





An Efficient Generic Insider Secure Signcryption with Non-Interactive Non-Repudiation

Ngarenon Togde  and Augustin P. Sarr 

Laboratoire ACCA, UFR SAT, Université Gaston Berger, Saint-Louis, Senegal
{ngarenon.togde, augustin-pathe.sarr}@ugb.edu.sn

Abstract. We present a generic construction of an insider secure signcryption scheme with non-interactive non-repudiation. Our construction uses as building blocks a signature scheme, a key encapsulation mechanism (KEM), a keyed hash function, a symmetric encryption scheme, and a pseudo-random function. We show that our construction is insider secure in the dynamic multi-user model, without resorting the random oracle or the key registration model. Our generic scheme provides also non-interactive non-repudiation.

Keywords: generic signcryption · insider security · dynamic multi-user model · non-interactive non-repudiation

1 Introduction

Signcryption schemes provide both the functionalities of signature and encryption schemes. These schemes were proposed for the first time by Zheng [24]. Since Zheng’s seminal work, many designs have been proposed, e.g. [2, 5, 7, 8, 10, 12, 18–22]. For the analysis of signcryption schemes, two important lines of separations in the security definitions are: two-party versus multi-party models, and outsider versus insider security models [1, 3, 4]. Broadly, in a two-party security model, only one sender and one receiver are considered. Whereas in a multi-party model, an attacker can use any public key of its choice. In an outsider model, it is assumed that an attacker cannot access a legitimate sender or receiver long-term secret. In an insider model, an attacker has access to all the secrets except the one “being attacked”; for confidentiality, it is assumed that the attacker knows the sender’s static private key, and for unforgeability that the attacker knows the receiver’s static private key. The strongest among these models is insider security in the (dynamic) multi-user model.

Some “natural” constructions of signcryption schemes are “encrypt and sign (E&S), “Encrypt then Sign” (EtS) and Sign then Encrypt (StE). Unfortunately, these natural constructions do not yield secure signcryption schemes in the dynamic multi-user insider model [1, Sect. 2.3]. For instance, In an E&S construction, the signature may reveal the encrypted message, confidentiality is not then achieved. In the EtS and StE constructions the difficulty is to maintain the security of the operation performed first. For instance in the EtS construction, for confidentiality, an attacker (a probabilistic polynomial time machine) which knows the sender’s static private key can resign and submit the resigned signcrypted text to a decryption oracle. In the StE construction, for

unforgeability, an adversary which knows the receiver static private key can decrypt the ciphertext and re-encrypt and submit the resulting signcrypted text as a forgery.

A nice property of signcryption schemes is Non-Interactive Non-Repudiation (NINR), which allows a third party to settle a non-repudiation dispute without engaging a costly protocol. NINR is a main advantage of signcryption schemes compared to one pass key exchange protocols, which often outperform signcryption schemes.

Building high-level secure and efficient cryptographic schemes from low-level primitives is a main focus in modern cryptography. In the case of signcryption schemes, insider security appears to be the right security definition [3]. As far as we are aware, there are only three works, that aim to propose generic insider secure constructions of signcryption schemes in the dynamic multi-user model, [10, 19] and [2]. Unfortunately the designs from [19] and [2] are shown to be secure in the registered key model, wherein an attacker has to show that it knows the private keys corresponding to the public keys it uses. This model does not capture some realistic attacks on certificate authorities, e.g. [11, 13]. In [10], Chiba *et al.* propose two generic StE type constructions that they show to be insider secure in the dynamic multi-user model, without resorting the random oracle or registered key model. As their constructions are StE, they inherit NINR from the base signature scheme.

In this work, we build a simple and efficient generic EtS signcryption scheme with NINR (SCNINR), termed SN (Signcryption with *Non*-interactive non-repudiation). We propose a detailed analysis of our construction, in the insider dynamic multi-user model, without using the random oracle or registered key model.

This paper is organized as follows. In Sect. 2, we present some preliminaries on signcryption schemes and on the building blocks we use in our design. In Sect. 3, we propose our generic SCNINR scheme. In Sect. 4, we propose a detailed security analysis of our construction in the dynamic multi-user model. We compare our design with the previous proposals in Sect. 5.

2 Preliminaries

If S is a set, $a \leftarrow_{\mathbf{r}} S$ means that a is chosen uniformly at random from S ; we write $a, b, c, \dots \leftarrow_{\mathbf{r}} S$ as a shorthand for $a \leftarrow_{\mathbf{r}} S; b \leftarrow_{\mathbf{r}} S$, etc. We denote by $\text{sz}(a)$ the number of bits required to represent a . If S and S' are two sets, $\text{Func}(S, S')$ denotes the set of functions with domain S and range S' .

For a probabilistic algorithm \mathcal{A} with parameters u_1, \dots, u_n and output $V \in \mathbf{V}$, we write $V \leftarrow_{\mathbf{r}} \mathcal{A}(u_1, \dots, u_n)$. We denote by $\{\mathcal{A}(u_1, \dots, u_n)\}$ the set $\{v \in \mathbf{V} : \Pr(V = v) \neq 0\}$. If x_1, x_2, \dots, x_k are objects belonging to different structures (group, bit-string, etc.) (x_1, x_2, \dots, x_k) denotes a representation as a bit-string of the tuple such that each element can be unequivocally parsed. For a list L , $\text{Apd}(L, X)$ adds X to L . For a positive integer n , $[n]$ denotes the set $\{1, 2, \dots, n\}$.

A Symmetric Encryption. A symmetric encryption scheme $\mathcal{E} = (\text{E}, \text{D}, \mathbf{K}(k), \mathbf{M}(k), \mathbf{C}(k))$ is a pair of efficient algorithms (E, D) , an encryption and a decryption algorithm, together with a triple of sets $(\mathbf{K}, \mathbf{M}, \mathbf{C})$, which depend on a security parameter k , such that for all $\tau \in \mathbf{K}$ and all $m \in \mathbf{M}$, it holds that $\text{E}(\tau, m) \in \mathbf{C}$ and $m = \text{D}(\tau, \text{E}(\tau, m))$.

Definition 1. Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an adversary against \mathcal{E} and let

$$\Pr(O_{i,i=0,1}) = \Pr \left[\begin{array}{l} (m_0, m_1, st) \leftarrow_R \mathcal{A}_1(k); \tau \leftarrow_R \mathbf{K}; c \leftarrow_R \mathbf{E}(\tau, m_i); \\ \hat{b} \leftarrow_R \mathcal{A}_2(k, c, st) \end{array} : \hat{b} = 1 \right]$$

and $\text{Adv}_{\mathcal{A}, \mathcal{E}}^{\text{SS}}(k)$ denote the quantity

$$\text{Adv}_{\mathcal{A}, \mathcal{E}}^{\text{SS}}(k) = |\Pr(O_0) - \Pr(O_1)|,$$

where $m_0, m_1 \in \mathbf{M}$ are distinct messages of equal length. The scheme \mathcal{E} is said to be $(t(k), \varepsilon(k))$ -semantically secure if for all adversaries \mathcal{A} running in time $t(k)$, it holds that $\text{Adv}_{\mathcal{A}, \mathcal{E}}^{\text{SS}}(k) \leq \varepsilon(k)$.

We will need also the following definition.

Definition 2. Let $\mathcal{E} = (\mathbf{E}, \mathbf{D}, \mathbf{K}(k), \mathbf{M}(k), \mathbf{C}(k))$ be an encryption scheme. The scheme \mathcal{E} is said to be $(t(k), \varepsilon(k))$ -secure against key clustering attacks if for all adversaries \mathcal{A} running in time $\leq t(k)$,

$$\Pr[(m, \tau, \tau') \leftarrow_R \mathcal{A}(k) : \tau \neq \tau' \text{ and } \mathbf{E}(\tau, m) = \mathbf{E}(\tau', m)] \leq \varepsilon(k).$$

Pseudo-Random Function (PRF). A PRF is a deterministic algorithm Prf together with a triple of sets $(\mathbf{K}(k), \mathbf{D}(k), \mathbf{R}(k))$ (which depends on the security parameter k) such that for all $\tau \in \mathbf{K}$ and all $m \in \mathbf{D}$, $\text{Prf}(\tau, m) \in \mathbf{R}$. Notice that for all fixed $\tau \in \mathbf{K}$, $\text{Prf}(\tau, \cdot) \in \text{Func}(\mathbf{D}, \mathbf{R})$.

Definition 3. Let Prf be a pseudo-random function and \mathcal{A} be an adversary,

$$\Pr(O_0) = \Pr \left[\tau \leftarrow_R \mathbf{K}; f \leftarrow \text{Prf}(\tau, \cdot); \hat{b} \leftarrow_R \mathcal{A}^{\mathcal{O}_f(\cdot)}(k) : \hat{b} = 1 \right],$$

$$\Pr(O_1) = \Pr \left[f \leftarrow_R \text{Func}(\mathbf{D}, \mathbf{R}); \hat{b} \leftarrow_R \mathcal{A}^{\mathcal{O}_f(\cdot)}(k) : \hat{b} = 1 \right],$$

and

$$\text{Adv}_{\mathcal{A}, \text{Prf}}(k) = |\Pr(O_0) - \Pr(O_1)|.$$

The PRF Prf is said to be $(t(k), \varepsilon(k))$ -secure if for all efficient adversaries \mathcal{A} running in time $\leq t$, it holds that $\text{Adv}_{\mathcal{A}, \text{Prf}}(k) \leq \varepsilon(k)$.

Collision Resistant Hash Function. Let $\mathbf{K}(k)$, $\mathbf{M}'(k)$ and $\mathbf{T}(k)$ be sets which depend on a security parameter k and \mathbf{H} be a keyed hash function defined over $(\mathbf{K}, \mathbf{M}', \mathbf{T})$, i. e. \mathbf{H} takes as inputs $\tau_0 \in \mathbf{K}$ and $m \in \mathbf{M}'$ and outputs $t \in \mathbf{T}$; we write $t \leftarrow \mathbf{H}(\tau_0, m)$.

Definition 4. A keyed hash function $\mathbf{H} : \mathbf{K} \times \mathbf{M}' \rightarrow \mathbf{T}$ is said to be $(t(k), \varepsilon(k))$ collision resistant if for all efficient adversaries \mathcal{A} running in time $\leq t(k)$,

$$\Pr[\tau_0 \leftarrow_R \mathbf{K}; (m_0, m_1) \leftarrow_R \mathcal{A}(k, \tau_0) : m_0 \neq m_1 \wedge \mathbf{H}(\tau_0, m_0) = \mathbf{H}(\tau_0, m_1)] \leq \varepsilon(k).$$

Definition 5. Let $H : \mathbf{K} \times \mathbf{M}' \rightarrow \mathbf{T}$ be a keyed hash function and Pfx be a subset of $\{0, 1\}^*$. H is said to be $(t(k), \varepsilon(k))$ resistant to collisions with identical prefix from Pfx , if for all efficient adversaries \mathcal{A} running in time $\leq t(k)$,

$$\Pr \left[\tau_0 \leftarrow_R \mathbf{K}; (p, m_0, m_1) \leftarrow_R \mathcal{A}(k, \tau_0) : \begin{cases} p \in \text{Pfx}, \\ m_0 \neq m_1 \text{ and} \\ H(\tau_0, (p, m_0)) = H(\tau_0, (p, m_1)) \end{cases} \right] \leq \varepsilon(k).$$

Notice that resistance to collisions with identical prefix may be a weaker assumption than classical collision resistance. We consider now the following game parameterized by a pseudo-random function Prf .

Game 1 Pre-image with chosen prefix and suffix

- 1) The challenger Chall chooses $\tau_0 \leftarrow_R \mathbf{K}$ and sends τ_0 to \mathcal{A} .
 - 2) \mathcal{A} chooses $p_0 \in \text{Pfx}$, $s_0 \in \text{Sfx}$ and $m_0 \in \mathbf{M}$ and sends (p_0, m_0, s_0) to Chall .
 - 3) Chall chooses $(\tau, \tau') \leftarrow_R \mathbf{K}^2$, computes $\tau'' \leftarrow \text{Prf}(\tau, m_0)$ and $\hat{m}_0 \leftarrow H(\tau_0, (p_0, \tau, \tau', \tau'', s_0))$, and sends \hat{m}_0 to \mathcal{A} .
 - 4) \mathcal{A} outputs $(\tau^*, \tau'^*) \in \mathbf{K}^2$.
 - 5) \mathcal{A} succeeds if $\hat{m}_0 = H(\tau_0, (p_0, \tau^*, \tau'^*, \tau''^*, s_0))$ wherein $\tau''^* \leftarrow \text{Prf}(\tau^*, m_0)$.
-

Definition 6. Let $H : \mathbf{K} \times \mathbf{M}' \rightarrow \mathbf{T}$ be a keyed hash function and Pfx and Sfx be respectively some sets of message prefixes and suffixes. For an adversary \mathcal{A} playing Game 1, let $\text{Succ}_{\mathcal{A}, H}(k)$ denote the event “ \mathcal{A} wins Game 1”. H is said to be $(t(k), \varepsilon(k))$ secure against pre-image attacks with chosen prefix from Pfx and suffix from Sfx , if for all efficient adversaries \mathcal{A} running in time $\leq t(k)$, $\Pr(\text{Succ}_{\mathcal{A}, H}(k)) \leq \varepsilon(k)$.

