A Practical and Insider Secure Signcryption with Non-interactive Non-repudiation

Augustin P. Sarr(✉), Papa B. Seye, and Togdé Ngarenon

Lacca, UFR SAT, Université Gaston Berger de Saint-Louis, Saint-Louis, Senegal
aug.sarr@gmail.com

Abstract. Signcryption with non-interactive non-repudiation is a public key primitive which aims at combining the functionalities of encryption and signature schemes, while offering to a judge the ability to settle a repudiation dispute without engaging in a costly multi-round protocol. We propose a new RSA based identification scheme together with a strongly unforgeable signature scheme. We derive a practical and efficient signcryption scheme with non-interactive non-repudiation we show to be insider secure, under the RSA assumption and the Random Oracle model. The communication overhead of our signcryption scheme, compared to the corresponding signature scheme is one group element.

Keywords: Identification · Signature · Signcryption · Insider security · Non-interactive non-repudiation · Signed quadratic residues

1 Introduction

Signcryption is a public key primitive introduced by Zheng [23], with the aim of combining the functionalities of encryption and signature schemes. Since Zheng's seminal work, many security models and constructions have been proposed [3]. In a recent work, Badertscher *et al.* [2] consider, from an application-centric perspective, the security goals a signcryption scheme should achieve depending on the secret keys the attacker knows. They conclude, in opposition to [3, p. 29], that insider security should be considered as the standard security goal.

An important attribute which is not considered in the "standard" insider security model is *non-interactive non-repudiation*. As discussed in [2], the natural usage of signcrytion is to achieve a confidential and authenticated channel between two parties over an insecure network. The same can be achieved using non-interactive or one pass-key exchange protocols, which often outperform signcryption schemes. So, a major benefit of signcryption schemes compared

This Research was supported by the African Center of Excellence in Mathematics, Computer Science and ICT of UGB.

C. Carlet et al. (Eds.): C2SI 2019, LNCS 11445, pp. 409–429, 2019.
https://doi.org/10.1007/978-3-030-16458-4_24

to non-interactive and one-pass key exchange is non-interactive non-repudiation (NINR), *i.e.* a non-repudiation attribute wherein a judge does not have to engage in a costly multi-round interactive protocol to settle a repudiation dispute.

A first attempt to achieve NINR in a signcryption design was proposed by Bao and Deng [6]. Unfortunately their scheme fails in providing both NINR and confidentiality [17,22]. In [17], Malone–Lee propose a design with NINR. However, he analyses his design, under the Gap Diffie–Hellman Assumption [19] and the Random Oracle (RO) model [8], in a security definition which is closer to the outsider model than to the insider one [3, Chap. 2–4]. Fan *et al.* [11] propose a strengthening of Malone–Lee's security model which considers, not only confidentiality and unforgeability in the insider model, but also soundness and unforgeability of non-repudiation evidence. They propose a design they show to be insider secure under the Decisional Bilinear Diffie–Hellman assumption, without resorting to the RO model.

In this paper, we propose a new identification scheme, inspired from the FXCR [20,21] and Guillou–Quisquater (GQ) [13] schemes, over the group of signed quadratic residues [14].

We derive a signature scheme which is strongly unforgeable against chosen message attacks. A significant advantage of our signature scheme, compared to the FXCR or GQ schemes is that it is defined over a group wherein the strong Diffie–Hellman assumption is known to hold under the factoring assumption [14]. Then, using a variant of Cash *et al.*'s trapdoor test technique [10], we derive a signcryption scheme with non-interactive non-repudiation (SCNINR) we show to be insider secure, under the RSA assumption and the RO model, in a variant of Fan *et al.*'s security definition [11].

This paper is organized as follows. In Sect. 2, we present some preliminaries. In Sect. 3, we propose the identification scheme, discuss its attributes, and derive the signature scheme. We present the new SCNINR scheme and its security arguments in Sect. 4.

2 Preliminaries

Notations. If n is an integer, $|n|$ denotes its bit-length and $[n]$ denotes the set $\{0, \cdots, n\}$. For a real l, $\lceil l \rceil$ denotes the smallest integer which is greater than or equal to l. We refer to the length of a list $\mathcal{L}$ by $|\mathcal{L}|$, and to the cardinality of a set S by $|S|$. If P is a probabilistic algorithm which takes as parameters $u_1, \cdots, u_n$ and outputs a result V which belongs to a set $\mathbf{V}$, we write $V \leftarrow_{\mathrm{R}} P(u_1, \cdots, u_n)$. We denote by $\{P(u_1, \cdots, u_n)\}$ the set $\{v \in \mathbf{V} : \Pr(V = v) \neq 0\}$. If S is a set, the notation $a \leftarrow_{\mathrm{R}} S$ means that a is chosen uniformly at random from S. $\mathsf{Exp}(\mathbb{Z}_N, t, l)$ denotes the computational effort required to perform t exponentiations with l bit exponents in $\mathbb{Z}_N$; $\mathsf{Exp}(\mathbb{Z}_N, l)$ stands for $\mathsf{Exp}(\mathbb{Z}_N, 1, l)$. $\mathsf{Jcb}(\mathbb{Z}_N)$ denotes the effort required to compute a Jacobi symbol in $\mathbb{Z}_N$. For two bit strings m_1 and m_2, $m_1||m_2$ denotes their concatenation; ϵ denotes the empty string. If $x_1, x_2, \cdots, x_k$ are objects belonging to different structures (group, bit-string, etc.) $(x_1, x_2, \cdots, x_k)$ denotes a representation of the tuple such that each component can be unequivocally parsed.

RSA Public Key Generator. Let k be a security parameter, $n(k)$ be a function of k and $0 \leqslant \delta < 1/2$ be a constant. An algorithm RSAGen (which may be distributed) is said to be a $(n(k), \delta)$ RSA public key generator if on input 1^k, it outputs a $n(k)$ bit Blum integer $N = pq$ together with a public exponent e such that all the prime factors of $\phi(N)/4$ are: (i) pairwise distinct, and (ii) at least δn bit integers, and (iii) e is a $(k+1)$ bit prime.

RSA and Factoring Assumptions. Let $\mathcal{A}$ be an algorithm. We define the quantity

$$\mathsf{Adv}^{\mathrm{RSA}}_{\mathcal{A},\mathsf{RSAGen}}(k) = \Pr\left[\begin{array}{l}(N,e) \leftarrow_{\mathrm{R}} \mathsf{RSAGen}(1^k); x \leftarrow_{\mathrm{R}} \mathbb{Z}_N; \\ y \leftarrow x^e \mod N; \hat{x} \leftarrow_{\mathrm{R}} \mathcal{A}(N,e,y)\end{array} : \hat{x} = x\right].$$

The RSA assumption for an $(n(k), \delta)$ RSA public key generator is said to hold if for all efficient adversary $\mathcal{A}$, $\mathsf{Adv}^{\mathrm{RSA}}_{\mathcal{A},\mathsf{RSAGen}}(k)$ is negligible. For an instance $(N,e) \leftarrow_{\mathrm{R}} \mathsf{RSAGen}(1^k)$ and an efficiently sampleable and recognizable subset J of $\mathbb{Z}_N$, we say that the RSA problem is $(t(k), \varepsilon(k))$ hard in J, if for all $\mathcal{A}$ running in time at most t, $\Pr[x \leftarrow_{\mathrm{R}} \mathsf{J}; y \leftarrow x^e \mod N; \hat{x} \leftarrow_{\mathrm{R}} \mathcal{A}(N,e,y) : \hat{x} = x] \leqslant \varepsilon$.

Let $\mathcal{A}$ be a factoring algorithm and

$$\mathsf{Adv}^{\mathsf{fac}}_{\mathcal{A},\mathsf{RSAGen}}(k) = \Pr\left[\begin{array}{l}(N,e) \leftarrow_{\mathrm{R}} \mathsf{RSAGen}(k); \\ p \leftarrow_{\mathrm{R}} \mathcal{A}(N,e)\end{array} : \ p \mid n \text{ and } p \notin \{\pm N, \pm 1\}\right].$$

The factoring assumption for an (n, δ) RSA public key generator is said to hold if for all efficient adversary $\mathcal{A}$, $\mathsf{Adv}^{\mathsf{fac}}_{\mathcal{A},\mathsf{RSAGen}}(k)$ is negligible.

Diffie–Hellman Assumptions. Let $\mathcal{G} = \langle G \rangle$ be a cyclic group, which order is a function of the security parameter k and is not necessarily known. For $X \in \mathcal{G}$, $\log_G X$ denotes the smallest non-negative integer x such that $G^x = X$. For, $X, Y \in \mathcal{G}$, we denote $G^{(\log_G X)(\log_G Y)}$ by $\mathrm{CDH}(X,Y)$. The *computational Diffie–Hellman (CDH) Assumption* is said to hold in $\mathcal{G}$ if for all efficient algorithm $\mathcal{A}$,

$$\mathsf{Adv}^{\mathrm{CDH}}_{\mathcal{A}}(\mathcal{G}) = \Pr[X \leftarrow_{\mathrm{R}} \mathcal{G}; Y \leftarrow_{\mathrm{R}} \mathcal{G}; Z \leftarrow_{\mathrm{R}} \mathcal{A}(G,X,Y) : Z = \mathrm{CDH}(X,Y)]$$

is negligible in k. The *strong Diffie–Hellman (sCDH) assumption* is said to hold in $\mathcal{G}$ if the CDH assumption holds even if $\mathcal{A}$ is endowed with a decisional Diffie–Hellman oracle $\mathcal{O}_{\mathrm{DDH},X}(\cdot,\cdot)$ for a some fixed X, which on input $U, V \in \mathcal{G}$ outputs 1 if $V = \mathrm{CDH}(X,U)$ and 0 otherwise.

Signed Quadratic Residues. For an odd integer N, we consider $\{-(N-1)/2, \cdots, (N-1)/2\}$ as a set of representatives of the residue classes modulo N. We denote by $\mathbb{J}_N$ the subgroup of elements of $\mathbb{Z}^*_N$ with Jacobi symbol 1, and consider the quotient group $\mathbb{J}_N/\{-1,1\}$. We define $\mathbb{J}^+_N = \mathbb{J}_N \cap \{1, \cdots, (N-1)/2\}$, and the binary operation $\circ$ over $\mathbb{J}^+_N$ by $X \circ Y = |X \cdot Y \mod N|$. For $X \in \mathbb{J}^+_N$ and $t \in \mathbb{N}$, we write $X^{\underline{t}}$ for $\overbrace{X \circ \cdots \circ X}^{t \text{ times}} = |X^t \mod N| \in \mathbb{J}^+_N$. Then $(\mathbb{J}^+_N, \circ)$ is a group, termed group of signed quadratic residues. Moreover the mapping which associates $\{-X, X\} \in \mathbb{J}_N/\{-1,1\}$ to $|X| \in \mathbb{J}^+_N$ is an isomorphism. We identify the quotient group $\mathbb{J}_N/\{-1,1\}$ with $\mathbb{J}^+_N$. From [14], we have the following Lemma.

Lemma 1. *If N is a Blum integer then (a) $(\mathbb{J}_N^+, \circ)$ is a subgroup of $\mathbb{Z}_N^*$ of order $\phi(N)/4$; (b) $\mathbb{J}_N^+$ is efficiently recognizable given only N; and (c) if $\mathbb{J}_N$ is cyclic then so is $\mathbb{J}_N^+$.*

Canonical Identification Schemes

Definition 1. *A* canonical identification scheme $\mathcal{I} = (\mathsf{Gen}, \mathsf{P}, \mathsf{V}, \mathsf{ChSet})$ *is a triple of algorithms together with a challenge set, such that:*

- Gen *is a probabilistic algorithm which takes as input a domain parameters dp and returns a key pair (sk, pk).*
- $\mathsf{P} = (\mathsf{P}_1, \mathsf{P}_2)$ *is a pair of algorithms such that: (i) P_1 takes as input a secret key sk and outputs a commitment X together with a state st; and (ii) P_2 takes as inputs a private key sk, a commitment X, a challenge $c \in \mathsf{ChSet}$, and a state st and outputs a response $s \in \{0,1\}^*$.*
- V *is a deterministic verification algorithm which takes as inputs a public key pk, a commitment X, a challenge c, and a response s and outputs $d \in \{0,1\}$.*
- *And, for all $(sk, pk) \in \{\mathsf{Gen}(dp)\}$, all $(X, st) \in \{\mathsf{P}_1(sk)\}$, all $c \in \mathsf{ChSet}$, and all $s \in \{\mathsf{P}_2(sk, X, c, st)\}$, $\mathsf{V}(pk, X, c, s) = 1$.*

A transcript (X, c, s) is said to be *accepting* with respect to pk if $\mathsf{V}(pk, X, c, s) = 1$.

An identification scheme is said to be *unique* if for all $(sk, pk) \in \{\mathsf{Gen}(dp)\}$, all $(X, st) \in \{\mathsf{P}_1(sk)\}$, and all $c \in \mathsf{ChSet}$, there is at most one $s \in \{0,1\}^*$ such that $\mathsf{V}(pk, X, c, s) = 1$. It is said to have α-bits of min entropy if for all $(sk, pk) \in \{\mathsf{Gen}(dp)\}$, the commitments generated through $\mathsf{P}_1(sk)$ are chosen from a distribution with min entropy at least α; *i.e.*, for all commitment X_0, if $(X, st) \leftarrow_{\mathrm{R}} \mathsf{P}_1(sk)$ was honestly generated then $\Pr(X = X_0) \leqslant 2^{-\alpha}$.

