Formal Proofs for the Security of Signcryption

Joonsang Baek[1], Ron Steinfeld[1], and Yuliang Zheng[2]

[1] School of Network Computing, Monash University,
McMahons Road, Frankston, VIC 3199, Australia
{joonsang.baek,ron.steinfeld}@infotech.monash.edu.au
[2] Dept. Software and Info. Systems,
UNC Charlotte, NC 28223, USA
yzheng@uncc.edu

Abstract. Signcryption is a public key or asymmetric cryptographic method that provides simultaneously both message confidentiality and unforgeability at a lower computational and communication overhead. In this paper, we propose a sound security model for signcryption that admits rigorous formal proofs for the confidentiality and unforgeablity of signcryption. A conclusion that comes out naturally from this work is that, as an asymmetric encryption scheme, signcryption is secure against adaptive chosen ciphertext attack in the random oracle model relative to, quite interestingly, the Gap Diffie-Hellman problem, and as a digital signature scheme, signcryption is existentially unforgeable against adaptive chosen message attack in the random oracle model relative to the discrete logarithm problem.

1 Introduction

1.1 Motivation for Research

To achieve message confidentiality and authenticity, there have been a great number of proposals of cryptographic building blocks both in the symmetric and asymmetric settings. Furthermore, ever since provable security with respect to strong attack models was regarded as important for proposals of new schemes, intensive efforts have been made in this line of research. In the early stage, Zheng and Seberry [26] proposed several practical asymmetric encryption schemes secure against adaptive chosen ciphertext attack, where the adversary is allowed to make queries to a decryption oracle to learn any information about a target ciphertext with the only restriction that the target ciphertext itself cannot be queried to the decryption oracle. Afterwards, schemes with security proofs against such an attack in the reductionist style (in other words, proofs of reduction from attacking the asymmetric encryption schemes to solving a computationally hard problems) under the heuristic assumption called random oracle model [6], were followed [5] [10] [17]. Moreover, the asymmetric encryption scheme with security proof without such an assumption was also proposed [7] and received great attention. For provable security of digital signature schemes,

D. Naccache and P. Paillier (Eds.): PKC 2002, LNCS 2274, pp. 80–98, 2002.

slight modifications of the schemes in [8] and [19] were proved [18] [15] to be existentially unforgeable against adaptive chosen message attack [12] in the random oracle model.

There has been growing interest in the integration of message confidentiality with authenticity. In the symmetric setting, some heuristic methods to support confidentiality and authenticity at the same time for transmitted data were considered in the internet standards such as IPSec [13] and SSL [9], and recently, these methods have been analyzed in [4] and [14]. In the asymmetric setting, Zheng [23] proposed a scheme called 'signcryption' which simultaneously and efficiently provides message confidentiality and unforgeability. Due to the potential of signcryption, especially in applications that demand high speed and low communication overhead, it is important to research into *rigorous security proofs in the reductionist style* for signcryption schemes.

The aim of this paper is to propose a precise security model for signcryption and provide rigorous proofs based on the proposed model. As a result of this work, we conclude that signcryption does meet strong security requirements with respect to message confidentiality and unforgeability under known cryptographic assumptions.

1.2 Related Work

At PKC '98, Tsiounis and Yung [22] studied a variant of a strengthened ElGamal encryption scheme originally proposed in [26], where Schnorr signature is used to provide non-malleability for the ElGamal encryption. However, the security goal of their scheme is to provide confidentiality and consequently, strong authentication for message origin is not supported in their scheme. The same scheme was also analyzed by Schnorr and Jakobsson [20] under both the generic and the random oracle models.

At ISW 2000, Steinfeld and Zheng [21] proposed the first signcryption scheme whose security is based on integer factorization. They provided a formal security model and security proof for the unforgeability of the proposed scheme. However, they left open a formal proof of the confidentiality of their scheme.

In a separate development, various researchers have made some interesting observations in the symmetric setting. At Asiacrypt 2000, Bellare and Namprepre [4] proposed formal security models for the compositions of symmetric encryption and message authentication. They concluded that only 'Encrypt-then-MAC (EtM)' composition is *generically* secure against chosen ciphertext attack and existentially unforgeable against chosen message attack. Krawczyk [14] considered the same problem while examining how to build secure channels over insecure networks. He showed that 'MAC-then-Encrypt (MtE)' composition was secure too under the assumption that the encryption method employed was either a secure CBC mode or a stream cipher that XORs the data with a random pad.

Very recently [1] (and independently of our work), in [1] it has been shown that earlier results in [4] and [14] can be extended to the asymmetric setting. Although security notions of [1] bear some similarities to ours, the generic anal-

ysis given in that paper does not appear to be applicable in deriving our security results for signcryption, primarily due to the special structure of signcryption.