The following lemma shows that the pre-image resistance (from Definition 6) follows from identical prefix collision resistance. The proof is given in the appendix.

Lemma 1. Let $H : \mathbf{K} \times \mathbf{M}' \rightarrow \mathbf{T}$ be a keyed hash function. If H is $(t(k), \varepsilon(k))$ secure against collisions with identical prefix from Pfx , then it is $(t(k), \varepsilon'(k))$ secure in the sense of Definition 6, where

$$\varepsilon'(k) \leq |\mathbf{T}|/|\mathbf{K}|^2 + \varepsilon(k).$$

Key Encapsulation Mechanism (KEM). A KEM is a four-tuple of efficient algorithms $\mathcal{K} = (\text{Setup}_{\mathcal{K}}, \text{Gen}_{\mathcal{K}}, \text{Ecp}, \text{Dcp})$ together with a key space $\mathbf{K}'(k)$ and encapsulated keys space \mathbf{C}' , such that:

- $\text{Setup}_{\mathcal{K}}$ is a probabilistic algorithm which takes as input a security parameter k and outputs a domain parameter $dp_{\mathcal{K}}$;
- $\text{Gen}_{\mathcal{K}}$ is a key pair generator, it takes as input the domain parameter $dp_{\mathcal{K}}$ and outputs a key pair $(sk_{\mathcal{K}}, pk_{\mathcal{K}})$;
- Ecp is a probabilistic algorithm which takes as input a public key $pk_{\mathcal{K}}$ and outputs a key $\tau \in \mathbf{K}'$ together with an encapsulated key $c \in \mathbf{C}'$, we write $(\tau, c) \leftarrow_R \text{Ecp}(pk_{\mathcal{K}})$;
- Dcp takes as inputs a private key $sk_{\mathcal{K}}$ together with an encapsulated key c and outputs $\tau \in \mathbf{K}'$ or an error symbol \perp .

It is required that for all $k \in \mathbb{N}^*$, all $dp_K \in \{\text{Setup}_K(k)\}$, all $(sk_K, pk_K) \in \{\text{Gen}_K(dp_K)\}$, if $(\tau, c) \in \{\text{Ecp}(pk_K)\}$, $\Pr[\text{Dcp}(sk_K, c) = \tau] = 1$.

Definition 7. Let \mathcal{K} be a KEM, and \mathcal{A} an adversary against \mathcal{K} . Let

$$\Pr(U_{b,b=0,1}) = \Pr \left[\begin{array}{l} dp_K \leftarrow_R \text{Setup}_K(k); (sk_K, pk_K) \leftarrow_R \text{Gen}_K(dp_K); \\ (\tau_0, c) \leftarrow_R \text{Ecp}(pk_K); \tau_1 \leftarrow_R \mathbf{K}'; \\ \hat{b} \leftarrow_R \mathcal{A}^{\mathcal{O}_{\text{Dcp}}(sk_K, \cdot)}(k, dp_K, pk_K, \tau_b, c) \end{array} : \hat{b} = 1 \right] \quad (1)$$

wherein the notation $\mathcal{A}^{\mathcal{O}_{\text{Dcp}}(sk_K, \cdot)}$ means that \mathcal{A} is given access to a decapsulation oracle $\mathcal{O}_{\text{Dcp}}(sk_K, \cdot)$ which, on input $c' \neq c$, outputs $\text{Dcp}(sk_K, c')$ (\mathcal{A} is not allowed to issue $\text{Dcp}(sk_K, c)$). Let $\text{Adv}_{\mathcal{A}, \mathcal{K}}^{\text{cca}}(k) = |\Pr(U_0) - \Pr(U_1)|$. \mathcal{K} is said to be $(t(k), \varepsilon(k))$ indistinguishable against chosen-ciphertext attacks (IND-CCA), if for all efficient adversaries \mathcal{A} running in time $\leq t(k)$, it holds that $\text{Adv}_{\mathcal{A}, \mathcal{K}}^{\text{cca}}(k) \leq \varepsilon(k)$.

Remark 1. In a KEM security experiment, we refer to the challenge (τ_0, c) and (τ_1, c) defined in (1) by $\text{Chall}_{\mathcal{K}_{E_0}}$ and $\text{Chall}_{\mathcal{K}_{E_1}}$, respectively.

Digital Signature. A digital signature scheme is a four-tuple of efficient algorithms $\mathcal{S} = (\text{Setup}_S, \text{Gen}_S, \text{Sign}, \text{Vrfy})$ together with a message space \mathbf{M}_S , such that:

- Setup_S takes as input a security parameter k and outputs a domain parameter dp_S ;
- Gen_S is a probabilistic algorithm which takes as input a domain parameter dp_S and outputs a key pair (sk_S, pk_S) ;
- Sign takes as inputs a secret key sk_S and a message $m \in \mathbf{M}_S$ and outputs a signature $\sigma \in \Sigma$;
- Vrfy is deterministic; it takes as inputs a public key pk_S , a message m , and a signature σ and outputs $d \in \{0, 1\}$; and
- \mathcal{S} is such that for all $k \in \mathbb{N}^*$, all $dp_S \in \{\text{Setup}_S(k)\}$, all $(sk_S, pk_S) \in \{\text{Gen}_S(dp_S)\}$, and all $m \in \mathbf{M}_S$, $\Pr[\text{Vrfy}(pk_S, m, \text{Sign}(sk_S, m)) = 1] = 1$.

Game 2 sUF-CMA security game

- 1) $dp_S \leftarrow_R \text{Setup}_S(k); (sk_S, pk_S) \leftarrow_R \text{Gen}_S(dp_S); \mathbf{L} \leftarrow ()$.
 - 2) For $j = 1, 2, \dots$, \mathcal{A} submits $m_j \in \mathbf{M}_S$ to the challenger which:
 - a) outputs $\sigma_j \leftarrow_R \text{Sign}(sk_S, m_j)$, and
 - b) appends (m_j, σ_j) to \mathbf{L} .
 - 3) \mathcal{A} outputs $(m, \sigma) \in \mathbf{M}_S \times \Sigma$.
 - \mathcal{A} succeeds in sUF-CMA if: i) $\text{Vrfy}(pk_S, m, \sigma) = 1$, and ii) $(m, \sigma) \notin \mathbf{L}$.
-

Definition 8. Let \mathcal{S} be a signature scheme; \mathcal{S} is said to be $(t(k), Q_{\text{Sign}}(k), \varepsilon(k))$ strongly Unforgeable against Chosen Message Attacks if for any adversary \mathcal{A} playing Game 2, if \mathcal{A} runs in time at most $t(k)$ and issues at most $Q_{\text{Sign}}(k)$ queries to the signing oracle, then it succeeds in the sUF-CMA game with probability $\leq \varepsilon(k)$.

Notice that when $\varepsilon(k)$ does not depend on $Q_{\text{Sign}}(k)$, we say simply that \mathcal{S} is $(t(k), \varepsilon(k))$ -secure. We will need also the following notion, which is not captured in the sUF-CMA security definition, although it seems naturally achieved by many usual

signature schemes (which uses a hash function), such as the Full Domain Hash, for instance.

Definition 9. A signature scheme is said to be $(t(k), \varepsilon(k))$ secure against colliding signatures if for all efficient adversaries \mathcal{A} running in time $\leq t(k)$,

$$\Pr \left[\begin{array}{l} dp_S \leftarrow_R \text{Setup}_S(k); \\ (pk_S, m_1, m_2, \sigma) \leftarrow_R \mathcal{A}(dp_S) \end{array} : \begin{array}{l} m_1 \neq m_2, \\ \text{Vrfy}(pk_S, m_1, \sigma) = 1, \text{ and} \\ \text{Vrfy}(pk_S, m_2, \sigma) = 1 \end{array} \right] \leq \varepsilon(k).$$

2.1 Insider Security for SCNINR

This subsection deals with the syntax of a SCNINR scheme and the insider security definitions in the dynamic Multi-User model [2] (also termed the Flexible Signcryption/ Flexible Unsigncryption Oracle (FSO/FUO) model [5]).

Definition 10. A signcryption scheme is a quintuple of algorithms $SC = (\text{Setup}, \text{Gen}_{sd}, \text{Gen}_{rcv}, \text{Sc}, \text{Usc})$ where:

- a) Setup takes a security parameter k as input, and outputs a public domain parameter dp ;
- b) Gen_{sd} takes as input dp and outputs a sender key pair (sk_{sd}, pk_{sd}) , sk_{sd} is the signcrypting key;
- c) Gen_{rcv} takes dp as input and outputs a receiver key pair (sk_{rcv}, pk_{rcv}) ;
- d) Sc takes as inputs a sender private key sk_{sd} , a receiver public key pk_{rcv} , and a message m , and outputs a signcryptext C ; we write $C \leftarrow_R \text{Sc}(sk_{sd}, pk_{rcv}, m)$;
- e) Usc is a deterministic algorithm. It takes as inputs dp , a receiver secret key sk_{rcv} , a sender public key pk_{sd} , and a signcryptext C , and outputs either a valid message $m \in \mathbf{M}$ or an error symbol $\perp \notin \mathbf{M}$.

And, for all $dp \in \{\text{Setup}(k)\}$, all $m \in \mathbf{M}$, all $(sk_{sd}, pk_{sd}) \in \{\text{Gen}_{sd}(dp)\}$, and all $(sk_{rcv}, pk_{rcv}) \in \{\text{Gen}_{rcv}(dp)\}$, $m = \text{Usc}(sk_{rcv}, pk_{sd}, \text{Sc}(sk_{sd}, pk_{rcv}, m))$. The scheme is said to provide NINR if there are two algorithms \mathbf{N} and \mathbf{PV} , a non-repudiation evidence generation and a public verification algorithms, such that:

- \mathbf{N} takes as inputs a receiver secret key sk_{rcv} , a sender public key pk_{sd} , and a signcryptext C , and outputs a non-repudiation evidence nr or a failure symbol \perp .
- \mathbf{PV} takes as inputs a signcryptext C , a message m , a non-repudiation evidence nr , a sender public key pk_{sd} , and a receiver public key pk_{rcv} , and outputs $d \in \{0, 1\}$.
- For all $dp \in \{\text{Setup}(k)\}$, all $C \in \{0, 1\}^*$, all $(sk_{sd}, pk_{sd}) \in \{\text{Gen}_{sd}(dp)\}$, and all $(sk_{rcv}, pk_{rcv}) \in \{\text{Gen}_{rcv}(dp)\}$, if $\perp \neq m \leftarrow \text{Usc}(sk_{rcv}, pk_{sd}, C)$ and $nr \leftarrow \mathbf{N}(sk_{rcv}, pk_{sd}, C)$ then $1 = d \leftarrow \mathbf{PV}(C, m, nr, pk_{sd}, pk_{rcv})$.

Definition 11 (Confidentiality in the dM-IND-iCCA). A SCNINR SC is said to be $(t(k), q_{\text{Usc}}(k), q_{\mathbf{N}}(k), \varepsilon(k))$ dM-IND-iCCA-secure, if for all adversaries \mathcal{A} playing Game 3, running in time $\leq t(k)$, and issuing at most respectively $q_{\text{Usc}}(k)$ and $q_{\mathbf{N}}(k)$ queries to the unsigncryption and non-repudiation evidence generation oracles, $\text{Adv}_{\mathcal{A}, SC}^{\text{cca}2}(k) \leq \varepsilon(k)$.

Game 3 Insider Confidentiality in the Dynamic Multi-User model (dM-IND-iCCA)

We consider the experiments E_0 and E_1 , described hereunder, wherein $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ is a two-stage adversary against a SCNINR scheme.