Definition 2. *Let $\mathcal{I} = (\mathsf{Gen}, \mathsf{P}, \mathsf{V}, \mathsf{ChSet})$ be a canonical identification scheme.*

(a) $\mathcal{I}$ is said to provide special soundness (SpS) *if there exists an efficient deterministic algorithm* Ext *(an* extractor*) such that for all accepting conversations with respect to a public key pk, (X, c, s) and (X, c', s'), if $c \neq c'$ then $sk^* \leftarrow \mathsf{Ext}(pk, X, c, s, c', s')$ is such that $(sk^*, pk) \in \{\mathsf{Gen}(dp)\}$.*

(b) It is said to be honest verifier zero knowledge (HVZK) *if there exists an efficient probabilistic algorithm* sim *(a* simulator*) such that for all $(sk, pk) \in \{\mathsf{Gen}(dp)\}$, the output distribution of* sim *on input pk is identical to that of a real transcript between $\mathsf{P}(sk)$ and $\mathsf{V}(pk)$.*

(c) It is said to be random self reducible (RSR) *if there is a probabilistic algorithm* Rerand *together with two deterministic algorithms* Tran *and* Derand *such that for all $(sk, pk) \in \{\mathsf{Gen}(dp)\}$:*
  - *if $(\tau, pk_1) \leftarrow_R \mathsf{Rerand}(pk)$ and $(sk_2, pk_2) \leftarrow_R \mathsf{Gen}(dp)$ then pk_1 and pk_2 have the same distribution;*
  - *for all $(sk_1, pk_1) \in \{\mathsf{Gen}(dp)\}$, for all τ such that $(\tau, pk_1) \in \{\mathsf{Rerand}(pk)\}$, if $sk^* \leftarrow \mathsf{Derand}(pk, pk_1, sk_1, \tau)$ then $(sk^*, pk) \in \{\mathsf{Gen}(dp)\}$;*

– *for all* $(sk_1, pk_1) \in \{\mathsf{Gen}(dp)\}$ *and all* (X, c, s_1) *such that* $\mathsf{V}(pk_1, X, c, s_1) = 1$, *if* $(X, c, s) \leftarrow \mathsf{Tran}(pk, pk_1, \tau, (X, c, s_1))$ *then* $\mathsf{V}(pk, X, c, s) = 1$.

Definition 3. *A canonical identification scheme* $\mathcal{I} = (\mathsf{Gen}, \mathsf{P}, \mathsf{V}, \mathsf{ChSet})$ *is said to be* (t, ε)*-secure against Key Recovery against Key Only Attacks (KR-KOA), if for all adversary* $\mathcal{A}$ *running in time at most* t

$$\Pr\left[(sk, pk) \leftarrow_R \mathsf{Gen}(dp); sk^* \leftarrow_R \mathcal{A}(pk) : (sk^*, pk) \in \{\mathsf{Gen}(dp)\}\right] \leqslant \varepsilon.$$

Symmetric Encryption, Digital Signature

Definition 4. *A symmetric encryption scheme* $\mathcal{E} = (\mathsf{E}, \mathsf{D}, \mathbf{K}(k), \mathbf{M}(k), \mathbf{C}(k))$ *is a pair of efficient algorithms* (E, D) *together with a triple of sets* $(\mathbf{K}(k), \mathbf{M}(k), \mathbf{C}(k))$ *such that for all* $\tau \in \mathbf{K}$ *and all* $m \in \mathbf{M}$, $\mathsf{E}(\tau, m) \in \mathbf{C}$, $m = \mathsf{D}(\tau, \mathsf{E}(\tau, m))$.

Definition 5. *Let* $\mathcal{A}$ *be an adversary against an encryption scheme* $\mathcal{E}$*; its* semantic security advantage *is*

$$\mathsf{Adv}^{\mathsf{ss}}_{\mathcal{A},\mathcal{E}}(k) = \left| \Pr\left[\begin{matrix}(m_0, m_1) \leftarrow_R A(1^k); \tau \leftarrow_R \mathbf{K}; b \leftarrow_R \{0,1\}; \\ c \leftarrow \mathsf{E}(\tau, m_b); \hat{b} \leftarrow_R \mathcal{A}(1^k, c)\end{matrix} : \hat{b} = b\right] - \frac{1}{2} \right|,$$

where $m_0, m_1 \in \mathbf{M}$ *are distinct equal length messages. The scheme* $\mathcal{E}$ *is said to be* (t, ε)-semantically secure *if for all adversary* $\mathcal{A}$ *running in time* t $\mathsf{Adv}^{\mathsf{ss}}_{\mathcal{A},\mathcal{E}}(k) \leqslant \varepsilon$.

Definition 6. *A signature scheme* $\mathcal{S} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ *is a triple of efficient algorithms together with a message space* $\mathbf{M}$*, such that:*

- Gen *is probabilistic algorithm which takes as input a domain parameter* dp *and returns a key pair* (sk, pk)*;*
- Sign *is a probabilistic algorithm which takes as inputs a secret key* sk *and a message* $m \in \mathbf{M}$ *and outputs a signature* σ*;*
- Vrfy *is a deterministic algorithm which takes as inputs a public key* pk*, a message* m*, and a signature* σ *and outputs* $d \in \{0, 1\}$*; and*
- *for all* $(sk, pk) \in \{\mathsf{Gen}(dp)\}$*, all* $m \in \mathbf{M}$, $\Pr[\mathsf{Vrfy}(pk, m, \mathsf{Sign}(sk, m)) = 1] = 1$.

Game 1. MU-SUF-CMA security game

1) For $i \in [U]$, $(sk_i, pk_i) \leftarrow_{\mathrm{R}} \mathsf{Gen}(dp)$;
2) $(i_0, m_0, \sigma_0) \leftarrow_{\mathrm{R}} \mathcal{A}^{\mathcal{O}_{\mathsf{H}}(\cdot), \mathcal{O}_{\mathsf{Sign}}(\cdot,\cdot)}(pk_1, \cdots, pk_U)$, wherein $\mathcal{O}_{\mathsf{H}}(\cdot)$ is a hashing oracle and $\mathcal{O}_{\mathsf{Sign}}(\cdot, \cdot)$ a signing oracle which takes as inputs an index $j \in [U]$ together with a message m and outputs $\sigma \leftarrow_{\mathrm{R}} \mathsf{Sign}(sk_j, m)$.
3) $\mathcal{A}$ succeeds if : (a) $i_0 \in [U]$ and $\mathsf{Vrfy}(pk_{i_0}, m_0, \sigma_0) = 1$, and (b) σ_0 was not received from the oracle $\mathcal{O}_{\mathsf{Sign}}(\cdot, \cdot)$ on a query on (i_0, m_0).

Definition 7. *Let* $\mathcal{S} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ *be a signature scheme such that the execution of* Sign *involves the computation of one digest value, at least.* $\mathcal{S}$ *is said to be* $(t, U, Q_{\mathsf{Sign}}, Q_{\mathsf{H}}, \varepsilon)$ *multi-user strongly unforgeable against chosen message* *attacks (MU-SUF-CMA) in the RO model, if for all adversary $\mathcal{A}$ playing Game 1 (wherein we consider U and dp as implicit parameters), if $\mathcal{A}$ runs in time at most t, issues at most Q_{Sign} and Q_{H} queries to the signing and hashing oracles respectively, the probability it succeeds is at most ε.*

Signcryption Schemes

Definition 8. *A signcryption scheme is a quintuple of algorithms $\mathcal{SC} = (\mathsf{Setup}, \mathsf{Gen}_S, \mathsf{Gen}_R, \mathsf{Sc}, \mathsf{Usc})$ wherein:*

(a) Setup *is a probabilistic algorithm which takes a security parameter 1^k as input, and outputs a domain parameter dp.*
(b) Gen_S *is a probabilistic algorithm which takes as input a domain parameter dp and outputs a sender key pair (sk_S, pk_S) wherein sk_S is the signing key.*
(c) Gen_R *is a probabilistic algorithm which takes dp as input and outputs a receiver key pair (sk_R, pk_R).*
(d) Sc *is a probabilistic algorithm which takes as inputs dp, a sender private key sk_S and a receiver public key pk_R, and outputs a signcrypted text C. We consider dp as an implicit parameter and write $C \leftarrow_R \mathsf{Sc}(sk_S, pk_R, m)$.*
(e) Usc *is a deterministic algorithm which takes as input dp, a sender public key pk_S, a receiver secret key sk_R and outputs either a message $m \in \mathcal{M}$ or an error symbol $\perp \notin \mathcal{M}$.*

The above algorithms are such that for all $dp \in \{\mathsf{Setup}(1^k)\}$, all $m \in \mathcal{M}$, all $(sk_S, pk_S) \in \{\mathsf{Gen}_S(dp)\}$, and all $(sk_R, pk_R) \in \{\mathsf{Gen}_R(dp)\}$, $m = \mathsf{Usc}(sk_R, pk_S, \mathsf{Sc}(sk_S, pk_R, m))$. The scheme is said to provide NINR if there is a non-repudiation evidence generation algorithm N *together with a* pubic verification *algorithm* PV *such that:*

– N *takes as inputs a receiver secret key sk_R, a sender public key pk_S, and a signcrypted text C, and outputs a* non-repudiation evidence *nr or a failure symbol $\perp$; we write $nr \leftarrow \mathsf{N}(sk_R, pk_S, C)$.*
– PV *takes as inputs a signcryptext C a message m, a non-repudiation evidence nr, and two public keys pk_S and pk_R and outputs, a decision $d \in \{0, 1\}$; we write $d \leftarrow \mathsf{PV}(C, m, nr, pk_S, pk_R)$.*
– *And, for all $dp \in \{\mathsf{Setup}(1^k)\}$, all $C \in \{0, 1\}^*$, all $(sk_S, pk_S) \in \{\mathsf{Gen}_S(dp)\}$, and all $(sk_R, pk_R) \in \{\mathsf{Gen}_R(dp)\}$, if $\perp \neq m \leftarrow \mathsf{Usc}(sk_R, pk_S, C)$ and $nr \leftarrow \mathsf{N}(sk_R, pk_S, C)$ then $1 = d \leftarrow \mathsf{PV}(C, m, nr, pk_S, pk_R)$.*

Confidentiality. We propose in Game 2 an extension of the Secret Key Ignorant Multi-User (SKI-MU) insider confidentiality in the Flexible Signcryption/Unsigncryption Oracle (FSO/FUO) model [4,5] geared to SCNINR.

Game 2. SKI–MU Insider Confidentiality in the FSO/FUO–IND–CCA2 sense

$\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ is a two–stage adversary against $\mathcal{SC}$; dp is the domain parameter.

1) The challenger computes $(sk_R, pk_R) \leftarrow_{\mathrm{R}} \mathsf{Gen}_R(dp)$;
2) $\mathcal{A}_1$ is provided with dp and pk_R together with two oracles: (a) $\mathcal{O}_{\mathsf{Usc}}(\cdot,\cdot)$, which takes as inputs a public key pk and a signcrypted text C and outputs $m \leftarrow \mathsf{Usc}(sk_R, pk, C)$; (b) $\mathcal{O}_{\mathsf{N}(\cdot,\cdot)}$ which takes as inputs a public key pk and a signcrypted text C and outputs $nr \leftarrow \mathsf{N}(sk_R, pk, C)$.
3) $\mathcal{A}_1$ outputs a four–tuple $(m_0, m_1, st, pk_S) \leftarrow_{\mathrm{R}} \mathcal{A}_1^{\mathcal{O}_{\mathsf{Usc}}(\cdot,\cdot), \mathcal{O}_{\mathsf{N}}(\cdot,\cdot)}(pk_R)$ wherein $m_0, m_1 \in \mathcal{M}$ are distinct equal length messages, st is a state, and pk_S is the attacked sender public key.
4) The challenger chooses $b \leftarrow_{\mathrm{R}} \{0,1\}$, computes $C^* \leftarrow_{\mathrm{R}} \mathsf{Sc}(sk_S, pk_R, m_b)$.
5) $\mathcal{A}_2$ outputs $b' \leftarrow_{\mathrm{R}} \mathcal{A}_2^{\mathcal{O}_{\mathsf{Sc}}(\cdot,\cdot), \mathcal{O}_{\mathsf{Usc}}(\cdot,\cdot), \mathcal{O}_{\mathsf{N}}(\cdot,\cdot)}(C^*, st)$, where $\mathcal{O}_{\mathsf{Usc}}(\cdot,\cdot)$ and $\mathcal{O}_{\mathsf{N}}(\cdot,\cdot)$ are as in step 2, and $\mathcal{O}_{\mathsf{Sc}}(\cdot,\cdot)$ takes as inputs $pk \in \{\mathsf{Gen}_R(dp)\}$ and $m \in \mathcal{M}$ and outputs $C \leftarrow_{\mathrm{R}} \mathsf{Sc}(sk_S, pk, m)$.
6) $\mathcal{A}$ wins the game if: (a) $\mathcal{A}_2$ never issued $\mathcal{O}_{\mathsf{Usc}}(pk_S, C^*)$ or $\mathcal{O}_{\mathsf{N}}(pk_S, C^*)$, and (b) $b = b'$.

We denote by $\mathsf{Succ}_{\mathcal{A}}^{\mathsf{cca2}}$ the event "conditions (6a) and (6b) are satisfied", and define $\mathcal{A}$'s advantage by $\mathsf{Adv}_{\mathcal{A},\mathcal{SC}}^{\mathsf{cca2}}(1^k) = \mid \Pr(\mathsf{Succ}_{\mathcal{A}}^{\mathsf{cca2}}) - 1/2 \mid$.

Definition 9. *A SCNINR $\mathcal{SC}$ is said to be $(t, q_{\mathsf{Sc}}, q_{\mathsf{Usc}}, q_{\mathsf{N}}, \varepsilon)$-secure in the SKI-MU insider confidentiality in the FSO/FUO-IND-CCA2 sense if for all adversary $\mathcal{A}$ playing Game 2, if $\mathcal{A}$ runs in time t, and issues respectively q_{Sc}, q_{Usc}, and q_{N} queries to the signcryption, unsigncryption, and non-repudiation evidence generation oracles then $\mathsf{Adv}_{\mathcal{A},\mathcal{SC}}^{\mathsf{cca2}}(1^k) \leqslant \varepsilon$.*

Unforgeability. We recall here the multi-user insider unforgeability in the FSO/FUO-sUF-CMA sense for SCNINR.

