A Computational Diffie-Hellman based Insider Secure Signcryption with Non-Interactive Non-Repudiation

Augustin P. Sarr, Ngarenon Togde

A Computational Diffie–Hellman based Insider Secure Signcryption with Non–Interactive Non–Repudiation

Ngarenon TOGDE and Augustin P. SARR

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

Abstract. An important advantage of signcryption schemes compared to one pass key exchange protocols is non–interactive non–repudiation (NINR). This attribute offers to the receiver of a signcrypted ciphertext the ability to generate a non–repudiation evidence, that can be verified by a third party without executing a costly multi–round protocol. We propose a computational Diffie–Hellman based insider secure signcryption scheme with non–interactive non–repudiation. Namely, we show that under the computational Diffie–Hellman assumption and the random oracle model, our scheme is *tightly* insider secure, provided the underlying encryption scheme is semantically secure. Compared to a large majority of the previously proposed signcryption schemes with NINR, our construction is more efficient and it does not use any specificity of the underlying group, such as pairings. The communication overhead of our construction, compared to Chevallier Mâmes' signature scheme is one group element.

Keywords: signcryption, non–interactive non–repudiation, insider security, computational Diffie–Hellman, random oracle model.

1 Introduction

A signcryption scheme provides simultaneously the functionalities of encryption and signature schemes [23]. A natural use of a signcryption scheme is to build an asynchronous secure channel *i. e.* a confidential and authenticated asynchronous channel. Given the similar uses of signcryption and (one pass) Key Exchange Protocols (KEP), to build confidential and authenticated channels, it appears, from a real world perspective, that the right security definition for signcryption schemes is insider security [3]. Informally, insider security ensures (*i*) *confidentiality* even if the sender's static private key is revealed to the attacker, and (*ii*) *unforgeability* even if the receiver's static private key is disclosed.

A signcryption scheme is said to provide *non–repudiation*, if the receiver of a signcrypted ciphertext has the ability to generate a non–repudiation evidence, that can be verified by a third party (a judge, for instance); as a result, a message sender cannot deny having signcrypted the message. The non–repudiation attribute is said to be *non–interactive*, if a non–repudiation evidence can be

generated and verified without executing a multi–round protocol. An important advantage of signcryption schemes, compared to one pass KEP, which often outperforms signcryption schemes, is non–interactive non–repudiation (NINR).

A signcryption scheme with the aim to provide NINR was proposed for the first time by Bao and Deng [5]; unfortunately their design fails in achieving confidentiality [19]. Malone–Lee [19] proposes an efficient design with NINR he analyses in the Random Oracle (RO) model. The scheme achieves confidentiality under the computational Diffie–Hellman (cDH) assumption, and unforgeability under the gap Diffie–Hellman Assumption. Unfortunately, the security model he uses is closer to the outsider than to the insider model. Indeed, the scheme fails in providing insider confidentiality. In [8], Bjørstad and Dent (BD) propose a design based on Chevallier Mâmes' (CM) signature scheme they show to tightly achieve insider unforgeability under the cDH assumption and *outsider* confidentiality under the gap DH assumption. Unfortunately, as for the ML scheme, it can be shown that the BD scheme does not achieve insider confidentiality.

In subsequent works [2, 13, 14, 20, 22], several insider secure schemes with NINR have been proposed. The designs offer a superior security, compared to the ML or BD schemes. However, they are less efficient and often assume some specificities of the underlying groups, such as the existence of a bilinear pairing. In [2], Arriaga *et al.* propose a generic insider secure signcryption scheme, with randomness reuse, in the standard model. They exhibit an insider secure instantiation of their design, under the Decisional Bilinear and the q–Strong Diffie–Hellman (DBDH and q–sDH) assumptions. Unfortunately, the unforgeability is achieved in the registered key model [20], wherein an attacker is required to register the *keys pairs* it uses in its attack. Matsuda *et al.* [20] propose a generic composition of signature and tag based encryption schemes, which yields to different shades of security depending on the security attributes of the base schemes. They exhibit two constructions with NINR that fully achieve insider confidentiality (under the cDH and the gap DH assumptions respectively) and unforgeability (under the co–cDH assumption). Chiba *et al.* [13] propose a generic construction of signcryption schemes, and exhibit two insider secure constructions with NINR under the DBDH and the q–sDH assumptions. In [14], Fan *et al.* propose a signcryption scheme with non–interactive non–repudiation (SCNINR), based on Boneh *et al.*'s signature scheme [10], they show to be insider secure under the DBDH assumption, without resorting the RO model. Sarr *et al.* [22] propose, over the group of signed quadratic residues, a SCNINR, based on a signature scheme of their own design, they show to be insider secure under the RSA assumption and the RO model.

The basic design principle in the SCNINR schemes from [8, 14, 19, 22], is (*i*) a Diffie–Hellman (DH) secret derivation, using ephemeral keys from the sender and the receiver's static public key, followed by (*ii*) an encryption using some part of the derived secret, and (*iii*) a signature generation, using the sender's private key, on the plain text and some part of the derived DH secret. One may notice also that these schemes assume rather specific groups or have loose security reductions. As tightly secure cDH based signature schemes exist [12, 15, 17], we

investigate whether such schemes can be leveraged as building blocks for tightly (multi–user) insider secure cDH based SCNINR schemes. As we aim at an efficient design, we use the random oracle (RO) model. We propose a new SCNINR, termed $\mathcal{SC}_{\mathsf{edl}}$, based on a variant of Chevallier–Mâmes' signature scheme [12], tailored to (*i*) be combined with Cash *et al.*'s twin Diffie–Hellman key exchange [11], (*ii*) and to allow a use of the same randomness in the DH key exchange and in the signature generation. And, using the trapdoor test technique [11], we show that $\mathcal{SC}_{\mathsf{edl}}$ is tightly insider secure under the cDH assumption and the RO model, provided the underlying symmetric encryption scheme is semantically secure. Even better, we show the insider confidentiality attribute in the *secret key ignorant* multi–user model, *i. e.*, when the sender public key is chosen by the adversary and the challenger does not know the corresponding private key. Compared to the ML and BD schemes, which do not require any specificity of the underlying group and do not achieve insider security, $\mathcal{SC}_{\mathsf{edl}}$ offers a stronger security, even if it is less efficient. And, compared to the schemes from [2, 13, 14, 20, 22], $\mathcal{SC}_{\mathsf{edl}}$ offers a tight security reduction, a better efficiency, and a comparable or a superior security.

This paper is organized as follows. In Section 2, we present some preliminaries on the syntax of SCNINR schemes and the insider security definitions for SCNINR. In Section 3, we propose the $\mathcal{SC}_{\mathsf{edl}}$ scheme. We propose a detailed security analysis in Section 4, and compare our design with previous constructions in Section 5.

2 Preliminaries

Notations. $\mathcal{G} = \langle G \rangle$ is a cyclic group of prime order p, $\mathcal{G}^*$ denotes the set $\mathcal{G} \setminus \{1\}$. We denote by $\mathsf{Exp}(\mathcal{G}, t)$ the computational effort required to perform t exponentiations with $|p|$–bits exponents in $\mathcal{G}$; $\mathsf{Exp}(\mathcal{G})$ denotes $\mathsf{Exp}(\mathcal{G}, 1)$. For an integer n, $[n]$ denotes the set $\{0, \cdots, n\}$. If S is a set, $a \leftarrow_{\mathrm{R}} S$ means that a is chosen uniformly at random from S; we write $a, b, c, \cdots \leftarrow_{\mathrm{R}} S$ as a shorthand for $a \leftarrow_{\mathrm{R}} S; b \leftarrow_{\mathrm{R}} S$, etc. We denote by $\mathsf{sz}(S)$ the number of bits required to represent $a \in S$. For a probabilistic algorithm $\mathcal{A}$ with parameters $u_1, \cdots, u_n$ and output $V \in \mathbf{V}$, we write $V \leftarrow_{\mathrm{R}} \mathcal{A}(u_1, \cdots, u_n)$. We denote by $\{\mathcal{A}(u_1, \cdots, u_n)\}$ the set $\{v \in \mathbf{V} : \Pr(V = v) \neq 0\}$. 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 as a bit–string of the tuple such that each element can be unequivocally parsed.

The cDH *Assumption.* We assume the existence of an algorithm $\mathsf{Setup}_{\mathsf{grp}}(\cdot)$, which on input a security parameter k outputs a system parameter Π_k which fully identifies a group $\mathcal{G} = \langle G \rangle$ together with its order. For $X \in \mathcal{G}$, we denote the smallest non–negative integer x such that $G^x = X$ by $\log_G X$. For, $X, Y \in \mathcal{G}$, we denote $G^{(\log_G X)(\log_G Y)}$ by $\mathsf{cDH}(X, Y)$; if $B \in \mathcal{G}$, we denote $(\mathsf{cDH}(X, B), \mathsf{cDH}(Y, B))$ by $\mathsf{2DH}(X, Y, B)$. The cDH assumption is said to hold in $\mathcal{G}$ if for all efficient algorithms $\mathcal{A}$,

$$\mathsf{Adv}^{\mathsf{cDH}}_{\mathcal{A}}(\mathcal{G}) = \Pr\left[X, Y \leftarrow_{\mathrm{R}} \mathcal{G}; Z \leftarrow_{\mathrm{R}} \mathcal{A}(G, X, Y) : Z = \mathsf{cDH}(X, Y)\right] \text{ is negligible in } k.$$

A Symmetric Encryption scheme $\mathcal{E} = (\mathsf{E}, \mathsf{D}, \mathbf{K}, \mathbf{M}, \mathbf{C})$ is a pair of efficient algorithms (E, D) together with a triple of sets $(\mathbf{K}, \mathbf{M}, \mathbf{C})$, which depend on the security parameter k, such that for all $\tau \in \mathbf{K}$ and all $m \in \mathbf{M}$, it holds that $\mathsf{E}(\tau, m) \in \mathbf{C}$ and $m = \mathsf{D}(\tau, \mathsf{E}(\tau, m))$. Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an adversary against $\mathcal{E}$ and let

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

then $\mathsf{Adv}^{\mathsf{ss}}_{\mathcal{A},\mathcal{E}}(k)$ denotes the quantity $\mathsf{Adv}^{\mathsf{ss}}_{\mathcal{A},\mathcal{E}}(k) = |\Pr(O_0) - \Pr(O_1)|$, where $m_0, m_1 \in \mathbf{M}$ are distinct equal length messages. The scheme $\mathcal{E}$ is said to be $(t, \varepsilon(k))$–*semantically secure* if for all adversaries $\mathcal{A}$ running in time t, $\mathsf{Adv}^{\mathsf{ss}}_{\mathcal{A},\mathcal{E}}(k) \leqslant \varepsilon(k)$.

2.1 Insider Security for SCNINR

We recall the syntax of a SCNINR scheme and the insider security definitions in the Flexible Signcryption / Flexible Unsigncryption Oracle (FSO/FUO) model [4], also termed dynamic Multi–User model [2].

Definition 1. *A* signcryption scheme *is a quintuple of algorithms* $\mathcal{SC} = (\mathsf{Setup}, \mathsf{Gen}_S, \mathsf{Gen}_R, \mathsf{Sc}, \mathsf{Usc})$ *where:*

a) Setup *takes a security parameter* k *as input, and outputs a public domain parameter* dp.
b) Gen_S *is the sender key pair generation algorithm. It takes* dp *as input and outputs a key pair* (sk_S, pk_S)*, wherein* sk_S *is the signcrypting key.*
c) Gen_R *is the receiver key pair generation algorithm; it takes* dp *as input and outputs a key pair* (sk_R, pk_R).
d) Sc *takes as inputs* dp *(an implicit parameter), a sender private key* sk_S*, a receiver public key* pk_R*, and a message* m*, and outputs a signcryptext* C*. We write* $C \leftarrow_{R} \mathsf{Sc}(sk_S, pk_R, m)$.
e) Usc *is a deterministic algorithm. It takes as inputs* dp*, a receiver secret key* sk_R*, a sender public key* pk_S*, 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 $dp \in \{\mathsf{Setup}(k)\}$*, all* $m \in \mathbf{M}$*, 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 are two algorithms* N *and* PV*, termed* non–repudiation evidence generation *and* pubic verification algorithms *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*, a sender public key* pk_S*, and a receiver public key* pk_R*, and outputs* $d \in \{0, 1\}$*; we write* $d \leftarrow \mathsf{PV}(C, m, nr, pk_S, pk_R)$.
- *For all* $dp \in \{\mathsf{Setup}(k)\}$*, 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* $1 = d \leftarrow \mathsf{PV}(C, m, nr, pk_S, pk_R)$.

Game 1 SKI–MU Insider Confidentiality in the FSO/FUO–IND–CCA2 sense

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 $\mathcal{SC}$;

1) The challenger generates $dp \leftarrow_{\mathrm{R}} \mathsf{Setup}(k)$ and $(sk_R, pk_R) \leftarrow_{\mathrm{R}} \mathsf{Gen}_R(dp)$;
2) $\mathcal{A}_1$ is provided with dp and pk_R, and is given access to: (a) an unsigncryption oracle $\mathcal{O}_{\mathsf{Usc}}(\cdot,\cdot)$, which takes as inputs a public key pk and a signcrypted text C, and outputs $m \leftarrow \mathsf{Usc}(sk_R, pk, C)$, and ($b$) a non–repudiation evidence generation oracle $\mathcal{O}_{\mathsf{N}}(\cdot,\cdot)$ 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 $(m_0, m_1, pk_S, st) \leftarrow_{\mathrm{R}} \mathcal{A}_1^{\mathcal{O}_{\mathsf{Usc}}(\cdot,\cdot),\mathcal{O}_{\mathsf{N}}(\cdot,\cdot)}(pk_R)$ where $m_0, m_1 \in \mathbf{M}$ are distinct equal length messages, st is a state, and pk_S is the attacked sender public key (sk_S is unknown to the challenger).
4) In the experiment $E_{b,b=0,1}$, the challenger computes $C^* \leftarrow_{\mathrm{R}} \mathsf{Sc}(sk_S, pk_R, m_b)$.
5) $\mathcal{A}_2$ outputs $b' \leftarrow_{\mathrm{R}} \mathcal{A}_2^{\mathcal{O}_{\mathsf{Usc}}(\cdot,\cdot),\mathcal{O}_{\mathsf{N}}(\cdot,\cdot)}(C^*, st)$ ($\mathcal{O}_{\mathsf{Usc}}(\cdot,\cdot)$ and $\mathcal{O}_{\mathsf{N}}(\cdot,\cdot)$ are as in step 2).
6) For $E_{b,b=0,1}$, out_b denotes the event: (i) $\mathcal{A}_2$ never issued $\mathcal{O}_{\mathsf{Usc}}(pk_S, C^*)$ or $\mathcal{O}_{\mathsf{N}}(pk_S, C^*)$, and (ii) $b' = 1$.