- 1) The challenger generates $dp \leftarrow_{\mathcal{R}} \text{Setup}(k)$ and $(sk_{rcv}, pk_{rcv}) \leftarrow_{\mathcal{R}} \text{Gen}_{rcv}(dp)$.
- 2) \mathcal{A}_1 is provided with dp and pk_{rcv} , and is given access to:
 - (a) an unsigncryption oracle $\mathcal{O}_{\text{Usc}}(\cdot, \cdot)$, which takes as inputs a sender public key pk and a signcrypted text C , and outputs $m \leftarrow \text{Usc}(sk_{rcv}, pk, C)$, and
 - (b) a non-repudiation evidence generation oracle $\mathcal{O}_{\text{N}}(\cdot, \cdot)$ which takes as inputs a sender public key pk and a signcrypted text C and outputs $nr \leftarrow \text{N}(sk_{rcv}, pk, C)$.
- 3) \mathcal{A}_1 outputs $(m_0, m_1, sk_{sd}, pk_{sd}, st) \leftarrow_{\mathcal{R}} \mathcal{A}_1^{\mathcal{O}_{\text{Usc}}(\cdot, \cdot), \mathcal{O}_{\text{N}}(\cdot, \cdot)}(dp, pk_{rcv})$ where $m_0, m_1 \in \mathbf{M}$, $m_0 \neq m_1$ and $|m_0| = |m_1|$, st is a state, and $(sk_{sd}, pk_{sd}) \in \{\text{Gen}_{sd}(dp)\}$ is the attacked sender key pair.
- 4) In the experiment $E_{b, b=0,1}$, the challenger computes $C^* \leftarrow_{\mathcal{R}} \text{Sc}(sk_{sd}, pk_{rcv}, m_b)$.
- 5) \mathcal{A}_2 outputs $b' \leftarrow_{\mathcal{R}} \mathcal{A}_2^{\mathcal{O}_{\text{Usc}}(\cdot, \cdot), \mathcal{O}_{\text{N}}(\cdot, \cdot)}(C^*, st)$ ($\mathcal{O}_{\text{Usc}}(\cdot, \cdot)$ and $\mathcal{O}_{\text{N}}(\cdot, \cdot)$ are as in step 2).
- 6) For $E_{b, b=0,1}$, out_b denotes the event: (i) \mathcal{A}_2 never issued $\mathcal{O}_{\text{Usc}}(pk_{sd}, C^*)$ or $\mathcal{O}_{\text{N}}(pk_{sd}, C^*)$, and (ii) $b' = 1$.

And, $\text{Adv}_{\mathcal{A}, \text{SC}}^{\text{cca2}}(k) = |\Pr(\text{out}_0) - \Pr(\text{out}_1)|$ denotes \mathcal{A} 's dM-IND-iCCA advantage.

Game 4 Strong Unforgeability in the Dynamic Multi-User model (dM-sUF-iCCA)

\mathcal{A} is a forger against SC .

- 1) The challenger computes $dp \leftarrow_{\mathcal{R}} \text{Setup}(k)$, $(sk_{sd}, pk_{sd}) \leftarrow_{\mathcal{R}} \text{Gen}_{sd}(dp)$, $L \leftarrow ()$.
- 2) \mathcal{A} runs with inputs (dp, pk_{sd}) and is given a flexible signcryption oracle $\mathcal{O}_{\text{Sc}}(\cdot, \cdot)$, which on inputs a valid public receiver key pk and a message m :
 - (i) computes $C \leftarrow_{\mathcal{R}} \text{Sc}(sk_{sd}, pk, m)$,
 - (ii) appends (pk, m, C) to L ,
 - (iii) and outputs C .
- 3) \mathcal{A} outputs $((sk_{rcv}, pk_{rcv}), C^*) \leftarrow_{\mathcal{R}} \mathcal{A}^{\mathcal{O}_{\text{Sc}}(\cdot, \cdot)}(dp, pk_{sd})$. \mathcal{A} wins the game if:
 - (i) $\perp \neq m^* \leftarrow \text{Usc}(sk_{rcv}, pk_{sd}, C^*)$, and
 - (ii) $(pk_{rcv}, m^*, C^*) \notin L$.

We denote by $\text{Adv}_{\mathcal{A}, \text{SC}}^{\text{uf}}(k) = \Pr(\text{Succ}_{\mathcal{A}}^{\text{uf}})$ the probability that \mathcal{A} wins the game.

Definition 12 (Unforgeability in the dM-sUF-iCCA model). A SCNINR is said to be $(t(k), q_{\text{Sc}}(k), \varepsilon(k))$ unforgeable in the dM-sUF-iCCA model if for all attackers \mathcal{A} playing Game 4, running in time $\leq t(k)$, and issuing at most $q_{\text{Sc}}(k)$ signcryption queries, $\text{Adv}_{\mathcal{A}, \text{SC}}^{\text{uf}}(k) \leq \varepsilon(k)$.

Game 5 Soundness of non-repudiation

- 1) The challenger computes $dp \leftarrow_{\mathcal{R}} \text{Setup}(k)$.
- 2) \mathcal{A} runs with input dp and outputs $(C^*, pk_{sd}, sk_{rcv}, pk_{rcv}, m', nr^*) \leftarrow_{\mathcal{R}} \mathcal{A}(dp)$.
- 3) \mathcal{A} wins the game if:
 - (i) $\perp \neq m \leftarrow \text{Usc}(sk_{rcv}, pk_{sd}, C^*)$, and
 - (ii) $m \neq m'$ and $1 = d \leftarrow \text{PV}(C^*, m', nr^*, pk_{sd}, pk_{rcv})$.

We denote by $\text{Adv}_{\mathcal{A}, \text{SC}}^{\text{snr}}(k)$ the probability that \mathcal{A} wins the game.

Definition 13 (Soundness of non-repudiation). A SCNINR is said to achieve $(t(k), \varepsilon(k))$ -computational soundness of non-repudiation if for any adversary \mathcal{A} playing Game 5 and running in time $\leq t(k)$, $\text{Adv}_{\mathcal{A}, \text{SC}}^{\text{nr}}(k) \leq \varepsilon(k)$.

Game 6 Unforgeability of non-repudiation evidence

\mathcal{A} is an attacker against SC .

- 1) The challenger computes $dp \leftarrow_{\mathcal{R}} \text{Setup}(k)$, $(sk_{\text{sd}}, pk_{\text{sd}}) \leftarrow_{\mathcal{R}} \text{Gen}_{\text{sd}}(dp)$; and $(sk_{\text{rcv}}, pk_{\text{rcv}}) \leftarrow_{\mathcal{R}} \text{Gen}_{\text{rcv}}(dp)$.
- 2) \mathcal{A} runs with inputs $(dp, pk_{\text{sd}}, pk_{\text{rcv}})$ and is given access to a signcryption, an unsigncryption, and a non-repudiation evidence generation oracles. It outputs $(C^*, m^*, nr^*) \leftarrow_{\mathcal{R}} \mathcal{A}^{\mathcal{O}_{\text{Sc}}(\cdot, \cdot), \mathcal{O}_{\text{Usc}}(\cdot, \cdot), \mathcal{O}_{\text{N}}(\cdot, \cdot)}(dp, pk_{\text{sd}}, pk_{\text{rcv}})$.
- 3) \mathcal{A} wins if:
 - (i) C^* was generated through the $\mathcal{O}_{\text{Sc}}(\cdot, \cdot)$ oracle on inputs (pk_{rcv}, m) for some m ,
 - (ii) $1 = d \leftarrow \text{PV}(C^*, m^*, nr^*, pk_{\text{sd}}, pk_{\text{rcv}})$, and
 - (iii) $\mathcal{O}_{\text{N}}(pk_{\text{sd}}, C^*)$ was not issued by \mathcal{A} .

$\text{Adv}_{\mathcal{A}, \text{SC}}^{\text{nr}}(k)$ denotes the probability that \mathcal{A} wins the game.

Definition 14 (Unforgeability of non-repudiation evidence). A SCNINR is said to achieve $(t, q_{\text{Sc}}, q_{\text{Usc}}, q_{\text{N}}, \varepsilon)$ unforgeability of non-repudiation evidence if for all adversaries \mathcal{A} playing Game 6, running in time t , and issuing respectively q_{Sc} , q_{Usc} , and q_{N} queries to the signcryption, unsigncryption, and non-repudiation evidence generation oracles, $\text{Adv}_{\mathcal{A}, \text{SC}}^{\text{nr}}(k) \leq \varepsilon$.

3 An Efficient Generic Insider Secure SCNINR

We present our generic SCNINR design termed SN; it uses as building blocks (i) a KEM $\mathcal{K} = (\text{Setup}_{\mathcal{K}}, \text{Gen}_{\mathcal{K}}, \text{Ecp}, \text{Dcp})$, (ii) a symmetric encryption scheme $\mathcal{E} = (\text{E}, \text{D}, \mathbf{K}, \mathbf{M}, \mathbf{C})$, (iii) a PRF Prf defined over $(\mathbf{K}, \mathbf{D}, \mathbf{R} = \mathbf{K})$, (iv) a hash function H defined over $(\mathbf{K}, \mathbf{M}', \mathbf{T})$, and (v) a signature scheme $\mathcal{S} = (\text{Setup}_{\mathcal{S}}, \text{Gen}_{\mathcal{S}}, \text{Sign}, \text{Vrfy})$ with message space $\mathbf{M}_{\mathcal{S}}$. We assume that $\mathbf{M} \subset \mathbf{D}$, $\Sigma \subset \mathbf{D}$, $\mathbf{T} \subset \mathbf{M}_{\mathcal{S}}$, and that for all $(\tau, \tau', \tau'') \in \mathbf{K}^2$, all $c' \in \mathbf{C}'$, all $c \in \mathbf{C}$, all pk_{sd} such that $(sk_{\text{sd}}, pk_{\text{sd}}) \in \{\text{Gen}_{\mathcal{S}}(dp_2)\}$ for some sk_{sd} , and all pk_{rcv} such that $(sk_{\text{rcv}}, pk_{\text{rcv}}) \in \{\text{Gen}_{\mathcal{S}}(dp_1)\}$, $(pk_{\text{sd}}, \tau, \tau', \tau'', c, c', pk_{\text{rcv}}) \in \mathbf{M}'$. We assume that the KEM is such that $\mathbf{K}' = \mathbf{K}^4$ (this can be achieved by using, if needed, an appropriate key derivation function and/or a pseudo-random generator), and that $dp_{\mathcal{K}}$ defines both \mathbf{K}' and \mathbf{C}' .

In an encrypt-then-sign design (which aims also at NINR), the signed data cannot be the plain-text m (or publicly depend on it), as otherwise even outsider confidentiality cannot be achieved. Moreover, for insider confidentiality (wherein the attacker knows the sender's private key) it should not be possible to recover the signed data from the sender's private key, as an attacker could resign the data and submit the resulting signcrypted cipher-text for decryption, and then succeed in an insider confidentiality game. To overcome these difficulties, we compute the signed data as a function of the encapsulated key and the plain text m such that it cannot be recovered by an attacker which

does not know the receiver's private key. Besides, we append a (PFR based MAC) tag of the signature, to make a "re-signing attack" not feasible. The design we obtain is described hereunder.