Game 3. Multi–User insider Unforgeability in the FSO/FUO–sUF–CMA sense

$\mathcal{A}$ is a forger against $\mathcal{SC}$, dp is the domain parameter.

1) The challenger computes $(sk_S, pk_S) \leftarrow_{\mathrm{R}} \mathsf{Gen}_S(dp)$.
2) $\mathcal{A}$ takes pk_S as input and is given access to a FSO $\mathcal{O}_{\mathsf{Sc}}(\cdot,\cdot)$, as in step 5 of Game 2.
3) $\mathcal{A}$ outputs $((sk_R, pk_R), C^*) \leftarrow_{\mathrm{R}} \mathcal{A}^{\mathcal{O}_{\mathsf{Sc}}(\cdot,\cdot)}(pk_S)$. He wins the game if: (a) $\perp \neq m \leftarrow \mathsf{Usc}(sk_R, pk_S, C^*)$, and (b) $\mathcal{A}$ never received C^* from the oracle $\mathcal{O}_{\mathsf{Sc}}(\cdot,\cdot)$ on a query on (pk_R, m).

$\mathsf{Adv}_{\mathcal{A},\mathcal{SC}}^{\mathsf{suf}}(1^k) = \Pr(\mathsf{Succ}_{\mathcal{A}}^{\mathsf{suf}})$ denotes the probability that $\mathcal{A}$ wins the game.

Definition 10. *A SCNINR is said to be $(t, q_{\mathsf{Sc}}, \varepsilon)$* multi-user insider unforgeable in the FSO/FUO-sUF-CMA sense *if for all attacker $\mathcal{A}$ playing Game 3, if $\mathcal{A}$ runs in time t and issues q_{Sc} signcryption queries then $\mathsf{Adv}_{\mathcal{A},\mathcal{SC}}^{\mathsf{suf}}(1^k) \leqslant \varepsilon$.*

Soundness of Non-repudiation. This attribute ensures that public verification always yields a correct result.

Game 4. Soundness of non–repudiation

$\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ is an attacker against $\mathcal{SC}$, dp is the domain parameter.

1) $\mathcal{A}_1$ executes with parameter dp and outputs $(st, pk_S) \leftarrow_{\mathrm{R}} \mathcal{A}_1(dp)$, wherein st is a state and pk_S a sender public key.
2) $\mathcal{A}_2$ executes with inputs st and pk_S and is given access to a FSO. It outputs $(sk_R, pk_R, C^*, m', nr) \leftarrow_{\mathrm{R}} \mathcal{A}_2^{\mathcal{O}_{\mathsf{Sc}}(\cdot,\cdot)}(st, pk_S)$.
3) $\mathcal{A}$ wins the game if: (a) C^* is valid, *i. e.* $\perp \neq m \leftarrow \mathsf{Usc}(sk_R, pk_S, C^*)$, and (b) $m \neq m'$ and $1 = d \leftarrow \mathsf{PV}(C^*, m', nr, pk_S, pk_R)$.

We denote by $\mathsf{Adv}^{\mathsf{snr}}_{\mathcal{A},\mathcal{SC}}(1^k)$ the probability that $\mathcal{A}$ wins the game.

Definition 11. *A signcryption scheme $\mathcal{SC}$ is said to achieve $(t, q_{\mathsf{Sc}}, \varepsilon)$* computational soundness of non-repudiation *if for all adversary $\mathcal{A}$ playing Game 4, if $\mathcal{A}$ runs in time t and issues q_{Sc} signcryption queries then $\mathsf{Adv}^{\mathsf{snr}}_{\mathcal{A},\mathcal{SC}}(1^k) \leqslant \varepsilon$.*

Unforgeability of Non-repudiation (NR) Evidence. Contrary to Malone–Lee [17], Fan *et al.* [11] consider unforgeability of non-repudiation evidence. However, their definition seems too restrictive. Indeed, they consider the capability of both the sender and receiver of a signcrypted text to generate a non-repudiation evidence as a security weakness. As a motivating example, they consider a malicious patient who receives a signcrypted medical report from his doctor, generates a non-repudiation evidence, and exposes the signcryted text together with the NR evidence. The patient can then claim that the doctor has exposed his report. In such a situation a judge cannot decide who, among the patient and the doctor, exposed the report.

As for us, non-repudiation ensures that a message sender (the doctor in the example) cannot deny that the message in the signcryted text (the medical record) is from him. The question considered in the example is *not* about the non-repudiation of the signcrypted message (the report), but about the non-repudiation of the (non-repudiation) evidence. Moreover in many settings, a non-repudiation evidence may be used both for *credit* (the ability of the sender to later claim being the sender of the message) and *responsibility* (the ability of the receiver to hold the sender accountable for the message contents) [9, Chap. 3]. It seems then important that NR evidences can be generated by both the sender (at signcrypted text generation) and the receiver of a signcrypted text.

Game 5. Unforgeability of non–repudiation evidence

$\mathcal{A}$ is an attacker against $\mathcal{SC}$, dp is the domain parameter.

1) The challenger computes $(sk_S, pk_S) \leftarrow_{\mathrm{R}} \mathsf{Gen}_S(dp)$; $(sk_R, pk_R) \leftarrow_{\mathrm{R}} \mathsf{Gen}_R(dp)$;
2) $\mathcal{A}$ runs with inputs pk_S and pk_R, and is given access to the oracles $\mathcal{O}_{\mathsf{Sc}}(\cdot,\cdot), \mathcal{O}_{\mathsf{Usc}}(\cdot,\cdot)$, and $\mathcal{O}_{\mathsf{N}}(\cdot,\cdot)$ as in step 5 of Game 2. It outputs $(C^*, m^*, nr^*) \leftarrow_{\mathrm{R}} \mathcal{A}^{\mathcal{O}_{\mathsf{Sc}}(\cdot,\cdot),\mathcal{O}_{\mathsf{Usc}}(\cdot,\cdot),\mathcal{O}_{\mathsf{N}}(\cdot,\cdot)}(pk_S, pk_R)$.
3) $\mathcal{A}$ wins the game if: (a) C^* was generated through $\mathcal{O}_{\mathsf{Sc}}(\cdot,\cdot)$ and (b) $1 = d \leftarrow \mathsf{PV}(C^*, m^*, nr^*, pk_S, pk_R)$, and nr^* was not generated by the oracle $\mathcal{O}_{\mathsf{N}}(\cdot,\cdot)$ on a query on (pk_S, C^*).

We denote by $\mathsf{Adv}^{\mathsf{unr}}_{\mathcal{A},\mathcal{SC}}(1^k)$ the probability that $\mathcal{A}$ wins the game.

Definition 12. *A SCNINR is said to achieve* $(t, q_{\mathsf{Sc}}, q_{\mathsf{Usc}}, q_{\mathsf{N}}, \varepsilon)$ unforgeability of non-repudiation evidence *if for all adversary* $\mathcal{A}$ *playing Game 5, if* $\mathcal{A}$ *runs in time* t *and issues respectively* q_{Sc}, q_{Usc}, *and* q_{N} *queries to the signcryption, unsigncryption, and non-repudiation evidence generation oracles then* $\mathsf{Adv}^{\mathsf{unr}}_{\mathcal{A},\mathcal{SC}}(1^k) \leqslant \varepsilon$.

3 New Identification and Signature Schemes

A domain parameter is given by $dp = (N, G, R, e, k)$ wherein

- $N = pq$ is an RSA modulus, $p = 2p' + 1$ and $q = 2q' + 1$ being safe primes.
- e is a $(k + 1)$ bit prime. To improve the scheme's efficiency, it can be chosen to be a sparse prime. It is used as an RSA public exponent.
- R is a generator of $\mathbb{J}_N^+$, and $G = R^{\underline{e}}$.
- k is a security parameter, $n(k) = |N|$ is chosen such that the best known algorithm for factoring N runs in time $\approx 2^k$.

For *domain parameter generation*, if there is a party which is trusted by all the users, he can generate the domain parameter. Alternatively, an perhaps *preferably*, the domain parameter may be generated by a set of parties such that each user of the scheme trusts at least one of them. In this case, the parties generating the domain parameter may perform as follows:

(1) They run the distributed shared RSA modulus generation following the protocol given in [1], to get product of two safe primes N, while each party has a share of the primes.
(2) They choose a $(k+1)$ bit prime e and $R \leftarrow_{\mathrm{R}} \mathbb{J}_N^+$, and compute $G = R^{\underline{e}}$ (R is a generator of $\mathbb{J}_N^+$, with all but negligible probability).
(3) The domain parameter is $dp = (N, G, R, e, k)$.

Description of the Scheme. Let $dp = (N, G, R, e, k)$ be a domain parameter, and $l = \lceil N/4 \rceil$. We derive the scheme $\mathcal{I}_{SSN} = (\mathsf{Gen}, \mathsf{P}, \mathsf{V}, \mathsf{ChSet})$ wherein Gen, $\mathsf{P} = (\mathsf{P}_1, \mathsf{P}_2)$, and V are as described hereunder; we denote $[2^k - 1]$ by ChSet.

$\underline{\mathsf{Gen}(dp)}$: $a \leftarrow_{\mathrm{R}} [l]$; $(sk, pk) \leftarrow (R^{\underline{a}}, G^{\underline{a}})$; Return (sk, pk).
$\underline{\mathsf{P}_1(sk)}$: $x \leftarrow_{\mathrm{R}} [l]$; $(X, st) \leftarrow (G^{\underline{x}}, R^{\underline{x}})$; Return (X, st).
$\underline{\mathsf{P}_2(sk, X, c, st)}$: $Y \leftarrow st$; $s \leftarrow Y \circ sk^{\underline{c}}$; Return s.
$\underline{\mathsf{V}(pk, X, c, s)}$: If $s^{\underline{e}} = X \circ pk^{\underline{c}}$ then Return 1, Else return 0.

For all $(sk, pk) \in \{\mathsf{Gen}(dp)\}$, if (X, c, s) is a transcript generated through P then $1 = \mathsf{V}(pk, X, c, s)$, as $s^{\underline{e}} = (R^{\underline{x+ca}})^{\underline{e}} = (R^{\underline{e}})^{\underline{x+ca}} = G^{\underline{x+ca}} = G^{\underline{x}} \circ (G^{\underline{a}})^{\underline{c}} = X \circ pk^{\underline{c}}$.

Uniqueness and Min Entropy. As the function $\mathrm{Exp}_e : \mathbb{J}_N^+ \to \mathbb{J}_N^+$ which maps Y to $Y^{\underline{e}}$ is bijective, for all $X, pk \in \mathbb{J}_N^+$, all $c \in \mathsf{ChSet}$, there is one and only one $s \in \mathbb{J}_N^+$ such that $s^{\underline{e}} = X \circ pk^{\underline{c}}$. Let δ_0 denote $\max(1/p', 1/q')$. If $x_1 \leftarrow_{\mathrm{R}} \left[|\mathbb{J}_N^+|\right]$ and $x_2 \leftarrow_{\mathrm{R}} [l]$ the statistical distance between x_1 and x_2 is $\Delta(x_1, x_2) \leqslant \frac{N/4 - \phi(N)/4}{N/4} \leqslant \delta_0$. So, if $X_1 \leftarrow G^{\underline{x_1}}$ and $X_2 \leftarrow G^{\underline{x_2}}$, then $\Delta(X_1, X_2) \leqslant \delta_0$. Then, if X is generated through $\mathsf{P}_1(\cdot)$, the statistical distance between the distribution of X and the uniform distribution over $\mathbb{J}_N^+$ is not greater than δ_0. And then for all $X_0 \in \mathbb{J}_N^+$, if X is generated through $\mathsf{P}_1(\cdot)$, $|\Pr(X = X_0) - 1/|\mathbb{J}_N^+|| \leqslant \delta_0$; the identification scheme has $\alpha \approx -\log_2(\delta_0)$ bits of min-entropy.

Special Soundness. If (X, c, s) and (X, c', s') are two accepting transcripts with respect to a public key pk such that $c \neq c'$ then $s \circ s'^{\underline{-1}} = sk^{\underline{c-c'}}$, and then $\left(s \circ s'^{\underline{-1}}\right)^{\underline{e}} = pk^{\underline{c-c'}}$. Now, as $c, c' \in \mathsf{ChSet} = [2^k - 1]$, and $e > 2^k$ is prime, it follows that $\gcd(e, c - c') = 1$. Let $\alpha, \beta \in \mathbb{Z}$ be such that $e\alpha + (c - c')\beta = 1$ and $sk^* = pk^{\underline{\alpha}} \circ \left(s \circ s'^{\underline{-1}}\right)^{\underline{\beta}}$, then $(sk^*)^{\underline{e}} = \left(pk^{\underline{\alpha}} \circ \left(s \circ s'^{\underline{-1}}\right)^{\underline{\beta}}\right)^{\underline{e}} = pk^{\underline{e\alpha}} \circ \left(s \circ s'^{\underline{-1}}\right)^{\underline{e\beta}} = pk^{\underline{e\alpha + (c-c')\beta}} = pk$.

Honest Verifier Zero Knowledge. For all public key $pk \in \mathbb{J}_N^+$, the following simulator yields transcripts with the same distribution as real transcripts.

$\underline{\mathsf{sim}(pk)}$: $c \leftarrow_{\mathrm{R}} \mathsf{ChSet}$; $z \leftarrow_{\mathrm{R}} [l]$; $s \leftarrow R^{\underline{z}}$; $X \leftarrow s^{\underline{e}} \circ pk^{\underline{-c}}$; Return (X, c, s).