And, $\mathsf{Adv}^{\mathsf{cca2}}_{\mathcal{A},\mathcal{SC}}(k) = \mid \Pr(\mathsf{out}_0) - \Pr(\mathsf{out}_1) \mid$ denotes $\mathcal{A}$'s CCA2 insider security advantage.

Definition 2 (*Secret Key Ignorant* Multi-User Insider Confidentiality). *A SCNINR $\mathcal{SC}$ is said to be $(t, q_{\mathsf{Usc}}, q_{\mathsf{N}}, \varepsilon)$–secure in the Secret Key Ignorant Multi–User (SKI–MU) insider confidentiality in the FSO/FUO IND–CCA2 sense, if for all adversaries $\mathcal{A}$ playing Game 1, running in time t, and issuing respectively q_{Usc} and q_{N} queries to the unsigncryption and non–repudiation evidence generation oracles, $\mathsf{Adv}^{\mathsf{cca2}}_{\mathcal{A},\mathcal{SC}}(k) \leqslant \varepsilon$.*

Game 2 MU Insider Unforgeability in the FSO/FUO–sUF–CMA sense

$\mathcal{A}$ is a forger, $dp \leftarrow_{\mathrm{R}} \mathsf{Setup}(k)$ still denotes the public domain parameter.

1) The challenger computes $(sk_S, pk_S) \leftarrow_{\mathrm{R}} \mathsf{Gen}_S(dp)$.
2) $\mathcal{A}$ runs with inputs (dp, pk_S) and is given a FSO $\mathcal{O}_{\mathsf{Sc}}(\cdot,\cdot)$, which takes as inputs a valid public receiver key pk and a message m and outputs $C \leftarrow_{\mathrm{R}} \mathsf{Sc}(sk_S, pk, m)$.
3) $\mathcal{A}$ outputs $((sk_R, pk_R), C^*) \leftarrow_{\mathrm{R}} \mathcal{A}^{\mathcal{O}_{\mathsf{Sc}}(\cdot,\cdot)}(dp, pk_S)$. It succeeds if:
   (i) $\perp \neq m \leftarrow \mathsf{Usc}(sk_R, pk_S, C^*)$, and
   (ii) it never received C^* from $\mathcal{O}_{\mathsf{Sc}}(\cdot,\cdot)$ on a query on (pk_R, m).

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

Definition 3 (Multi–User Strong Insider Unforgeability). *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 attackers $\mathcal{A}$ playing Game 2, running in time t, and issuing q_{Sc} queries to the signcryption oracle, $\mathsf{Adv}^{\mathsf{suf}}_{\mathcal{A},\mathcal{SC}}(k) \leqslant \varepsilon$.*

Confidentiality and unforgeability are natural security goals for signcryption schemes. The soundness and unforgeability of non–repudiation evidence attributes are specific to SCNINR schemes.

Game 3 Soundness of non–repudiation

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

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

Definition 4 (Soundness of non–repudiation). *A SCNINR is said to achieve* (t, ε)–computational soundness of non–repudiation *if for all attackers* $\mathcal{A}$ *playing Game 3 and running in time* t, $\mathsf{Adv}^{\mathsf{snr}}_{\mathcal{A},\mathcal{SC}}(k) \leqslant \varepsilon$.

Game 4 Unforgeability of non–repudiation evidence

$\mathcal{A}$ is an attacker against $\mathcal{SC}$, $dp \leftarrow_{\mathrm{R}} \mathsf{Setup}(k)$ 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 (dp, pk_S, pk_R), and outputs $(C^*, m^*, nr^*) \leftarrow_{\mathrm{R}} \mathcal{A}^{\mathcal{O}_{\mathsf{Sc}}(\cdot,\cdot),\mathcal{O}_{\mathsf{Usc}}(\cdot,\cdot),\mathcal{O}_{\mathsf{N}}(\cdot,\cdot)}(dp, pk_S, pk_R)$.
3) $\mathcal{A}$ wins if:
   (i) C^* was generated through the $\mathcal{O}_{\mathsf{Sc}}(\cdot,\cdot)$ oracle on inputs (pk_R, m) for some m,
   (ii) $1 = d \leftarrow \mathsf{PV}(C^*, m^*, nr^*, pk_S, pk_R)$, and
   (iii) nr^* was not generated by the oracle $\mathcal{O}_{\mathsf{N}}(\cdot,\cdot)$ on a query on (pk_S, C^*).

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

Definition 5 (Unforgeability of non–repudiation evidence). *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 adversaries* $\mathcal{A}$ *playing Game 4, 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,* $\mathsf{Adv}^{\mathsf{unr}}_{\mathcal{A},\mathcal{SC}}(k) \leqslant \varepsilon$.

3 The New Construction

We consider the following variant of Chevallier–Mâmes' (CM) signature scheme [12]; $\mathsf{H}_1 : \{0,1\}^* \to \mathcal{G}$, $\mathsf{H}_2 : \{0,1\}^* \to \mathbf{K}$, and $\mathsf{H}_3 : \{0,1\}^* \to [p-1]$ are hash functions, aux denotes some auxiliary information.

A Variant of Chevallier–Mâmes' signature scheme

1 $\underline{\mathsf{Setup}_{\mathsf{Sign}}(k)}$: the setup outputs a description of the group $\mathcal{G}$, a generator G of $\mathcal{G}$, its prime order p, together with descriptions of the hash functions $\mathsf{H}_{i,i=1,2,3}$.

2 $\underline{\mathsf{Gen}(dp)}$: $sk \leftarrow_{\mathrm{R}} [p-1]$; $pk \leftarrow G^{sk}$; return (sk, pk);

3 $\underline{\mathsf{Sign}(sk, m)}$: $x_1, x_2 \leftarrow_{\mathrm{R}} [p-1]$; $X_1 \leftarrow G^{x_1}$; $X_2 \leftarrow G^{x_2}$; $R \leftarrow \mathsf{H}_1(X_1, X_2)$; $V \leftarrow R^{x_1}$;

4 $W \leftarrow R^{sk}$; $h \leftarrow \mathsf{H}_3(m, X_1, X_2, G, R, V, W, pk, \mathsf{aux})$; $\sigma \leftarrow x_1 + h \cdot sk$; return (X_2, W, σ, h);

$\underline{\mathsf{Vrfy}(pk, (X_2, W, \sigma, h), m)}$: $X_1 \leftarrow G^{\sigma} pk^{-h}$; $R \leftarrow \mathsf{H}_1(X_1, X_2)$; $V \leftarrow R^{\sigma} W^{-h}$;
if $h = \mathsf{H}_3(m, X_1, X_2, G, R, V, W, pk, \mathsf{aux})$ **then** return 1; **else** return 0;

As for CM, in the RO model, the signature generation can be efficiently simulated, and the scheme can be shown to be unforgeable under cDH assumption. An interesting property of this scheme is that when it comes to extend it to a SCNINR, in a simulation of a signcrypted text generation, we can generate $X_1, X_2 \leftarrow_{\mathrm{R}} \mathcal{G}$ such that for all $(B, Z_1, Z_2) \in \mathcal{G}^3$, using the trapdoor test technique [11], we can efficiently decide whether $\mathsf{2DH}(X_1, X_2, B) = (Z_1, Z_2)$ or not. Then, if $(B_1, B_2) \in \mathcal{G}^2$ is a receiver public key, and a twin Diffie–Hellman key exchange [11] is performed using (X_1, X_2) and (B_1, B_2), we can use a trapdoor test at both the sender and the receiver. Then, as for the signature scheme's unforgeability, we can show the signcryption scheme to tightly achieve insider security (confidentiality and unforgeability) under the cDH assumption. The scheme is as described hereunder; we omit the subgroup membership tests.

The $\mathcal{SC}_{\mathsf{edl}}$ Scheme

$\underline{\mathsf{Setup}(k)}$: the algorithm defines a group $\mathcal{G} = \langle G \rangle$ of prime order p, together with an encryption scheme $\mathcal{E} = (\mathsf{E}, \mathsf{D}, \mathbf{K}, \mathbf{M}, \mathbf{C})$ and the hash functions $\mathsf{H}_1 : \{0,1\}^* \to \mathcal{G}$, $\mathsf{H}_2 : \{0,1\}^* \to \mathbf{K}$, and $\mathsf{H}_3 : \{0,1\}^* \to [p-1]$. The domain parameter is $dp = (\mathcal{G}, \mathcal{E}, \mathsf{H}_1, \mathsf{H}_2, \mathsf{H}_3)$. We assume $p \geqslant |\mathbf{K}|$.

$\underline{\mathsf{Gen}_S(dp)}$: $a \leftarrow_{\mathrm{R}} [p-1]$; $(sk_S, pk_S) \leftarrow (a, G^a)$; return (sk_S, pk_S);

$\underline{\mathsf{Gen}_R(dp)}$: $b_1, b_2 \leftarrow_{\mathrm{R}} [p-1]$; $(sk_R, pk_R) \leftarrow \big((b_1, b_2), (G^{b_1}, G^{b_2})\big)$; return (sk_R, pk_R);

$\underline{\mathsf{Sc}(sk_S, pk_R, m)}$: Parse pk_R as (B_1, B_2); $x_1, x_2 \leftarrow_{\mathrm{R}} [p-1]$; $X_1 \leftarrow G^{x_1}$; $X_2 \leftarrow G^{x_2}$;
$R \leftarrow \mathsf{H}_1(X_1, X_2)$; $V \leftarrow R^{x_1}$; $W \leftarrow R^{sk_S}$;
$Z_1 \leftarrow B_1^{x_1}$; $Z_2 \leftarrow B_2^{x_1}$; $Z_3 \leftarrow B_1^{x_2}$; $Z_4 \leftarrow B_2^{x_2}$;
$\tau_1 \leftarrow \mathsf{H}_2(X_1, X_2, Z_1, Z_2, Z_3, Z_4, pk_S, pk_R)$; $\tau_2 \leftarrow \mathsf{H}_2(X_2, X_1, Z_3, Z_4, Z_1, Z_2, pk_S, pk_R)$;
$c \leftarrow \mathsf{E}(\tau_2, m)$; $h \leftarrow \mathsf{H}_3(m, \tau_1, c, X_1, X_2, G, R, V, W, pk_S, pk_R)$;
$\sigma \leftarrow x_1 + h \cdot sk_S \mod p$; return (X_2, W, σ, h, c);

$\underline{\mathsf{Usc}(sk_R, pk_S, C)}$: Parse sk_R as $(b_1, b_2) \in [p-1]^2$;
Parse C as $(X_2, W, \sigma, h, c) \in \mathcal{G}^2 \times [p-1]^2 \times \mathbf{C}$.
$X_1 \leftarrow G^{\sigma} pk_S^{-h}$; $Z_1 \leftarrow X_1^{b_1}$; $Z_2 \leftarrow X_1^{b_2}$; $Z_3 \leftarrow X_2^{b_1}$; $Z_4 \leftarrow X_2^{b_2}$;
$\tau_1 \leftarrow \mathsf{H}_2(X_1, X_2, Z_1, Z_2, Z_3, Z_4, pk_S, pk_R)$; $\tau_2 \leftarrow \mathsf{H}_2(X_2, X_1, Z_3, Z_4, Z_1, Z_2, pk_S, pk_R)$;
$m \leftarrow \mathsf{D}(\tau_2, c)$; $R \leftarrow \mathsf{H}_1(X_1, X_2)$; $V \leftarrow R^{\sigma} W^{-h}$;
if $h = \mathsf{H}_3(m, \tau_1, c, X_1, X_2, G, R, V, W, pk_S, pk_R)$ **then** return m; **else** return $\perp$;

$\underline{\mathsf{N}(sk_R, pk_S, C)}$: Parse sk_R as (b_1, b_2); Parse C as (X_2, W, σ, h, c).
$X_1 \leftarrow G^{\sigma} pk_S^{-h}$; $Z_1 \leftarrow X_1^{b_1}$; $Z_2 \leftarrow X_1^{b_2}$; $Z_3 \leftarrow X_2^{b_1}$; $Z_4 \leftarrow X_2^{b_2}$;
$\tau_1 \leftarrow \mathsf{H}_2(X_1, X_2, Z_1, Z_2, Z_3, Z_4, pk_S, pk_R)$; $\tau_2 \leftarrow \mathsf{H}_2(X_2, X_1, Z_3, Z_4, Z_1, Z_2, pk_S, pk_R)$;
$m \leftarrow \mathsf{D}(\tau_2, c)$; $R \leftarrow \mathsf{H}_1(X_1, X_2)$; $V \leftarrow R^{\sigma} W^{-h}$;
if $h = \mathsf{H}_3(m, \tau_1, c, X_1, X_2, G, R, V, W, pk_S, pk_R)$ **then** return (τ_1, τ_2); **else** return $\perp$;

$\underline{\mathsf{PV}(C, m, nr, pk_S, pk_R)}$: Parse C as (X_2, W, σ, h, c) and nr as (τ_1, τ_2);
$m' \leftarrow \mathsf{D}(\tau_2, c)$;
if $m' \neq m$ **then** return 0;
$X_1 \leftarrow G^{\sigma} pk_S^{-h}$; $R \leftarrow \mathsf{H}_1(X_1, X_2)$; $V \leftarrow R^{\sigma} W^{-h}$;
if $h = \mathsf{H}_3(m, \tau_1, c, X_1, X_2, G, R, V, W, pk_S, pk_R)$ **then** return 1; **else** return 0;

For the consistency of $\mathcal{SC}_{\mathsf{edl}}$, one can observe that, as $\sigma = x_1 + h \cdot sk_S$, $G^{\sigma} pk_S^{-h}$ yields X_1; similarly $R^{\sigma} W^{-h}$ yields V. Then, if $C \leftarrow_{\mathrm{R}} \mathsf{Sc}(sk_S, pk_R, m)$ the same Z_i's are computed in the signcryption and unsigncryption algorithms. And, the same values of τ_1 and τ_2 are derived both in $\mathsf{Sc}(sk_S, pk_R, m)$ and $\mathsf{Usc}(sk_R, pk_S, C)$. The remaining part in the definition of Sc (resp. Usc) is essentially a proof (resp. verification) of equality of discrete logarithms (edl) modified to include m, τ_1 and c. Doing so, for all $dp \in \{\mathsf{Setup}(k)\}$, 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 $1 = d \leftarrow \mathsf{PV}(C, m, nr, pk_S, pk_R)$.