The Generic SN Signcryption Scheme

- 10 Setup(k): The algorithm computes $dp_K \leftarrow_R \text{Setup}_K(k)$; $dps \leftarrow_R \text{Setup}_S(k)$; it defines also $\mathcal{E} = (E, D, \mathbf{K} = \{0, 1\}^k, \mathbf{M}, \mathbf{C})$, a pseudo-random function Prf over $(\mathbf{K}, \mathbf{M}, \mathbf{K})$, and a hash function H over $(\mathbf{K}, \mathbf{M}', \mathbf{T})$.
 - 11 $\tau_0 \leftarrow_R \mathbf{K}$; $dp \leftarrow (dp_K, dps, \mathcal{E}, \text{Prf}, H, \tau_0)$; **return** dp ;
 - 12 Gensd(dp):
 - 13 Parse dp as $(dp_K, dps, \mathcal{E}, \text{Prf}, H, \tau_0)$; $(sk_{sd}, pk_{sd}) \leftarrow_R \text{Gens}(dps)$; **return** (sk_{sd}, pk_{sd}) ;
 - 14 Genrcv(dp):
 - 15 Parse dp as $(dp_K, dps, \mathcal{E}, \text{Prf}, H, \tau_0)$; $(sk_{rcv}, pk_{rcv}) \leftarrow_R \text{Gen}_K(dp_K)$; **return** (sk_{rcv}, pk_{rcv}) ;
 - 16 Sc(sk_{sd}, pk_{rcv}, m):
 - 17 $((\tau_1, \tau'_1, \tau_2, \tau'_2), c_1) \leftarrow_R \text{Ecp}(pk_{rcv})$; $c_2 \leftarrow_R E(\tau_1, m)$; $\tau_3 \leftarrow \text{Prf}(\tau_2, m)$;
 - 18 $\hat{m} \leftarrow H(\tau_0, (pk_{sd}, \tau_2, \tau'_2, \tau_3, c_1, c_2, pk_{rcv}))$; $\sigma \leftarrow_R \text{Sign}(sk_{sd}, \hat{m})$;
 - 19 $t \leftarrow \text{Prf}(\tau'_1, \sigma)$; **return** (t, σ, c_1, c_2) ;
 - 20 Usc(sk_{rcv}, pk_{sd}, C):
 - 21 Parse C as (t, σ, c_1, c_2) ; $(\tau_1, \tau'_1, \tau_2, \tau'_2) \leftarrow \text{Dcp}(sk_{rcv}, c_1)$;
 - 22 $m \leftarrow D(\tau_1, c_2)$; $\tau_3 \leftarrow \text{Prf}(\tau_2, m)$;
 - 23 $\hat{m} \leftarrow H(\tau_0, (pk_{sd}, \tau_2, \tau'_2, \tau_3, c_1, c_2, pk_{rcv}))$; $d \leftarrow \text{Vrfy}(pk_{sd}, \hat{m}, \sigma)$; $t' \leftarrow \text{Prf}(\tau'_1, \sigma)$;
 - 24 **if** $d = 1$ and $t = t'$ **then** **return** m ; **else** **return** \perp ;
 - 25 N(sk_{rcv}, pk_{sd}, C):
 - 26 Parse C as (t, σ, c_1, c_2) ; $(\tau_1, \tau'_1, \tau_2, \tau'_2) \leftarrow \text{Dcp}(sk_{rcv}, c_1)$;
 - 27 $m \leftarrow D(\tau_1, c_2)$; $\tau_3 \leftarrow \text{Prf}(\tau_2, m)$;
 - 28 $\hat{m} \leftarrow H(\tau_0, (pk_{sd}, \tau_2, \tau'_2, \tau_3, c_1, c_2, pk_{rcv}))$; $d \leftarrow \text{Vrfy}(pk_{sd}, \hat{m}, \sigma)$; $t' \leftarrow \text{Prf}(\tau'_1, \sigma)$;
 - 29 **if** $d = 1$ and $t = t'$ **then** **return** $(\tau_1, \tau'_1, \tau_2, \tau'_2)$; **else** **return** \perp ;
 - 30 PV($C, m, nr, pk_{sd}, pk_{rcv}$):
 - 31 Parse C as (t, σ, c_1, c_2) and nr as $(\tau_1, \tau'_1, \tau_2, \tau'_2)$; $m' \leftarrow D(\tau_1, c_2)$;
 - 32 **if** $m' \neq m$ **then** **return** 0;
 - 33 $\tau_3 \leftarrow \text{Prf}(\tau_2, m)$; $\hat{m} \leftarrow H(\tau_0, (pk_{sd}, \tau_2, \tau'_2, \tau_3, c_1, c_2, pk_{rcv}))$;
 - 34 $d \leftarrow \text{Vrfy}(pk_{sd}, \hat{m}, \sigma)$; $t' \leftarrow \text{Prf}(\tau'_1, \sigma)$;
 - 35 **if** $d = 1$ and $t = t'$ **then** **return** 1; **else** **return** 0.
-

For the consistency of the scheme, one can observe that as $\text{Dcp}(sk_{rcv}, c_1)$ yields $(\tau_1, \tau'_1, \tau_2, \tau'_2)$, the receiver can compute $\tau_3 \leftarrow \text{Prf}(\tau_2, m)$ and \hat{m} , and then verify whether $1 = \text{Vrfy}(pk_{sd}, \hat{m}, \sigma)$ and $t = \text{Prf}(\tau'_1, \sigma)$ to accept or reject m . So, for all $dp \in \{\text{Setup}(k)\}$, all $m \in \mathcal{M}$, all $(sk_{sd}, pk_{sd}) \in \{\text{Gensd}(dp)\}$, and all $(sk_{rcv}, pk_{rcv}) \in \{\text{Genrcv}(dp)\}$, $m = \text{Usc}(sk_{rcv}, pk_{sd}, \text{Sc}(sk_{sd}, pk_{rcv}, m))$. Besides, if $nr \leftarrow \text{N}(sk_{rcv}, pk_{sd}, \text{Sc}(sk_{sd}, pk_{rcv}, m))$ then $1 = d \leftarrow \text{PV}(C, m, nr, pk_{sd}, pk_{rcv})$. Our construction is a signcryption scheme with non-interactive non-repudiation.

4 Security Analysis of the SN Scheme

We propose in this section a detailed security analysis of our generic construction.

4.1 Insider Confidentiality

Theorem 1. *If the encryption scheme \mathcal{E} is $(t(k), \varepsilon_{\text{ss}}(k))$ -semantically secure, the pseudo random function Prf is $(t(k), \varepsilon_{\text{Prf}}(k))$ -secure, the key encapsulation mechanism is $(t(k), \varepsilon_{\mathcal{K}}(k))$ -secure, and the signature scheme is $(t(k), \varepsilon_{\mathcal{S}}(k))$ resistant against colliding signatures, then the SN signcryption scheme is $(t(k), \varepsilon(k))$ -dM-IND-iCCA secure, where*

$$\varepsilon(k) \leq \varepsilon_{\text{ss}}(k) + 2(\varepsilon_{\mathcal{K}}(k) + \varepsilon_{\mathcal{S}}(k) + \varepsilon_{\text{H}}(k) + 2\varepsilon_{\text{Prf}}(k) + (q_{\text{Usc}} + q_{\text{N}})/|\mathbf{K}|), \quad (2)$$

wherein q_{Usc} and q_{N} are upper bounds on the number of unsigncryption and non-repudiation evidence generation queries the attacker issues.

Proof. We denote the steps (1) and (2), (3) and (4), and (5) and (6) of Game 3 by PRE-CHALLENGE, CHALLENGE, and POST-CHALLENGE stages respectively. We consider the following simulator to answer \mathcal{A} 's queries. The Initialization procedure is executed once at the beginning of the game. The Finalization procedure is also executed once, after \mathcal{A} produces its output, at the end of the game. To keep the description simple, we omit public key validations.

Simulation for the experiments E_0 and $E_0^{(1)}$, $E_0^{(2)}$, and $E_0^{(3)}$ in the dM-IND-iCCA Game

100 Initialization: $dp \leftarrow (dp_{\mathcal{K}}, dp_{\mathcal{S}}, \mathcal{E}, \text{Prf}, \text{H}, \tau_0) \leftarrow_{\text{R}} \text{Setup}(k)$;

101 $\boxed{E_0}$ $(sk_{\text{rcv}}, pk_{\text{rcv}}) \leftarrow_{\text{R}} \text{Gen}_{\text{rcv}}(dp)$; $\boxed{E_0^{(1)}, E_0^{(2)}, E_0^{(3)}}$ receive pk_{rcv} from the KEM challenger;

102 $\boxed{E_0}$ $((\bar{\tau}_1, \bar{\tau}'_1, \bar{\tau}_2, \bar{\tau}'_2), \bar{c}_1) \leftarrow_{\text{R}} \text{Ecp}(pk_{\text{rcv}})$; $\boxed{E_0^{(1)}}$ $((\bar{\tau}_1, \bar{\tau}'_1, \bar{\tau}_2, \bar{\tau}'_2), \bar{c}_1) \leftarrow_{\text{R}} \text{Chall}_{\mathcal{K}_{E_0}}$;

$\boxed{E_0^{(2)}, E_0^{(3)}}$ $((\bar{\tau}_1, \bar{\tau}'_1, \bar{\tau}_2, \bar{\tau}'_2), \bar{c}_1) \leftarrow_{\text{R}} \text{Chall}_{\mathcal{K}_{E_1}}$;

PRE-CHALLENGE PHASE \mathcal{A} is provided with (dp, pk_{rcv}) and is given access to the following oracles.

103 $\mathcal{O}_{\text{Usc}}(pk, C)$: $\mathcal{O}_{\text{N}}(pk, C)$:

104 Parse C as (t, σ, c_1, c_2) ; ► Return \perp if the parsing fails

105 $\boxed{E_0^{(1)}, E_0^{(2)}, E_0^{(3)}}$ **if** $c_1 = \bar{c}_1$ **then return** \perp ;

106 $\boxed{E_0}$ $(\tau_1, \tau'_1, \tau_2, \tau'_2) \leftarrow \text{Dcp}(sk_{\text{rcv}}, c_1)$; $\boxed{E_0^{(1)}, E_0^{(2)}, E_0^{(3)}}$ $(\tau_1, \tau'_1, \tau_2, \tau'_2) \leftarrow \mathcal{O}_{\text{Dcp}}(c_1)$;

107 $m \leftarrow \text{D}(\tau_1, c_2)$; $\tau_3 \leftarrow \text{Prf}(\tau_2, m)$; $\hat{m} \leftarrow \text{H}(\tau_0, (pk, \tau_2, \tau'_2, \tau_3, c_1, c_2, pk_{\text{rcv}}))$;

108 $d \leftarrow \text{Vrfy}(pk, \hat{m}, \sigma)$; $t' \leftarrow \text{Prf}(\tau'_1, \sigma)$;

```

109 if  $d = 1$  and  $t = t'$  then  $\mathcal{O}_{\text{Usc}}$  return  $m$ ;  $\mathcal{O}_{\text{N}}$  return  $(\tau_1, \tau'_1, \tau_2, \tau'_2)$ ; else return  $\perp$ ;
    CHALLENGE PHASE
110  $(m_0, m_1, sk_{sd}, pk_{sd}, st) \leftarrow_{\mathcal{R}} \mathcal{A}_1^{\mathcal{O}_{\text{Usc}}, \mathcal{O}_{\text{N}}}(dp, pk_{rcv})$ ; ▶  $|m_0| = |m_1|$ 
111  $((\tau_1, \tau'_1, \tau_2, \tau'_2), c_1) \leftarrow ((\bar{\tau}_1, \bar{\tau}'_1, \bar{\tau}_2, \bar{\tau}'_2), \bar{c}_1)$ ;
     $E_0, E_0^{(i)}, i = 1, 2, 3$ 
112  $c_2 \leftarrow_{\mathcal{R}} \mathbf{E}(\tau_1, m_0)$ ;
     $E_0, E_0^{(1)}, E_0^{(2)}$ 
113  $\tau_3 \leftarrow \text{Prf}(\tau_2, m_0)$ ;  $E_0^{(3)}$   $\tau_3 \leftarrow_{\mathcal{R}} \mathbf{K}$ ; ▶  $\tau_3 \leftarrow_{\mathcal{R}} \mathbf{K}$  is equivalent to  $f \leftarrow_{\mathcal{R}} \text{Func}(\mathbf{M}, \mathbf{K})$ ;  $\tau_3 \leftarrow f(m_0)$ ;
114  $\hat{m} \leftarrow \mathbf{H}(\tau_0, (pk_{sd}, \tau_2, \tau'_2, \tau_3, c_1, c_2, pk_{rcv}))$ ;  $\sigma \leftarrow \text{Sign}(sk_{sd}, \hat{m})$ ;  $t \leftarrow \text{Prf}(\tau'_1, \sigma)$ ;
     $C^* \leftarrow_{\mathcal{R}} (t, \sigma, c_1, c_2)$ ;

    POST-CHALLENGE PHASE
     $\mathcal{A}_2$  runs with inputs  $(C^*, st)$ . It has access to the oracles  $\mathcal{O}_{\text{Usc}}(\cdot, \cdot)$ ,  $\mathcal{O}_{\text{N}}(\cdot, \cdot)$ ,
    The simulation aborts if  $\mathcal{A}$  issues  $\mathcal{O}_{\text{Usc}}(pk_{sd}, C^*)$  or  $\mathcal{O}_{\text{N}}(pk_{sd}, C^*)$ .
115  $\hat{b} \leftarrow_{\mathcal{R}} \mathcal{A}_2^{\mathcal{O}_{\text{Usc}}(\cdot, \cdot), \mathcal{O}_{\text{N}}(\cdot, \cdot)}(C^*, st)$ ;

116 Finalization: return  $\hat{b}$ ;

```

At lines 103 to 109 we describe simultaneously the $\mathcal{O}_{\text{Usc}}(\cdot, \cdot)$ and $\mathcal{O}_{\text{N}}(\cdot, \cdot)$ oracles. When one of the oracles is queried, at line 109, the boxed instruction with corresponding header is executed.