Random Self Reducibility. The Rerand, Tran and Derand algorithms are:

$\underline{\mathsf{Rerand}(pk)}$: $z \leftarrow_{\mathrm{R}} [l]$; $\tau \leftarrow R^{\underline{z}}$; $pk_1 \leftarrow \tau^{\underline{e}} \circ pk$; Return (τ, pk_1);
$\underline{\mathsf{Derand}(pk, pk_1, sk_1, \tau)}$: $sk^* \leftarrow sk_1 \circ \tau^{\underline{-1}}$; Return sk^*;
$\underline{\mathsf{Tran}(pk, pk_1, \tau, (X, c, s_1))}$: $Z \leftarrow \tau^{\underline{-c}}$; $s \leftarrow Z \circ s_1$; Return (X, c, s).

The Rerand algorithm outputs a public key pk_1 which has the same distribution as the keys generated through $\mathsf{Gen}(dp)$. The Derand algorithm provides the static private key corresponding to pk. The Tran algorithm produces a valid transcript with respect to the public key pk.

$\mathsf{KR\text{-}KOA}$ *Security.* For $sk, pk \in \mathbb{J}_N^+$, if $sk^{\underline{e}} = pk$ then $(\pm sk)^e = pk$. Then under the RSA assumption over $\mathbb{J}_N^+$, $\mathcal{I}_{SSN}$ is secure against KR-KOA.

Lemma 2. *If the RSA problem is (t, ε)-hard over $\mathbb{J}_N^+$ then the identification scheme $\mathcal{I}_{SSN}$ is (t, ε)-*$\mathsf{KR\text{-}KOA}$*-secure.*

The Signature Scheme. As the identification scheme is commitment recoverable, using the (alternative) Fiat–Shamir transform [12], we derive the signature scheme $\mathcal{S}_{SSN} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ we describe hereunder. $\mathsf{H}_1 : \{0,1\}^* \to \mathsf{ChSet}$ is a hash function.

$\underline{\mathsf{Gen}(dp)}$: $a \leftarrow_{\mathrm{R}} [l]$; $(sk, pk) \leftarrow (R^{\underline{a}}, G^{\underline{a}})$; Return (sk, pk).
$\underline{\mathsf{Sign}(sk, m)}$: $x \leftarrow_{\mathrm{R}} [l]$; $X \leftarrow G^{\underline{x}}$; $h \leftarrow \mathsf{H}_1(X, m)$ $s \leftarrow R^{\underline{x}} \circ sk^{\underline{h}}$; Return (h, s).
$\underline{\mathsf{Vrfy}(pk, m, \sigma)}$: Parse σ as $(h, s) \in \mathsf{ChSet} \times \mathbb{Z}_N$; $X \leftarrow s^{\underline{e}} \circ pk^{\underline{-h}}$; $h' = \mathsf{H}_1(X, m)$.
If $pk, s \in \mathbb{J}_N^+$ and $h = h'$ then Return 1; Else Return 0.

Security and Efficiency of the Signature Scheme. We have the following theorem; its proof follows straightly from the SpS, HVZN, RSR, min-entropy, and KR-KOA security attributes of the identification scheme and Theorem 3.1 from [15].

Theorem 1. *If the RSA problem is* (t, ε) *hard on* (N, e)*, then the scheme* $\mathcal{S}_{SSN}$ *is* $(t', \varepsilon', U, Q_s, Q_h)$-MU-SUF-CMA *secure in the random oracle model, where* $\varepsilon'/t' \leqslant 24(Q_h + 1) \cdot \varepsilon/t + Q_s/2^\alpha + 1/2^k$.

Although efficient, the signature scheme is slightly less efficient than the GQ scheme [13]. A key pair generation requires $\mathsf{Exp}(\mathbb{Z}_N, 2, l)$ operations for our scheme while it requires $\mathsf{Exp}(\mathbb{Z}_N, k)$ operations for the GQ scheme. We stress that, using simultaneous exponentiation techniques [18, Sect. 14.6], $\mathsf{Exp}(\mathbb{Z}_N, 2, l) \approx 1.17 \cdot \mathsf{Exp}(\mathbb{Z}_N, l)$. A $\mathcal{S}_{SSN}$ signature generation can be performed in $1.17 \cdot \mathsf{Exp}(\mathbb{Z}_N, l)$ $+\mathsf{Exp}(\mathbb{Z}_N, k)$ operations, while it requires $2 \cdot \mathsf{Exp}(\mathbb{Z}_N, k)$ operations for the GQ scheme. In both schemes, only $\mathsf{Exp}(\mathbb{Z}_N, k)$ operations need to be performed online, all the other operations can be performed offline. A signature verification requires $2 \cdot \mathsf{Jcb}(N)$ + $\mathsf{Exp}(\mathbb{Z}_N, 2, k)$ operations for $\mathcal{S}_{SSN}$ and $\mathsf{Exp}(\mathbb{Z}_N, 2, k)$ operations for the GQ scheme.

4 The Signcryption Scheme

From the $\mathcal{S}_{SSN}$ scheme, which has the advantage of being defined over a group wherein the strong DH assumption is known to hold under the factoring assumption [14], we derive $\mathcal{SC}_{SSN} = (\mathsf{Setup}, \mathsf{Gen}_S, \mathsf{Gen}_R, \mathsf{Sc}, \mathsf{Usc}, \mathsf{N}, \mathsf{PV})$. The Setup algorithm generates a domain parameter dp' as in Sect. 3, together with an encryption scheme $\mathcal{E}$ and two hash functions $\mathsf{H}_1 : \{0,1\}^* \to \mathsf{ChSet}$ and $\mathsf{H}_2 : \{0,1\}^* \to \mathbf{K}$. We consider $dp = (dp', \mathsf{H}_1, \mathsf{H}_2, \mathcal{E})$ as an implicit parameter.

$\underline{\mathsf{Gen}_S(dp)}$: $a \leftarrow_{\mathrm{R}} [l]$; $(sk_S, pk_S) \leftarrow (\underline{R^a}, \underline{G^a})$; Return (sk_S, pk_S);
$\underline{\mathsf{Gen}_R(dp)}$: $b \leftarrow_{\mathrm{R}} [l]$; $(sk_R, pk_R) \leftarrow (b, \underline{G^b})$; Return (sk_R, pk_R);
$\underline{\mathsf{Sc}(sk_S, pk_R, m)}$: $x_1, x_2 \leftarrow_{\mathrm{R}} [l]$; $X_1 \leftarrow \underline{G^{x_1}}$; $Z_1 \leftarrow \underline{pk_R{}^{x_1}}$; $X_2 \leftarrow \underline{G^{x_2}}$; $Z_2 \leftarrow \underline{pk_R{}^{x_2}}$;
$\tau_1 \leftarrow \mathsf{H}_2(X_1, X_2, Z_1, Z_2, pk_S, pk_R)$; $\tau_2 \leftarrow \mathsf{H}_2(X_2, X_1, Z_2, Z_1, pk_S, pk_R)$;
$h \leftarrow \mathsf{H}_1(X_1, X_2, m, \tau_1)$; $c \leftarrow \mathsf{E}(\tau_2, m)$; $s \leftarrow \underline{R^{x_1}} \circ \underline{sk_S{}^h}$; Return (h, X_2, s, c);
$\underline{\mathsf{Usc}(sk_R, pk_S, C)}$: Parse C as (h, X_2, s, c). If $X_2, pk_S \notin \mathbb{J}_N^+$ then Return $\perp$;
$X_1 \leftarrow \underline{s^e} \circ \underline{pk_S{}^{-h}}$; $Z_1 \leftarrow \underline{X_1{}^{sk_R}}$; $Z_2 \leftarrow \underline{X_2{}^{sk_R}}$; $\tau_1 \leftarrow \mathsf{H}_2(X_1, X_2, Z_1, Z_2, pk_S, pk_R)$; $\tau_2 \leftarrow \mathsf{H}_2(X_2, X_1, Z_2, Z_1, pk_S, pk_R)$; $m \leftarrow \mathsf{D}(\tau_2, c)$;
If $h = h' \leftarrow \mathsf{H}_1(X_1, X_2, m, \tau_1)$ then Return m; Else return $\perp$;
$\underline{\mathsf{N}(sk_R, pk_S, C)}$: Parse C as (h, X_2, s, c). If $X_2, pk_S \notin \mathbb{J}_N^+$ then Return $\perp$;
$X_1 \leftarrow \underline{s^e} \circ \underline{pk_S{}^{-h}}$; $Z_1 \leftarrow \underline{X_1{}^{sk_R}}$; $Z_2 \leftarrow \underline{X_2{}^{sk_R}}$; $\tau_1 \leftarrow \mathsf{H}_2(X_1, X_2, Z_1, Z_2, pk_S, pk_R)$; $\tau_2 \leftarrow \mathsf{H}_2(X_2, X_1, Z_2, Z_1, pk_S, pk_R)$; $m \leftarrow \mathsf{D}(\tau_2, c)$;
If $h = h' \leftarrow \mathsf{H}_1(X_1, X_2, m, \tau_1)$ then Return (τ_1, τ_2); Else return $\perp$;
$\underline{\mathsf{PV}(C, m, nr, pk_S, pk_R)}$: Parse C as (h, X_2, s, c) and nr as (τ_1, τ_2); $m' \leftarrow \mathsf{D}(\tau_2, c)$;
If $m' \neq m$ then Return 0; $X_1 \leftarrow \underline{s^e} \circ \underline{pk_S{}^{-h}}$;
If $h = h' \leftarrow_{\mathrm{R}} \mathsf{H}_1(X_1, X_2, m, \tau_1)$ then Return 1; Else return 0;

For the consistency of the scheme, one can observe that for all $dp \in \{\mathsf{Setup}(1^k)\}$, all $m \in \mathcal{M}$, all $(sk_S, pk_S) \in \{\mathsf{Gen}_S(dp)\}$, and all $(sk_R, pk_R) \in \{\mathsf{Gen}_R(dp)\}$, $m = \mathsf{Usc}(sk_R, pk_S, \mathsf{Sc}(sk_S, pk_R, m))$. Moreover, if $nr \leftarrow \mathsf{N}(sk_R, pk_S, \mathsf{Sc}(sk_S, pk_R, m))$ then $1 = d \leftarrow \mathsf{PV}(C, m, nr, pk_S, pk_R)$.

Efficiency of the Scheme. Since Malone–Lee's scheme [17] is defined over any Diffie–Hellman group, and Fan *et al.*'s [11] design makes use of bilinear pairings, it is rather difficult to compare the efficiency of these schemes with our (we use an RSA instance), without considering concrete instances. Nonetheless, our design is a practical and efficient one; it uses the RSA primitive, which remains probably the most widely deployed public key primitive [16]. A sender key pair generation requires $\mathsf{Exp}(\mathbb{Z}_n, 2, l)$ operations (the exponentiations use the same exponent); a receiver key pair generation requires $\mathsf{Exp}(\mathbb{Z}_n, l)$ operations. A signcryption generation requires $\mathsf{Exp}(\mathbb{Z}_n, 6, l)$ operations (we neglect the cost of the three digest operations together with the symmetric encryption). Five of the six exponentiations can be performed off-line. Moreover, three of the five off-line exponentiations share the same exponent, and the remaining two exponentiations have also the same exponent. An unsigncryption or a non-repudiation evidence generation requires four exponentiations; we recall that e can be chosen to be a sparse prime so that exponentiations involving e can be performed using few multiplications. A public verification requires $\mathsf{Exp}(\mathbb{Z}_n, 2, l)$ operations. Assuming that $|c| = |m|$, the communication overhead compared to a signature is one group element.

4.1 Confidentiality of the $\mathcal{SC}_{SSN}$ Signcryption Scheme

We need the following result, its proof is given in the full version of this paper.

Theorem 2. *If X_1, r, s be mutually independent random variables, such r and s are uniformly distributed over $[N/4]$. Let X_2 be defined by $X_2 \leftarrow G^{\underline{s}} \circ X_1^{\underline{-r}}$, and suppose that Y, Z_1, and Z_2 are random variables taking values in $\mathbb{J}_N^+$, and are defined as some functions of X_1 and X_2, then: (a) the statistical distance between X_2 and the uniform distribution over $\mathbb{J}_N^+$ is not greater than $2\delta_0$;(b) If $X_1 = G^{\underline{x_1}}$ and $X_2 = G^{\underline{x_2}}$, then the probability that the truth value of*

$$Z_1^{\underline{r}} Z_2 = G^{\underline{s}} \tag{1}$$

does not agree with

$$Z_1 = Y^{\underline{x_1}} \text{ and } Z_2 = Y^{\underline{x_2}} \tag{2}$$

is at most $5\delta_0$; and if (2) holds then so does (1).

Theorem 3. *Under the RO model, if the factorization of N is $(t(k), \varepsilon_{\mathsf{fac}}(k))$-hard and the encryption scheme $\mathcal{E}$ is $(t(k), \varepsilon_{\mathsf{ss}}(k))$-semantically secure, then $\mathcal{SC}_{SSN}$ is $(t(k), q_{\mathsf{Sc}}, q_{\mathsf{Usc}}, q_{\mathsf{N}}, \varepsilon'(k))$-secure in the SKI-MU insider confidentiality in the FSO/FUO-IND-CCA2 sense, wherein*

$$\varepsilon'(k) = \varepsilon_{\mathsf{ss}}(k) + \varepsilon_{\mathsf{fac}}(k) + (1 + 1/2 \cdot q_{\mathsf{Sc}}(q_{\mathsf{Sc}} - 1))\,(p'q')^{-2}|\mathbf{K}|^{-1} + (5q_{\mathsf{Sc}} + 2)\delta_0.$$

Proof. We call the steps (1) and (2), (3) and (4), and (5) and (6) of Game 2 the pre-challenge, challenge, and post-challenge phases respectively. We provide a simulator which answers to $\mathcal{A}$'s queries in all phases. The Initialization procedure is executed at the beginning of the game. When the variable abort is set to 1, the whole simulation fails. If the simulation does not fail, the Finalization procedure is executed at the end of the game. The oracle $\mathrm{DDH}_{Y_0}(\cdot,\cdot)$ takes $U, V \in \mathbb{J}_N^+$ as inputs and outputs 1 if $\mathrm{CDH}(Y_0, U) = V$ and 0 otherwise. For a list L and an element X, $\mathsf{Apd}(L, X)$ adds X to L.