4 Security Arguments of the $\mathcal{SC}_{\mathsf{edl}}$ Scheme

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

Theorem 1. *We assume the RO model. If q_X, with $X \in \{\mathsf{H}_2, \mathsf{Usc}, \mathsf{N}\}$, is an upper bound on the number of times $\mathcal{A}$ issues the $\mathcal{O}_X$ oracle in Game 1, the* cDH *problem is $(t(k), \varepsilon_{\mathsf{cDH}}(k))$–hard in $\mathcal{G}$, and the encryption scheme $\mathcal{E}$ is $(t(k), \varepsilon_{\mathsf{ss}}(k))$–semantically secure, then $\mathcal{SC}_{\mathsf{edl}}$ is $(t(k), q_{\mathsf{Usc}}, q_{\mathsf{N}}, \varepsilon(k))$–secure in the SKI–MU insider confidentiality in the FSO/FUO–IND–CCA2 sense, where*

$$\varepsilon(k) \leqslant \varepsilon_{\mathsf{cDH}}(k) + \varepsilon_{\mathsf{ss}}(k) + 4(q_{\mathsf{H}_2} + 2q_{\mathsf{Usc}} + 2q_{\mathsf{N}} + 1)/p + 2q_{\mathsf{H}_3}/|\mathbf{K}|. \qquad (1)$$

Proof. We call the steps (1) and (2), (3) and (4), and (5) and (6) of Game 1 pre–challenge, challenge, and post–challenge stages respectively. The simulator answers $\mathcal{A}$'s queries in all phases as described. The Initialization procedure is executed once at the beginning of the game. When abort is set to 1, the whole simulation fails. If the simulation does not fail, the Finalization procedure is executed once, at the end of the game. For a list L, $\mathsf{Apd}(L, X)$ adds X to L. We omit the subgroup membership tests.

Simulation for the experiments $E_{i,i=0,1}$ in the SKI–MU insider confidentiality game

Input: $dp = (\mathcal{G}, \mathcal{E}, \mathsf{H}_1, \mathsf{H}_2, \mathsf{H}_3) \leftarrow_{\mathrm{R}} \mathsf{Setup}(k)$ where $\mathcal{E} = (\mathsf{E}, \mathsf{D}, \mathbf{K}, \mathbf{M}, \mathbf{C})$; $X_0, Y_0 \leftarrow_{\mathrm{R}} \mathcal{G}$.

100 Initialization: $r_0, s_0 \leftarrow_{\mathrm{R}} [p-1]$; $Y_0' \leftarrow G^{s_0} Y_0^{-r_0}$; $pk_R \leftarrow (Y_0, Y_0')$; $\mathcal{S}_{\mathsf{H}_1} \leftarrow ()$; $\mathcal{S}_{\mathsf{k}} \leftarrow ()$; $\mathcal{S}_{\mathsf{H}_2} \leftarrow ()$; $\mathcal{S}_{\mathsf{H}_3} \leftarrow ()$; $\mathsf{abort} \leftarrow 0$;

Pre–Challenge Phase

101 $\mathcal{O}_{\mathsf{H}_1}(d)$:

102 **if** $\nexists\, R : (d, R) \in \mathcal{S}_{\mathsf{H}_1}$ **then** return R; **else** $R \leftarrow_{\mathrm{R}} \mathcal{G}$; $\mathsf{Apd}(\mathcal{S}_{\mathsf{H}_1}, (d, R))$; return R;

103 $\mathcal{O}_{\mathsf{H}_2}(d)$:

104 **if** $\nexists\, \tau : (d, \tau) \in \mathcal{S}_{\mathsf{H}_2}$ **then** return τ;

105 **else**

106 **if** d has format $(X_1, X_2, Z_1, Z_2, Z_3, Z_4, pk, pk' = pk_R) \in \mathcal{G}^7 \times \mathcal{G}^2$ **then**

107 **if** $\exists \tau : ((X_1, X_2, pk, pk_R), \tau) \in \mathcal{S}_{\mathsf{k}}$ **then**

108 **if** $Z_1^{r_0} Z_2 = X_1^{s_0}$ and $Z_3^{r_0} Z_4 = X_2^{s_0}$ **then** $\mathsf{Apd}(\mathcal{S}_{\mathsf{H}_2}, (d, \tau))$; return τ;

109 $\tau \leftarrow_{\mathrm{R}} \mathbf{K}$; $\mathsf{Apd}(\mathcal{S}_{\mathsf{H}_2}, (d, \tau))$; return τ;

$\mathcal{O}_{\mathsf{H}_3}(d)$:
if $\exists\, h : (d,h) \in \mathcal{S}_{\mathsf{H}_3}$ **then** return h;
else $h \leftarrow_{\mathrm{R}} [p-1]$; $\mathsf{Apd}(\mathcal{S}_{\mathsf{H}_3}, (d,h))$; return h;

$\mathcal{O}_{\mathsf{Usc}}(pk, C)$: $\quad$ $\mathcal{O}_{\mathsf{N}}(pk, C)$:
Parse C as $(X_2, W, \sigma, h, c) \in \mathcal{G}^2 \times [p-1]^2 \times \mathbf{C}$; ▶ *Return $\perp$ if the parsing fails*
$X_1 \leftarrow G^{\sigma} pk^{-h}$;
if $\exists\, Z_1, Z_2, Z_3, Z_4 \in \mathcal{G}$ and $\tau \in \mathbf{K} : ((X_1, X_2, Z_1, Z_2, Z_3, Z_4, pk, pk_R), \tau) \in \mathcal{S}_{\mathsf{H}_2}$ and $Z_1^{r_0} Z_2 = X_1^{s_0}$ and $Z_3^{r_0} Z_4 = X_2^{s_0}$ **then**
  $\tau_1 \leftarrow \tau$; ▶ *$\mathsf{H}_2(X_1, X_2, Z_1, Z_2, Z_3, Z_4, pk, pk_R)$ was issued where $\mathsf{2DH}(Y_0, Y_0', X_1) = (Z_1, Z_2)$ and $\mathsf{2DH}(Y_0, Y_0', X_2) = (Z_3, Z_4)$*
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 the ephemeral keys used in the generation of C' are (X_1, X_2) 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 $\exists\, Z_1, Z_2, Z_3, Z_4 \in \mathcal{G}, \tau \in \mathbf{K} : ((X_2, X_1, Z_3, Z_4, Z_1, Z_2, pk, pk_R), \tau) \in \mathcal{S}_{\mathsf{H}_2}$ and $Z_1^{r_0} Z_2 = X_1^{s_0}$ and $Z_3^{r_0} Z_4 = X_2^{s_0}$ **then**
  $\tau_2 \leftarrow \tau$; ▶ *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 $\tau_2 \leftarrow_{\mathrm{R}} \mathbf{K}$; $\mathsf{Apd}(\mathcal{S}_{\mathsf{k}}, ((X_2, X_1, pk, pk_R), \tau_2))$;
$R \leftarrow \mathcal{O}_{\mathsf{H}_1}(X_1, X_2)$; $V = R^{\sigma} W^{-h}$;
$m \leftarrow \mathsf{D}(\tau_2, c)$; $h' \leftarrow \mathcal{O}_{\mathsf{H}_3}(m, \tau_1, c, X_1, X_2, G, R, V, W, pk_S, pk_R)$;
if $h = h'$ **then** ($\mathcal{O}_{\mathsf{Usc}}$) return m; ($\mathcal{O}_{\mathsf{N}}$) return (τ_1, τ_2); **else** return $\perp$;

Challenge Phase
$(m_0, m_1, pk_S, st) \leftarrow_{\mathrm{R}} \mathcal{A}_1^{\mathcal{O}_{\mathsf{Usc}}, \mathsf{N}, \mathsf{H}_1, \mathsf{H}_2, \mathsf{H}_3}(pk_R)$; ▶ $|m_0| = |m_1|$
$\hat{\beta}, \hat{\sigma}, \hat{h}, \hat{u}_0 \leftarrow [p-1]$; $\hat{X}_1 \leftarrow G^{\hat{\sigma}} pk_S^{-\hat{h}}$; $\hat{X}_2 \leftarrow X_0 G^{\hat{u}_0}$;
if $\exists R' : \left((\hat{X}_1, \hat{X}_2), R'\right) \in \mathcal{S}_{\mathsf{H}_1}$ **then** abort $\leftarrow 1$;
$\hat{R} \leftarrow G^{\hat{\beta}}$; $\mathsf{Apd}(\mathcal{S}_{\mathsf{H}_1}, ((\hat{X}_1, \hat{X}_2), \hat{R}))$; $\hat{W} \leftarrow pk_S^{\hat{\beta}}$; $\hat{V} \leftarrow \hat{R}^{\hat{\sigma}} \hat{W}^{-\hat{h}}$; ▶ $\log_G pk_S = \log_{\hat{R}} \hat{W}$ *$(= sk_S$ which is unknown)*
$\hat{\tau}_1 \leftarrow_{\mathrm{R}} \mathbf{K}$; $\hat{\tau}_2 \leftarrow_{\mathrm{R}} \mathbf{K}$;
(E_0, E_{int}) $\hat{c} \leftarrow \mathsf{E}(\hat{\tau}_2, m_0)$; ($E_1$) $\hat{c} \leftarrow \mathsf{E}(\hat{\tau}_2, m_1)$;
if $\exists\, h', m', \tau_1', R', V', W', pk_S' : ((m', \tau_1', \hat{X}_1, \hat{X}_2, G, R', V', W', pk_S', pk_R), h') \in \mathcal{S}_{\mathsf{H}_3}$ **then** abort $\leftarrow 1$; ▶ *$(\hat{X}_1, \hat{X}_2)$ were already used as ephemeral keys*
$\mathsf{Apd}(\mathcal{S}_{\mathsf{k}}, ((\hat{X}_1, \hat{X}_2, pk_S, pk_R), \hat{\tau}_1))$; $\mathsf{Apd}(\mathcal{S}_{\mathsf{k}}, ((\hat{X}_2, \hat{X}_1, pk_S, pk_R), \hat{\tau}_2))$;
(E_0,) $\mathsf{Apd}(\mathcal{S}_{\mathsf{H}_3}, ((m_0, \hat{\tau}_1, \hat{c}, \hat{X}_1, \hat{X}_2, G, \hat{R}, \hat{V}, \hat{W}, pk_S, pk_R), \hat{h}))$; ($E_1, E_{\mathrm{int}}$) $\mathsf{Apd}(\mathcal{S}_{\mathsf{H}_3}, ((m_1, \hat{\tau}_1, \hat{c}, \hat{X}_1, \hat{X}_2, G, \hat{R}, \hat{V}, \hat{W}, pk_S, pk_R), \hat{h}))$;
$C^* \leftarrow (\hat{X}_2, \hat{W}, \hat{\sigma}, \hat{h}, \hat{c})$;

Post–Challenge Phase
$\mathcal{A}_2$ runs with inputs (C^*, st). It has access to the oracles $\mathcal{O}_{\mathsf{Usc}}(\cdot,\cdot)$, $\mathcal{O}_{\mathsf{N}}(\cdot,\cdot)$, $\mathcal{O}_{\mathsf{H}_1}(\cdot)$, $\mathcal{O}_{\mathsf{H}_2}(\cdot)$ and $\mathcal{O}_{\mathsf{H}_3}(\cdot)$.
$b' \leftarrow_{\mathrm{R}} \mathcal{A}_2^{\mathcal{O}_{\mathsf{Usc}}, \mathsf{N}, \mathsf{H}_1, \mathsf{H}_2, \mathsf{H}_3}(C^*, st)$;

140 Finalization:
141 **if** $\exists\, \hat{Z}_1, \hat{Z}_2, \hat{Z}_3, \hat{Z}_4 \in \mathcal{G} : \big(((\hat{X}_1, \hat{X}_2, \hat{Z}_1, \hat{Z}_2, \hat{Z}_3, \hat{Z}_4, pk_S, pk_R), \hat{\tau}_1) \in \mathcal{S}_{\mathsf{H}_2}$ or
142 $((\hat{X}_2, \hat{X}_1, \hat{Z}_3, \hat{Z}_4, \hat{Z}_1, \hat{Z}_2, pk_S, pk_R), \hat{\tau}_2) \in \mathcal{S}_{\mathsf{H}_2}\big)$ and $\hat{Z}_1^{r_0}\hat{Z}_2 = \hat{X}_1^{s_0}$ and $\hat{Z}_3^{r_0}\hat{Z}_4 = \hat{X}_2^{s_0}$
then return $\hat{Z}_3 Y_0^{-\hat{u}_0}$; **else** return $\perp$;

In the Initialization procedure, the simulator stores $r_0, s_0 \leftarrow_{\mathrm{R}} [p-1]$, and computes Y_0' (see at line 100). Doing so, the receiver public key (Y_0, Y_0'), is such that given $(X_1, X_2, Z_1, Z_2, Z_3, Z_4) \in \mathcal{G}^6$, we can decide, using the trapdoor test, whether $(Z_1, Z_2) = \mathsf{2DH}(Y_0, Y_0', X_1)$ and $(Z_3, Z_4) = \mathsf{2DH}(Y_0, Y_0', X_2)$ or not. We describe at lines 113–128 both the $\mathcal{O}_{\mathsf{Usc}}(\cdot,\cdot)$ and $\mathcal{O}_{\mathsf{N}}(\cdot,\cdot)$ oracles. In the execution of $\mathcal{O}_{\mathsf{Usc}}(\cdot,\cdot)$ (resp. $\mathcal{O}_{\mathsf{N}}(\cdot,\cdot)$), the instruction return (τ_1, τ_2) (resp. return m) at line 128 is omitted. In the challenge phase, depending on whether the simulation is for E_0 or for E_1, the corresponding boxed code is executed (see at lines 134 and 137).