Besides the experiment E_0 in the dM-IND-iCCA security game, we define three other experiments $E_0^{(1)}$, $E_0^{(2)}$, and $E_0^{(3)}$. For each experiment, at a line with boxed codes, only the code with corresponding header is executed. The simulator is efficient in all the experiments. We give a summary of the changes between the experiments hereunder.

- 1) From E_0 to $E_0^{(1)}$:
 - a) in $E_0^{(1)}$, the simulator Sim does not generate (sk_{rcv}, pk_{rcv}) , instead it receives pk_{rcv} from a KEM challenger (see at line 101),
 - b) to compute $\text{Dcp}(sk_{rcv}, c_1)$, the simulator sends c_1 to the KEM challenger and receives $(\tau_1, \tau'_1, \tau_2, \tau'_2) \leftarrow \text{Dcp}(sk_{rcv}, c_1)$ from the challenger (see at line 106),
 - c) and in the challenge phase, the value of $((\tau_1, \tau'_1, \tau_2, \tau'_2), c) \leftarrow_{\mathcal{R}} \text{Ecp}(pk_{rcv})$ is received from the KEM challenger; we note $((\tau_1, \tau'_1, \tau_2, \tau'_2), c) \leftarrow \text{Chall}_{\mathcal{K}_{E_0}}$.
 - d) Besides, in the Usc and N oracles, whenever \mathcal{A} provides the simulator with a signcrypted cipher-text $C = (t, \sigma, c_1, c_2)$ with $c_1 = \bar{c}_1$, the simulator considers t as an invalid PRF based MAC and returns \perp (see at line 105).
- 2) From $E_0^{(1)}$ to $E_0^{(2)}$, the only change is at line 102 of the challenge phase, wherein the KEM challenger provides \mathcal{S} with $\text{Chall}_{\mathcal{K}_{E_1}}$ instead of $\text{Chall}_{\mathcal{K}_{E_0}}$.
- 3) From $E_0^{(2)}$ to $E_0^{(3)}$, the change is in the challenge phase, where τ_3 is computed as $\tau_3 \leftarrow \text{Prf}(\tau_2, m_0)$ in $E_0^{(2)}$, and as $\tau_3 \leftarrow_{\mathcal{R}} \mathbf{K}$ in $E_0^{(3)}$ (see at line 113). Notice that $\tau_3 \leftarrow_{\mathcal{R}} \mathbf{K}$ is equivalent to $f \leftarrow_{\mathcal{R}} \text{Func}(\mathbf{M}, \mathbf{K})$; $\tau_3 \leftarrow f(m_0)$.

Let $\Pr(\text{out}_0)$ and $\Pr(\text{out}_0^{(i)})$, for $i \in \{1, 2, 3\}$ denote the probability that \mathcal{A} outputs 1 in the experiments E_0 and $E_0^{(i)}$, respectively. Notice that the FINALIZATION procedure outputs exactly whatever \mathcal{A} returns. Given the difference between E_0 and $E_0^{(1)}$, whenever \mathcal{A} provides the \mathcal{O}_{usc} or \mathcal{O}_{N} oracles with a valid $C = (t, \sigma, c_1, c_2)$ with $c_1 = \bar{c}_1$ then:

- a) If this occurs before the challenge phase, t is a no-message (PRF-based) MAC forgery.
- b) If this occurs after the challenge phase (with the restriction $C \neq C^*$), if $(t, \sigma) \neq (t^*, \sigma^*)$, then (t^*, σ^*) is MAC forgery. Otherwise, we necessarily have $(pk, c_1, c_2) \neq (pk_{\text{sd}}, c_1^*, c_2^*)$. And then, if $\hat{m} = \hat{m}^*$, we have a H collision, otherwise we have colliding signatures.

So, using [9, Theorem 6.2, p. 224], it holds that

$$|\Pr(\text{out}_0) - \Pr(\text{out}_0^{(1)})| \leq \varepsilon_{\text{Prf}}(k) + (q_{\text{usc}} + q_{\text{N}})/|\mathbf{K}| + \varepsilon_{\text{H}}(k) + \varepsilon_{\text{S}}(k). \quad (3)$$

The difference between $E_0^{(1)}$ and $E_0^{(2)}$ is: in $E_0^{(1)}$ the simulator receives $\text{Chall}_{\mathcal{K}_{E_0}}$ from the KEM challenger, while it receives $\text{Chall}_{\mathcal{K}_{E_1}}$ in $E_0^{(2)}$. As \mathcal{K} is $(t(k), \varepsilon_{\mathcal{K}}(k))$ -secure, it follows that

$$|\Pr(\text{out}_0^{(1)}) - \Pr(\text{out}_0^{(2)})| \leq \varepsilon_{\mathcal{K}}(k). \quad (4)$$

Also, given that Prf is $(t(k), \varepsilon_{\text{Prf}}(k))$ -secure, we have

$$|\Pr(\text{out}_0^{(2)}) - \Pr(\text{out}_0^{(3)})| \leq \varepsilon_{\text{Prf}}(k). \quad (5)$$

Now, we consider the experiments $E_1^{(3)}, E_1^{(2)}, E_1^{(1)}$ and E_1 where the only difference between E_1 (resp. $E_1^{(3)}, E_1^{(2)}, E_1^{(1)}$) and E_0 (resp. $E_0^{(3)}, E_0^{(2)}, E_0^{(1)}$) is that the lines 112 and 113 in the challenge phase are modified, to use m_1 instead of m_0 , as hereunder:

$$\begin{array}{l} \boxed{E_1, E_1^{(i)}, i = 1, 2, 3} \\ 112 \quad c_2 \leftarrow_{\text{R}} \bar{\text{E}}(\tau_1, m_1); \\ \boxed{E_1, E_1^{(1)}, E_1^{(2)}} \quad \boxed{E_1^{(3)}} \\ 113 \quad \tau_3 \leftarrow \text{Prf}(\tau_2, m_1); \quad \tau_3 \leftarrow_{\text{R}} \mathbf{K}^2; \end{array}$$

With similar arguments, applied to the experiments E_1 and $E_1^{(i)}$, $i = 1, 2, 3$, we obtain

$$|\Pr(\text{out}_1) - \Pr(\text{out}_1^{(1)})| \leq \varepsilon_{\text{Prf}}(k) + (q_{\text{usc}} + q_{\text{N}})/|\mathbf{K}| + \varepsilon_{\text{H}}(k) + \varepsilon_{\text{S}}(k), \quad (6)$$

$$|\Pr(\text{out}_1^{(1)}) - \Pr(\text{out}_1^{(2)})| \leq \varepsilon_{\mathcal{K}}(k), \quad (7)$$

and

$$|\Pr(\text{out}_1^{(2)}) - \Pr(\text{out}_1^{(3)})| \leq \varepsilon_{\text{Prf}}(k). \quad (8)$$

We consider now, the challenge phases in the experiments $E_b^{(3)}$, $b = 0, 1$, wherein the secret key τ_1 is used only in the encryption $c_2 \leftarrow_{\text{R}} \bar{\text{E}}(\tau_1, m_b)$. Recall that in $E_{b,b=0,1}^{(3)}$, $(\tau_1, \tau_1', \tau_2, \tau_2')$ is computed at the KEM challenger as $(\tau_1, \tau_1', \tau_2, \tau_2') \leftarrow_{\text{R}} \mathbf{K}^3$. Now,

we consider the experiments $E_{b,b=0,1}^{(3a)}$, such that the difference between $E_{b,b=0,1}^{(3)}$ and $E_{b,b=0,1}^{(3a)}$ is that in $E_{b,b=0,1}^{(3a)}$ the simulator ignores the value of τ_1 generated by the KEM challenger; it does not compute c_2 . Instead, it receives c_2 from a semantic security challenger. The challenger computes c_2 using the instructions: $\tau \leftarrow_{\mathbf{R}} \mathbf{K}; c_2 \leftarrow_{\mathbf{R}} \mathbf{E}(\tau, m_b)$. Given the change, it holds that

$$\Pr(\text{out}_b^{(3)}) = \Pr(\text{out}_b^{(3a)}), \text{ for } b = 0, 1 \quad (9)$$

and the difference between $E_0^{(3a)}$ and $E_1^{(3a)}$ is that in $E_0^{(3a)}$ c_2 is computed as $c_2 \leftarrow_{\mathbf{R}} \mathbf{E}(\tau, m_0)$ wherein $\tau \leftarrow_{\mathbf{R}} \mathbf{K}$, while in $E_0^{(3a)}$ it is computed as $c_2 \leftarrow_{\mathbf{R}} \mathbf{E}(\tau, m_0)$, it then follows that

$$|\Pr(\text{out}_0^{(3)}) - \Pr(\text{out}_1^{(3)})| = |\Pr(\text{out}_0^{(3a)}) - \Pr(\text{out}_1^{(3a)})| \leq \varepsilon_{\text{ss}}(k). \quad (10)$$

From the inequalities (3) to (10), we obtain

$$|\Pr(\text{out}_0) - \Pr(\text{out}_1)| \leq \varepsilon_{\text{ss}}(k) + 2(\varepsilon_{\mathcal{K}}(k) + \varepsilon_{\mathcal{S}}(k) + \varepsilon_{\mathcal{H}}(k) + 2\varepsilon_{\text{Prf}}(k) + (q_{\text{usc}} + q_{\mathbf{N}})/|\mathbf{K}|).$$

□

4.2 Unforgeability of the SN Scheme

Theorem 2. *If the signature scheme is $(t(k), \varepsilon_{\mathcal{S}}(k))$ -sUF-CMA secure and the hash function \mathbf{H} is $(t(k), \varepsilon_{\mathcal{H}}(k))$ collision resistant, then the SN signcryption scheme is $(t(k), \varepsilon(k))$ dM-sUF-iCCA-secure, where $\varepsilon(k) \leq \varepsilon_{\mathcal{H}}(k) + \varepsilon_{\mathcal{S}}(k)$.*

Proof. We consider the following simulation to answer \mathcal{A} 's queries.

Simulation for the dM-sUF-iCCA security game

200 **Initialization:** $dp \leftarrow (dp_{\mathcal{K}}, dp_{\mathcal{S}}, \mathcal{E}, \text{Prf}, \mathbf{H}, \tau_0) \leftarrow_{\mathbf{R}} \text{Setup}(k); \mathbf{L} \leftarrow (); \mathbf{L}_1 \leftarrow (); \mathbf{L}_2 \leftarrow ();$

201 $\overset{E_0}{(sk_{\text{sd}}, pk_{\text{sd}})} \leftarrow_{\mathbf{R}} \text{Gen}_{\text{sd}}(dp); \overset{E_1}{\text{Get } pk_{\text{sd}} \text{ from the challenger for signature unforgeability;}}$

202 $\mathcal{O}_{\text{Sc}}(pk, m):$

203 $((\tau_1, \tau'_1, \tau_2, \tau'_2), c_1) \leftarrow_{\mathbf{R}} \text{Ecp}(pk); c_2 \leftarrow_{\mathbf{R}} \mathbf{E}(\tau_1, m);$

204 $\tau_3 \leftarrow \text{Prf}(\tau_2, m); \hat{m} \leftarrow \mathbf{H}(\tau_0, (pk_{\text{sd}}, \tau_2, \tau'_2, \tau_3, c_1, c_2, pk));$

205 $\overset{E_0}{\sigma} \leftarrow_{\mathbf{R}} \text{Sign}(sk_{\text{sd}}, \hat{m}); \overset{E_1}{\text{Send } \hat{m} \text{ to the signing oracle and receive } \sigma;}$

206 $t \leftarrow \text{Prf}(\tau'_1, \sigma)$

207 $\text{Apd}(\mathbf{L}, (pk, m, (t, \sigma, c_1, c_2))); \text{Apd}(\mathbf{L}_1, (\sigma, \hat{m}));$

208 $\text{Apd}(\mathbf{L}_2, (t, pk_{\text{sd}}, m, \hat{m}, \sigma, \tau_1, \tau'_1, \tau_2, \tau'_2, \tau_3, c_1, c_2, pk));$
 return $(t, \sigma, c_1, c_2);$

209 **Finalization:**

210 **if** \mathcal{A} outputs $(sk_{\text{rcv}}, pk_{\text{rcv}}, C^*)$ such that