1.3 Differences between Our Model and Previous Models

To address the significant difference between security implication of the compositions of encryption and authentication in the symmetric setting and that in the asymmetric setting, we consider the confidentiality of the 'Encrypt-then-MAC (EtM)' and 'Encrypt-and-MAC (EaM)' compositions in the symmetric setting, and the security of the corresponding *simple* asymmetric versions, namely, 'Encrypt-then-Sign (SimpleEtS)' and 'Encrypt-and-Sign (SimpleEaS)', defined in the natural way, with the signer's public key being appended. As was independently observed in [1], it is not hard to see that while the symmetric composition EtM is secure against chosen ciphertext attack (indeed, EtM is generically secure as shown in [4]), the simple asymmetric version SimpleEtS is *completely insecure against adaptive chosen ciphertext attack*, even if the underlying encryption scheme is secure against adaptive chosen ciphertext attack. The reason is that in the asymmetric versions, a ciphertext in the composed scheme contains an additional component (not present in the symmetric versions), namely the *sender's signature public key*. The fact that this component is easily malleable implies the insecurity of the asymmetric version SimpleEtS under adaptive chosen ciphertext attack.

As an example, let us assume that Alice encrypts and signs her message m following the SimpleEtS composition. That is, she encrypts the message m using an asymmetric encryption algorithm $\mathcal{E}_{pk_B}(\cdot)$ and computes $c = \mathcal{E}_{pk_B}(m)$. Then she signs on c using her digital signature algorithm $\mathcal{S}_{sk_A}(\cdot)$ to produce $\sigma = \mathcal{S}_{sk_A}(c)$. Now the ciphertext C is (c, σ). However, an adversary Eve now generates her own public and private key pair (pk_E, sk_E) and signs on c obtained by eavesdropping the ciphertext C en route from Alice to Bob. Namely, she can produce $C' = (c, \mathcal{S}_{sk_E}(c))$ where $\mathcal{S}_{sk_E}(\cdot)$ is Eve's digital signature algorithm. Then she hands in her public key pk_E (which may be contained in Eve's digital certificate) to Bob. Now notice that C' which is different from C is always verified as being valid using Eve's public key pk_E. Thus Bob decrypts C' into m. Hence Eve succeeds in her chosen ciphertext attack on the SimpleEtS scheme even if the underlying asymmetric encryption scheme is strong, say, secure against adaptive chosen ciphertext attack.

1.4 Signcryption: A *Variant* of Encrypt-and-Sign (EaS)

The most attractive feature of signcryption is its efficiency. To achieve this goal, signcryption can be viewed as an instantiation of the *Encrypt-and-Sign (EaS)* paradigm. Besides efficiency gains, however, signcryption has some important security-related improvements on the (insecure) SimpleEtS and SimpleEaS compositions. That is, signcryption seems to 'fix', intuitively, the following two problems with the confidentiality of those simple compositions. The first problem is with the malleability of SimpleEtS discussed above in the asymmetric setting.

The second problem (which has pointed out in [4] and [14] for the scheme EaM in the symmetric setting), is that the EaS composition cannot be *generically* secure because the signature part can reveal some information about the plaintext message, and this may be true even though the underlying signature scheme is unforgeable. However, the result in [4] and [14] does not mean that every EaS composition is insecure. Rather one should read it as that security of cryptographic schemes employing EaS ought to be analyzed on a case by case basis.

1.5 Our Contributions

As mentioned earlier, signcryption has features which intuitively fix the above mentioned confidentiality problems of the SimpleEtS and SimpleEaS compositions. A main contribution of this paper is to provide a *rigorous proof that this intuition is indeed correct*, under known cryptographic assumptions in the random oracle model for the underlying hash functions. More specifically, we define a strong security notion that is similar to the well known 'IND-CCA2' [3] notion for standard public-key encryption schemes, and prove the confidentiality of Zheng's original signcryption schemes in the security notion. Our notion for confidentiality is even stronger than the direct adaptation of 'IND-CCA2' to the setting of signcryption, since we allow the attacker to query the signcryption oracle, as well as the unsigncryption oracle. We also prove the unforgeability of signcryption in a strong sense, namely existential unforgeability against adaptive chosen message attack.

2 Preliminaries

2.1 The Gap Diffie-Hellman Problem

At PKC 2001, Okamoto and Pointcheval proposed a new class of computational problems, called gap problems [16]. A gap problem is dual of inverting and decisional problems . More precisely, this problem is to solve an inverting problem with the help of an oracle for a decisional problem. In this paper, we only recall the Gap Diffie-Hellman (GDH) problem, among the various gap problems discussed in [16], on which the confidentiality of signcryption is based.