Simulation for the SKI MU insider confidentiality game

Input: $dp = (N, G, R, e, k)$, $\mathcal{E} = (\mathsf{E}, \mathsf{D}, \mathbf{K}, \mathbf{M}, \mathbf{C})$, and $X_0, Y_0 \leftarrow_{\mathrm{R}} \mathbb{J}_N^+$.

External Oracles: $\mathrm{DDH}_{Y_0}(\cdot,\cdot)$;

Initialization: $pk_R \leftarrow Y_0$; $\mathcal{S}_{\mathsf{H}_1} \leftarrow ()$; $\mathcal{S}_{\mathsf{k}} \leftarrow ()$; $\mathcal{S}_{\mathsf{k\&r}} \leftarrow ()$; $\mathcal{S}_{\mathsf{H}_2} \leftarrow ()$; abort $\leftarrow 0$;

Pre–Challenge Phase

$\mathcal{O}_{\mathsf{H}_1}(s)$:
if $\exists\, h : (s, h) \in \mathcal{S}_{\mathsf{H}_1}$ **then** return h; **else** $h \leftarrow_{\mathrm{R}} \mathsf{ChSet}$; $\mathsf{Apd}(\mathcal{S}_{\mathsf{H}_1}, (s, h))$; return h;
$\mathcal{O}_{\mathsf{H}_2}(s)$:
if $\exists\, \tau : (s, \tau) \in \mathcal{S}_{\mathsf{H}_2}$ **then** return τ;
else if s has format $(X_1, X_2, Z_1, Z_2, pk, pk' = pk_R) \in \left(\mathbb{J}_N^+\right)^6$ **then**
  if $\exists\, \tau : ((X_1, X_2, pk, pk_R), \tau) \in \mathcal{S}_{\mathsf{k}}$ **then**
    if $\mathrm{DDH}_{Y_0}(X_1, Z_1) = \mathrm{DDH}_{Y_0}(X_2, Z_2) = 1$ **then** $\mathsf{Apd}(\mathcal{S}_{\mathsf{H}_2}, (s, \tau))$; return τ;
else $\tau \leftarrow_{\mathrm{R}} \mathsf{ChSet}$; $\mathsf{Apd}(s_{\mathsf{H}_2}, (s, \tau))$; return τ;

$\mathcal{O}_{\mathsf{Usc}}(pk, C)$: $\mathcal{O}_{\mathsf{N}}(pk, C)$:
if $pk \notin \mathbb{J}_N^+$ **then** return $\perp$;
Parse C as $(h, X_2, s, c) \in \mathsf{ChSet} \times \mathbb{J}_N^+ \times \mathbb{J}_N^+ \times \mathbf{C}$; ▶ *$\perp$ is returned if the parsing fails*
$X_1 \leftarrow s^{\underline{e}} \circ pk^{\underline{-h}}$;
if $\exists\, Z_1, Z_2 \in \mathbb{J}_N^+, \tau \in \mathbf{K} : ((X_1, X_2, Z_1, Z_2, pk, pk_R), \tau) \in \mathcal{S}_{\mathsf{H}_2}$ and $\mathrm{DDH}_{Y_0}(X_1, Z_1) = \mathrm{DDH}_{Y_0}(X_2, Z_2) = 1$ **then**
  $\tau_1 \leftarrow \tau$; ▶ $\mathsf{H}_2(X_1, X_2, Z_1, Z_2, pk, pk_R)$ *was issued*
else if $\exists\, \tau : ((X_1, X_2, pk, pk_R), \tau) \in \mathcal{S}_{\mathsf{k}}$ **then**
  $\tau_1 \leftarrow \tau$; ▶ $\mathsf{Usc}(pk, C')$ *or* $\mathsf{N}(pk, C')$ *such that C' parses as (h, X_2, s, c') was issued*
else $\tau_1 \leftarrow_{\mathrm{R}} \mathbf{K}$; $\mathsf{Apd}(\mathcal{S}_{\mathsf{k}}, ((X_1, X_2, pk, pk_R), \tau_1))$;
if $\exists\, Z_2, Z_1 \in \mathbb{J}_N^+, \tau \in \mathbf{K} : ((X_2, X_1, Z_2, Z_1, pk, pk_R), \tau) \in \mathcal{S}_{\mathsf{H}_2}$ and $\mathrm{DDH}_{Y_0}(X_1, Z_1) = \mathrm{DDH}_{Y_0}(X_2, Z_2) = 1$ **then** $\tau_2 \leftarrow \tau$; ▶ *the same treatment as for τ_1*
else if $\exists\, \tau : ((X_2, X_1, pk, pk_R), \tau) \in \mathcal{S}_{\mathsf{k}}$ **then** $\tau_2 \leftarrow \tau$;
else $\tau_2 \leftarrow_{\mathrm{R}} \mathbf{K}$; $\mathsf{Apd}(\mathcal{S}_{\mathsf{k}}, ((X_2, X_1, pk, pk_R), \tau_2))$;
$m \leftarrow \mathsf{D}(\tau_2, c)$; $h' \leftarrow \mathcal{O}_{\mathsf{H}_1}(X_1, X_2, m, \tau_1)$;
if $h = h'$ **then** $\overset{\mathcal{O}_{\mathsf{Usc}}}{\boxed{\text{return } m}}$ $\overset{\mathcal{O}_{\mathsf{N}}}{\boxed{\text{return } (\tau_1, \tau_2)}}$ **else** return $\perp$;

Challenge Phase

$(m_0, m_1, st, pk_S) \leftarrow_{\mathrm{R}} \mathcal{A}_1^{\mathcal{O}_{\mathsf{Usc}}(\cdot,\cdot), \mathcal{O}_{\mathsf{N}}(\cdot,\cdot), \mathcal{O}_{\mathsf{H}_1}(\cdot), \mathcal{O}_{\mathsf{H}_2}(\cdot)}(pk_R)$;
$\hat{h} \leftarrow_{\mathrm{R}} \mathsf{ChSet}$; $\hat{z} \leftarrow_{\mathrm{R}} [l]$; $\hat{s} \leftarrow R^{\underline{\hat{z}}}$; $\hat{X}_1 \leftarrow \hat{s}^{\underline{e}} \circ pk_S^{\underline{-\hat{h}}}$; $\hat{X}_2 \leftarrow X_0$;
$b \leftarrow_{\mathrm{R}} \{0, 1\}$; $\hat{\tau}_1 \leftarrow_{\mathrm{R}} \mathbf{K}$; $\hat{\tau}_2 \leftarrow_{\mathrm{R}} \mathbf{K}$; $\hat{c} \leftarrow \mathsf{E}(\hat{\tau}_2, m_b)$;
if $\exists\, h', m' : ((\hat{X}_1, \hat{X}_2, m', \hat{\tau}_1), h') \in \mathcal{S}_{\mathsf{H}_1}$ **then** abort $\leftarrow 1$;

27 $\mathsf{Apd}(\mathcal{S}_{\mathsf{H}_1}, ((\hat{X}_1, \hat{X}_2, m_b, \hat{\tau}_1), \hat{h}))$; $\mathsf{Apd}(\mathcal{S}_{\mathsf{k}}, ((\hat{X}_1, \hat{X}_2, pk_S, pk_R), \hat{\tau}_1))$;
28 $\mathsf{Apd}(\mathcal{S}_{\mathsf{k}}, ((\hat{X}_2, \hat{X}_1, pk_S, pk_R), \hat{\tau}_2))$; $C^* \leftarrow (\hat{h}, \hat{X}_2, \hat{s}, \hat{c})$;

<u>Post–Challenge Phase</u>
$\mathcal{A}_2$ is run with input (C^*, st). It has access to the oracles $\mathcal{O}_{\mathsf{Sc}}(\cdot,\cdot)$, $\mathcal{O}_{\mathsf{Usc}}(\cdot,\cdot)$, $\mathcal{O}_{\mathsf{N}}(\cdot,\cdot)$, $\mathcal{O}_{\mathsf{H}_1}(\cdot)$, and $\mathcal{O}_{\mathsf{H}_2}(\cdot)$. Only changes compared to the pre–challenge phase are drawn.
29 $\underline{\mathcal{O}_{\mathsf{Sc}}(pk, m)}$:
30 $h \leftarrow_{\mathrm{R}} \mathsf{ChSet}$; $z \leftarrow_{\mathrm{R}} [l]$; $s_1 \leftarrow R^{\underline{z}}$; $X_1 \leftarrow s_1^{\underline{e}} \circ pk_S^{\underline{-h}}$;
31 $r \leftarrow_{\mathrm{R}} [l]$; $s_2 \leftarrow_{\mathrm{R}} [l]$; $X_2 \leftarrow G^{\underline{s_2}} \circ X_1^{\underline{-r}}$; $\tau_1 \leftarrow_{\mathrm{R}} \mathbf{K}$; $\tau_2 \leftarrow_{\mathrm{R}} \mathbf{K}$;
32 **if** $\exists\, h', m' : ((X_1, X_2, m', \tau_1), h') \in \mathcal{S}_{\mathsf{H}_1}$ **then** abort $\leftarrow 1$;
33 $\mathsf{Apd}\,(\mathcal{S}_{\mathsf{H}_1}, ((X_1, X_2, m, \tau_1), h))$;
34 **if** $pk = pk_R$ **then** $\mathsf{Apd}(\mathcal{S}_{\mathsf{k}}, ((X_1, X_2, pk_S, pk_R), \tau_1))$; $\mathsf{Apd}(\mathcal{S}_{\mathsf{k}}, ((X_2, X_1, pk_S, pk_R), \tau_2))$;
35 **else** $\mathsf{Apd}\,(\mathcal{S}_{\mathsf{k\&r}}, ((X_1, X_2, pk_S, pk), (r, s_2, \tau_1, \tau_2)))$;
36 $c \leftarrow \mathsf{E}(\tau_2, m)$; $C \leftarrow (h, X_2, s_1, c)$; return C;

37 $\underline{\mathcal{O}_{\mathsf{H}_2}(s)}$:
38 **if** $\exists\, \tau : (s, \tau) \in \mathcal{S}_{\mathsf{H}_2}$ **then** return τ;
39 **else if** s has format $(X_1, X_2, Z_1, Z_2, pk, pk_R) \in \left(\mathbb{J}_N^+\right)^6$ **then**
40   **if** $\exists\, \tau : ((X_1, X_2, pk, pk_R), \tau) \in \mathcal{S}_{\mathsf{k}}$ **then**
41   **if** $\mathrm{DDH}_{Y_0}(X_1, Z_1) = \mathrm{DDH}_{Y_0}(X_2, Z_2) = 1$ **then** $\mathsf{Apd}(\mathcal{S}_{\mathsf{H}_2}, (s, k))$; return τ;
42 **else if** s has format $(X_1, X_2, Z_1, Z_2, pk_S, pk) \in \left(\mathbb{J}_N^+\right)^6$ **then**
43   **if** $\exists\, r, s, \tau_1, \tau_2 : ((X_1, X_2, pk_S, pk), (r, s, \tau_1, \tau_2)) \in \mathcal{S}_{\mathsf{k\&r}}$ **then**
44   **if** $Z_1^{\underline{r}} \circ Z_2 = pk^{\underline{s}}$ **then** return τ_1; ▶ *$2DH(X_1, X_2, pk) = (Z_1, Z_2)$ with all but negligible probability.*
45   **if** $\exists\, r, s, \tau_1, \tau_2 : ((X_2, X_1, pk_S, pk), (r, s, \tau_1, \tau_2)) \in \mathcal{S}_{\mathsf{k\&r}}$ **then**
46   **if** $Z_2^{\underline{r}} \circ Z_1 = pk^{\underline{s}}$ **then** return τ_2;
47 **else** $\tau \leftarrow_{\mathrm{R}} \mathsf{ChSet}$; $\mathsf{Apd}(s_{\mathsf{H}_2}, (s, \tau))$; return τ;
48 <u>Finalization:</u>
49 **if** $\exists\, \hat{Z}_1, \hat{Z}_2 \in \mathbb{J}_N^+ : \big(((\hat{X}_1, \hat{X}_2, \hat{Z}_1, \hat{Z}_2, pk_S, pk_R), \hat{\tau}_1) \in \mathcal{S}_{\mathsf{H}_2}$ or $((\hat{X}_2, \hat{X}_1, \hat{Z}_2, \hat{Z}_1, pk_S, pk_R), \hat{\tau}_2) \in \mathcal{S}_{\mathsf{H}_2}\big)$ and $\mathrm{DDH}_{Y_0}(\hat{X}_1, \hat{Z}_1) = \mathrm{DDH}_{Y_0}(\hat{X}_2, \hat{Z}_2) = 1$ **then** return $\hat{Z}_2$;
50 **else** return $\perp$;

In the pre-challenge phase, the simulator answers to $\mathcal{O}_{\mathsf{H}_1}(\cdot)$, $\mathcal{O}_{\mathsf{H}_2}(\cdot)$, $\mathcal{O}_{\mathsf{Usc}}(\cdot,\cdot)$, and $\mathcal{O}_{\mathsf{N}}(\cdot,\cdot)$ queries. The lines 10–22 describe both $\mathcal{O}_{\mathsf{Usc}}(\cdot,\cdot)$ and $\mathcal{O}_{\mathsf{N}}(\cdot,\cdot)$. When executing $\mathcal{O}_{\mathsf{Usc}}(\cdot,\cdot)$ (resp. $\mathcal{O}_{\mathsf{N}}(\cdot,\cdot)$), the instruction return (τ_1, τ_2) (resp. return m) at line 22 is omitted. Digest queries are answered using input-output tables. The $\mathcal{O}_{\mathsf{H}_2}(\cdot)$ digest values of strings with format $(X_1, X_2, Z_1, Z_2, pk, pk_R)$ are not only assigned by the $\mathcal{O}_{\mathsf{H}_2}(\cdot)$ oracle, but also through executions of $\mathcal{O}_{\mathsf{Usc}}(\cdot,\cdot)$ and $\mathcal{O}_{\mathsf{N}}(\cdot,\cdot)$; in the latter two cases $Z_1 = \mathrm{CDH}(X_1, pk_R)$ and $Z_2 = \mathrm{CDH}(X_2, pk_R)$ are unknown. So, for consistency, in addition to $\mathcal{S}_{\mathsf{H}_2}$, we use a list $\mathcal{S}_{\mathsf{k}}$ to store the values of $\mathcal{O}_{\mathsf{H}_2}(X_1, X_2, Z_1, Z_2, pk, pk_R)$ which was assigned while Z_1 and Z_2 are unknown (see at lines 14–20). Doing so, the simulator consistently answers to all digest queries with the help of the $\mathrm{DDH}_{Y_0 = pk_R}(\cdot,\cdot)$ oracle (see at lines 6–8).