(i) $\perp \neq m^* \leftarrow \text{Usc}(sk_{\text{rcv}}, pk_{\text{sd}}, C^*)$ and

(ii) $(pk_{\text{rcv}}, m^*, C^*) \notin \mathcal{L}$

then

```

211 Parse  $C^*$  as  $(t^*, \sigma^*, c_1^*, c_2^*)$ ;
212  $(\tau_1^*, \tau_1'^*, \tau_2^*, \tau_2'^*) \leftarrow \text{Dcp}(sk_{\text{rcv}}, c_1^*)$ ;  $m^* \leftarrow \text{D}(\tau_1^*, c_2^*)$ ;  $\tau_3^* \leftarrow \text{Prf}(\tau_2^*, m^*)$ ;
213  $\hat{m}^* \leftarrow \text{H}(\tau_0^*, (pk_{\text{sd}}, \tau_2^*, \tau_2'^*, \tau_3^*, c_1^*, c_2^*, pk_{\text{rcv}}))$ ;
214 if  $(\sigma^*, \hat{m}^*) \notin \mathbf{L}_1$  then
215     return  $(\sigma^*, \hat{m}^*)$ ; ▶  $(\sigma^*, \hat{m}^*)$  is a signature forgery;
216 else ▶  $(\sigma^*, \hat{m}^*) \in \mathbf{L}_1$ 
217     Find  $(t, pk_{\text{sd}}, m, \hat{m}, \sigma, \tau_1, \tau_1', \tau_2, \tau_2', \tau_3, c_1, c_2, pk) \in \mathbf{L}_2$  such that  $(\sigma, \hat{m}) =$ 
218      $(\sigma^*, \hat{m}^*)$ ;
219      $x_1 \leftarrow (pk_{\text{sd}}, \tau_2^*, \tau_2'^*, \tau_3^*, c_1^*, c_2^*, pk_{\text{rcv}})$ ;  $x_2 \leftarrow (pk_{\text{sd}}, \tau_2, \tau_2', \tau_3, c_1, c_2, pk)$ ;
220     if  $(pk, c_1, c_2) \neq (pk_{\text{rcv}}, c_1^*, c_2^*)$  then
221         return  $(x_1, x_2)$ ; ▶ This yields a collision,  $x_1 \neq x_2$  and  $\text{H}(\tau_0, x_1) = \text{H}(\tau_0, x_2)$ .
222     else return  $\perp$ ; ▶
223      $(\sigma^*, \hat{m}^*) \in \mathbf{L}_1, pk = pk_{\text{rcv}}, c_1 = c_1^*,$  and  $c_2 = c_2^*$ , so we have  $(\tau_1, \tau_1', \tau_2, \tau_2') = (\tau_1^*, \tau_1'^*, \tau_2^*, \tau_2'^*)$ , then
224      $m = m^* = \text{D}(\tau_1, c_2)$ , and then  $(pk_{\text{rcv}}, m^*, C^* = (t, \sigma^*, c_1^*, c_2^*)) \in \mathbf{L}$ ; this cannot occur (see condition (ii) at
225     line 210).

```

In experiment E_0 the simulator answers \mathcal{A}' 's queries exactly as in an dM–sUF–iCCA security game. In E_1 , we modify the simulator such that it receives pk_{sd} from a signature challenger, and whenever \mathcal{S} needs a signature on some \hat{m} , it sends it to its signature challenger and receives the corresponding signature (see at line 205). Let $\text{Ev}_{b,b=0,1}$ be the event: “the conditions (i) and (ii) in the FINALIZATION procedure are satisfied in experiment E_b .” It is clear that $\Pr(\text{Ev}_0) = \Pr(\text{Ev}_1)$. Let Coll be the event simulator outputs (x_1, x_2) such that $\text{H}(\tau_0, x_1) = \text{H}(\tau_0, x_2)$.

$$\Pr(\text{Ev}_1 \wedge \text{Coll}) \leq \Pr(\text{Coll}) \leq \varepsilon_{\text{H}}(k).$$

And, if $\text{Ev}_1 \wedge \neg \text{Coll}$ occurs, the simulator outputs a signature forgery, *i. e.*

$$\Pr(\text{Ev}_1 \wedge \neg \text{Coll}) \leq \varepsilon_{\mathcal{S}}(k).$$

It follows that $\varepsilon(k) = \Pr(\text{Ev}) \leq \varepsilon_{\text{H}}(k) + \varepsilon_{\mathcal{S}}(k)$. □

4.3 Soundness of Non-Repudiation

Theorem 3. *If the hash function H is $(t(k), \varepsilon_{\text{H}}(k))$ -collision resistant and the signature scheme is $(t(k), \varepsilon_{\mathcal{S}}(k))$ secure against colliding signatures, then the SN scheme achieves $(t(k), \varepsilon(k))$ soundness of non-repudiation, where $\varepsilon(k) \leq \varepsilon_{\text{H}}(k) + \varepsilon_{\mathcal{S}}(k)$.*

Proof. We consider the following simulator.

Simulation for Soundness of non-repudiation

```

300 Initialization:  $dp \leftarrow (dp_{\text{K}}, dp_{\text{S}}, \mathcal{E}, \text{Prf}, \text{H}, \tau_0) \leftarrow_{\text{R}} \text{Setup}(k)$ ;
301 The attacker  $\mathcal{A}$  outputs  $(C^*, pk_{\text{sd}}, sk_{\text{rcv}}, pk_{\text{rcv}}, m', nr^*) \leftarrow_{\text{R}} \mathcal{A}(dp)$ ;
302 Finalization:
303 if  $\mathcal{A}$  outputs  $(C^*, pk_{\text{sd}}, sk_{\text{rcv}}, pk_{\text{rcv}}, m', nr^*)$  such that

```

```

(i)  $\perp \neq m \leftarrow \text{Usc}(sk_{rcv}, pk_{sd}, C^*)$ , and
(ii)  $m \neq m'$  and  $1 = d \leftarrow \text{PV}(C^*, m', nr^*, pk_{sd}, pk_{rcv})$ ;
then
304   Parse  $C^*$  as  $(t^*, \sigma^*, c_1^*, c_2^*)$  and  $nr^*$  as  $(\tau_1^*, \tau_1'^*, \tau_2^*, \tau_2'^*)$ ;
305    $nr \leftarrow \text{N}(sk_{rcv}, pk_{sd}, C^*)$ ; parse  $nr$  as  $(\tau_1, \tau_1', \tau_2, \tau_2')$ ;
306    $\tau_3^* \leftarrow \text{Prf}(\tau_2^*, m')$ ;  $\tau_3 \leftarrow \text{Prf}(\tau_2, m)$ ;
307    $s_1 \leftarrow (pk_{sd}, \tau_2^*, \tau_2'^*, \tau_3^*, c_1, c_2, pk_{rcv})$ ;  $\hat{m}^* \leftarrow \text{H}(\tau_0, s_1)$ ;
308    $s_2 \leftarrow (pk_{sd}, \tau_2, \tau_2', \tau_3, c_1, c_2, pk_{rcv})$ ;  $\hat{m} \leftarrow \text{H}(\tau_0, s_2)$ ; ▶ As  $m \neq m'$  we necessarily have
    $\tau_1 \neq \tau_1^*$ . Also, as  $m \neq m'$ ,  $\tau_2 = \tau_2^*$  implies  $\tau_3 \neq \tau_3^*$ , so it holds that  $(\tau_2, \tau_3) \neq (\tau_2^*, \tau_3^*)$ ;
309   if  $\hat{m} = \hat{m}^*$  then return  $(s_1, s_2)$ ; ▶ A collision is found for H
310   else return  $(pk_{sd}, \hat{m}, \hat{m}^*, \sigma^*)$ ; ▶ Colliding signatures for  $\hat{m}$  and  $\hat{m}^*$ :
311 else return  $\perp$ ;

```

Clearly, our simulator is efficient and if \mathcal{A} succeeds in the soundness of non-repudiation game, its output $(C^*, pk_{sd}, sk_{rcv}, pk_{rcv}, C^*, m', nr^*)$ is such that the conditions (i) and (ii) at line 303 are satisfied. Then the simulator outputs either (s_1, s_2) such that $s_1 \neq s_2$ and $\text{H}(\tau_0, s_1) = \text{H}(\tau_0, s_2)$, or $(pk_{sd}, \hat{m}, \hat{m}^*, \sigma^*)$ such that $\hat{m} \neq \hat{m}^*$ and $1 = \text{Vrfy}(pk, \hat{m}, \sigma^*) = \text{Vrfy}(pk, \hat{m}^*, \sigma^*)$. Hence, $\varepsilon(k) \leq \varepsilon_H(k) + \varepsilon_S(k)$. \square

4.4 Unforgeability of Non-Repudiation Evidence

Theorem 4. *If the encryption scheme is $(t(k), \varepsilon_E(k))$ resistant to clustering key attacks, the signature scheme is $(t(k), \varepsilon_S(k))$ resistant to colliding signatures, the hash function is $(t(k), \varepsilon_H(k))$ resistant to collisions with identical prefix, and the KEM is $(t(k), \varepsilon_K(k))$ IND-CCA secure, then SN achieves $(t(k), \varepsilon(k))$ unforgeability of non-repudiation evidence with*

$$\varepsilon(k) \leq q_{Sc}(\varepsilon_{Prf}(k) + (q_{Usc} + q_N + 1)/|\mathbf{K}|) + \varepsilon_S(k) + \varepsilon_K(k) + 2\varepsilon_H(k) \quad (11)$$

wherein q_{Sc} , q_{Usc} , and q_N are upper bounds on the number of times the attacker issues respectively the signcryption, unsigncryption, and non-repudiation evidence generation oracles.

Proof. Let Ev be the event: \mathcal{A} outputs (C^*, m^*, nr^*) such that the conditions

- (i) $C^* \leftarrow_{\mathbf{R}} \mathcal{O}_{Sc}(pk_{rcv}, m)$ was issued by \mathcal{A} , for some $m \in \mathbf{M}$;
- (ii) $1 = d \leftarrow \text{PV}(C^*, m^*, nr^*, pk_{sd}, pk_{rcv})$;
- (iii) $\mathcal{O}_N(pk_{sd}, C^*)$ was never issued by \mathcal{A} .

We consider the following simulation; when abort is set to true, the simulation aborts.

Simulation for Unforgeability of non-repudiation evidence

```

400 Initialization:  $dp \leftarrow (dp_K, dp_S, \mathcal{E}, \text{Prf}, \text{H}, \tau_0) \leftarrow_{\mathbf{R}} \text{Setup}(k)$ ;  $L \leftarrow ()$ ;  $L_1 \leftarrow ()$ ;
401  $(sk_{rcv}, pk_{rcv}) \leftarrow_{\mathbf{R}} \text{Gen}_{rcv}(dp)$ ;  $(sk_{sd}, pk_{sd}) \leftarrow_{\mathbf{R}} \text{Gen}_{sd}(dp)$ ;
402  $i_0 \leftarrow_{\mathbf{R}} [q_{Sc}]$ ;  $\text{cnt} \leftarrow 0$ ;  $\text{abort} \leftarrow \text{false}$ ;

```

403 $\boxed{E_0}$ $((\bar{\tau}_1, \bar{\tau}'_1, \bar{\tau}_2, \bar{\tau}'_2), \bar{c}_1) \leftarrow_{\mathbf{R}} \text{Ecp}(pk_{\text{rcv}});$ $\boxed{E_1}$ $((\bar{\tau}_1, \bar{\tau}'_1, \bar{\tau}_2, \bar{\tau}'_2), \bar{c}_1) \leftarrow_{\mathbf{R}} \text{Chall}_{\mathcal{K}_{E_0}};$

$\boxed{E_2, E_3}$ $((\bar{\tau}_1, \bar{\tau}'_1, \bar{\tau}_2, \bar{\tau}'_2), \bar{c}_1) \leftarrow_{\mathbf{R}} \text{Chall}_{\mathcal{K}_{E_1}};$

404 $m_0 \leftarrow \perp; \hat{m}_0 \leftarrow \perp; C_0 \leftarrow \perp;$

405 $\mathcal{O}_{\text{Sc}}(pk, m):$

406 $\text{cnt} \leftarrow \text{cnt} + 1;$

407 **if** $\text{cnt} = i_0$ **then**

408 **if** $pk \neq pk_{\text{rcv}}$ **then** $\text{abort} \leftarrow \text{true};$ ► *The guess is incorrect.*