Definition 1 (The Gap Diffie-Hellman Problem). Let A_{gdh} be an adversary for the GDH Problem. Consider a following experimental algorithm that takes a security parameter $k \in \mathbb{N}$. A_{gdh}'s job is to compute the Diffie-Hellman key $g^{xy} \bmod p$ of $g^x \bmod p$ and $g^y \bmod p$ with the help of Decisional Diffie-Hellman (DDH) oracle $DDH_g(\cdot,\cdot,\cdot)$. Note that this DDH oracle tests whether a given tuple is a Diffie-Hellman tuple (DH-tuple) or not. For example, if (X, Y, Z) is a DH-tuple, $DDH_g(X, Y, Z) = 1$. Otherwise, it returns 0.

Experiment $\mathbf{GDHExp}^{\text{invert}}_{\mathsf{GDH},A_{gdh}}(k)$
Choose primes (p, q) such that $|p| = k$ and $q|(p-1)$
Choose $g \in (\mathbb{Z}/p\mathbb{Z})^*$ such that $\mathrm{Ord}(g) = q$

$x \leftarrow_R \mathbb{Z}/q\mathbb{Z}$; $X \leftarrow g^x \bmod p$
$y \leftarrow_R \mathbb{Z}/q\mathbb{Z}$; $Y \leftarrow g^y \bmod p$
$g^{xy} \bmod p \leftarrow A_{gdh}^{DDH_g(\cdot,\cdot,\cdot)}(k, g, X, Y)$
`return` $g^{xy} \bmod p$

Now let $\mathbf{Succ}^{\text{invert}}_{\mathsf{GDH},A_{gdh}}(k) \stackrel{\text{def}}{=} \Pr[\mathbf{GDHExp}^{\text{invert}}_{\mathsf{GDH},A_{gdh}}(k) = g^{xy} \bmod p]$. Then define an advantage function of A_c as follows.

$$\mathbf{Adv}^{\text{invert}}_{\mathsf{GDH}}(k, t, q_{ddh}) \stackrel{\text{def}}{=} \max_{A_{gdh}}\{\mathbf{Succ}^{\text{invert}}_{\mathsf{GDH},A_{gdh}}(k)\},$$

where the maximum is taken by all A_{gdh} with execution time t and making q_{ddh} queries to the DDH oracle. We say the GDH problem is secure if $\mathbf{Adv}^{\text{invert}}_{\mathsf{GDH}}(k, t, q_{ddh})$ is a negligible function in k[1] for any adversary A_{gdh} with polynomial time bound in k and whose queries are polynomial in k.

2.2 Description of the Original Signcryption Scheme (SDSS1-Type)

Note that the signcryption scheme described in this section is the one derived from the shorthand digital signature scheme (SDSS1) (named by the author of [23]) which is a variant of ElGamal based signature schemes. Another signcryption scheme derived from SDSS2 can be described and analyzed in a very similar manner presented in this paper. So we only consider the SDSS1-type signcryption scheme. Note also that the hash functions used in the signcryption scheme are assumed to be random oracles [6] in this paper. And the *bind* information, which is hashed in the signcryption process, contains such information as Alice and Bob's public keys. We remark that κ, which is a Diffie-Hellman key, is directly provided as input to random oracle H without being hashed by the random oracle G in our description. Since hashing Diffie-Hellman key in signcryption is allowed to be done quite flexibly as noted in [23] and [24], we do not regard this as a major modification of the scheme.

Definition 2. Let $\mathsf{SC} = (\mathcal{COM}, \mathcal{K}_A, \mathcal{K}_B, \mathcal{SC}, \mathcal{USC})$ be a signcryption scheme. Let $k \in \mathbb{N}$ be a security parameter. Suppose that $\mathsf{H} : \{0,1\}^* \rightarrow \mathbb{Z}/q\mathbb{Z}$ and $\mathsf{G} : \{0,1\}^* \rightarrow \{0,1\}^l$ are random oracles. Note that l is a security parameter, i.e. key length, for a symmetric encryption scheme described below. Let $\mathsf{E}_\alpha(\cdot)$ denote a symmetric encryption function under some key α and $\mathsf{D}_\alpha(\cdot)$ denote a decryption function of the symmetric encryption. (We assume that there is a one-to-one correspondence between l and k. We also assume that $\mathsf{D}_\alpha(\cdot)$ is one-to-one over some ciphertext space $\mathcal{C}$ for all α. (This implies that the symmetric encryption is deterministic.)) Also, note that $|\cdot|$ indicates the number of bits in the binary representation of an integer.