We simulate digest queries using associative lists. The main technical subtlety is that the $\mathcal{O}_{\mathsf{H}_2}(\cdot)$ digest values for strings with format $(X_1, X_2, Z_1, Z_2, Z_3, Z_4, pk, pk_R) \in \mathcal{G}^7 \times \mathcal{G}^2$ are not only assigned by the $\mathcal{O}_{\mathsf{H}_2}(\cdot)$ oracle, but also through the executions of $\mathcal{O}_{\mathsf{Usc}}(\cdot,\cdot)$ and $\mathcal{O}_{\mathsf{N}}(\cdot,\cdot)$. In these latter cases, the values of $Z_{i,i=1,2,3,4}$ are unknown to the simulator. To keep the simulation consistent, besides $\mathcal{S}_{\mathsf{H}_2}$, we use a list $\mathcal{S}_{\mathsf{k}}$ to store the values of $\mathcal{O}_{\mathsf{H}_2}(X_1, X_2, Z_1, Z_2, Z_3, Z_4, pk, pk_R)$, together with (X_1, X_2, pk, pk_R), which was assigned while the $Z_{i,i=1,2,3,4}$ are unknown (see at lines 120 and 125). As a consequence, using the trapdoor test Theorem [11], all the $\mathcal{O}_{\mathsf{H}_2}(\cdot)$ queries can be consistently answered, with all but negligible probability. In the challenge phase, we essentially simulate an encryption and a signature generation in our CM variant, wherein, $\hat{X}_2$ is set to $X_0 G^{\hat{u}_0}$, and some savings are performed for consistency in digests.

We assume, without loss of generality, that for all $\tau \in \mathbf{K}$, whenever $\mathcal{A}$ issues $\mathsf{H}_3(m_0, \tau, \hat{c}, \hat{X}_1, \hat{X}_2, G, \hat{R}, \hat{V}, \hat{W}, pk_S, pk_R)$ it issues also $\mathsf{H}_3(m_1, \tau, \hat{c}, \hat{X}_1, \hat{X}_2, G, \hat{R}, \hat{V}, \hat{W}, pk_S, pk_R)$ and vice versa. Let bad_1 be the event: "the simulator aborts" (see at lines 131 and 135), then, assuming $t \leqslant \sqrt{p}$,

$$\Pr(\mathsf{bad}_1) \leqslant (q_{\mathsf{H}_1} + q_{\mathsf{Usc}} + q_{\mathsf{N}})/p^2 + (q_{\mathsf{H}_3} + q_{\mathsf{Usc}} + q_{\mathsf{N}})/p^2 \leqslant 1/p.$$

Let bad_2 be the event: "in at least one of the executions of $\mathcal{O}_{\mathsf{H}_2}(\cdot)$, $\mathcal{O}_{\mathsf{Usc}}(\cdot)$, or $\mathcal{O}_{\mathsf{N}}(\cdot)$ the $Z_{i,i=1,2,3,4}$'s are such that
a) $Z_1^{r_0} Z_2 = X_1^{s_0}$ and $Z_3^{r_0} Z_4 = X_2^{s_0}$, while
b) $\mathsf{2DH}(Y_0, Y_0', X_1) \neq (Z_1, Z_2)$ or $\mathsf{2DH}(Y_0, Y_0', X_2) \neq (Z_3, Z_4)$" (see at lines 108, 116, and 121), then from the trapdoor test Theorem [11]

$$\Pr(\mathsf{bad}_2) \leqslant 2(q_{\mathsf{H}_2} + 2q_{\mathsf{Usc}} + 2q_{\mathsf{N}})/p.$$

And, if $\mathsf{bad} = \mathsf{bad}_1 \vee \mathsf{bad}_2$, then

$$\Pr(\mathsf{bad}) \leqslant (2q_{\mathsf{H}_2} + 4q_{\mathsf{Usc}} + 4q_{\mathsf{N}} + 1)/p. \tag{2}$$

Let $\mathsf{out}_0^{\mathsf{sim}}$ denote the event: "the conditions 6i and 6ii of Game 1 are satisfied in the simulated environment for E_0". Then, under the RO model, if $\neg\mathsf{bad}$, $\mathcal{A}$'s

views in the real and simulated environments are the same. So, $\Pr(\mathsf{out}_0 \wedge \neg\mathsf{bad}) = \Pr(\mathsf{out}_0^{\mathsf{sim}} \wedge \neg\mathsf{bad})$, and then

$$|\Pr(\mathsf{out}_0) - \Pr(\mathsf{out}_0^{\mathsf{sim}})| \leqslant \Pr(\mathsf{bad}) \tag{3}$$

We consider the intermediate simulated experiment E_{int}, wherein the line 134 is the same as in the simulation E_0, and the line 137 is as in E_1. In short, compared to the simulation E_0, in the computation of $\hat{h}$ in the challenge phase, the challenger commits to m_1 instead of m_0. Let bad' be the event "$\mathcal{A}$ issues $\mathcal{O}_{\mathsf{H}_3}(m_1, \hat{\tau}_1, \hat{c}, \hat{X}_1, \hat{X}_2, G, \hat{R}, \hat{V}, \hat{W}, pk_S, pk_R)$", or equivalently "$\mathcal{A}$ issues $\mathcal{O}_{\mathsf{H}_3}(m_0, \hat{\tau}_1, \hat{c}, \hat{X}_1, \hat{X}_2, G, \hat{R}, \hat{V}, \hat{W}, pk_S, pk_R)$". Under the RO model, assuming that a digest query requires one time unit, as $\hat{\tau}_1$ is chosen uniformly at random from $\mathbf{K}$, $\Pr(\mathsf{bad}') \leqslant 2q_{\mathsf{H}_3}/|\mathbf{K}|$.

If $\mathsf{out}_{\text{int}}^{\mathsf{sim}}$ denotes the event: "the conditions 6i and 6ii of Game 1 are satisfied in the simulated environment for E_{int}." As if $\neg\mathsf{bad}'$ occurs, $\mathcal{A}$'s views in the simulations for E_0 and E_{int} are the same, it holds that $\Pr(\mathsf{out}_0^{\mathsf{sim}} \wedge \neg\mathsf{bad}') = \Pr(\mathsf{out}_{\text{int}}^{\mathsf{sim}} \wedge \neg\mathsf{bad}')$, and then

$$|\Pr(\mathsf{out}_0^{\mathsf{sim}}) - \Pr(\mathsf{out}_{\text{int}}^{\mathsf{sim}})| \leqslant \Pr(\mathsf{bad}') \tag{4}$$

We now consider the simulated environment for the experiment E_1. Notice that to obtain it from the simulation E_{int}, only the line 134 has to be changed, with $\hat{c}$ computed as $\mathsf{E}(\hat{\tau}_2, m_1)$ instead of $\mathsf{E}(\hat{\tau}_2, m_0)$. If $\mathsf{out}_1^{\mathsf{sim}}$ denotes the event: "the conditions 6i and 6ii of Game 1 are satisfied in the simulated environment for E_1" then, using the same arguments as for E_0, it holds that

$$|\Pr(\mathsf{out}_1) - \Pr(\mathsf{out}_1^{\mathsf{sim}})| \leq \Pr(\mathsf{bad}). \tag{5}$$

Let E denote the event "Finalization outputs $\hat{Z}_3 Y_0^{-\hat{u}_0} \neq \perp$". This means that $\mathcal{A}$ issues $\mathcal{O}_{\mathsf{H}_2}(\hat{X}_1, \hat{X}_2, \hat{Z}_1, \hat{Z}_2, \hat{Z}_3, \hat{Z}_4, pk_S, pk_R)$ or $\mathcal{O}_{\mathsf{H}_2}(\hat{X}_2, \hat{X}_1, \hat{Z}_3, \hat{Z}_4, \hat{Z}_1, \hat{Z}_2, pk_S, pk_R)$ such that $\hat{Z}_1^{r_0}\hat{Z}_2 = \hat{X}_1^{s_0}$ and $\hat{Z}_3^{r_0}\hat{Z}_4 = \hat{X}_2^{s_0}$. And, if E occurs, let bad'' be the event: "in the run of the Finalization procedure, the $\hat{Z}_{i,i=1,2,3,4}$'s are such that
a) $\hat{Z}_1^{r_0}\hat{Z}_2 = \hat{X}_1^{s_0}$ and $\hat{Z}_3^{r_0}\hat{Z}_4 = \hat{X}_2^{s_0}$ while
b) $\mathsf{2DH}(Y_0, Y_0', X_1) \neq (Z_1, Z_2)$ or $\mathsf{2DH}(Y_0, Y_0', X_2) \neq (Z_3, Z_4)$", then

$$\Pr(\mathsf{bad}'') \leqslant 2/p. \tag{6}$$

As if $\neg\mathsf{bad}'' \wedge \mathsf{E}$ occurs, the simulator outputs $\mathsf{cDH}(X_0, Y_0)$, it holds that

$$\Pr(\mathsf{out}_1^{\mathsf{sim}} \wedge \neg\mathsf{bad}'' \wedge \mathsf{E}) \leqslant \mathsf{Adv}_{\mathcal{B}_1}^{\mathsf{cDH}}(\mathcal{G}).$$

Similar arguments show that

$$\Pr(\mathsf{out}_{\text{int}}^{\mathsf{sim}} \wedge \neg\mathsf{bad}'' \wedge \mathsf{E}) \leqslant \mathsf{Adv}_{\mathcal{B}_1}^{\mathsf{cDH}}(\mathcal{G}).$$

where $\mathcal{B}_1$ is a cDH adversary obtained from $\mathcal{A}$ and the simulator.
Now, if $\mathsf{out}_1^{\mathsf{sim}} \wedge \neg\mathsf{bad}'' \wedge \neg\mathsf{E}$ occurs, then the adversary never obtains $\hat{\tau}_2$ from the $\mathcal{O}_{\mathsf{H}_2}$ oracle. The difference between the probability of the events $\mathsf{out}_{\text{int}}^{\mathsf{sim}} \wedge \neg\mathsf{bad}'' \wedge \neg\mathsf{E}$ and $\mathsf{out}_1^{\mathsf{sim}} \wedge \neg\mathsf{bad}'' \wedge \neg\mathsf{E}$ induced by this change is essentially a

semantic security advantage. Using $\mathcal{A}$ and the simulator we obtain an adversary $\mathcal{B}_2$ against $\mathcal{E}$ such that

$$|\Pr(\mathsf{out}_{\mathrm{int}}^{\mathsf{sim}} \wedge \neg \mathsf{bad}'' \wedge \neg \mathsf{E}) - \Pr(\mathsf{out}_{1}^{\mathsf{sim}} \wedge \neg \mathsf{bad}'' \wedge \neg \mathsf{E})| \leqslant \mathsf{Adv}^{\mathsf{ss}}_{\mathcal{B}_2,\mathcal{E}}(k), \tag{7}$$

and then

$$|\Pr(\mathsf{out}_{\mathrm{int}}^{\mathsf{sim}}) - \Pr(\mathsf{out}_{1}^{\mathsf{sim}})| \leqslant \varepsilon_{\mathsf{ss}}(k) + \varepsilon_{\mathsf{cDH}}(k) + \Pr(\mathsf{bad}''); \tag{8}$$

As

$$\begin{aligned}|\Pr(\mathsf{out}_0) - \Pr(\mathsf{out}_1)| \leqslant\ & |\Pr(\mathsf{out}_0) - \Pr(\mathsf{out}_0^{\mathsf{sim}})| + |\Pr(\mathsf{out}_0^{\mathsf{sim}}) - \Pr(\mathsf{out}_{\mathrm{int}}^{\mathsf{sim}})| \\ & + |\Pr(\mathsf{out}_{\mathrm{int}}^{\mathsf{sim}}) - \Pr(\mathsf{out}_1^{\mathsf{sim}})| + |\Pr(\mathsf{out}_1^{\mathsf{sim}}) - \Pr(\mathsf{out}_1)|,\end{aligned} \tag{9}$$

The result follows from the inequalities (2) to (9).

4.2 Unforgeability of the $\mathcal{SC}_{\mathsf{edl}}$ Signcryption Scheme

Theorem 2. *Let q_X, where $X \in \{\mathsf{H}_1, \mathsf{H}_2, \mathsf{H}_3, \mathsf{Sc}\}$, be an upper bound on the number of times $\mathcal{A}$ issues the $\mathcal{O}_X$ oracle in Game 2. Under the RO model, if the* cDH *problem is $(t(k), \varepsilon_{\mathsf{cDH}}(k))$–hard in $\mathcal{G}$, then $\mathcal{SC}_{\mathsf{edl}}$ is $(t(k), q_{\mathsf{Sc}}(k), \varepsilon(k))$–MU insider unforgeable in the FSO/FUO–sUF–CMA sense, where*

$$\varepsilon \leqslant \varepsilon_{\mathsf{cDH}} + ((q_{\mathsf{Sc}} + q_{\mathsf{H}_3})^2 + q_{\mathsf{Sc}}^2)/2p + (q_{\mathsf{H}_3} + 2q_{\mathsf{H}_2} + 1)/p.$$

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

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

Input: $dp = (\mathcal{G}, \mathcal{E}, \mathsf{H}_1, \mathsf{H}_2, \mathsf{H}_3) \leftarrow_{\mathrm{R}} \mathsf{Setup}(k)$ where $\mathcal{E} = (\mathsf{E}, \mathsf{D}, \mathbf{K}, \mathbf{M}, \mathbf{C})$; $X_0, Y_0 \leftarrow_{\mathrm{R}} \mathcal{G}$;

200 Initialization: $pk_S \leftarrow Y_0$; $\mathcal{S}_{\mathsf{H}_1} \leftarrow ()$; $\mathcal{S}_{\mathsf{k\&r}} \leftarrow ()$; $\mathcal{S}_{\mathsf{H}_2} \leftarrow ()$; $\mathcal{S}_{\mathsf{H}_3} \leftarrow ()$; abort $\leftarrow 0$; $\mathcal{S}_{\mathsf{Sc}} \leftarrow ()$;

201 $\mathcal{O}_{\mathsf{H}_1}(s)$:

202 **if** $\exists\, R, d : (s, R, \text{“1”}, d) \in \mathcal{S}_{\mathsf{H}_1}$ or $(s, R, \text{“2”}, \perp) \in \mathcal{S}_{\mathsf{H}_1}$ **then** return R; ▶ *We use the indicators “1” and “2” to differentiate the H_1 digest values assigned through the $\mathcal{O}_{\mathsf{H}_1}(\cdot)$ oracle from the ones assigned through the $\mathcal{O}_{\mathsf{Sc}}(\cdot)$ oracle.*

203 **else** $d \leftarrow_{\mathrm{R}} [p-1]$; $R \leftarrow X_0 G^d$; $\mathsf{Apd}(\mathcal{S}_{\mathsf{H}_1}, (s, R, \text{“1”}, d))$; return R;

204 $\mathcal{O}_{\mathsf{H}_2}(s)$:

205 **if** $\exists\, \tau : (s, \tau) \in \mathcal{S}_{\mathsf{H}_2}$ **then** return τ;

206 **else if** s has format $(X_1, X_2, Z_1, Z_2, Z_3, Z_4, pk_S, pk) \in \mathcal{G}^7 \times \mathcal{G}^2$ **then**