In the challenge phase, we essentially simulate a signature generation (at line 24), then X_2 is set to X_0 (the simulator takes X_0 and $Y_0 = pk_R$ as input). The secret keys, τ_1 and τ_2 are chosen uniformly at random from $\mathbf{K}$, and savings are performed for $\mathcal{O}_{\mathsf{H}_2}(\cdot)$ digests consistency (lines 27–28). In the post-challenge phase, the changes, compared to the pre-challenge phase, are the (re)definitions of the $\mathcal{O}_{\mathsf{Sc}}(\cdot,\cdot)$ and $\mathcal{O}_{\mathsf{H}_2}(\cdot)$ oracles. When computing $\mathcal{O}_{\mathsf{Sc}}(pk, m)$, the simulator ignores both sk_S and the secret key corresponding to pk. For consistency, we simulate a signature generations (see at line 30), choose r and s_2, and generate X_2 (see at line 31) such that: (i) the statistical distance between the distribution of the X_2 we generate in this way and the distribution of X_2 we obtain through a real execution of $\mathsf{Sc}(\cdot,\cdot,\cdot)$ is not greater than $2\delta_0 = 2\max(1/p', 1/q')$; ($ii$) if Z_1 and Z_2 are such that $Z_1^{\underline{r}} Z_2 = G^{\underline{s}}$, then $Z_1 = \mathrm{CDH}(X_1, pk)$ and $Z_2 = \mathrm{CDH}(X_2, pk)$ with overwhelming probability (see Theorem 2). Doing so, we have a way to assign values to τ_1 and τ_2, while keeping the outputs of $\mathcal{O}_{\mathsf{H}_2}(\cdot)$ consistent (see at lines 31–35 and 43–46). Let bad be the event: "(a) the simulator aborts (see at lines 26 and 32) or (b) in some execution of $\mathcal{O}_{\mathsf{H}_2}(\cdot)$, Z_1 and Z_2 are such that $Z_1^{\underline{r}} \circ Z_2 = pk^{\underline{s}}$ while $\mathrm{CDH}(X_1, pk) \neq Z_1$ or $\mathrm{CDH}(X_2, pk) \neq Z_2$ (see at lines 43–46)." Then, from Theorem 2

$$\Pr(\mathsf{bad}) \leqslant (p'q')^{-2}|\mathbf{K}|^{-1} + q_{\mathsf{Sc}}(q_{\mathsf{Sc}} - 1)\left(2(p'q')^2|\mathbf{K}|\right)^{-1} + 5q_{\mathsf{Sc}}\delta_0. \tag{3}$$

Let $\mathsf{Succ}^{\mathsf{cca2}}_{\mathcal{A},\mathsf{sim}}$ denote the event "$\mathcal{A}$ succeeds in the simulated environment". Under the RO model, if $\neg\mathsf{bad}$ then, $\mathcal{A}$'s views in the real and simulated environments are the same; so, $\Pr(\mathsf{Succ}^{\mathsf{cca2}}_{\mathcal{A}} \wedge \neg\mathsf{bad}) = \Pr(\mathsf{Succ}^{\mathsf{cca2}}_{\mathcal{A},\mathsf{sim}} \wedge \neg\mathsf{bad})$. Then

$$\mathsf{Adv}^{\mathsf{cca2}}_{\mathcal{A}}(1^k) = |\Pr(\mathsf{Succ}^{\mathsf{cca2}}_{\mathcal{A}}) - 1/2| \leqslant |\Pr(\mathsf{Succ}^{\mathsf{cca2}}_{\mathcal{A}} \wedge \neg\mathsf{bad}) - 1/2| + \Pr(\mathsf{bad}). \tag{4}$$

Let $\mathsf{CDHfound}$ be the event the "$\underline{\mathsf{Finalization}}$ procedure outputs $\hat{Z}_2 \neq \perp$". By the definition of $\mathsf{CDHfound}$, $\Pr(\mathsf{Succ}^{\mathsf{cca2}}_{\mathcal{A},\mathsf{sim}} \wedge \neg\mathsf{bad} \wedge \mathsf{CDHfound}) \leqslant \mathsf{Adv}^{\mathrm{sCDH}}_{\mathcal{B}_1}(\mathbb{J}_N^+)$, where $\mathcal{B}_1$ is obtained from $\mathcal{A}$ and the simulator. Using [14, Theorem 2], we obtain

$$\Pr(\mathsf{Succ}^{\mathsf{cca2}}_{\mathcal{A},\mathsf{sim}} \wedge \neg\mathsf{bad} \wedge \mathsf{CDHfound}) \leqslant \mathsf{Adv}^{\mathsf{fac}}_{\mathcal{B}_1,\mathsf{RSAGen}}(k) + 1/p' + 1/q'. \tag{5}$$

Now, if $\mathsf{Succ}^{\mathsf{cca2}}_{\mathcal{A},\mathsf{sim}} \wedge \neg\mathsf{bad} \wedge \neg\mathsf{CDHfound}$, then $\mathcal{A}$ is essentially playing a semantic security game against $\mathcal{E}$, so using $\mathcal{A}$ and the simulator we build an adversary $\mathcal{B}_2$ against $\mathcal{E}$ such that

$$|\Pr(\mathsf{Succ}^{\mathsf{cca2}}_{\mathcal{A},\mathsf{sim}} \wedge \neg\mathsf{bad} \wedge \neg\mathsf{CDHfound}) - 1/2| = \mathsf{Adv}^{\mathsf{ss}}_{\mathcal{B}_2,\mathcal{E}}(k). \tag{6}$$

The result follows from (3)–(6). □

4.2 Unforgeability of the $\mathcal{SC}_{SSN}$ Scheme

Theorem 4. *Under the RO model, if the RSA problem is $(t(k), \varepsilon_0(k))$-hard over $\mathbb{J}_N^+$, then $\mathcal{SC}_{SSN}$ is $(t, q_{\mathsf{Sc}}, \varepsilon')$-MU insider unforgeable in the FSO/FUO-sUF-CMA sense, with $\varepsilon' \leqslant \sqrt{q\varepsilon_0} + (q+1)|\mathsf{ChSet}|^{-1} + q_{\mathsf{Sc}}(q_{\mathsf{Sc}} - 1)\left(2(p'q')^2|\mathbf{K}|\right)^{-1} + 5q_{\mathsf{Sc}}\delta_0$, with $q = q_{\mathsf{H}_1} + q_{\mathsf{Sc}}$ wherein q_{H_1} is an upper bound on the number of $\mathcal{O}_{\mathsf{H}_1}(\cdot)$ queries the adversary issues.*

Proof. Let q_{H_1} and q_{Sc} be upper bounds on the number of queries $\mathcal{A}$ issues to the $\mathcal{O}_{\mathsf{H}_1}(\cdot)$ and $\mathcal{O}_{\mathsf{Sc}}(\cdot,\cdot)$ oracles respectively, and $q = q_{\mathsf{H}_1} + q_{\mathsf{Sc}}$. In addition to the domain parameter and $Y_0 \leftarrow_{\mathrm{R}} \mathbb{J}_N^+$, the simulator takes as an additional input $L_{\mathsf{H}_1} = (h_1, \cdots, h_q)$ such that for all i, $h_i \leftarrow_{\mathrm{R}} \mathsf{ChSet}$.

Simulation for the MU insider Unforgeability in the FSO/FUO–sUF–CMA sense

Input: $dp = (N, G, R, e, k)$, $\mathcal{E} = (\mathsf{E}, \mathsf{D}, \mathbf{K}, \mathbf{M}, \mathbf{C})$, $Y_0 \leftarrow_{\mathrm{R}} \mathbb{J}_N^+$, $L_{\mathsf{H}_1} = (h_1, h_2, \cdots, h_q)$.

100 Initialization: $pk_S \leftarrow Y_0$; $\mathcal{S}_{\mathsf{H}_1} \leftarrow ()$; $\mathsf{cnt} \leftarrow 0$; $\mathcal{S}_{\mathsf{k\&r}} \leftarrow ()$; $\mathcal{S}_{\mathsf{H}_2} \leftarrow ()$; $\mathsf{abort} \leftarrow 0$;
101 $\mathcal{O}_{\mathsf{H}_1}(s)$:
102 **if** $\exists\, h : (s, h) \in \mathcal{S}_{\mathsf{H}_1}$ **then** return h;
103 **else** $\mathsf{cnt} \leftarrow \mathsf{cnt} + 1$; $h \leftarrow L_{\mathsf{H}_1}[\mathsf{cnt}]$; $\mathsf{Apd}(\mathcal{S}_{\mathsf{H}_1}, (s, h, \mathsf{cnt}))$; return h;
104 $\mathcal{O}_{\mathsf{H}_2}(s)$:
105 **if** $\exists\, \tau : (s, \tau) \in \mathcal{S}_{\mathsf{H}_2}$ **then** return τ
106 **else if** s has format $(X_1, X_2, Z_1, Z_2, pk_S, pk) \in \left(\mathbb{J}_N^+\right)^6$ **then**
107   **if** $\exists\, r, s, \tau_1, \tau_2 : ((X_1, X_2, pk_S, pk), (r, s, \tau_1, \tau_2)) \in \mathcal{S}_{\mathsf{k\&r}}$ **then**
108   **if** $Z_1^{\underline{r}} \circ Z_2 = pk^{\underline{s}}$ **then** return τ_1;
109   **if** $\exists\, r, s, \tau_1, \tau_2 : ((X_2, X_1, pk_S, pk), (r, s, \tau_1, \tau_2)) \in \mathcal{S}_{\mathsf{k\&r}}$ **then**
110   **if** $Z_2^{\underline{r}} \circ Z_1 = pk^{\underline{s}}$ **then** return τ_2;
111 **else** $\tau \leftarrow_{\mathrm{R}} \mathsf{ChSet}$; $\mathsf{Apd}(s_{\mathsf{H}_2}, (s, \tau))$; return τ;
112 $\mathcal{O}_{\mathsf{Sc}}(pk, m)$:
113 $\mathsf{cnt} \leftarrow \mathsf{cnt} + 1$; $h \leftarrow L_{\mathsf{H}_1}[\mathsf{cnt}]$; $z \leftarrow_{\mathrm{R}} [l]$; $s_1 \leftarrow R^{\underline{z}}$; $X_1 \leftarrow s_1{}^{\underline{e}} \circ pk_S{}^{\underline{-h}}$;
114 $r \leftarrow_{\mathrm{R}} [l]$; $s_2 \leftarrow_{\mathrm{R}} [l]$; $X_2 \leftarrow G^{\underline{s_2}} \circ X_1{}^{\underline{-r}}$; $\tau_1 \leftarrow_{\mathrm{R}} \mathbf{K}$; $\tau_2 \leftarrow_{\mathrm{R}} \mathbf{K}$;
115 **if** $\exists\, h', m', j : ((X_1, X_2, m', \tau_1), h', j) \in \mathcal{S}_{\mathsf{H}_1}$ **then** $\mathsf{abort} \leftarrow 1$;
116 $\mathsf{Apd}\,(\mathcal{S}_{\mathsf{H}_1}, ((X_1, X_2, m, \tau_1), h, \mathsf{cnt}))$; $\mathsf{Apd}\,(\mathcal{S}_{\mathsf{k\&r}}, ((X_1, X_2, pk_S, pk), (r, s_2, \tau_1, \tau_2)))$;
117 $c \leftarrow \mathsf{E}(\tau_2, m)$; $C \leftarrow (h, X_2, s_1, c)$; return C;
118 Finalization:
119 **if** $\mathcal{A}$ outputs (sk_R, pk_R, C^*) such that $\perp \neq \hat{m} \leftarrow \mathcal{O}_{\mathsf{Usc}}(sk_R, C^*)$ and $\mathcal{O}_{\mathsf{Sign}}(pk_R, \hat{m})$ was never issued **then**
120   Parse C^* as $(\hat{h}, \hat{X}_2, \hat{s}, \hat{c})$;
121     $\hat{X}_1 \leftarrow \hat{s}^{\underline{e}} \circ pk_S{}^{\underline{-\hat{h}}}$; $\hat{Z}_1 \leftarrow \hat{X}_1^{\underline{sk_R}}$; $\hat{Z}_2 \leftarrow \hat{X}_2^{\underline{sk_R}}$; $\hat{\tau}_1 \leftarrow \mathcal{O}_{\mathsf{H}_2}(\hat{X}_1, \hat{X}_2, \hat{Z}_1, \hat{Z}_2, pk_S, pk_R)$;
122   **if** $\exists j_0 \ : (\hat{X}_1, \hat{X}_2, \hat{m}, \hat{\tau}_1), \hat{h}, j_0) \in \mathcal{S}_{\mathsf{H}_1}$ **then** return $(j_0, \hat{X}_1, \hat{s})$;
123 return $(0, \epsilon, \epsilon)$;

As in the previous analysis, bad denotes the event: "(a) abort is set to 1 (see at line 115) or (b) in the execution of $\mathcal{O}_{\mathsf{H}_2}(\cdot)$, Z_1 and Z_2 are such that (see at lines 108 and 110) $Z_1^{\underline{r}} \circ Z_2 = pk^{\underline{s}}$ and $\mathrm{CDH}(X_1, pk) \neq Z_1$ or $\mathrm{CDH}(X_2, pk) \neq Z_2$."