[1] We say a probability function $f : \mathbb{N} \rightarrow \mathbb{R}_{[0,1]}$ is negligible in k if, for all $c > 0$, there exists $k_0 \in \mathbb{N}$ such that $f(k) \leq \frac{1}{k^c}$ whenever $k \geq k_0$. Here, $\mathbb{R}_{[0,1]} = \{x \in \mathbb{R} \mid 0 \leq x \leq 1\}$.

Signcryption SC

Common parameter generation $\mathcal{COM}(k)$
Choose prime p such that $|p| = k$
Choose prime $q|(p-1)$ such that $q > 2^{l_q(k)}$
where $l_q(k) \in \mathsf{N}$ for some function l_q
Choose $g \in (\mathbb{Z}/p\mathbb{Z})^*$ such that $\mathrm{Ord}(g) = q$
$cp_{sc} \leftarrow (p, q, g)$
`return` cp_{sc}
Alice's key generation $\mathcal{K}_A(k, cp_{sc})$
$x_A \leftarrow_R \mathbb{Z}/q\mathbb{Z}$; $y_A \leftarrow g^{x_A} \bmod p$
`return` (y_A, x_A)
Bob's key generation $\mathcal{K}_B(k, cp_{sc})$
$x_B \leftarrow_R \mathbb{Z}/q\mathbb{Z}$; $y_B \leftarrow g^{x_B} \bmod p$
`return` (y_B, x_B)
Signcryption $\mathcal{SC}^{\mathsf{G},\mathsf{H}}_{x_A,y_B}(m)$ by Alice the Sender
$x \leftarrow_R \mathbb{Z}/q\mathbb{Z}$; $\kappa \leftarrow y_B^x \bmod p$; $\tau \leftarrow \mathsf{G}(\kappa)$
Get $y_A (= g^{x_A} \bmod p)$; $bind \leftarrow y_A || y_B$
$c \leftarrow \mathsf{E}_\tau(m)$; $r \leftarrow \mathsf{H}(m||bind||\kappa)$; $s \leftarrow x/(r + x_A) \bmod q$
$C \leftarrow (c, r, s)$
`return` C
Unsigncryption $\mathcal{USC}^{\mathsf{G},\mathsf{H}}_{y_A,x_B}(C)$ by Bob the Recipient
Parse C as (c, r, s); Check whether $r, s \in \mathbb{Z}/q\mathbb{Z}$ and $c \in \mathcal{C}$
`if` (c, r, s) is not in correct spaces
`return` "reject"
`else`
$\omega \leftarrow (y_A g^r)^s \bmod p$; $\kappa \leftarrow \omega^{x_B} \bmod p$; $\tau \leftarrow \mathsf{G}(\kappa)$
$m \leftarrow \mathsf{D}_\tau(c)$
Get $y_B (= g^{x_B} \bmod p)$; $bind \leftarrow y_A || y_B$
`if` $\mathsf{H}(m||bind||\kappa) = r$ `return` m
`else return` "reject"

3 Security Notions for Signcryption Scheme

3.1 Security Notions for Confidentiality of Signcryption

Taking into account all the aspects of confidentiality issues discussed in the first section, we now explain in detail a confidentiality attack model for signcryption, which we call the *Flexible Unsigncryption Oracle (FUO)*-model. In this model, the adversary Eve's goal is to break the confidentiality of messages between Alice and Bob. Eve is given Alice's public key pk_A and Bob's public key pk_B, and has access to Alice's signcryption oracle (with Bob as recipient), as well as a *flexible* unsigncryption oracle, which on input a signcrypted text C, returns output after performing unsigncryption under sender's public key $pk_{A'}$ chosen by Eve at her will (Eve may choose sender's public key as Alice's public key

pk_A, say, $pk_{A'} = pk_A$.) and Bob's private key sk_B. In other words, the flexible unsigncryption oracle is not constrained to be executed only under pk_A and sk_B – Alice's public key can be replaced by the public key generated by Eve. Accordingly, the FUO-model gives Eve the full chosen-ciphertext power with the ability to choose the sender's public key as well as the signcrypted text.

Note, however, that in the FUO-model for signcryption Eve also has access to Alice's signcryption oracle. This can be useful to Eve because Alice's private key, which is involved in the signcryption process, can be exploited to achieve Eve's goal, namely to decrypt signcrypted texts from Alice to Bob. This is an important difference between the FUO attack model for signcryption and the standard chosen-ciphertext attack model for traditional asymmetric encryption schemes (where the attacker can simulate the encryption oracle by himself).