$\boxed{E_0, E_1, E_2}$ $((\tau_1, \tau'_1, \tau_2, \tau'_2), c_1) \leftarrow ((\bar{\tau}_1, \bar{\tau}'_1, \bar{\tau}_2, \bar{\tau}'_2), \bar{c}_1);$ $\boxed{E_3}$ $\tau_1 \leftarrow \bar{\tau}_1; \tau'_1 \leftarrow \bar{\tau}'_1; c_1 \leftarrow \bar{c}_1;$

409 $c_2 \leftarrow_{\mathbf{R}} \text{E}(\tau_1, m);$

$\boxed{E_0, E_1, E_2}$ $\tau_3 \leftarrow \text{Prf}(\tau_2, m);$

$\boxed{E_3}$ Send $(p, m, s) \leftarrow (pk_{\text{sd}}, m, (c_1, c_2, pk_{\text{rcv}}))$ to the pre-image challenger;

$\boxed{E_0, E_1, E_2}$ $\hat{m} \leftarrow \text{H}(\tau_0, (pk_{\text{sd}}, \tau_2, \tau'_2, \tau_3, c_1, c_2, pk));$ $\boxed{E_3}$ Receive \hat{m} from the pre-image challenger;

412 $m_0 \leftarrow m; \hat{m}_0 \leftarrow \hat{m};$

413 **else**

414 $((\tau_1, \tau'_1, \tau_2, \tau'_2), c_1) \leftarrow_{\mathbf{R}} \text{Ecp}(pk); c_2 \leftarrow_{\mathbf{R}} \text{E}(\tau_1, m);$

415 $\tau_3 \leftarrow \text{Prf}(\tau_2, m); \hat{m} \leftarrow \text{H}(\tau_0, (pk_{\text{sd}}, \tau_2, \tau'_2, \tau_3, c_1, c_2, pk));$

416 $\sigma \leftarrow_{\mathbf{R}} \text{Sign}(sk_{\text{sd}}, \hat{m}); t \leftarrow \text{Prf}(\tau'_1, \sigma); \text{Apd}(\text{L}, (pk, m, (t, \sigma, c_1, c_2)));$

417 **if** $\text{cnt} = i_0$ **then**

418 $C_0 \leftarrow (t, \sigma, c_1, c_2);$

419 **return** $(t, \sigma, c_1, c_2);$

420 $\mathcal{O}_{\text{Usc}}(pk, C):$ $\mathcal{O}_{\mathbf{N}}(pk, C):$

421 **if** $pk = pk_{\text{sd}}$ and $C = C_0 \neq \perp$ **then**

$\boxed{\mathcal{O}_{\text{Usc}}}$ $\text{return } m_0;$ $\boxed{\mathcal{O}_{\mathbf{N}}}$ $\text{abort} \leftarrow 1;$

422 $\text{return } m_0;$ $\text{abort} \leftarrow 1;$

423 Parse C as $(t, \sigma, c_1, c_2);$

$\boxed{E_1, E_2, E_3}$ **if** $c_1 = \bar{c}_1$ **then** $\text{return } \perp;$

424 $(\tau_1, \tau'_1, \tau_2, \tau'_2) \leftarrow \text{Dcp}(sk_{\text{rcv}}, c_1); m \leftarrow \text{D}(\tau_1, c_2);$

425 $\tau_3 \leftarrow \text{Prf}(\tau_2, m); \hat{m} \leftarrow \text{H}(\tau_0, (pk, \tau_2, \tau'_2, \tau_3, c_1, c_2, pk_{\text{rcv}}));$

426 $d \leftarrow \text{Vrfy}(pk_{\text{sd}}, \hat{m}, \sigma); t' \leftarrow \text{Prf}(\tau'_1, \sigma);$

427 **if** $d = 1$ and $t = t'$ **then**

$\boxed{\mathcal{O}_{\text{Usc}}}$ $\text{return } m;$ $\boxed{\mathcal{O}_{\mathbf{N}}}$ $nr \leftarrow (\tau_1, \tau'_1, \tau_2, \tau'_2); \text{Apd}(\text{L}_1, (pk, nr, C)); \text{return } nr;$

428 $\text{return } \perp;$

429 **else** $\text{return } \perp;$

430 **Finalization:**

431 **if** \mathcal{A} outputs (C^*, m^*, nr^*) such that:

(i) $(pk_{\text{rcv}}, m, C^*) \in \text{L}$ for some $m \in \mathbf{M}$, ► C^* was generated by $\mathcal{O}_{\text{Sc}}(\cdot, \cdot)$ on input (pk_{rcv}, m) .

(ii) $1 = d \leftarrow \text{PV}(C^*, m^*, nr^*, pk_{\text{sd}}, pk_{\text{rcv}})$, and

(iii) $(pk_{\text{sd}}, nr^*, C^*) \notin \mathbf{L}_1$, ▶ nr^* was not generated by $\mathcal{O}_N(\cdot, \cdot)$ on a query on (pk_{sd}, C^*) ;
 (iv) and $C^* = C_0 = (\bar{t}, \bar{\sigma}, \bar{c}_1, \bar{c}_2)$; ▶ the simulator guessed correctly;
then
 434 Parse nr^* as $(\tau_1^*, \tau_1'^*, \tau_2^*, \tau_2'^*)$; **return** $(\tau_2^*, \tau_2'^*)$
 435 **else return** \perp ;

We consider the experiments E_i , for $i = 0, 1, 2, 3$. In E_0 \mathcal{A} plays Game 6; the simulator guesses the execution of the signcryption oracle wherein C^* will be generated, and answers \mathcal{A} 's queries consistently. Let Ev be the event \mathcal{A} succeeds and Guess be the event the simulator's guess is correct. If $\text{Ev} \wedge \text{Guess}$ occurs, the simulator outputs $(\tau_2^*, \tau_2'^*)$ such that $\hat{m}_0 \leftarrow \text{H}(\tau_0, (pk, \tau_2^*, \tau_2'^*, \tau_3^*, c_1, c_2, pk_{\text{rcv}}))$ wherein $\tau_3^* \leftarrow \text{Prf}(\tau_2^*, m_0)$. As the guess's correctness is independent from \mathcal{A} 's success,

$$\Pr(\text{Ev} \wedge \text{Guess}) = \Pr(\text{Ev})/q_{\text{Sc}}. \quad (12)$$

Let out_i denote the event $\text{Ev} \wedge \text{Guess}$ in experiment E_i , for $i = 0, 1, 2, 3$. We now consider the experiment E_1 , wherein instead of generating $(\bar{\tau}_1, \bar{\tau}_1', \bar{\tau}_2, \bar{\tau}_2', \bar{c}_1)$ for the guessed Sc query (see at lines 403 and 409), the simulator receives $(\bar{\tau}_1, \bar{\tau}_1', \bar{\tau}_2, \bar{\tau}_2', \bar{c}_1)$ from a KEM challenger as $\text{Chall}_{\mathcal{K}_{E_0}}$. In E_1 , when \mathcal{A} provides the \mathcal{O}_{Usc} or \mathcal{O}_N oracles with a signcrypted cipher text (t, σ, c_1, c_2) with $c_1 = \bar{c}_1$, the simulator returns \perp . Indeed, for such a query to succeeds (except C_0 , which is allowed only for \mathcal{O}_{Usc}), it must hold that $t = t' \leftarrow \text{Prf}(\bar{\tau}_1', \sigma)$. As $(t, \sigma, c_1, c_2) \neq (\bar{t}, \bar{\sigma}, \bar{c}_1, \bar{c}_2)$, if $(t, \sigma) \neq (\bar{t}, \bar{\sigma})$, this yields a PRF MAC forgery, otherwise (we must have $(c_1, c_2) \neq (\bar{c}_1, \bar{c}_2)$) we obtain a collision for H or colliding signatures. Hence

$$|\Pr(\text{out}_0) - \Pr(\text{out}_1)| \leq \varepsilon_{\text{Prf}}(k) + (q_{\text{Usc}} + q_N)/|\mathbf{K}| + \varepsilon_{\mathcal{S}}(k) + \varepsilon_{\text{H}}(k).$$

We consider the experiment E_2 , where the only difference compared to E_1 is that $(\bar{\tau}_1, \bar{\tau}_1', \bar{\tau}_2, \bar{\tau}_2', c_1)$ is received from a KEM challenger as $\text{Chall}_{\mathcal{K}_{E_1}}$ instead of $\text{Chall}_{\mathcal{K}_{E_0}}$. It holds that

$$|\Pr(\text{out}_1) - \Pr(\text{out}_2)| \leq \varepsilon_{\mathcal{K}}(k).$$

In experiment E_3 , the challenger receives $(\bar{\tau}_1, \bar{\tau}_1', \bar{\tau}_2, \bar{\tau}_2', c_1)$ as $\text{Chall}_{\mathcal{K}_{E_1}}$ from the KEM challenger, however it does not use $\bar{\tau}_2$ and $\bar{\tau}_2'$, instead the values of $\bar{\tau}_2$ and $\bar{\tau}_2'$ are generated by a pre-image challenger, as $\bar{\tau}_2$ and $\bar{\tau}_2'$ are generated following the same distribution as at the KEM challenger, it follows that

$$\Pr(\text{out}_2) = \Pr(\text{out}_3).$$

Now if out_3 occurs, the simulator succeeds in its pre-image game. So, from Lemma 1,

$$\Pr(\text{out}_3) \leq 1/|\mathbf{K}| + \varepsilon_{\text{H}}(k).$$

And then,

$$\begin{aligned} \Pr(\text{Ev})/q_{\text{Sc}} = \Pr(\text{out}_0) &\leq |\Pr(\text{out}_0) - \Pr(\text{out}_1)| + |\Pr(\text{out}_1) - \Pr(\text{out}_3)| + \Pr(\text{out}_3) \\ &\leq \varepsilon_{\text{Prf}}(k) + (q_{\text{Usc}} + q_N + 1)/|\mathbf{K}| + \varepsilon_{\mathcal{S}}(k) + \varepsilon_{\mathcal{K}}(k) + 2\varepsilon_{\text{H}}(k). \end{aligned}$$

□

5 Comparison with Previous Constructions

As far as we are aware, only Chiba *et al.* [10] propose generic constructions of insider secure signcryption schemes (in the dynamic multi-user model) in the standard model. They propose two generic designs, we refer to by CMSM1 [10, Sect. 4.1] and CMSM2 [10, Sect. 4.2]. Both constructions use as building blocks:

- an IND-CCA-secure symmetric encryption scheme (only semantic security is required for CMSM2), and
- a sUF-CMA-secure signature scheme.

The construction CMSM1 uses also an IND-CCA-secure tag-based-KEM (a KEM which takes a tag as additional input for encapsulation/decapsulation).

The design CMSM2 uses as additional building blocks:

- an IND-CCA-secure KEM, and
- a one-to-one and sUF-OT secure MAC.

In comparison, in our design, we use as building blocks:

- a semantically secure symmetric encryption scheme,
- a sUF-CMA-secure signature scheme,
- an IND-CCA-secure KEM,
- a collision resistant hash function, and
- a secure pseudo-random function.

Although tag-based-KEMs can be built from any IND-CCA-secure public key encryption scheme [10], KEMs seem to be more common. For instance, cryptography standards, such as HPKE [6], use KEMs as building block, not tag-based-KEMs. And, any tag-based KEM can be transformed into a KEM, by using an empty tag. In this respect, compared to CMSM1, the SN scheme uses more common low level primitives.

The construction CMSM2 uses very common low level primitives. Unfortunately, to achieve strong unforgeability, there is a significant restriction on the MAC, which is required to be one-to-one, *i. e.* it is required that given a key τ and a message m , there is *one and only one* t such that $\text{MAC}(\tau, m) = t$. This requirement excludes a large class of hash based MACs such as HMAC [16], UMAC [17], or KMAC [15]. The same restriction exists on the encryption scheme; this precludes the use a randomized encryption scheme, such as a bloc cipher with a mode of operation using a (pseudo-)random initialization vector, for instance. In comparison, in the SN construction, we require the signature scheme to be resistant against colliding signatures and the encryption scheme to be resistant against clustering key attacks. In many signatures, wherein the message to be signed is hashed first (the Full Domain Hash [14], for instance), colliding signatures yield a digest collision. The requirement is then naturally achieved in usual signature schemes. And, given the commonly required avalanche effect in substitution permutation network based encryption schemes (each cipher-text bit is changed with probability $1/2$, when a single bit of the key is modified), one can reasonably expect common encryption schemes to be resistant against key clustering attacks. To instantiate the PFR, given the public parameter τ_0 and a secure block cipher, from the PFR-PRP switching lemma [9, p. 134], $\text{Prf}(\tau, x)$ can be computed using the instructions:

```

500 Prf( $\tau, x$ ):
501  $x' \leftarrow H(\tau_0, x)$ ;  $t \leftarrow E(\tau, x')$ ; return  $t$ ;

```

It appears that, compared to CMSM2, the SN scheme offers a wider range of choices for an instantiation of the low level primitives. This may be of prime importance in a constrained environment wherein only a limited number of low level primitives can be implemented.