Then

$$\Pr(\mathsf{bad}) \leqslant q_{\mathsf{Sc}}(q_{\mathsf{Sc}} - 1)\left(2(p'q')^2|\mathbf{K}|\right)^{-1} + 5q_{\mathsf{Sc}}\delta_0, \tag{7}$$

and then

$$\mathsf{Adv}^{\mathsf{suf}}_{\mathcal{A},\mathcal{SC}}(1^k) \leqslant \Pr(\mathsf{Succ}^{\mathsf{suf}}_{\mathcal{A}} \wedge \neg\mathsf{bad}) + q_{\mathsf{Sc}}(q_{\mathsf{Sc}} - 1)\left(2(p'q')^2|\mathbf{K}|\right)^{-1} + 5q_{\mathsf{Sc}}\delta_0. \tag{8}$$

Let fail be the event "the Finalization procedure outputs $(0, \epsilon, \epsilon)$". If the event $\mathsf{Succ}^{\mathsf{suf}}_{\mathcal{A}} \wedge \neg\mathsf{bad} \wedge \mathsf{fail}$ occurs then the oracle $\mathcal{O}_{\mathsf{H}_1}(\cdot)$ was never queried with value $(\hat{X}_1, \hat{X}_2, \hat{m}, \hat{\tau}_1)$. Which means that $\mathcal{A}$ successfully guessed $\mathcal{O}_{\mathsf{H}_1}(\hat{X}_1, \hat{X}_2, \hat{m}, \hat{\tau}_1)$. Under the RO model,

$$\Pr(\mathsf{Succ}^{\mathsf{suf}}_{\mathcal{A}} \wedge \neg\mathsf{bad} \wedge \mathsf{fail}) \leqslant |\mathsf{ChSet}|^{-1}. \tag{9}$$

Using $\mathcal{A}$ and the simulator, we obtain a machine $\mathcal{B}$ which takes $(dp, \mathcal{E}, Y_0, L_{\mathsf{H}_1} = (h_1, \cdots, h_q))$ as input and outputs $(j_0, \hat{X}_1, \hat{s})$ such that $\hat{s}^{\underline{e}} = X_1 Y_0^{\overline{h_{j_0}}}$ with probability $\varepsilon_1 = \Pr(\mathsf{Succ}^{\mathsf{suf}}_{\mathcal{A}} \wedge \neg \mathsf{bad} \wedge \neg \mathsf{fail})$. Let F_B be the forking algorithm [7, Sect. 3] associated to $\mathcal{B}$. By the General Forking Lemma [7, Lemma 1], from F_B's output, we have $(h_{j_0}, h'_{j_0}, X_1, \hat{s}, \hat{s'})$ such that $h_{j_0} \neq h'_{j_0}$, $\hat{s}^{\underline{e}} = X_1 Y_0^{\overline{h_{j_0}}}$, and $\hat{s'}^{\underline{e}} = X_1 Y_0^{\overline{h'_{j_0}}}$ with probability $\varepsilon_0 \geqslant \varepsilon_1(\varepsilon_1/q - 1/|\mathsf{ChSet}|)$. Then, using F_B and Shamir's trick (we use on page 9 when proving that $\mathcal{I}_{SSN}$ provides special soundness), we obtain a machine $\mathcal{B}_2$ which, on input Y_0, outputs X_0 such that $X_0^{\underline{e}} = Y_0$ with probability ε_0. Again, from the General Forking Lemma [7, Lemma 1],

$$\varepsilon_1 \leqslant q|\mathsf{ChSet}|^{-1} + \sqrt{q\varepsilon_0}. \tag{10}$$

The result follows from (8)–(10).

4.3 Soundness of Non-repudiation

Theorem 5. *Under the RO model, $\mathcal{SC}_{SSN}$ achieves $(t, q_{\mathsf{Sc}}, \varepsilon)$-computational soundness of non-repudiation, with $\varepsilon \leqslant 1/2 \cdot q(q-1)|\mathsf{ChSet}|^{-1} + 1/2 \cdot q_{\mathsf{Sc}}(q_{\mathsf{Sc}} - 1)(p'q')^{-2}|\mathbf{K}|^{-1} + 5q_{\mathsf{Sc}}\delta_0$, where $q = q_{\mathsf{H}_1} + q_{\mathsf{Sc}}$, wherein q_{H_1} is an upper bound on the number of $\mathcal{O}_{\mathsf{H}_1}(\cdot)$ queries $\mathcal{A}$ issues.*

Proof. First, we provide a simulation for Game 4. The simulator takes $dp = (N, G, R, e, k)$ and $\mathcal{E} = (\mathsf{E}, \mathsf{D}, \mathbf{K}, \mathbf{M}, \mathbf{C})$ as inputs. The initialization simply sets $\mathcal{S}_{\mathsf{H}_1} \leftarrow ()$; $\mathcal{S}_{\mathsf{k}} \leftarrow ()$; $\mathcal{S}_{\mathsf{k\&r}} \leftarrow ()$; $\mathcal{S}_{\mathsf{H}_2} \leftarrow ()$. The $\mathcal{O}_{\mathsf{H}_1}(\cdot)$ oracle is as described in lines 2–3 in the simulation for the confidentiality game. The $\mathcal{O}_{\mathsf{H}_2}(\cdot)$ and $\mathcal{O}_{\mathsf{Sc}}(\cdot,\cdot)$ oracles are as in lines 104–111 and 112–117 in the simulation for the unforgeability game, except that the lines 113 and 115 are replaced respectively with the lines 200 and 201, hereunder:

200 $h \leftarrow_{\mathrm{R}} \mathsf{ChSet}$;
201 **if** $\exists\, h', m' : ((X_1, X_2, m', \tau_1), h') \in \mathcal{S}_{\mathsf{H}_1}$ **then** abort $\leftarrow 1$.

Defining bad as in the proof of Theorem 4, the inequality (7) still holds. Then

$$\mathsf{Adv}^{\mathsf{snr}}_{\mathcal{A},\mathcal{SC}}(1^k) \leqslant \Pr(\mathsf{Succ}^{\mathsf{snr}}_{\mathcal{A}} \wedge \neg\mathsf{bad}) + q_{\mathsf{Sc}}(q_{\mathsf{Sc}} - 1)\left(2(p'q')^2|\mathbf{K}|\right)^{-1} + 5q_{\mathsf{Sc}}\delta_0. \tag{11}$$

If $\mathcal{A}$ succeeds and $\neg$bad, $\mathcal{A}$ outputs $(sk_R, pk_R, C^*, m', nr)$ such that $m' \neq m \leftarrow \mathsf{Usc}(sk_R, pk_S, C^*)$ and $1 = d \leftarrow \mathsf{PV}(C^*, m', nr, pk_S, pk_R)$. Let $C^* = (\hat{h}, \hat{X}_2, \hat{s}, \hat{c})$, $nr = (\tau_1, \tau_2)$, $\widehat{nr} = (\hat{\tau}_1, \hat{\tau}_2) \leftarrow \mathsf{N}(sk_R, pk_S, C^*)$, and $\hat{X}_1 \leftarrow s^{\underline{e}} \circ pk_S^{\overline{-h}}$. As $m \neq m'$ and $1 = d \leftarrow \mathsf{PV}(C^*, m', nr, pk_S, pk_R) = d' \leftarrow \mathsf{PV}(C^*, m, \widehat{nr}, pk_S, pk_R)$. $\mathcal{A}$ have found $(m, \hat{\tau}_1)$ and (m', τ_1) such that $\hat{h} = h_1 \leftarrow \mathcal{O}_{\mathsf{H}_1}(\hat{X}_1, \hat{X}_2, m, \hat{\tau}_1) = h_2 \leftarrow \mathcal{O}_{\mathsf{H}_1}(\hat{X}_1, \hat{X}_2, m', \tau_1)$. Then

$$\Pr(\mathsf{Succ}^{\mathsf{snr}}_{\mathcal{A}} \wedge \neg\mathsf{bad}) \leqslant q(q-1)(2 \cdot |\mathsf{ChSet}|)^{-1}. \tag{12}$$

The Theorem follows from (11) and (12). $\square$

4.4 Unforgeability of Non-repudiation Evidence

Theorem 6. *Under the RO model, if the factoring problem is* $(t(k), \varepsilon(k))$ *hard, then the* $\mathcal{SC}_{SSN}$ *scheme achieves* $(t, q_{\mathsf{Sc}}, q_{\mathsf{Usc}}, q_{\mathsf{N}}, \varepsilon')$ *unforgeability of non-repudiation evidence with* $\varepsilon' \leqslant \varepsilon + |\mathbf{K}|^{-1} + q_{\mathsf{Sc}}(q_{\mathsf{Sc}} - 1)\left(2(p'q')^2\right)^{-1} + (5q_{\mathsf{Sc}} + 2)\delta_0$.

Proof. We consider the following simulation.

Simulation for Unforgeability of non–repudiation evidence

Input: $dp = (N, G, R, e, k)$, $\mathcal{E} = (\mathsf{E}, \mathsf{D}, \mathbf{K}, \mathbf{M}, \mathbf{C})$, $X_0, Y_0 \leftarrow_{\mathrm{R}} \mathbb{J}_N^+$, $L_{\mathsf{H}_1} = (h_1, h_2, \cdots, h_q)$.
External Oracles: $\mathrm{DDH}_{Y_0}(\cdot,\cdot)$

<u>Initialization:</u> $a \leftarrow [l]$; $(sk_S, pk_S) \leftarrow (R^{\underline{a}}, G^{\underline{a}})$; $pk_R \leftarrow Y_0$; $\mathcal{S}_{\mathsf{H}_1} \leftarrow ()$; $\mathsf{cnt} \leftarrow 0$; $\mathcal{S}_{\mathsf{k}} \leftarrow ()$; $\mathcal{S}_{\mathsf{k\&r}} \leftarrow ()$; $\mathcal{S}_{\mathsf{H}_2} \leftarrow ()$;
<u>$\mathcal{O}_{\mathsf{H}_1}(s)$</u>: is defined as in the simulation for the confidentiality game, at lines 2–3.
<u>$\mathcal{O}_{\mathsf{H}_2}(s)$</u>:
if $\exists\, \tau : (s, \tau) \in \mathcal{S}_{\mathsf{H}_2}$ **then** return τ;
else if s has format $(X_1, X_2, Z_1, Z_2, pk, pk' = pk_R) \in \left(\mathbb{J}_N^+\right)^6$ **then**
  if $pk = pk_S$ and $\exists \tau, x : ((X_1, X_2, Z_1, \epsilon, pk_S, pk_R), \tau, x)) \in \mathcal{S}_{\mathsf{k\&r}}$ and $\mathrm{DDH}_{Y_0}(X_2, Z_2) = 1$ **then** $\mathsf{Apd}(\mathcal{S}_{\mathsf{H}_2}, (s, \tau))$; return τ;
  if $pk = pk_S$ and $\exists \tau, x : ((X_1, X_2, \epsilon, Z_2, pk_S, pk_R), \tau, x)) \in \mathcal{S}_{\mathsf{k\&r}}$ and $\mathrm{DDH}_{Y_0}(X_1, Z_1) = 1$ **then** $\mathsf{Apd}(\mathcal{S}_{\mathsf{H}_2}, (s, \tau))$; return τ;
  if $\exists\, \tau : ((X_1, X_2, pk, pk_R), \tau) \in \mathcal{S}_{\mathsf{k}}$ and $\mathrm{DDH}_{Y_0}(X_1, Z_1) = \mathrm{DDH}_{Y_0}(X_2, Z_2) = 1$ **then** $\mathsf{Apd}(\mathcal{S}_{\mathsf{H}_2}, (s, \tau))$; return τ;
else $\tau \leftarrow_{\mathrm{R}} \mathsf{ChSet}$; $\mathsf{Apd}(s_{\mathsf{H}_2}, (s, \tau))$; return τ;

<u>$\mathcal{O}_{\mathsf{Sc}}(pk, m)$</u>:
$x_1 \leftarrow_{\mathrm{R}} [l]$; $X_1 \leftarrow G^{\underline{x_1}}$; $Z_1 = pk_R{}^{\underline{x_1}}$; $x_2 \leftarrow_{\mathrm{R}} [l]$;
if $pk \neq pk_R$ **then**
  $X_2 \leftarrow G^{\underline{x_2}}$; $Z_2 = pk_R{}^{\underline{x_2}}$;
  $\tau_1 \leftarrow \mathcal{O}_{\mathsf{H}_2}(X_1, X_2, Z_1, Z_2, pk_S, pk)$; $\tau_2 \leftarrow \mathcal{O}_{\mathsf{H}_2}(X_2, X_1, Z_2, Z_1, pk_S, pk)$;
else
  $X_2 \leftarrow X_0 \circ G^{\underline{x_2}}$; $\tau_1 \leftarrow_{\mathrm{R}} \mathbf{K}$; $\tau_2 \leftarrow_{\mathrm{R}} \mathbf{K}$; ▶ *The simulator takes X_0, Y_0 as inputs*
  $\mathsf{Apd}(\mathcal{S}_{\mathsf{k\&r}}, ((X_1, X_2, Z_1, \epsilon, pk_S, pk_R), \tau_1, x_2))$; ▶ *$pk = pk_R$;*
  $\mathsf{Apd}(\mathcal{S}_{\mathsf{k\&r}}, ((X_2, X_1, \epsilon, Z_1, pk_S, pk_R), \tau_2, x_2))$;
$h \leftarrow \mathcal{O}_{\mathsf{H}_1}(X_1, X_2, m, \tau_1)$; $c \leftarrow \mathsf{E}(\tau_2, m)$; $s \leftarrow R^{\underline{x_1}} \circ sk_S{}^{\underline{h}}$; return (h, X_2, s, c);