207 Parse pk as $(B_1, B_2) \in \mathcal{G}^2$;

208 **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**

209 **if** $Z_1^r Z_3 = B_1^s$ and $Z_2^r Z_4 = B_2^s$ **then** return τ_1; ▶ $\mathsf{2DH}(X_1, X_2, B_1) = (Z_1, Z_3)$ *and* $\mathsf{2DH}(X_1, X_2, B_2) = (Z_2, Z_4)$ *with all but negligible probability.*

210 **else 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**

211 **if** $Z_3^r Z_1 = B_1^s$ and $Z_4^r Z_2 = B_2^s$ **then** return τ_2;

212 **else** $\tau \leftarrow_{\mathrm{R}} \mathbf{K}$; $\mathsf{Apd}(\mathcal{S}_{\mathsf{H}_2}, (s, \tau))$; return τ;

213 **else** $\tau \leftarrow_{\mathrm{R}} \mathbf{K}$; $\mathsf{Apd}(\mathcal{S}_{\mathsf{H}_2}, (s, \tau))$; return τ;

214 $\mathcal{O}_{\mathsf{H}_3}(d)$:
215 **if** $\exists\, h : (d,h) \in \mathcal{S}_{\mathsf{H}_3}$ **then** return h; **else** $h \leftarrow_{\mathrm{R}} [p-1]$; $\mathsf{Apd}(\mathcal{S}_{\mathsf{H}_3}, (d,h))$; return h;
216 $\mathcal{O}_{\mathsf{Sc}}(pk, m)$:
217 $\beta, \sigma, h, s, r \leftarrow_{\mathrm{R}} [p-1]$; $X_1 \leftarrow G^{\sigma} pk_S^{-h}$; $X_2 \leftarrow G^{s} X_1^{-r}$;
218 **if** $\exists\, R', i, j : \left((X_1, X_2), R', i, j\right) \in \mathcal{S}_{\mathsf{H}_1}$ **then** abort $\leftarrow 1$; ▶ *A digest on (X_1, X_2) was issued or (X_1, X_2) were previously used as outgoing ephemeral keys*
219 $R \leftarrow G^{\beta}$; $W \leftarrow pk_S^{\beta}$; $V \leftarrow R^{\sigma} W^{-h}$; ▶ *$\log_G pk_S = \log_R W (= sk_S$, which is unknown)*
220 $\mathsf{Apd}\,(\mathcal{S}_{\mathsf{H}_1}, ((X_1, X_2), R, \text{“2”}, \perp))$; ▶ *We define $\mathcal{O}_{\mathsf{H}_1}(X_1, X_2)$ to be R*
221 **if** $\exists\, h', m', c', \tau_1', R', V', W', pk' \;:\; ((m', \tau_1', c', X_1, X_2, G, R', V', W', pk_S, pk'), h') \in \mathcal{S}_{\mathsf{H}_3}$ **then** abort $\leftarrow 1$; ▶ *(X_1, X_2) were already used as ephemeral keys*
222 $\tau_1 \leftarrow_{\mathrm{R}} \mathbf{K}$; $\tau_2 \leftarrow_{\mathrm{R}} \mathbf{K}$; $c \leftarrow \mathsf{E}(\tau_2, m)$; $\mathsf{Apd}\,(\mathcal{S}_{\mathsf{k\&r}}, ((X_1, X_2, pk_S, pk), (r, s, \tau_1, \tau_2)))$;
223 $\mathsf{Apd}(\mathcal{S}_{\mathsf{H}_3}, ((m, \tau_1, c, X_1, X_2, G, R, V, W, pk_S, pk), h))$; $C \leftarrow (X_2, W, \sigma, h, c)$;
224 $\mathsf{Apd}(\mathcal{S}_{\mathsf{Sc}}, ((m, \tau_1, X_1, X_2, G, R, V, W, pk_S, pk), C))$;
225 return C;
226 Finalization:
227 **if** $\mathcal{A}$ outputs (sk_R, pk_R, C^*) such that
(i) $\perp \neq \hat{m} \leftarrow \mathsf{Usc}(sk_R, pk_S, C^*)$ and
(ii) and $\mathcal{A}$ never received C^* from $\mathcal{O}_{\mathsf{Sign}}(\cdot,\cdot)$ on a query on $(pk_R, \hat{m})$
then
228 Parse C^* as $(\hat{X}_2, \hat{W}, \hat{\sigma}, \hat{h}, \hat{c})$;
229 $\hat{X}_1 \leftarrow G^{\hat{\sigma}} pk_S^{-\hat{h}}$; $\hat{R} \leftarrow \mathcal{O}_{\mathsf{H}_1}(\hat{X}_1, \hat{X}_2)$;
230 **if** $\exists \hat{d} : ((\hat{X}_1, \hat{X}_2), \hat{R}, \text{“1”}, \hat{d}) \in \mathcal{S}_{\mathsf{H}_1}$, for some $\hat{d}$ **then** ▶ *the value of $\mathcal{O}_{\mathsf{H}_1}(\hat{X}_1, \hat{X}_2)$ was assigned through the $\mathcal{O}_{\mathsf{H}_1}$ oracle*
231 $Z_0 \leftarrow \hat{W} pk_S^{-\hat{d}}$; return Z_0;
232 **else**
233 **if** $\exists (X_1', X_2') \neq (\hat{X}_1, \hat{X}_2) : (X_1', X_2', \hat{R}, \text{“2”}, \perp) \in \mathcal{S}_{\mathsf{H}_1}$ **then** abort $\leftarrow 1$;
234 **else** ▶ *We have necessarily $(\hat{X}_1, \hat{X}_2, \hat{R}, \text{“2”}, \perp) \in \mathcal{S}_{\mathsf{H}_1}$*
235 Lookup $((m', \tau_1', \hat{X}_1, \hat{X}_2, G, \hat{R}, V', W', pk_S, pk'), C') \in \mathcal{S}_{\mathsf{Sc}}$ such that $(C', pk', m') \neq (C^*, pk_R, \hat{m})$; ▶ *Such an element of $\mathcal{S}_{\mathsf{Sc}}$ can be found, as $\mathcal{A}$ never received C^* on $\mathcal{O}_{\mathsf{Sc}}(pk_R, \hat{m})$ query;*
236 Parse C' as $(\hat{X}_2, W', \sigma', h', c')$;
237 **if** $\hat{h} = h'$ **then** abort $\leftarrow 1$;
238 **else** ▶ *$\hat{h} \neq h'$ and $\hat{X}_1 = G^{\hat{\sigma}} pk_S^{-\hat{h}} = G^{\sigma'} pk_S^{-h'}$*
239 $y_0 \leftarrow (\hat{\sigma} - \sigma')(\hat{h} - h')^{-1} \mod p$; $Z_0 \leftarrow X_0^{y_0}$; return Z_0;
240 return $\perp$;

The main trick in this simulation remains the use of the trapdoor test technique [11] for consistency in the digest values.
Let bad_{1a} denote the event: “the simulator aborts before the execution of the Finalization procedure” (see at lines 218 and 221), then

$$\Pr(\mathsf{bad}_{1a}) \;\leqslant\; (q_{\mathsf{Sc}} + q_{\mathsf{H}_1})^2/2p^2 + (q_{\mathsf{Sc}} + q_{\mathsf{H}_3})^2/2p^2 \;\leqslant\; 1/p.$$

If bad_{1b} is the event: “in an execution of the $\mathcal{O}_{\mathsf{H}_2}(\cdot)$ procedure, a tuple $(X_1, X_2, Z_1, Z_2, Z_3, Z_4, r, s)$ is such that
A) a) $Z_1^r Z_3 = B_1^s$ and $Z_2^r Z_4 = B_2^s$, while
b) $\mathsf{2DH}(X_1, X_2, B_1) \neq (Z_1, Z_3)$ or $\mathsf{2DH}(X_1, X_2, B_2) \neq (Z_2, Z_4)$,

or

B) a') $Z_3^r Z_1 = B_1^s$ and $Z_4^r Z_2 = B_2^s$, while
b') $\mathsf{2DH}(X_2, X_1, B_1) \neq (Z_3, Z_1)$ or $\mathsf{2DH}(X_2, X_1, B_2) \neq (Z_4, Z_2)$."

Then, still using the trapdoor test Theorem [11], we have $\Pr(\mathsf{bad}_{1b}) \leqslant 2q_{\mathsf{H}_2}/p$.

Let E be the event: "the conditions (i) and (ii) in the Finalization procedure are satisfied" and $\mathsf{bad}_1 = \mathsf{bad}_{1a} \vee \mathsf{bad}_{1b}$. Then, under the RO model,

$$\begin{aligned}\mathsf{Adv}^{\mathsf{suf}}_{\mathcal{A},\mathcal{SC}}(k) &= \Pr(\mathsf{Succ}^{\mathsf{suf}}_{\mathcal{A}} \wedge \neg\mathsf{bad}_1) + \Pr(\mathsf{Succ}^{\mathsf{suf}}_{\mathcal{A}} \wedge \mathsf{bad}_1)\\ &\leqslant \Pr(\mathsf{E}) + (2q_{\mathsf{H}_2} + 1)/p. \end{aligned} \tag{10}$$

Let E_1 be the event: "$\mathsf{E} \wedge \exists \hat{d} : ((\hat{X}_1, \hat{X}_2), \hat{R}, \text{"1"}, \hat{d}) \in \mathcal{S}_{\mathsf{H}_1}$" (see at line 230), and E_2 be the event: "$\mathsf{E} \wedge \nexists \hat{d} : ((\hat{X}_1, \hat{X}_2), \hat{R}, \text{"1"}, \hat{d}) \in \mathcal{S}_{\mathsf{H}_1}$." Notice that as $\mathsf{E} = \mathsf{E}_1 \vee \mathsf{E}_2$ and the union is disjoint, $\Pr(\mathsf{E}) = \Pr(\mathsf{E}_1) + \Pr(\mathsf{E}_2)$. Now, if E_1 occurs, let bad_2 be the event: "$\hat{W} \neq \hat{R}^{sk_S}$". If $\mathsf{E}_1 \wedge \mathsf{bad}_2$ occurs, let $\hat{x}_1 = \log_G \hat{X}_1$, $x'_1 = \log_{\hat{R}} \hat{V}$, and $sk' = \log_{\hat{R}} \hat{W}$. As $\perp \neq \hat{m} \leftarrow \mathsf{Usc}(sk_R, pk_S, C^*)$, it holds that

$$\hat{x}_1 = \hat{\sigma} - \hat{h} \cdot sk_S \text{ and } x'_1 = \hat{\sigma} - \hat{h} \cdot sk'.$$

As $\hat{W} \neq \hat{R}^{sk_S}$, we have $sk_S \neq sk'$. It follows that

$$\hat{h} = (\hat{x}_1 - x'_1)(sk' - sk_S)^{-1} \mod p.$$

Hence, under the RO model, $\Pr(\mathsf{E}_1 \wedge \mathsf{bad}_2) \leqslant q_{\mathsf{H}_3}/p$. Then, if E_1, except with probability $\leqslant q_{\mathsf{H}_3}/p$, it holds that $\hat{W} = \hat{R}^{sk_S} = (X_0 G^{\hat{d}})^{sk_S}$; and $\hat{W} pk_S^{-\hat{d}} = \hat{W} Y_0^{-\hat{d}}$ yields $\mathsf{cDH}(X_0, Y_0)$.

If E_2 occurs, then $\mathcal{O}_{\mathsf{H}_1}(\hat{X}_1, \hat{X}_2)$ was assigned through the $\mathcal{O}_{\mathsf{Sc}}(\cdot,\cdot)$ oracle. Hence, we necessarily have $(\hat{X}_1, \hat{X}_2, \hat{R}, \text{"2"}, \perp) \in \mathcal{S}_{\mathsf{H}_1}$. Let bad_3 be the event "$\exists (X_1, X_2) \neq (\hat{X}_1, \hat{X}_2) : (X_1, X_2, \hat{R}, \text{"2"}, \perp) \in \mathcal{S}_{\mathsf{H}_1}$" (a collision occurred among the $\mathcal{O}_{\mathsf{H}_1}$ values assigned through the $\mathcal{O}_{\mathsf{Sc}}$ oracle, see at line 233), then

$$\Pr(\mathsf{E}_2 \wedge \mathsf{bad}_3) \leqslant q^2_{\mathsf{Sc}}/2p,$$

and then

$$\Pr(\mathsf{E}_2) \leqslant \Pr(\mathsf{E}_2 \wedge \neg\mathsf{bad}_3) + q^2_{\mathsf{Sc}}/2p. \tag{11}$$

Now, if $\mathsf{E}_2 \wedge \neg\mathsf{bad}_3$, as $\mathcal{O}_{\mathsf{H}_1}(\hat{X}_1, \hat{X}_2)$ was assigned through $\mathcal{O}_{\mathsf{Sc}}(\cdot,\cdot)$ there exists $((m', \tau'_1, X'_1 = \hat{X}_1, X'_2 = \hat{X}_2, G, \hat{R}, V', W', pk_S, pk'), C') \in \mathcal{S}_{\mathsf{Sc}}$ such that

- $C' = (\hat{X}_2, W', \sigma', h', c')$ is a valid signcrypted text with regard to pk_S and $pk' \in \mathcal{G}^2$ (a receiver public key), and
- $(C', pk', m') \neq (C^*, pk_R, \hat{m})$ (otherwise, the condition (ii) in the Finalization procedure is not satisfied – $\mathcal{A}$ received C^* from $\mathcal{O}_{\mathsf{Sc}}(\cdot,\cdot)$ on query $(pk_R, \hat{m})$).

Let bad_4 be the event: "$h' = \hat{h}$". It is clear that if bad_4 and $pk' \neq pk_R$, then a $\mathcal{O}_{\mathsf{H}_3}$ collision occurred. And, if bad_4 and $pk' = pk_R$ then:

(a) if $\hat{W} = R^{sk_S} = W'$ then as $\hat{R} = R'$, we necessarily have $\hat{\sigma} = \sigma'$. Now, as $(C', pk', m') \neq (C^*, pk_R, \hat{m})$, we have $(m', c') \neq (\hat{m}, \hat{c})$, and then a $\mathcal{O}_{\mathsf{H}_3}$ collision occurred.

(b) And, if $\hat{W} \neq W'$ a $\mathcal{O}_{\mathsf{H}_3}$ collision occurred also.

