.: Formal proofs for the security of signcryption. In: Naccache, D., Paillier, P. (eds.) PKC 2002. LNCS, vol. 2274, pp. 80–98. Springer, Heidelberg (2002)
6. Baek, J., Steinfeld, R., Zheng, Y.: Formal proofs for the security of signcryption. J. of Cryptology 20(2), 203–235 (2007)
7. 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)
8. Boneh, D., Boyen, X.: Secure identity based encryption without random oracles. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 443–459. Springer, Heidelberg (2004)

9. Boneh, D., Canetti, R., Halevi, S., Katz, J.: Chosen-ciphertext security from identity-based encryption. SIAM J. Computing 36(5), 1301–1328 (2007)
10. Boneh, D., Shen, E., Waters, B.: Strongly unforgeable signatures based on computational diffie-hellman. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T. (eds.) PKC 2006. LNCS, vol. 3958, pp. 229–240. Springer, Heidelberg (2006)
11. Boyen, X., Mei, Q., Waters, B.: Direct chosen ciphertext security from identity-based techniques. In: Proc. of CCS 2005, pp. 320–329. ACM, New York (2005)
12. Boyen, X., Mei, Q., Waters, B.: Direct chosen ciphertext security from identity-based techniques, Updated version of [11]. Cryptology ePrint Archive: Report 2005/288 (2005), `http://eprint.iacr.org/2005/288/`
13. Dent, A.: Hybrid signcryption schemes with outsider security (extended abstract). In: Zhou, J., López, J., Deng, R.H., Bao, F. (eds.) ISC 2005. LNCS, vol. 3650, pp. 203–217. Springer, Heidelberg (2005)
14. Hofheinz, D., Kiltz, E.: Practical chosen ciphertext secure encryption from factoring. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 313–332. Springer, Heidelberg (2009)
15. Hohenberger, S., Waters, B.: Short and stateless signatures from the RSA assumption. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 654–670. Springer, Heidelberg (2009)
16. Kiltz, E.: Chosen-ciphertext security from tag-based encryption. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 581–600. Springer, Heidelberg (2006)
17. Li, C., Yang, G., Wang, D., Deng, X., Chow, S.: An efficient signcryption scheme with key privacy. In: López, J., Samarati, P., Ferrer, J.L. (eds.) EuroPKI 2007. LNCS, vol. 4582, pp. 78–93. Springer, Heidelberg (2007)
18. Libert, B., Quisquater, J.-J.: Improved signcryption with key privacy from gap Diffie-Hellman groups. In: Bao, F., Deng, R., Zhou, J. (eds.) PKC 2004. LNCS, vol. 2947, pp. 187–200. Springer, Heidelberg (2004)
19. Libert, B., Quisquater, J.-J.: Improved signcryption with key privacy from gap Diffie-Hellman groups, Updated version of [18] (2004), `http://www.dice.usl.ac.be/~libert/`
20. MacKenzie, P.D., Reiter, M.K., Yang, K.: Alternatives to non-malleability: Definitions, constructions, and applications. In: Naor, M. (ed.) TCC 2004. LNCS, vol. 2951, pp. 171–190. Springer, Heidelberg (2004)
21. Matsuda, T., Matsuura, K., Schuldt, J.: Efficient constructions of signcryption schemes and signcryption composability. In: Roy, B., Sendrier, N. (eds.) INDOCRYPT 2009. LNCS, vol. 5922, pp. 321–342. Springer, Heidelberg (2009)
22. Phan, D., Pointcheval, D.: About the security of ciphers (semantic security and pseudo-random permutations). In: Handschuh, H., Hasan, M.A. (eds.) SAC 2004. LNCS, vol. 3357, pp. 182–197. Springer, Heidelberg (2004)
23. Tan, C.: Signcryption scheme in multi-user setting without random oracles. In: Matsuura, K., Fujisaki, E. (eds.) IWSEC 2008. LNCS, vol. 5312, pp. 64–82. Springer, Heidelberg (2008)
24. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005)
25. Zheng, Y.: Digital signcryption or how to achieve cost (Signature & encryption) ¡¡ = cost(Signature) + cost(Encryption). In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 165–179. Springer, Heidelberg (1997)

## A Additional Definitions

### A.1 Key Encapsulation Mechanism

A key encapsulation mechanism (KEM) is a special case of a TBKEM in which the tag is always given by an empty string i.e. the encapsulation and decapsulation algorithms do not support the additional input $\mathsf{tag}$. We use the notation $\mathsf{KSetup}$, $\mathsf{KKG}$, $\mathsf{Encap}$, and $\mathsf{Decap}$ to denote the setup, key generation, encapsulation and decapsulation algorithms, respectively. Like for a TBKEM, it is required for all $prm \leftarrow \mathsf{KSetup}(1^\kappa)$, all $(pk, sk) \leftarrow \mathsf{KKG}(prm)$, and all $(c, K) \leftarrow \mathsf{Encap}(prm, pk)$, that $K = \mathsf{Decap}(prm, sk, c)$. Lastly, IND-CCA security of a KEM is defined almost as IND-tag-CCA security of a TBKEM, except that decapsulation queries do not contain a tag, and the adversary is given a challenge encapsulation and key pair without choosing a challenge tag.

### A.2 Message Authentication Code

A message authentication code (MAC) is given by the following two algorithms $(\mathsf{Mac}, \mathsf{MVer})$: $\mathsf{Mac}$, which is given input a symmetric key $K \in \mathcal{K}$ and a message $m$, where $\mathcal{K}$ is a key space, returns a message authentication tag $\tau$; and $\mathsf{MVer}$, which given input a symmetric key $K \in \mathcal{K}$, a message $m$, and a message authentication tag $\tau$, returns $\top$ or $\bot$.

It is required for all $K \in \mathcal{K}$ and all $m$ that $\mathsf{MVer}(K, m, \mathsf{Mac}(K, m)) = \top$.

*sUF-OT security.* For a MAC, strong unforgeability against one time attacks (sUF-OT) is defined by the following game between an adversary $\mathcal{A}$ and a sUF-OT challenger $\mathcal{CH}$.

**Setup.** $\mathcal{CH}$ chooses a key $K \in \mathcal{K}$ at random.
**Query.** $\mathcal{A}$ can submit a single MAC query $m$ to $\mathcal{CH}$. $\mathcal{CH}$ responds to this query by returning $\tau \leftarrow \mathsf{Mac}(K, m)$.
**Output.** $\mathcal{A}$ outputs a message/tag pair $(m^*, \tau^*)$.

We define the sUF-OT advantage of $\mathcal{A}$ attacking the MAC $M$ as follows:

$$\mathsf{Adv}_{M,\mathcal{A}}^{\text{sUF-OT}} = \Pr[\mathsf{MVer}(K, m^*, \tau^*) = \top \wedge (m^*, \tau^*) \neq (m, \tau)]$$

**Definition 6.** *We say that a MAC $M$ is sUF-OT secure if* $\mathsf{Adv}_{M,\mathcal{A}}^{\text{sUF-OT}}$ *is negligible for any PPT adversary $\mathcal{A}$.*

*One-to-One Property.* A MAC is said to be *one-to-one* if given a key $K \in \mathcal{K}$ and a message $m$, there is only one message authentication tag $\tau$ such that $\mathsf{MVer}(K, m, \tau) = \top$. It is satisfied by many MAC schemes such as the CMAC, whose $\mathsf{Mac}$ algorithm is deterministic and the $\mathsf{MVer}$ algorithm re-computes the MAC tag from a key and given message, and compares it with the given tag.