Using the notion of indistinguishability of encryption (also known as semantic security) [11], we now formalize the concept of security against adaptive chosen ciphertext attack for signcryption with respect to the FUO-model. We say a signcryption scheme is secure in the sense of indistinguishability (abbreviated by 'ind'), if there is no polynomial-time adversary that can learn any information about the plaintext from the signcrypted text except for its length. Following a commonly accepted practice, we denote by FUO-IND-CCA2 the security of signcryption against adaptive chosen ciphertext attack with respect to the FUO-model under the indistinguishability notion.

Definition 3 (FUO-IND-CCA2). Let $\mathsf{SC} = (\mathcal{COM}, \mathcal{K}_A, \mathcal{K}_B, \mathcal{SC}, \mathcal{USC})$ be a signcryption scheme. Let A_c be an adversary that conducts adaptive chosen ciphertext attack. A_c is composed of a find-stage algorithm A_1 and a guess-stage algorithm A_2 and has access to random oracles, the signcryption oracle which performs signcryption under the fixed keys x_A and y_B and the flexible unsigncryption oracle. A_c's job is to correctly guess the bit b after making a number of queries to its oracles with restriction that A_2 is not allowed to query the target signcrypted text C to the unsigncryption oracle $\mathcal{USC}^{\mathsf{G},\mathsf{H}}_{y_A,x_B}(\cdot)$ in which the flexible unsigncryption oracle $\mathcal{USC}^{\mathsf{G},\mathsf{H}}_{x_B}(\cdot)$ executes unsigncryption under Alice's public key y_A and Bob's private key x_B. Note that in describing our attack model, the unsigncryption oracle is denoted by $\mathcal{USC}^{\mathsf{G},\mathsf{H}}_{x_B}(\cdot)$, namely, there is no specified sender's public key in the subscript. This is chosen intentionally to highlight that the sender's public key is given more *flexibly* to the unsigncryption oracle (or the recipient). (However, it is important to note that A_c is *allowed* to make the query C to the unsigncryption oracle $\mathcal{USC}^{\mathsf{G},\mathsf{H}}_{y_{A'},x_B}(\cdot)$ where the flexible unsigncryption oracle $\mathcal{USC}^{\mathsf{G},\mathsf{H}}_{x_B}(\cdot)$ performs unsigncryption under the public key $y_{A'}$ which is arbitrarily chosen by A_c and is different from y_A.) Let $k \in \mathbb{N}$ be a security parameter and s be state information. A specification for the experimental algorithm is as follows.

Experiment $\mathbf{Cca2Exp}^{\mathrm{fuo-ind-cca2}}_{\mathsf{SC},A_c}(k)$
$cp_{sc} \leftarrow \mathcal{COM}(k)$
Pick $\mathsf{G} : \{0,1\}^* \rightarrow \{0,1\}^l$ at random
Pick $\mathsf{H} : \{0,1\}^* \rightarrow \mathbb{Z}/q\mathbb{Z}$ at random

$(y_A, x_A) \leftarrow_R \mathcal{K}_A(k, cp_{sc})$
$(y_B, x_B) \leftarrow_R \mathcal{K}_B(k, cp_{sc})$
$(m_0, m_1, s) \leftarrow A_1^{\mathsf{G},\mathsf{H},\mathcal{SC}^{\mathsf{G},\mathsf{H}}_{x_A,y_B}(\cdot),\mathcal{USC}^{\mathsf{G},\mathsf{H}}_{x_B}(\cdot)}(k, \mathsf{find}, y_A, y_B)$
$b \leftarrow_R \{0,1\}$; $C \leftarrow \mathcal{SC}^{\mathsf{G},\mathsf{H}}_{x_A,y_B}(m_b)$
$b' \leftarrow A_2^{\mathsf{G},\mathsf{H},\mathcal{SC}^{\mathsf{G},\mathsf{H}}_{x_A,y_B}(\cdot),\ \mathcal{USC}^{\mathsf{G},\mathsf{H}}_{x_B}(\cdot)}(k, \mathsf{guess}, C, y_A, y_B, s)$
`if` $b' = b$ and C was never queried to $\mathcal{USC}^{\mathsf{G},\mathsf{H}}_{y_A,x_B}(\cdot)$
`return` 1
`else return` 0

Now let $\mathbf{Succ}^{\text{fuo-ind-cca2}}_{\mathsf{SC},A_c}(k) \stackrel{\text{def}}{=} 2\Pr[\mathbf{Cca2Exp}^{\text{fuo-ind-cca2}}_{\mathsf{SC},A_c}(k) = 1] - 1$. Then an advantage function of FUO-IND-CCA2 is defined as follows.