Then

$$\Pr(\mathsf{E}_2 \wedge \neg\mathsf{bad}_3 \wedge \mathsf{bad}_4) \leqslant (q_{\mathsf{Sc}} + q_{\mathsf{H}_3})^2/2p, \tag{12}$$

and then

$$\Pr(\mathsf{E}_2) \leqslant \Pr(\mathsf{E}_2 \wedge \neg\mathsf{bad}_3 \wedge \neg\mathsf{bad}_4) + ((q_{\mathsf{Sc}} + q_{\mathsf{H}_3})^2 + q_{\mathsf{Sc}}^2)/2p. \tag{13}$$

If $\mathsf{E}_2 \wedge \neg\mathsf{bad}_3 \wedge \neg\mathsf{bad}_4$, then as both C^* and C' are valid signcrypted texts with sender public key pk_S,

$$\hat{x}_1 = \hat{\sigma} - \hat{h} \cdot sk_S \text{ and } \hat{x}_1 = \sigma' - h' \cdot sk_S, \text{ wherein } \hat{x}_1 = \log_G \hat{X}_1.$$

Then if E_2, except with probability $\leqslant ((q_{\mathsf{Sc}} + q_{\mathsf{H}_3})^2 + q_{\mathsf{Sc}}^2)/2p$, the value $(\hat{\sigma} - \sigma')(\hat{h} - h')^{-1} \mod p$ yields $sk_S = y_0 = \log_G Y_0$ and then $X_0^{y_0} = \mathsf{cDH}(X_0, Y_0)$. Then it follows from the inequalities (10) to (13) that

$$\mathsf{Adv}^{\mathsf{suf}}_{\mathcal{A},\mathcal{SC}}(k) \leqslant \mathsf{Adv}^{\mathsf{cDH}}_{\mathcal{B}}(\mathcal{G}) + ((q_{\mathsf{Sc}} + q_{\mathsf{H}_3})^2 + q_{\mathsf{Sc}}^2)/2p + (q_{\mathsf{H}_3} + 2q_{\mathsf{H}_2} + 1)/p,$$

where $\mathcal{B}$ is a cDH solver obtained from $\mathcal{A}$ and the simulator. □

4.3 Soundness of Non–Repudiation

Theorem 3. *Under the RO model, the $\mathcal{SC}_{\mathsf{edl}}$ scheme achieves (t, ε)–computational soundness of non–repudiation, where $\varepsilon \leqslant q_{\mathsf{H}_3}^2/2p$ wherein q_{H_3}, is an upper bound on the number of times $\mathcal{A}$ issues queries to the $\mathcal{O}_{\mathsf{H}_3}$ oracle.*

Proof. We simulate the digest queries as hereunder.

Simulation for Soundness of non–repudiation

Input: $dp = (\mathcal{G}, \mathcal{E}, \mathsf{H}_1, \mathsf{H}_2, \mathsf{H}_3)$ where $\mathcal{E} = (\mathsf{E}, \mathsf{D}, \mathbf{K}, \mathbf{M}, \mathbf{C})$;

300 Initialization: $\mathcal{S}_{\mathsf{H}_1} \leftarrow ()$; $\mathcal{S}_{\mathsf{k\&r}} \leftarrow ()$; $\mathcal{S}_{\mathsf{H}_2} \leftarrow ()$; $\mathcal{S}_{\mathsf{H}_3} \leftarrow ()$; $\mathsf{abort} \leftarrow 0$; $\mathcal{S}_{\mathsf{Sc}} \leftarrow ()$;

301 $\mathcal{O}_{\mathsf{H}_1}(d)$: is defined as described at lines 101–102.

302 $\mathcal{O}_{\mathsf{H}_2}(d)$:

303 **if** $\exists\, R : (d, R) \in \mathcal{S}_{\mathsf{H}_2}$ **then** return R; **else** $R \leftarrow_{\mathrm{R}} \mathcal{G}$; $\mathsf{Apd}(\mathcal{S}_{\mathsf{H}_2}, (d, R))$; return R;

304 $\mathcal{O}_{\mathsf{H}_3}(d)$: is defined as described at lines 110–112.

305 The attacker $\mathcal{A}$ outputs $(C^*, pk_S, sk_R, pk_R, m', nr) \leftarrow_{\mathrm{R}} \mathcal{A}^{\mathcal{O}_{\mathsf{H}_1}(\cdot), \mathcal{O}_{\mathsf{H}_2}(\cdot), \mathcal{O}_{\mathsf{H}_3}(\cdot)}(dp)$;

306 Finalization:

307 **if** $\mathcal{A}$ outputs $(C^*, pk_S, sk_R, pk_R, m', nr)$ such that (i) $\perp \neq m \leftarrow \mathsf{Usc}(sk_R, pk_S, C^*)$, and (ii) $m \neq m'$ and $1 = d \leftarrow \mathsf{PV}(C^*, m', nr, pk_S, pk_R)$, **then**

308 Parse C^* as $(\hat{X}_2, \hat{W}, \hat{\sigma}, \hat{h}, \hat{c})$ and nr as (τ_1, τ_2);

309 $\hat{X}_1 \leftarrow G^{\hat{\sigma}} pk_S^{-\hat{h}}$; $\hat{R} \leftarrow \mathcal{O}_{\mathsf{H}_1}(\hat{X}_1, \hat{X}_2)$; $\hat{V} \leftarrow \hat{R}^{\hat{\sigma}} \hat{W}^{-\hat{h}}$;

310 $\hat{nr} \leftarrow \mathsf{N}(sk_R, pk_S, C)$; parse $\hat{nr}$ as $(\hat{\tau}_1, \hat{\tau}_2)$;

311 $s_1 \leftarrow (m, \hat{\tau}_1, \hat{c}, \hat{X}_1, \hat{X}_2, G, \hat{R}, \hat{V}, \hat{W}, pk_S, pk_R)$; $s_2 \leftarrow (m', \hat{\tau}_1, \hat{c}, \hat{X}_1, \hat{X}_2, G, \hat{R}, \hat{V}, \hat{W}, pk_S, pk_R)$;

312 return (s_1, s_2);

313 **else** return $\perp$;

If $\mathcal{A}$ outputs $(sk_R, pk_R, C^*, m', nr)$ is such that $m' \neq m \leftarrow \mathsf{Usc}(sk_R, pk_S, C^*)$ and $1 = d \leftarrow \mathsf{PV}(C^*, m', nr, pk_S, pk_R)$. As $m \neq m'$, the Finalization procedure outputs (s_1, s_2) such that $s_1 \neq s_1$ and $\mathcal{O}_{\mathsf{H}_3}(s_1) = \mathcal{O}_{\mathsf{H}_3}(s_2)$. Hence,

$$\Pr(\mathsf{Succ}^{\mathsf{snr}}_{\mathcal{A}} \wedge \neg\mathsf{bad}) \leqslant q_{\mathsf{H}_3}^2/2p. \tag{14}$$

4.4 Unforgeability of Non–Repudiation Evidence

Theorem 4. *Under the RO model, if the* cDH *problem is* $(t(k), \varepsilon_{\mathsf{cDH}}(k))$ *hard, then* $\mathcal{SC}_{\mathsf{edl}}$ *achieves* $(t, q_{\mathsf{Sc}}, q_{\mathsf{Usc}}, q_{\mathsf{N}}, \varepsilon)$ *unforgeability of non–repudiation evidence wherein* $\varepsilon \leqslant \varepsilon_{\mathsf{cDH}} + 1/|\mathbf{K}| + 3/(2p)$.

Proof. We define the simulator as indicated hereunder.

Simulation for Unforgeability of non–repudiation evidence

Input: $dp = (\mathcal{G}, \mathcal{E}, \mathsf{H}_1, \mathsf{H}_2, \mathsf{H}_3) \leftarrow_{\mathrm{R}} \mathsf{Setup}(k)$ wherein $\mathcal{E} = (\mathsf{E}, \mathsf{D}, \mathbf{K}, \mathbf{M}, \mathbf{C})$; $X_0, Y_0 \leftarrow_{\mathrm{R}} \mathcal{G}$;

400 Initialization: $a \leftarrow [p-1]$; $(sk_S, pk_S) \leftarrow (a, G^a)$; $r_0, s_0 \leftarrow_{\mathrm{R}} [p-1]$; $Y_0' \leftarrow G^{s_0} Y_0^{-r_0}$; $pk_R \leftarrow (Y_0, Y_0')$; $\mathcal{S}_{\mathsf{H}_1} \leftarrow ()$; $\mathcal{S}_{\mathsf{k}} \leftarrow ()$; $\mathcal{S}_{\mathsf{k\&r}} \leftarrow ()$; $\mathcal{S}_{\mathsf{H}_2} \leftarrow ()$; $\mathcal{S}_{\mathsf{H}_3} \leftarrow ()$; abort $\leftarrow 0$;

401 $\mathcal{O}_{\mathsf{H}_1}(s)$: is defined as at lines 101–102.

402 $\mathcal{O}_{\mathsf{H}_2}(s)$:

403 **if** $\exists\, \tau : (s, \tau) \in \mathcal{S}_{\mathsf{H}_2}$ **then** return τ;

404 **else if** s has format $(X_1, X_2, Z_1, Z_2, Z_3, Z_4, pk, pk' = pk_R) \in \mathcal{G}^7 \times G^2$ **then**

405 **if** $pk = pk_S$ and $\exists \tau, x : ((X_1, X_2, Z_1, Z_2, \epsilon, \epsilon, pk_S, pk_R), \tau, x)) \in \mathcal{S}_{\mathsf{k\&r}}$ and $Z_3^{r_0} Z_4 = X_2^{s_0}$ **then** $\mathsf{Apd}(\mathcal{S}_{\mathsf{H}_2}, (s, \tau))$; return τ; ▶ *ϵ denotes the empty string.*

406 **if** $pk = pk_S$ and $\exists \tau, x : ((X_1, X_2, \epsilon, \epsilon, Z_3, Z_4, pk_S, pk_R), \tau, x)) \in \mathcal{S}_{\mathsf{k\&r}}$ and $Z_1^{r_0} Z_2 = X_1^{s_0}$ **then** $\mathsf{Apd}(\mathcal{S}_{\mathsf{H}_2}, (s, \tau))$; return τ;

407 **if** $\exists\, \tau : ((X_1, X_2, pk, pk_R), \tau) \in \mathcal{S}_{\mathsf{k}}$ and $Z_1^{r_0} Z_2 = X_1^{s_0}$ and $Z_3^{r_0} Z_4 = X_2^{s_0}$ **then** $\mathsf{Apd}(\mathcal{S}_{\mathsf{H}_2}, (s, \tau))$; return τ;

408 **else** $\tau \leftarrow_{\mathrm{R}} \mathbf{K}$; $\mathsf{Apd}(\mathcal{S}_{\mathsf{H}_2}, (s, \tau))$; return τ;

409 **else** $\tau \leftarrow_{\mathrm{R}} \mathbf{K}$; $\mathsf{Apd}(\mathcal{S}_{\mathsf{H}_2}, (s, \tau))$; return τ;

410 $\mathcal{O}_{\mathsf{H}_3}(d)$: is defined as at lines 110–112.

411 $\mathcal{O}_{\mathsf{Sc}}(pk, m)$:

412 Parse pk as (B_1, B_2); $x_1, x_2 \leftarrow_{\mathrm{R}} [p-1]$; $X_1 \leftarrow G^{x_1}$; $Z_1 = B_1^{x_1}$; $Z_2 = B_2^{x_1}$;

413 **if** $pk \neq pk_R$ **then**

414 $X_2 \leftarrow G^{x_2}$; $Z_3 = B_1^{x_2}$; $Z_4 = B_2^{x_2}$;

415 **if** $\exists R : ((X_1, X_2), R) \in \mathcal{S}_{\mathsf{H}_1}$ **then** abort $\leftarrow 1$; ▶ *(X_1, X_2) were already used as ephemeral keys.*

416 $\tau_1 \leftarrow \mathcal{O}_{\mathsf{H}_2}(X_1, X_2, Z_1, Z_2, Z_3, Z_4, pk_S, pk)$; $\tau_2 \leftarrow \mathcal{O}_{\mathsf{H}_2}(X_2, X_1, Z_3, Z_4, Z_1, Z_2, pk_S, pk)$;

417 **else** ▶ *$pk = pk_R$;*

418 $X_2 \leftarrow X_0 G^{x_2}$; ▶ *The simulator takes X_0, Y_0 as inputs; we "embed" X_0 in X_2.*

419 **if** $\exists R : ((X_1, X_2), R) \in \mathcal{S}_{\mathsf{H}_1}$ **then** abort $\leftarrow 1$; ▶ *(X_1, X_2) were already used as ephemeral keys.*

420 $\tau_1 \leftarrow_{\mathrm{R}} \mathbf{K}$; $\tau_2 \leftarrow_{\mathrm{R}} \mathbf{K}$;

421 $\mathsf{Apd}(\mathcal{S}_{\mathsf{k\&r}}, ((X_1, X_2, Z_1, Z_2, \epsilon, \epsilon, pk_S, pk_R), \tau_1, x_2))$;

422 $\mathsf{Apd}(\mathcal{S}_{\mathsf{k\&r}}, ((X_2, X_1, \epsilon, \epsilon, Z_1, Z_2, pk_S, pk_R), \tau_2, x_2))$;

423 $R \leftarrow \mathcal{O}_{\mathsf{H}_1}(X_1, X_2)$; $V \leftarrow R^{x_1}$; $W \leftarrow R^{sk_S}$;

424 $c \leftarrow \mathsf{E}(\tau_2, m)$; $h \leftarrow \mathcal{O}_{\mathsf{H}_3}(m, \tau_1, c, X_1, X_2, G, R, V, W, pk_S, pk)$;

425 $\sigma \leftarrow x_1 + h \cdot sk_S$; return (X_2, W, σ, h, c);

426 $\mathcal{O}_{\mathsf{Usc}}(pk, C)$: $\mathcal{O}_{\mathsf{N}}(pk, C)$:

427 Parse C as $(X_2, W, \sigma, h, c) \in \mathcal{G}^2 \times [p-1]^2 \times \mathbf{C}$;

428 $X_1 \leftarrow G^{\sigma} pk^{-h}$;

429 **if** $\exists\, Z_1, Z_2, Z_3, Z_4 \in \mathcal{G}, \tau \in \mathbf{K} : ((X_1, X_2, Z_1, Z_2, Z_3, Z_4, pk, pk_R), \tau) \in \mathcal{S}_{\mathsf{H}_2}$ and $Z_1^{r_0} Z_2 = X_1^{s_0}$ and $Z_3^{r_0} Z_4 = X_2^{s_0}$ **then**

430 $\tau_1 \leftarrow \tau$; ▶ $\mathsf{H}_2(X_1, X_2, Z_1, Z_2, Z_3, Z_4, pk, pk_R)$ *was issued*