<u>$\mathcal{O}_{\mathsf{Usc}}(pk, C)$</u>: <u>$\mathcal{O}_{\mathsf{N}}(pk, C)$:</u>
if $pk \notin \mathbb{J}_N^+$ **then** return $\perp$;
Parse C as $(h, X_2, s, c) \in \mathsf{ChSet} \times \mathbb{J}_N^+ \times \mathbb{J}_N^+ \times \mathbf{C}$; $X_1 \leftarrow s^{\underline{e}} \circ pk^{\underline{-h}}$;
if $\exists\, Z_1, Z_2 \in \mathbb{J}_N^+, \tau \in \mathbf{K} : ((X_1, X_2, Z_1, Z_2, pk, pk_R), \tau) \in \mathcal{S}_{\mathsf{H}_2}$ and $\mathrm{DDH}_{Y_0}(X_1, Z_1) = \mathrm{DDH}_{Y_0}(X_2, Z_2) = 1$ **then**
  $\tau_1 \leftarrow \tau$; ▶ $\mathsf{H}_2(X_1, X_2, Z_1, Z_2, pk, pk_R)$ *was issued*
else if $pk = pk_R$ and $\exists\, \tau, x : ((X_1, X_2, Z_1, \epsilon, pk_S, pk_R), \tau, x) \in \mathcal{S}_{\mathsf{k\&r}}$ **then**
  $\tau_1 \leftarrow \tau$ ▶ *$\mathcal{O}_{\mathsf{Sc}}(\cdot,\cdot)$ returned (h, X_2, s, c') for some c'*
else if $\exists\, \tau : ((X_1, X_2, pk, pk_R), \tau) \in \mathcal{S}_{\mathsf{k}}$ **then**
  $\tau_1 \leftarrow \tau$; ▶ $\mathsf{Usc}(pk, C')$ *or* $\mathsf{N}(pk, C')$ *such that C' parses as (h, X_2, s, c') was issued*
else $\tau_1 \leftarrow_{\mathrm{R}} \mathbf{K}$; $\mathsf{Apd}(\mathcal{S}_{\mathsf{k}}, ((X_1, X_2, pk, pk_R), \tau_1))$;

328 **if** $\exists\, Z_2, Z_1 \in \mathbb{J}_N^+, \tau \in \mathbf{K} : ((X_2, X_1, Z_2, Z_1, pk, pk_R), \tau) \in \mathcal{S}_{\mathsf{H}_2}$ and $\mathrm{DDH}_{Y_0}(X_1, Z_1) = \mathrm{DDH}_{Y_0}(X_2, Z_2) = 1$ **then** $\tau_2 \leftarrow \tau$; ▶ *the same treatment as for* τ_1
329 **else if** $pk = pk_R$ and $\exists\, \tau, x : ((X_2, X_1, \epsilon, Z_2, pk_S, pk_R), \tau, x) \in \mathcal{S}_{\mathsf{k\&r}}$ **then** $\tau_2 \leftarrow \tau$
330 **else if** $\exists\, \tau : ((X_2, X_1, pk, pk_R), \tau) \in \mathcal{S}_{\mathsf{k}}$ **then** $\tau_2 \leftarrow \tau$;
331 **else** $\tau_2 \leftarrow_{\mathrm{R}} \mathbf{K}$; $\mathsf{Apd}(\mathcal{S}_{\mathsf{k}}, ((X_2, X_1, pk, pk_R), \tau_2))$;
332 $m \leftarrow \mathsf{D}(\tau_2, c)$; $h' \leftarrow \underline{\mathcal{O}_{\mathsf{H}_1}}(X_1, X_2, m, \tau_1)$;
333 **if** $h = h'$ **then** $\overset{\mathcal{O}_{\mathsf{Usc}}}{\boxed{\text{return } m}}$ $\overset{\mathcal{O}_{\mathsf{N}}}{\boxed{\text{return } (\tau_1, \tau_2)}}$ **else** return $\perp$;

334 Finalization:
335 **if** $\mathcal{A}$ outputs (C^*, m^*, nr^*) such that C^* was generated through $\mathcal{O}_{\mathsf{Sc}}(\cdot,\cdot)$, $1 = d \leftarrow \mathsf{PV}(C^*, m^*, nr^*, pk_S, pk_R)$ and nr^* was not generated by the oracle $\mathcal{O}_{\mathsf{N}}(\cdot,\cdot)$ on a query on (pk_S, C^*) **then**
336 Parse C^* as $(\hat{h}, \hat{X}_2, \hat{s}, \hat{c})$ and nr^* as $(\hat{\tau}_1, \hat{\tau}_2)$;
$\hat{X}_1 \leftarrow \hat{s}^{\underline{e}} \circ pk_S^{\underline{-\hat{h}}}$;
337 Recover $((\hat{X}_1, \hat{X}_2, \hat{Z}_1, \epsilon, pk_S, pk_R), \hat{\tau}, x)$ from $\mathcal{S}_{\mathsf{k\&r}}$ ▶ *As* C^* *was output by* $\mathcal{O}_{\mathsf{Sc}}(\cdot,\cdot)$ *there are some* $\hat{Z}_1, \hat{\tau}, x : ((\hat{X}_1, \hat{X}_2, \hat{Z}_1, \epsilon, pk_S, pk_R), \hat{\tau}, x)) \in \mathcal{S}_{\mathsf{k\&r}}$ *(see at line 316)*
338 **if** $\exists\, \hat{Z}_1, \hat{Z}_2 \in \mathbb{J}_N^+ : ((\hat{X}_1, \hat{X}_2, \hat{Z}_1, \hat{Z}_2, pk_S, pk_R), \hat{\tau}_1) \in \mathcal{S}_{\mathsf{H}_2}$ and $\mathrm{DDH}_{Y_0}(\hat{X}_2, \hat{Z}_2) = 1$ **then**
$U_0 \leftarrow Z_2 \circ pk_R^{\underline{-x}}$; return U_0;
339 return ϵ;

Let bad denote the event "the same couple (X_1, X_2) is generated in two executions of $\mathcal{O}_{\mathsf{Sign}}(\cdot,\cdot)$". Then, under the RO model,

$$\Pr(\mathsf{bad}) \leqslant \frac{1}{2} q_{\mathsf{Sc}}(q_{\mathsf{Sc}} - 1)(p'q')^{-2} + 5 q_{\mathsf{Sc}} \delta_0. \tag{13}$$

Let fail be the event "the Finalization procedure outputs ϵ". If $\mathsf{Succ}_{\mathcal{A}}^{\mathsf{unr}} \wedge \neg\mathsf{bad} \wedge \mathsf{fail}$ occurs, $\mathcal{A}$ never query the $\mathcal{O}_{\mathsf{H2}}$ oracle on $(\hat{X}_1, \hat{X}_2, \mathrm{CDH}(pk_R, \hat{X}_1), \mathrm{CDH}(pk_R, \hat{X}_2), pk_S, pk_R)$; then $\mathcal{A}$ successfully guessed the corresponding digest value. It follows

$$\Pr(\mathsf{Succ}_{\mathcal{A}}^{\mathsf{unr}} \wedge \neg\mathsf{bad} \wedge \mathsf{fail}) \leqslant |\mathbf{K}|^{-1}. \tag{14}$$

If $\mathsf{Succ}_{\mathcal{A}}^{\mathsf{unr}} \wedge \neg\mathsf{bad} \wedge \neg\mathsf{fail}$ occurs, as $\hat{X}_2 = X_0 \circ G^{\underline{x}}$ and $\hat{Z}_2 = \mathrm{CDH}(X_2, pk_R = Y_0)$

$$U_0 = \mathrm{CDH}(X_0, Y_0) = Z_2 \circ pk_R^{\underline{-x}}. \tag{15}$$

Using $\mathcal{A}$ and the simulator, we have a machine which takes X_0, Y_0 as input and outputs $\mathrm{CDH}(X_0, Y_0)$ with probability $\Pr(\mathsf{Succ}_{\mathcal{A}}^{\mathsf{unr}} \wedge \neg\mathsf{bad} \wedge \neg\mathsf{fail})$. The result follows from (13), (14), and [14, Theorem 2]. □

5 Concluding Remarks

We have proposed a new identification scheme over the group of signed quadratic residues, wherein the strong Diffie–Hellman assumption holds under the factoring assumption. Using the identification scheme, we derived a new signature scheme we have shown to be strongly unforgeable against chosen message attacks, under the RSA assumption and the Random Oracle model. We proposed an efficient signcryption scheme with non-interactive non-repudiation, we have shown to be insider secure, under the RSA assumption and the RO model, in a variant of Fan *et al.*'s security model. The communication overhead of the signcryption scheme, compared to the corresponding signature scheme is one group element.

Compared to Fan *et al.*'s design which uses bilinear maps, our scheme is RSA based and can be easily deployed in most of the existing platforms.

In a forthcoming stage, we will be interested in the conditions under which our design can be generalized to generic Diffie–Hellman groups. We will investigate also signcryption designs with a tight security reduction.

References

1. Algesheimer, J., Camenisch, J., Shoup, V.: Efficient computation modulo a shared secret with application to the generation of shared safe-prime products. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 417–432. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45708-9_27
2. 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
3. Baek, J., Steinfeld, R.: Security for signcryption: the multi-user model. In: Dent, A., Zheng, Y. (eds.) Practical Signcryption. ISC, pp. 43–53. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-540-89411-7_3
4. 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). https://doi.org/10.1007/3-540-45664-3_6
5. Baek, J., Steinfeld, R., Zheng, Y.: Formal proofs for the security of signcryption. J. Cryptol. **20**(2), 203–235 (2007)
6. 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
7. Bellare, M., Neven, G.: Multi-signatures in the plain public-key model and a general forking lemma. In: Proceedings of the 13th ACM Conference on Computer and Communications Security, pp. 390–399. ACM (2006)
8. Bellare, M., Rogaway, P.: Random oracle are practical: a paradigm for designing efficient protocols. In: ACM-CCS 1993, pp. 62–73. ACM (1993)
9. Boyd, C., Mathuria, A.: Protocols for Authentication and Key Establishment. Springer Science & Business Media, Heidelberg (2003). https://doi.org/10.1007/978-3-662-09527-0
10. Cash, D., Kiltz, E., Shoup, V.: The twin Diffie-Hellman problem and applications. J. Cryptol. **22**(4), 470–504 (2009)
11. 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
12. Fiat, A., Shamir, A.: How to prove yourself: practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (1987). https://doi.org/10.1007/3-540-47721-7_12

13. Guillou, L.C., Quisquater, J.-J.: A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory. In: Barstow, D., Brauer, W., Brinch Hansen, P., Gries, D., Luckham, D., Moler, C., Pnueli, A., Seegmüller, G., Stoer, J., Wirth, N., Günther, C.G. (eds.) EUROCRYPT 1988. LNCS, vol. 330, pp. 123–128. Springer, Heidelberg (1988). https://doi.org/10.1007/3-540-45961-8_11
14. Hofheinz, D., Kiltz, E.: The group of signed quadratic residues and applications. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 637–653. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-03356-8_37
15. Kiltz, E., Masny, D., Pan, J.: Optimal security proofs for signatures from identification schemes. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016. LNCS, vol. 9815, pp. 33–61. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53008-5_2
16. Lenstra, A.K., Hughes, J.P., Augier, M., Bos, J.W., Kleinjung, T., Wachter, C.: Public keys. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 626–642. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32009-5_37
17. Malone-Lee, J.: Signcryption with non-interactive non-repudiation. Des. Codes Crypt. **37**(1), 81–109 (2005)
18. Menezes, A., van Oorschot, P., Vanstone, S.: Handbook of Applied Cryptography. CRC Press, Boca Raton (1996)
19. Okamoto, T., Pointcheval, D.: The gap-problems: a new class of problems for the security of cryptographic schemes. In: Kim, K. (ed.) PKC 2001. LNCS, vol. 1992, pp. 104–118. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44586-2_8
20. Sarr, A.P., Elbaz–Vincent, P.: On the security of the (F)HMQV protocol. In: Pointcheval, D., Nitaj, A., Rachidi, T. (eds.) AFRICACRYPT 2016. LNCS, vol. 9646, pp. 207–224. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-31517-1_11
21. Sarr, A.P., Elbaz-Vincent, P., Bajard, J.-C.: A secure and efficient authenticated Diffie–Hellman protocol. In: Martinelli, F., Preneel, B. (eds.) EuroPKI 2009. LNCS, vol. 6391, pp. 83–98. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-16441-5_6
22. Shin, J.-B., Lee, K., Shim, K.: New DSA-verifiable signcryption schemes. In: Lee, P.J., Lim, C.H. (eds.) ICISC 2002. LNCS, vol. 2587, pp. 35–47. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36552-4_3
23. Zheng, Y.: Digital signcryption or how to achieve cost(signature & encryption) $\ll$ cost(signature) + cost(encryption). In: Kaliski, B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 165–179. Springer, Heidelberg (1997). https://doi.org/10.1007/BFb0052234