Contrary to Tan's design [23] and the generic constructions from [2] and [19], the SN scheme does not require the registered key model; it then offers a superior security. Also, compared to the constructions from [20–22], SN does not use the random oracle model. Another security advantage of the SN scheme compared to these constructions is its generic nature; it can be instantiated with adequate present and future (including quantum-resistant) primitives.

From an efficiency perspective, the computational cost of the CMSM1, CMSM2, and SN schemes, comes mainly from the asymmetric operations (the cost of the symmetric operations is usually neglected): encapsulation and signature for signcryption, and decapsulation and signature verification, for unsigncryption. Given that any tag-based-KEM can be transformed (for free) into a KEM, for any instantiation of CMSM1 or CMSM2, there is an instantiation of SN that achieves the same efficiency for the asymmetric operations, if not better. For a comparison with direct constructions [20–23], SN can be instantiated with any signature scheme \mathcal{S} and symmetric encryption scheme \mathcal{E} , and an appropriate KEM, PRF and hash function, provided \mathcal{S} is strongly unforgeable and \mathcal{S} is semantically secure and the KEM is IND-CCA-secure. Given that hash and PRF evaluations are negligible compared to signature and KEM operations, SN will yield a comparable efficiency.

The bit length of a CMSM1 signcrypted cipher-text corresponding to a message m is the bit length of m (assuming that the encryption scheme \mathcal{E} is length preserving) added with that of a signature on m and that of a encapsulated key, *i.e.* $sz(m) + sz(\text{Sign}(sk_{sd}, m)) + sz(\text{Ecp}(sk_{sd}, pk_{rcv}))$, where sk_{sd} and pk_{rcv} are respectively the sender's private key and the receiver's public key. The CMSM2 and SN schemes add to this quantity the size of a MAC (a PRF based MAC in the case of SN). So, the SN and CMSM2 have the same communication overhead, which is slightly greater than that of CMSM1.

An interesting feature of the SN scheme, is that all the security reductions are tight, except for the unforgeability of non-repudiation evidence wherein we use a guessing strategy. A concrete instance of SN may be re-analyzed for unforgeability of non-repudiation evidence, if the underlying KEM is build upon a random self-reducible problem.

A Proof of Lemma 1

Let \mathcal{A} be an adversary playing Game 1. We build an adversary \mathcal{B} against the collision (with identical prefix) resistance of H as follows.

- 1) \mathcal{B} receives $\tau_0 \leftarrow_{\mathbf{R}} \mathbf{K}$ from its challenger and sends τ_0 to \mathcal{A} .
- 2) When \mathcal{B} receives (p_0, m_0, s_0) from \mathcal{A} , it chooses $(\tau, \tau') \leftarrow_{\mathbf{R}} \mathbf{K}^2$ and computes $\tau_0'' \leftarrow \text{Prf}(\tau', m_0)$, $\hat{m}_0 \leftarrow H(\tau_0, (p_0, \tau, \tau', \tau_0'', s_0))$ and sends \hat{m}_0 to \mathcal{A} .

- 3) When \mathcal{A} outputs (τ^*, τ'^*) such that $\hat{m}_0 = \hat{m}_0^* \leftarrow H(\tau_0, (p_0, \tau^*, \tau'^*, \tau''^*, s_0))$ wherein $\tau''^* \leftarrow \text{Prf}(\tau'^*, m_0)$, if $(\tau, \tau') \neq (\tau^*, \tau'^*)$ then \mathcal{B} outputs (s, s') wherein $s = (p_0, \tau, \tau', \tau'', s_0)$ and $s' = (p_0, \tau^*, \tau'^*, \tau''^*, s_0)$ as messages with identical prefix p_0 and colliding hashes under τ_0 .

Let bad be the event: the chosen pair (τ, τ') is such that for all $(\bar{\tau}, \bar{\tau}') \neq (\tau, \tau')$, $\hat{m}_0 \neq H(\tau_0, (p_0, \bar{\tau}, \bar{\tau}', \bar{\tau}'', s_0))$, *i. e.* there is no other pair $(\bar{\tau}, \bar{\tau}') \in \mathbf{K}^2$ such that $H(\tau_0, (p_0, \bar{\tau}, \bar{\tau}', \bar{\tau}'', s_0)) = H(\tau_0, (p_0, \tau, \tau', \tau'', s_0))$. It holds that

$$\Pr(\text{bad}) \leq |\mathbf{T}|/|\mathbf{K}|^2.$$

If $\text{Succ}_{\mathcal{A}, \text{H}}$ denotes the event \mathcal{A} succeeds in Game 1,

$$\begin{aligned} \Pr(\text{Succ}_{\mathcal{A}, \text{H}}) &= \Pr(\text{Succ}_{\mathcal{A}, \text{H}} \wedge \text{bad}) + \Pr(\text{Succ}_{\mathcal{A}, \text{H}} \wedge \neg \text{bad}) \\ &\leq \Pr(\text{bad}) + \Pr(\text{Succ}_{\mathcal{A}, \text{H}} \wedge \neg \text{bad}). \end{aligned}$$

Now let Eq be the event $(\tau, \tau') = (\tau^*, \tau'^*)$.

$$\Pr(\text{Succ}_{\mathcal{A}, \text{H}} \wedge \neg \text{bad}) = \Pr(\text{Succ}_{\mathcal{A}, \text{H}} \wedge \neg \text{bad} \wedge \text{Eq}) + \Pr(\text{Succ}_{\mathcal{A}, \text{H}} \wedge \neg \text{bad} \wedge \neg \text{Eq}).$$

Now, as if $\text{Succ}_{\mathcal{A}, \text{H}} \wedge \neg \text{bad}$ occurs, there at least one $(\tau^*, \tau'^*) \neq (\tau, \tau')$ such that $\hat{m}_0 = \hat{m}_0^* \leftarrow H(\tau_0, (p_0, \tau^*, \tau'^*, \tau''^*, s_0))$, and \mathcal{A} has no information about (τ, τ') besides \hat{m}_0 , it holds that

$$\Pr(\text{Succ}_{\mathcal{A}, \text{H}} \wedge \neg \text{bad} \wedge \text{Eq}) \leq \Pr(\text{Succ}_{\mathcal{A}, \text{H}} \wedge \neg \text{bad} \wedge \neg \text{Eq}).$$

Hence

$$\Pr(\text{Succ}_{\mathcal{A}, \text{H}}) \leq |\mathbf{T}|/|\mathbf{K}|^2 + 2 \Pr(\text{Succ}_{\mathcal{A}, \text{H}} \wedge \neg \text{bad} \wedge \neg \text{Eq}).$$

And, whenever $\text{Succ}_{\mathcal{A}, \text{H}} \wedge \neg \text{bad} \wedge \neg \text{Eq}$ occurs \mathcal{B} outputs s, s' with identical prefix such that $H(\tau_0, s) = H(\tau_0, s')$. \square

References

1. An, J.H., Rabin, T.: Security for signcryption: the two-user model. In: Dent, A., Zheng, Y. (eds.) Practical Signcryption, pp. 21–42. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-540-89411-7_2
2. Arriaga, A., Barbosa, M., Farshim, P.: On the joint security of signature and encryption schemes under randomness reuse: efficiency and security amplification. In: Bao, F., Samarati, P., Zhou, J. (eds.) ACNS 2012. LNCS, vol. 7341, pp. 206–223. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31284-7_13
3. Badertscher, C., Banfi, F., Maurer, U.: A constructive perspective on signcryption security. In: Catalano, D., De Prisco, R. (eds.) SCN 2018. LNCS, vol. 11035, pp. 102–120. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-98113-0_6
4. Baek, J., Steinfeld, R.: Security for signcryption: the multi-user model. In: Dent, A., Zheng, Y. (eds.) Practical Signcryption, pp. 43–53. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-540-89411-7_3
5. Baek, J., Steinfeld, R., Zheng, Y.: Formal proofs for the security of signcryption. *J. Cryptol.* **20**(2), 203–235 (2007)

6. Barnes, R., Bhargavan, K., Lipp, B., Wood, C.: RFC 9180: Hybrid public key encryption (2022)
7. Bao, F., Deng, R.H.: A signcryption scheme with signature directly verifiable by public key. In: Imai, H., Zheng, Y. (eds.) PKC 1998. LNCS, vol. 1431, pp. 55–59. Springer, Heidelberg (1998). <https://doi.org/10.1007/BFb0054014>
8. Bjørstad, T.E., Dent, A.W.: Building better signcryption schemes with tag-KEMs. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T. (eds.) PKC 2006. LNCS, vol. 3958, pp. 491–507. Springer, Heidelberg (2006). https://doi.org/10.1007/11745853_32
9. Boneh, D., Shoup, V.: A graduate course in applied cryptography. Draft 0.6 (2023). <https://toc.cryptobook.us/>
10. Chiba, D., Matsuda, T., Schuldt, J.C.N., Matsuura, K.: Efficient generic constructions of signcryption with insider security in the multi-user setting. In: Lopez, J., Tsudik, G. (eds.) ACNS 2011. LNCS, vol. 6715, pp. 220–237. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21554-4_13
11. Ducklin, P.: Serious security: google finds fake but trusted SSL certificates for its domains, made in France. Naked Security-Award-Winning Computer Security, News, Opinion, Advice and Research from SOPHOS, pp. 09–12 (2013)
12. Fan, J., Zheng, Y., Tang, X.: Signcryption with non-interactive non-repudiation without random oracles. In: Gavrilova, M.L., Tan, C.J.K., Moreno, E.D. (eds.) Transactions on Computational Science X. LNCS, vol. 6340, pp. 202–230. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17499-5_9
13. Fisher, D.: Final Report on DigiNotar Hack Shows Total Compromise of CA Servers. Threatpost, 10/31/12. <https://threatpost.com/final-report-diginotar-hack-shows-total-compromise-ca-servers-103112/77170/>
14. Kakvi, S.A., Kiltz, E.: Optimal security proofs for full domain hash, revisited. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 537–553. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_32
15. Kelsey, J., Chang, S.J., Perlmutter, R.: SHA-3 derived functions: cSHAKE, KMAC, TupleHash, and ParallelHash. NIST Special Publication, vol. 800, p. 185 (2016)
16. Krawczyk, H., Bellare, M., Canetti, R.: RFC2104: HMAC: Keyed-hashing for message authentication (1997)
17. Krovetz, T. (Ed.): RFC 4418: UMAC: Message Authentication Code using Universal Hashing (2006)
18. Malone-Lee, J.: Signcryption with non-interactive non-repudiation. Des. Codes Cryptogr. **37**(1), 81–109 (2005)
19. Matsuda, T., Matsuura, K., Schuldt, J.C.N.: 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). https://doi.org/10.1007/978-3-642-10628-6_22
20. Ngarenon, T., Sarr, A.P.: A Computational Diffie-Hellman based Insider Secure Signcryption with Non Interactive Non Repudiation (full version) (2022). <https://hal.science/hal-03628351/document>
21. Ngarenon, T., Sarr, A.P.: A computational Diffie-Hellman based insider secure signcryption with non-interactive non-repudiation. In: Rushi Kumar, B., Ponnusamy, S., Giri, D., Thuraisingham, B., Clifton, C.W., Carminati, B. (eds.) ICMC 2022. Springer Proceedings in Mathematics & Statistics, vol. 415. Springer, Cham (2023). https://doi.org/10.1007/978-981-19-9307-7_8
22. Sarr, A.P., Seye, P.B., Ngarenon, T.: A practical and insider secure signcryption with non-interactive non-repudiation. In: Carlet, C., Guilley, S., Nitaj, A., Souidi, E.M. (eds.) C2SI 2019. LNCS, vol. 11445, pp. 409–429. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-16458-4_24

23. Tan, C.H.: 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). https://doi.org/10.1007/978-3-540-89598-5_5
24. Zheng, Y.: Digital signcryption or how to achieve $\text{cost}(\text{signature} \ \& \ \text{encryption}) \ll \text{cost}(\text{signature}) + \text{cost}(\text{encryption})$. In: Kaliski, B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 165–179. Springer, Heidelberg (1997). <https://doi.org/10.1007/BFb0052234>