431 **else if** $pk = pk_S$ and $\exists \tau, x, Z_3, Z_4 : ((X_1, X_2, \epsilon, \epsilon, Z_3, Z_4, pk_S, pk_R), \tau, x)) \in \mathcal{S}_{\mathsf{k\&r}}$ **then** $\tau_1 \leftarrow \tau$;

432 **else if** $\exists \tau : ((X_1, X_2, pk, pk_R), \tau) \in \mathcal{S}_{\mathsf{k}}$ **then**

433 $\tau_1 \leftarrow \tau$; ▶ $\mathsf{Usc}(pk, C')$ *or* $\mathsf{N}(pk, C')$ *such that* C' *parses as* (X_2, W', σ, h, c') *was issued.*

434 **else** $\tau_1 \leftarrow_{\mathrm{R}} \mathbf{K}$; $\mathsf{Apd}(\mathcal{S}_{\mathsf{k}}, ((X_1, X_2, pk, pk_R), \tau_1))$;

435 **if** $\exists\, Z_1, Z_2, Z_3, Z_4 \in \mathcal{G}, \tau \in \mathbf{K} : ((X_2, X_1, Z_3, Z_4, Z_1, Z_2, pk, pk_R), \tau) \in \mathcal{S}_{\mathsf{H}_2}$ and $Z_1^{r_0} Z_2 = X_1^{s_0}$ and $Z_3^{r_0} Z_4 = X_2^{s_0}$ **then** $\tau_2 \leftarrow \tau$; ▶ *The same treatment as for* τ_1

436 **else if** $pk = pk_S$ and $\exists \tau, x, Z_3, Z_4 : ((X_2, X_1, Z_3, Z_4, \epsilon, \epsilon, pk_S, pk_R), \tau, x)) \in \mathcal{S}_{\mathsf{k\&r}}$ **then** $\tau_2 \leftarrow \tau$;

437 **else if** $\exists \tau : ((X_2, X_1, pk, pk_R), \tau) \in \mathcal{S}_{\mathsf{k}}$ **then** $\tau_2 \leftarrow \tau$;

438 **else** $\tau_2 \leftarrow_{\mathrm{R}} \mathbf{K}$; $\mathsf{Apd}(\mathcal{S}_{\mathsf{k}}, ((X_2, X_1, pk, pk_R), \tau_2))$;

439 $m \leftarrow \mathsf{D}(\tau_2, c)$; $R \leftarrow \mathcal{O}_{\mathsf{H}_2}(X_1, X_2)$; $V \leftarrow R^{\sigma} W^{-h}$;

440 $h' \leftarrow \mathcal{O}_{\mathsf{H}_3}(m, \tau_1, c, X_1, X_2, G, R, V, W, pk_S, pk)$;

441 **if** $h = h'$ **then** $\overbrace{\text{return } m;}^{\mathcal{O}_{\mathsf{Usc}}}$ $\overbrace{\text{return } (\tau_1, \tau_2);}^{\mathcal{O}_{\mathsf{N}}}$ **else return** $\perp$;

442 Finalization:

443 **if** $\mathcal{A}$ outputs (C^*, m^*, nr^*) such that: (i) C^* was generated through $\mathcal{O}_{\mathsf{Sc}}(\cdot,\cdot)$, (ii) $1 = d \leftarrow \mathsf{PV}(C^*, m^*, nr^*, pk_S, pk_R)$, and (iii) nr^* was not generated by $\mathcal{O}_{\mathsf{N}}(\cdot,\cdot)$ on a query on (pk_S, C^*) **then**

444 Parse C^* as $(\hat{X}_2, \hat{W}, \hat{\sigma}, \hat{h}, \hat{c})$ and nr^* as $(\hat{\tau}_1, \hat{\tau}_2)$;

445 $\hat{X}_1 \leftarrow G^{\hat{\sigma}} pk_S^{-\hat{h}}$;

446 Recover $((\hat{X}_1, \hat{X}_2, \hat{Z}_1, \hat{Z}_2, \epsilon, \epsilon, pk_S, pk_R), \hat{\tau}, x_2)$ from $\mathcal{S}_{\mathsf{k\&r}}$ ▶ C^* *was generated through* $\mathcal{O}_{\mathsf{Sc}}(pk_R, m)$*, for some* m*, so there are some* $\hat{Z}_1, \hat{Z}_2, \hat{\tau}, x_2 : ((\hat{X}_1, \hat{X}_2, \hat{Z}_1, \hat{Z}_2, \epsilon, \epsilon, pk_S, pk_R), \hat{\tau}, x_2)) \in \mathcal{S}_{\mathsf{k\&r}}$ *(see at line 421)*

447 **if** $\exists\, \hat{Z}_3, \hat{Z}_4 \in \mathcal{G} : ((\hat{X}_1, \hat{X}_2, \hat{Z}_1, \hat{Z}_2, \hat{Z}_3, \hat{Z}_4, pk_S, pk_R), \hat{\tau}_1) \in \mathcal{S}_{\mathsf{H}_2}$ and $Z_3^{r_0} Z_4 = \hat{X}_2^{s_0}$ **then**

448 $U_0 \leftarrow Z_3 {Y_0}^{-x_2}$; **return** U_0;

449 **return** $\perp$;

We reuse the trapdoor test technique. Let bad_1 be the event: "the simulator aborts" (see at lines 415 and 419), then under the RO model

$$\Pr(\mathsf{bad}_1) \leqslant q_{\mathsf{Sc}}^2/(2p^2) \leqslant 1/(2p). \tag{15}$$

Let bad_2 be the event "the Finalization procedure outputs $\perp$.". If $\mathsf{Succ}_{\mathcal{A}}^{\mathsf{unr}} \wedge \neg\mathsf{bad}_1 \wedge \mathsf{bad}_2$ occurs, $\mathcal{A}$ never queried the $\mathcal{O}_{\mathsf{H}_3}$ oracle on $(\hat{X}_1, \hat{X}_2, \hat{Z}_1, \hat{Z}_2, \hat{Z}_3, \hat{Z}_4, pk_S, pk_R)$ such that $Z_3^{r_0} Z_4 = X_2^{s_0}$; then $\mathcal{A}$ successfully guessed the value of $\mathcal{O}_{\mathsf{H}_2}(\hat{X}_1, \hat{X}_2, \mathsf{2DH}(Y_0, Y_0', X_1), \mathsf{2DH}(Y_0, Y_0', X_2), pk_S, pk_R)$. Hence, under the RO model,

$$\Pr(\mathsf{Succ}_{\mathcal{A}}^{\mathsf{unr}} \wedge \neg\mathsf{bad}_1 \wedge \mathsf{bad}_2) \leqslant 1/|\mathbf{K}|. \tag{16}$$

If the event $\mathsf{Succ}_{\mathcal{A}}^{\mathsf{unr}} \wedge \neg\mathsf{bad}_1 \wedge \neg\mathsf{bad}_2$ occurs, as $Z_3^{r_0} Z_4 = \hat{X}_2^{s_0}$ it holds that $Z_3 = \mathsf{cDH}(\hat{X}_2, Y_0)$, except with probability $1/p$. And, as $\hat{X}_2 = X_0 G^{x_2}$ (see at line 418) $U_0 = \mathsf{cDH}(X_0, Y_0) = Z_3 Y_0^{-x_2}$. Using $\mathcal{A}$ and the simulator, we obtain a machine which takes (X_0, Y_0) as inputs and outputs $\mathsf{cDH}(X_0, Y_0)$ with probability $\Pr(\mathsf{Succ}_{\mathcal{A}}^{\mathsf{unr}} \wedge \neg\mathsf{bad}_1 \wedge \neg\mathsf{bad}_2) - 1/p$. As

$$\Pr(\mathsf{Succ}_{\mathcal{A}}^{\mathsf{unr}}) \leqslant \Pr(\mathsf{Succ}_{\mathcal{A}}^{\mathsf{unr}} \wedge \neg\mathsf{bad}_1 \wedge \neg\mathsf{bad}_2) + \Pr(\mathsf{bad}_1) + \Pr(\mathsf{bad}_2),$$

the result follows from (15) and (16). □

4.5 On the Concrete Choice of the Set of Domain Parameters

Recall that a concrete instance of a cryptographic problem is said to have k–bits of security if any adversary $\mathcal{A}$ running in time t and trying to solve the problem succeeds with probability $\varepsilon \leqslant t/2^k$. A cryptographic scheme is said to have k–bits of security with respect to some security attribute, if any attacker playing the security game that defines the attribute and running in time t, succeeds with probability $\leqslant t/2^k$.

In $\mathcal{SC}_{\mathsf{edl}}$, if the underlying group $\mathcal{G}$ and the encryption scheme $\mathcal{E}$ are chosen such that the cDH problem in $\mathcal{G}$ has $(k+1)$–bits of security and $\mathcal{E}$ has $(k+3)$–bits of security then, from (1), it follows that $\mathcal{SC}_{\mathsf{edl}}$ is $(t, q_{\mathsf{Sc}}, q_{\mathsf{Usc}}, q_{\mathsf{N}}, \varepsilon)$–secure in the SKI–MU insider confidentiality in the FSO/FUO–IND–CCA2 sense, where

$$\varepsilon \leqslant t/2^{k+1} + t/2^{k+3} + 4(q_{\mathsf{H}_2} + 2q_{\mathsf{Usc}} + 2q_{\mathsf{N}} + 1)/p + 2t/|\mathbf{K}|.$$

As an $\mathcal{O}(\sqrt{p})$ algorithm is known for the discrete logarithm problem, $\alpha\sqrt{p} \geqslant 2^{k+1}$ for some "moderate" constant α. As $q_{\mathsf{H}_2} + 2q_{\mathsf{Usc}} + 2q_{\mathsf{N}} + 1 \leqslant 2t$ and $|\mathbf{K}| \geqslant 2^{k+3}$, we obtain $\varepsilon \leqslant t/2^k$. Hence, $\mathcal{SC}_{\mathsf{edl}}$ has k–bits of security in the SKI–MU insider confidentiality in the FSO/FUO–IND–CCA2 sense. A similar analysis shows that under the same assumptions, $\mathcal{SC}_{\mathsf{edl}}$ has k–bits of security with regard to (*i*) the MU insider strong unforgeability in the FSO/FUO–sUF–CMA sense, (*ii*) the soundness of non–repudiation, and (*iii*) the unforgeability of non–repudiation evidence.

5 Comparison with other schemes

The design of $\mathcal{SC}_{\mathsf{edl}}$ integrates the randomness reuse idea suggested in [2, 20]. A $\mathcal{SC}_{\mathsf{edl}}$ sender (resp. receiver) key pair generation requires one (resp. two) exponentiations. An execution of the Sc algorithm requires $\mathsf{Exp}(\mathcal{G}, 8)$. Four of the 8 exponentiations can be performed *offline*, before the receiver public key and the plain text are provided. If the receiver public key is provided before the plain text (this may occur in email systems where the recipient is often typed before email text) *all* the 8 exponentiations can performed before the plain text is provided. The Usc and N algorithms require $\mathsf{Exp}(\mathcal{G}, 4)$ (two pairs of exponentiations with the same exponent) and two multi–exponentiations. The public verification algorithm requires two multi–exponentiations. If the encryption scheme $\mathcal{E}$ is such that a clear text and a corresponding ciphertext have the same length, the communication overhead of $\mathcal{SC}_{\mathsf{edl}}$, compared to the CM signature scheme is one group element. Notice that we neglected the group membership tests, as they have a negligible cost in $\mathbb{Z}_q^*$ and elliptic curve groups.

In [19], Malone–Lee (ML) proposes a very efficient design with NINR. Unfortunately, the design, which is analysed in the RO model under de cDH assumption, does not achieve insider security. Also the reduction uses the Forking

Lemma [6, 21]. Assuming $q_{\mathsf{H}} = 2^{32}$, for a security target of 128–bits, the underlying group $\mathcal{G}'$ must be chosen to offer 160–bits of security. In the case $\mathcal{G}'$ is a (sub)group of the rational points of an elliptic curve $\mathcal{G}' = E(\mathbb{F}_{q'})$, q' has to be chosen such that $|q'| \approx 320$. An execution of the ML Sc or Usc algorithm requires two exponentiations. As a modular multiplication (performed with the Karatsuba–Ofman algorithm) in $\mathbb{F}_{q'}$ has complexity $\approx |q'|^{1.585}$. Given the tightness of our reduction, in ECC, we need $|q| = 256$ to have 128 bits of security. As $\mathsf{Mult}(\mathbb{F}_{q'}) \approx 1.42 \cdot \mathsf{Mult}(\mathbb{F}_q)$, assuming that a group operation in $\mathcal{G}'$ requires $14 \cdot \mathsf{Mult}(\mathbb{F}_{q'})$ (see[1] [16, p. 96]), $\mathsf{Exp}(\mathcal{G}') \approx 6720 \cdot \mathsf{Mult}(\mathbb{F}_{q'}) \approx 9570 \cdot \mathsf{Mult}(\mathbb{F}_q) \approx 1.78 \cdot \mathsf{Exp}(\mathcal{G})$. The ML design is about (a) 2.25 times faster for signcryption, and (b) 1.25 times faster for unsigncryption than ours.

Bjørstad and Dent's (BD) design [8] tightly achieves, in the RO model, insider unforgeability under the cDH assumption and *outsider* confidentiality under the gap DH assumption. The scheme does not achieve insider confidentiality. The Sc algorithm requires $\mathsf{Exp}(\mathcal{G}, 3)$ operations, the Usc algorithm requires two multi–exponentiations. The BD construction is about 2.5 times faster then $\mathcal{SC}_{\mathsf{edl}}$ for signcrypted text generation and about 3 times faster for unsigncryption.

Some of the designs we consider hereunder assume the existence of groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ together with a bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. Recall that for a choice of the groups $\mathcal{G}, \mathbb{G}_1, \mathbb{G}_2$, and $\mathbb{G}_T$ (where $\mathcal{G}$ is a classical ECC group), with a target of 128–bits of security, the cost of a pairing evaluation is about $\approx \mathsf{Exp}(\mathcal{G}, 8)$, $\mathsf{Exp}(\mathbb{G}_1) \approx \mathsf{Exp}(\mathcal{G}, 3)$, and $\mathsf{Exp}(\mathbb{G}_2) \approx \mathsf{Exp}(\mathcal{G}, 6)$ [7, p. 126].