$$\mathbf{Adv}^{\text{fuo-ind-cca2}}_{\mathsf{SC}}(k, t, q_g, q_h, q_{sc}, q_{usc}) \stackrel{\text{def}}{=} \max_{A_c}\{\mathbf{Succ}^{\text{fuo-ind-cca2}}_{\mathsf{SC},A_c}(k)\},$$

where the maximum is taken over all A_c with execution time t. Note that q_{sc} is the number of queries to the signcryption oracle and q_{usc} is the number of queries to the unsigncryption oracle, respectively. Also, note that g_g and q_h are the number of queries to the random oracles G and H, respectively. We say SC is FUO-IND-CCA2 secure if $\mathbf{Adv}^{\text{fuo-ind-cca2}}_{\mathsf{SC}}(k, t, q_g, q_h, q_{sc}, q_{usc})$ is a negligible function in k for any adversary A_c with polynomial time bound in k and whose queries are polynomial in k.

Now we recall the definition of security against chosen plaintext attack for the symmetric encryption [2] used in the signcryption under the notion of indistinguishability.

Definition 4 (IND-CPA for Symmetric Encryption). Let $\mathsf{SC}^{\text{SYM}} = (\mathsf{K}, \mathsf{E}, \mathsf{D})$ be a symmetric encryption scheme. Let A'_p be an adversary for IND-CPA. A'_p is composed of a find-stage algorithm A'_1 and a guess-stage algorithm A'_2. Let $l \in \mathbb{N}$ be a security parameter and s be state information. A specification for the experimental algorithm is as follows.

Experiment $\mathbf{CpaExp}^{\text{ind-cpa}}_{\mathsf{SC}^{\text{SYM}},A'_p}(l)$
$\kappa \leftarrow_R \mathsf{K}(l)$
$(m_0, m_1, s) \leftarrow A_1'^{\mathsf{E}_\kappa(\cdot)}(l, \mathsf{find})$
$b \leftarrow_R \{0,1\}$; $c \leftarrow \mathsf{E}_\kappa(m_b)$
$b' \leftarrow A_2'^{\mathsf{E}_\kappa(\cdot)}(l, \mathsf{guess}, m_0, m_1, c, s)$
`if` $b' = b$ `return` 1 `else return` 0

Now let $\mathbf{Succ}^{\text{ind-cpa}}_{\mathsf{SC}^{\text{SYM}},A'_p}(l) \stackrel{\text{def}}{=} 2\Pr[\mathbf{CpaExp}^{\text{ind-cpa}}_{\mathsf{SC}^{\text{SYM}},A'_p}(l) = 1] - 1$. Then an advantage function of IND-CPA for symmetric encryption is defined as follows.

$$\mathbf{Adv}^{\text{ind-cpa}}_{\mathsf{SC}^{\text{SYM}}}(l, t, q_e) \stackrel{\text{def}}{=} \max_{A'_p}\{\mathbf{Succ}^{\text{ind-cpa}}_{\mathsf{SC}^{\text{SYM}},A'_p}(l)\},$$

where the maximum is taken over all A'_p with execution time t and q_e denotes the number of queries to the encryption oracle, made by A'_p during the attack.

We say $\mathsf{SC}^{\mathrm{SYM}}$ is IND-CPA secure if $\mathbf{Adv}^{\mathrm{ind-cpa}}_{\mathsf{SC}}(l, t, q_e)$ is a negligible function in l for any adversary A'_p whose time complexity is polynomial in l.

3.2 Security Notion for Unforgeability of Signcryption

Following the security notion for unforgeability of signcryption formalized in [21], we define unforgeability of the signcryption scheme SC. Since signcryption offers non-repudiation for the sender Alice, it is essential that even the receiver Bob cannot impersonate Alice and forge valid signcrypted texts from Alice to himself. To ensure that our proof of unforgeability covers this aspect, we allow the forger in our attack model to have access to Bob's private key as well as the corresponding public key. A formal definition is as follows.

Definition 5. An experiment of forgery for SC is realized by the following procedure that takes as input a security parameter $k = |p| \in \mathbb{N}$.