Arriaga *et al.*'s generic construction with NINR [2] is insider secure in the standard model. They propose an instantiation of their design which assumes the Decisional Bilinear and the q–Strong DH assumptions. Unfortunately, the unforgeability is achieved in the registered key model [20], wherein an attacker needs to register to the challenger the keys pairs it uses in its attack. The design assumes the existence of groups $\mathbb{G}, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ such that (i) $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ are of order q, (ii) there is a bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ and (iii) a one to one and efficiently invertible mapping from $\mathbb{G}$ to $\mathbb{Z}_q$. An evaluation of the Sc algorithm requires $\mathsf{Exp}(\mathbb{G}, 2) + \mathsf{Exp}(\mathbb{G}_1)$ and one multi–exponentiation in $\mathbb{G}$. The Usc algorithm requires two multi–exponentiations, one in $\mathbb{G}$ and one in $\mathbb{G}_2$, and a pairing evaluation. For a target of 128 bits of security, we expect $\mathcal{SC}_{\mathsf{edl}}$ to be 1.5 times faster for signcryption and 2.8 times faster for unsigncryption.

Matsuda *et al.* [20]'s two generic constructions with NINR are insider secure in the FSO/FUO model. The security reduction is provided in the RO model. The most efficient among the instanciations that achieve insider security in the FSO/FUO model, uses as base schemes, the DHIES encryption scheme [1] and the BLS signature scheme [9]. The construction assumes the existence of groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ together with a bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$. A Sc operation requires $\mathsf{Exp}(\mathbb{G}_1, 3)$, an Usc operation requires $\mathsf{Exp}(\mathbb{G}_2)$ and two pairing

[1] If $|\mathcal{G}| = 2^\lambda$, the cost of $\mathsf{Exp}(\mathcal{G})$ using the classical square–and–multiply algorithm is $\approx 1.5 \cdot \lambda$ operations in $\mathcal{G}$. And if $\mathcal{G}$ is such that the multiplication of two of its elements requires 14 multiplications in $\mathbb{F}_q$ then the computational cost of an exponentiation is $14 \cdot 1.5 \cdot \lambda$ multiplications in $\mathbb{F}_q$.

evaluations. Compared to $\mathcal{SC}_{\mathsf{edl}}$, for a target of 128 bits of security (given that $\mathsf{Exp}(\mathbb{G}_1) \approx \mathsf{Exp}(\mathcal{G}, 3)$, $\mathsf{Exp}(\mathbb{G}_2) \approx \mathsf{Exp}(\mathcal{G}, 6)$ and the cost of a pairing evaluation $\approx \mathsf{Exp}(\mathcal{G}, 8)$) we expect our design to be 1.12 times faster for signcryption, and about 3.6 times faster for unsigncryption.

For a comparison with Chiba *et al.*'s generic contruction with NINR [13], we consider the most efficient among the instantiations they propose. It achieves insider security in the FSO/FUO model, under the Decisional Bilinear and the q–strong DH assumptions. Although the insider security is shown in the standard model, the unforgeability is achieved in the registered key model. Besides, the scheme assumes the existence of a pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, with $\mathbb{G}_1 = \mathbb{G}_2$. The Sc algorithm requires $\mathsf{Exp}(\mathbb{G}_1, 3)$ together with a multi–exponentiation. The Usc operation requires one exponentiation, one multi–exponentiation, and one pairing evaluation. We expect $\mathcal{SC}_{\mathsf{edl}}$ to be about 1.5 times faster for signcryption, and about 2.3 times faster for unsigncryption.

Fan *et al.*'s design [14] assumes the existence of a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, where $\mathbb{G}$ and $\mathbb{G}_T$ are multiplicative cyclic groups. The Sc algorithm requires one pairing, $\mathsf{Exp}(\mathbb{G}, 4) + \mathsf{Exp}(\mathbb{G}_T)$, and $(n+1)/2$ group operations in $\mathbb{G}$, where n is the bit–length output of some collision resistant hash function $\mathsf{H} : \mathbb{G} \to \{0,1\}^n$ used in the design. The unsigncryption algorithm requires 3 pairings, $\mathsf{Exp}(\mathbb{G}, 2)$, and $(n/2+1)$ group operations in $\mathbb{G}$. A signcrypted text is an element of $\mathbb{G}_T \times \mathbb{G}^3$. For a choice of the groups $\mathcal{G}$, $\mathbb{G}$, and $\mathbb{G}_T$, with target 128–bits of security, we expect our design to be about (a) 2.5 times faster for signcryption, and (b) 7.5 times faster for unsigncryption than Fan *et al.*'s construction, in addition to having shorter signcrypted texts.

In the scheme from [22], defined over the (RSA based) group of signed quadratic residues $\mathbb{J}_N^+$, the Sc algorithm requires $\mathsf{Exp}(\mathbb{J}_N^+, 6)$ and the Usc algorithm requires $\mathsf{Exp}(\mathbb{Z}_N, 3)$ (we ignore the exponentiation with the RSA public exponent, which is often small and sparse). Unfortunately, the security reduction uses the Forking Lemma, which implies a $1/q_{\mathsf{H}}$ security degradation, where q_{H} is the number of digest queries the attacker issues. For $q_{\mathsf{H}} = 2^{32}$, if the target security is 128–bits, the RSA modulus needs to have a bitlength $|N| \approx 7864$ [18][2]. Then, considering a square–and–multiply based exponentiation, $\mathsf{Exp}(\mathbb{J}_N^+) \approx 11796 \cdot \mathsf{Mult}(\mathbb{Z}_N)$, where $\mathsf{Mult}(\mathbb{Z}_N)$ denotes the cost of a multiplication in $\mathbb{Z}_N$. In contrast $\mathcal{SC}_{\mathsf{edl}}$ can be instantiated over an elliptic curve (sub)group $\mathcal{G} = E(\mathbb{F}_q)$ such that $|q| \approx 256$ and $\mathcal{G}$ has 128–bits of security. Assuming that a group operation in $\mathcal{G}$ requires $14 \cdot \mathsf{Mult}(\mathbb{F}_q)$ [16, p. 96], $\mathsf{Exp}(\mathcal{G}) \approx 5376 \cdot \mathsf{Mult}(\mathbb{F}_q)$. As $\mathsf{Mult}(\mathbb{Z}_N) > 30 \cdot \mathsf{Mult}(\mathbb{F}_q)$, for a 128–bits security target, we expect $\mathcal{SC}_{\mathsf{edl}}$ over $\mathcal{G}$ to be at least 13 times faster (for key generation, signcryption, unsigncryption, etc.) than the design from [22].

Compared to the ML and BD schemes, which do not require any specificity of the underlying group and do not achieve insider security, $\mathcal{SC}_{\mathsf{edl}}$ offers a stronger security, even if it is less efficient. And, compared to the schemes from [2,13,14,20, 22], $\mathcal{SC}_{\mathsf{edl}}$ offers a tight security reduction, a better efficiency and a comparable or a superior security. We summarize in Table 1 some elements of comparisons.

[2] see also www.keylength.com

The column Assumptions indicates the computational assumptions used in the security reductions; FL and IS stand respectively for Forking Lemma and Insider Security (in the FSO/FUO model). The letters 'y' and 'n' stand for "yes" and "no", respectively; 'p' stands for "partial" (BD achieves insider unforgeability, but *outsider* confidentiality). In the column Computations $[a, b, c][a', b', c']$ means that a Sc (resp. Usc) operation requires a (resp. a') exponentiations, b (resp. b') multi–exponentiations, and c (resp. c') pairing evaluations. We recall that the number of exponentiations has to be considered in conjunction with the underlying mathematical structure. For instance, as previously said, if a scheme requires a bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$, for a target of 128 bits of security, it holds $\mathsf{Exp}(\mathbb{G}_1) \approx \mathsf{Exp}(\mathcal{G}, 3)$ and $\mathsf{Exp}(\mathbb{G}_2) \approx \mathsf{Exp}(\mathcal{G}, 6)$. The column Overhead indicates the signcrypted ciphertext overhead compared to the *cleartext*.

Table 1. Comparison of the proposed signcryption schemes with some SCNINR schemes from the litterature.

Scheme	Assumptions	FL	IS	Computations	Overhead
ML [19]	RO, cDH	y	n	[2, 0, 0] [2, 0, 0]	$2 \cdot \mathsf{sz}(\mathbb{Z}_p)$
BD [8]	RO, cDH	n	p	[2, 0, 0] [0, 2, 0]	$\mathsf{sz}(\mathcal{G}) + \mathsf{sz}(\mathbb{Z}_p)$
ABF [2]	DBDH, q-sDH	n	y	[3, 1, 0] [0, 2, 1]	$\mathsf{sz}(\mathbb{G}) + \mathsf{sz}(\mathbb{G}_1)$
MMS [20]	RO, GDH, co–cDH	n	y	[3, 0, 0] [1, 0, 2]	$\mathsf{sz}(\mathbb{G}_1) + \mathsf{sz}(\mathbb{G}_2)$
CMSM [13]	DBDH, q-sDH	n	y	[3, 1, 0] [1, 1, 2]	$\mathsf{sz}(\mathbb{Z}_p) + 4 \cdot \mathsf{sz}(\mathbb{G}_1)$
FZT [14]	DBDH, DL	n	y	[5, 0, 1] [2, 0, 3]	$\mathsf{sz}(\mathbb{Z}_p) + 2 \cdot \mathsf{sz}(\mathbb{G}_1)$
SSN [22]	RO, RSA	y	y	[6, 0, 0] [3, 0, 0]	$\mathsf{sz}(\mathbb{Z}_p) + 2 \cdot \mathsf{sz}(\mathbb{Z}_N)$
Ours: $\mathcal{SC}_{\mathsf{edl}}$	RO, cDH	n	y	[8, 0, 0] [4, 2, 0]	$2 \cdot \mathsf{sz}(\mathbb{Z}_p) + 2 \cdot \mathsf{sz}(\mathcal{G})$

References

1. Abdalla, M., Bellare, M., Rogaway, P.: The oracle Diffie-Hellman assumptions and an analysis of DHIES. In: Naccache, D. (ed.) CT-RSA 2001. LNCS, vol. 2020, pp. 143–158. Springer, Heidelberg (2001)
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) Applied Cryptography and Network Security. ACNS 2012. LNCS, vol 7341. Springer, Berlin, Heidelberg (2012)
3. Badertscher C., Banfi F., Maurer U.: A Constructive Perspective on Signcryption Security. In: Catalano D., De Prisco R. (eds) Security and Cryptography for Networks. SCN 2018. LNCS, vol 11035. Springer, Cham (2018)
4. Baek J., Steinfeld R., Zheng Y.: Formal Proofs for the Security of Signcryption. Journal of Cryptology, 20(2):203–235 (2007)
5. Bao F., Deng R.H.: A signcryption scheme with signature directly verifiable by public key. In: Imai H., Zheng Y. (eds) Public Key Cryptography. PKC 1998. LNCS, vol 1431. Springer, Berlin, Heidelberg (1998)
6. 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)

7. Benhamouda F., Couteau G., Pointcheval D., Wee H.: Implicit Zero-Knowledge Arguments and Applications to the Malicious Setting. In: Gennaro R., Robshaw M. (eds) Advances in Cryptology – CRYPTO 2015. CRYPTO 2015. LNCS, vol 9216. Springer, Berlin, Heidelberg (2015)
8. Bjørstad T.E., Dent A.W.: Building Better Signcryption Schemes with Tag-KEMs. In: Yung M., Dodis Y., Kiayias A., Malkin T. (eds) Public Key Cryptography - PKC 2006. PKC 2006. LNCS, vol 3958. Springer, Berlin, Heidelberg (2006)
9. Boneh, D., Lynn, B., Shacham, H.: Short signatures from the Weil pairing. J. Cryptology 17(4), 297–319 (2004)
10. Boneh D., Shen E., Waters B.: Strongly Unforgeable Signatures Based on Computational Diffie–Hellman. In: Yung M., Dodis Y., Kiayias A., Malkin T. (eds) Public Key Cryptography — PKC 2006. PKC 2006. LNCS, vol 3958. Springer, Berlin, Heidelberg (2006)
11. Cash D., Kiltz E., Shoup V.: The twin Diffie–Hellman problem and applications. Journal of Cryptology, 22(4), 470–504 (2009)
12. Chevallier–Mames B.: An Efficient CDH–Based Signature Scheme with a Tight Security Reduction. In: Shoup V. (eds) Advances in Cryptology — CRYPTO 2005. CRYPTO 2005. LNCS, vol 3621. Springer, Berlin, Heidelberg (2005)
13. 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) Applied Cryptography and Network Security. ACNS 2011. LNCS, vol 6715. Springer, Berlin, Heidelberg (2011)
14. Fan J., Zheng Y., Tang X.: Signcryption with non–interactive non–repudiation without random oracles. In: Transactions on computational science X, pp. 202–230. Springer, Berlin, Heidelberg (2010)
15. Goh EJ., Jarecki S.: A Signature Scheme as Secure as the Diffie–Hellman Problem. In: Biham E. (eds) Advances in Cryptology — EUROCRYPT' 03. EUROCRYPT 2003. LNCS, vol 2656. Springer, Berlin, Heidelberg (2003)
16. Hankerson D., Menezes A. J., Vanstone S.: Guide to elliptic curve cryptography. Springer (2004)
17. Katz J., Wang N.: Efficiency improvements for signature schemes with tight security reductions. In Proceedings of the 10th ACM conference on Computer and communications security, pp. 155–164. ACM, (2003)
18. Lenstra A. K.: Key lengths. Handbook of Information Security, vol. 2, pp. 617–635. Wiley (2005)
19. Malone–Lee J.: Signcryption with non–interactive non–repudiation. Designs, Codes and Cryptography, vol. 37, no 1, pp. 81–109. Springer (2005)
20. Matsuda T., Matsuura K., Schuldt J.C.N.: Efficient Constructions of Signcryption Schemes and Signcryption Composability. In: Roy B., Sendrier N. (eds) Progress in Cryptology - INDOCRYPT 2009. INDOCRYPT 2009. LNCS, vol 5922. Springer, Berlin, Heidelberg (2009)
21. Pointcheval D., Stern J.: Security Proofs for Signature Schemes. In: Maurer U. (eds) Advances in Cryptology — EUROCRYPT'96. EUROCRYPT 1996. LNCS, vol 1070. Springer, Berlin, Heidelberg (1996)
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. (eds) Codes, Cryptology and Information Security. C2SI 2019. LNCS, vol 11445. Springer, Cham (2019)
23. Zheng Y.: Digital signcryption or how to achieve cost(signature & encryption) $\ll$ cost(signature) + cost(encryption). In: Kaliski B.S. (eds) Advances in Crypto-

logy — CRYPTO '97. CRYPTO 1997. LNCS, vol. 1294. Springer, Berlin, Heidelberg (1997)