Experiment $\mathbf{ForgeExp}^{\mathrm{cma}}_{\mathsf{SC},F}(k)$
$cp_{sc} \leftarrow \mathcal{COM}(k)$
Pick $\mathsf{G} : \{0,1\}^* \rightarrow \{0,1\}^l$ at random
Pick $\mathsf{H} : \{0,1\}^* \rightarrow \mathbb{Z}/q\mathbb{Z}$ at random
$(y_A, x_A) \leftarrow_R \mathcal{K}_A(k, cp_{sc})$
$(y_B, x_B) \leftarrow_R \mathcal{K}_B(k, cp_{sc})$
`if` $F^{\mathsf{G},\mathsf{H},\mathcal{SC}^{\mathsf{G},\mathsf{H}}_{x_A,y_B}(\cdot)}(y_A, y_B, x_B)$ outputs (m, C) such that
(1) $\mathcal{USC}^{\mathsf{G},\mathsf{H}}_{y_A,x_B}(C) = m$ and
(2) m was never queried to $\mathcal{SC}^{\mathsf{G},\mathsf{H}}_{x_A,y_B}(\cdot)$
`return` 1 `else return` 0

Now let $\mathbf{Succ}^{\mathrm{cma}}_{\mathsf{SC},F}(k) \stackrel{\mathrm{def}}{=} \Pr[\mathbf{ForgeExp}^{\mathrm{cma}}_{\mathsf{SC},F}(k) = 1]$. Then an advantage function of F can be defined as follows.

$$\mathbf{Adv}^{\mathrm{cma}}_{\mathsf{SC}}(k, t, q_g, q_h, q_s) \stackrel{\mathrm{def}}{=} \max_F \{\mathbf{Succ}^{\mathrm{cma}}_{\mathsf{SC},F}(k)\},$$

where the maximum is taken over all F with execution time t and at most q_g, q_h and q_{sc} queries to the random oracles G, H and the signcryption oracle $\mathcal{SC}$, respectively, made by F. We say SC is existentially unforgeable against adaptive chosen message attack if $\mathbf{Adv}^{\mathrm{cma}}_{\mathsf{SC}}(k, t, q_g, q_h, q_{sc})$ is a negligible function in k for any forger F whose time complexity is polynomial in k (Also, its queries are polynomial in k).

4 Security Reductions

4.1 Confidentiality of Signcryption

In this section, we provide a proof of a security reduction that signcryption is FUO-IND-CCA2 secure in the random oracle model, relative to the GDH problem. We show that an adversary A_{gdh} using an adaptive chosen ciphertext

attacker A_c as a subroutine can solve the GDH problem. We assume that A_c is given the signcryption oracle and the flexible unsigncryption oracle described in the previous section. Note that the confidentiality of signcryption against adaptive chosen ciphertext is relative to the GDH problem. This is because, with the help of DDH oracle, the signcryption and unsigncryption oracles are successfully simulated. Now we state the results as a following theorem.

Theorem 1. *If the GDH problem is hard and the symmetric encryption scheme* $\mathsf{SC}^{\mathrm{SYM}}$ *in signcryption* SC *is IND-CPA secure, then* SC *is FUO-IND-CCA2 secure in the random oracle model. Concretely,*

$$\mathbf{Adv}^{\mathrm{fuo-ind-cca2}}_{\mathsf{SC}}(k, t, q_g, q_h, q_{sc}, q_{usc})$$
$$\leq 4\mathbf{Adv}^{\mathrm{invert}}_{\mathsf{GDH}}(k, t_1, q_{ddh}) + \mathbf{Adv}^{\mathrm{ind-cpa}}_{\mathsf{SC}^{\mathrm{SYM}}}(l, t_2, 0) + \frac{q_{sc}(q_g + q_h + 1) + q_{usc}}{2^{l_q(k)-1}}$$

where k *and* l *denote the security parameters,* t *denotes execution time for FUO-IND-CCA2 adversaries,* q_{sc} *and* q_{usc} *denote the number of queries to the signcryption and the unsigncryption oracles, respectively. Here,* $t_1 = O(t + time_g + time_h + time_{sc} + time_{usc})$ *and* $t_2 = O(t_1)$*, where* $time_g (= O(q_g^2 + 1))$*,* $time_h (= O(q_h^2 + 1))$*,* $time_{sc} (= O(k^3))$ *and* $time_{usc} (= O(k^3 + q_{usc})(q_g + q_h + time_d))$ *denote the simulation time for the random oracles* G *and* H*, the signcryption and the unsigncryption oracles, respectively. Here,* $time_d$ *is simulation time for the symmetric decryption function* D*. Also,* q_{ddh} *denotes the number of queries to DDH oracle made by the adversary for the GDH problem and satisfies* $q_{ddh} = O(q_g + q_h + q_{usc})$*.*

Proof 1. Suppose that $k (= |p|)$ is a security parameter. Let p and q be primes such that $q|(p-1)$ and g be element of order q. Let $X = g^x \bmod p$ and $Y = g^{x_B} \bmod p$. Now we construct an adversary A_{gdh} that given (k, p, q, g, X, Y), computes the Diffie-Hellman key $\kappa^* \stackrel{\mathrm{def}}{=} X^{x_B} \bmod p$ with the help of a Decisional Diffie-Hellman (DDH) oracle $DDH_g(\cdot, \cdot, \cdot)$, using FUO-IND-CCA2 adversary A_c. By definition, A_c consists of a find-stage algorithm A_1 and a guess-stage algorithm A_2.

Beginning of the Simulation. At the beginning of the simulation, A_{gdh} chooses a random string α^* for $\mathsf{G}(\kappa^*)$. (Note that A_{gdh} does not know the Diffie-Hellman key κ^* at this stage.) Then, A_{gdh} chooses random strings r^* and s^* from $\mathbb{Z}/q\mathbb{Z}$ and sets $(Xg^{-r^*s^*})^{\frac{1}{s^*}} \bmod p$ as Alice's public key y_A. Also A_{gdh} sets $y_B = Y$.

Simulation of Guess Stage and End of the Simulation. When A_1 outputs two plaintexts m_0 and m_1 after asking queries to the random oracles and signcryption/unsigncryption oracles during the find-stage, A_{gdh} chooses $b \in \{0, 1\}$ at random, computes $c^* = \mathsf{E}_{\alpha^*}(m_b)$. Then it answers A_1 with (c^*, r^*, s^*). When A_2 outputs a bit b' as its guess after asking queries to the random oracles and signcryption/unsigncryption oracles during the guess-stage, A_{gdh} returns κ^* which is a guess for the Diffie-Hellman key $X^{x_B} \bmod p$ and is a preimage of α^*.

Simulation of the Random Oracles. If A_1 or A_2 makes a query κ to its random oracle G, A_{gdh} runs a random simulator G-sim specified below and forwards the answers to A_1 or A_2. Note that two types of "query-answer" lists L_1^{G} and L_2^{G} are maintained for the simulation of the random oracle G, i.e., $L^{\mathsf{G}} = L_1^{\mathsf{G}} \cup L_2^{\mathsf{G}}$. The list L_1^{G} consists of simple input/output entries for G of the form (κ_i, τ_i), where $i \in \mathsf{N}$. But the list L_2^{G} consists of special input/output entries for G which are of the form $\omega_i||(?, \tau_i)$ and implicitly represents the input/output relation $\mathsf{G}(\omega_i^{x_B} \bmod p) = \tau_i$, although the input $\omega_i^{x_B}$ is not explicitly stored and hence is denoted by '?'. New entries are added to L_2^{G} by either signcryption or unsigncryption oracle simulators, which will be specified.

Meanwhile, if A_1 or A_2 makes a query μ to the random oracle H, A_{gdh} runs another random oracle simulator H-sim and answers A_1 or A_2 with the output of H-sim taking μ as input.

Similarly to G-sim, the simulator H-sim also makes use of two input/output lists L_1^{H} and L_2^{H}. The list L_1^{H} consists of simple input/output entries for H of the form (μ_i, r_i). The list L_2^{H} consists of special input/output entries for H which are of the form $\omega_i||(m_i||bind_i||?, \tau_i)$ and implicitly represents the input/output relation $\mathsf{H}(m_i||bind_i||\kappa_i) = \tau_i$, where $\kappa_i = \omega_i^{x_B} \bmod p$ is not explicitly stored and hence is denoted by '?'. New entries are also added to L_2^{H} by either signcryption or unsigncryption oracle simulators. Now we provide complete specifications for G-sim and H-sim.

G-sim(L^{G}, κ)
  if $DDH_g(X, y_B, \kappa)$=1
    then **return** $NULL$
  else if $DDH_g(\omega_i, y_B, \kappa)$=1
    for some $\omega_i||(?, \tau_i) \in L_2^{\mathsf{G}}$
    then **return** τ_i
  else if $\kappa = \kappa_i$ for some
    $(\kappa_i, \tau_i) \in L_1^{\mathsf{G}}$ then **return** τ_i
  else $\tau_i \leftarrow_R \{0,1\}^l$
    then **return** τ_i;
    $\kappa_i \leftarrow \kappa$; Put (κ_i, τ_i) into L_1^{G}

H-sim(L^{H}, μ)
  Parse μ as $m||bind||\kappa$,
    where κ is the rightmost k
    bits of μ
  if $DDH_g(X, y_B, \kappa) = 1$
    then **return** $NULL$
  else if $DDH_g(\omega_i, y_B, \kappa) = 1$
  and $m||bind = m_i||bind_i$ for some
  $\omega_i||(m_i||bind_i||?), r_i) \in L_2^{\mathsf{H}}$
    then **return** r_i
  else if $\mu = \mu_i$ for some
  $(\mu_i, r_i) \in L_1^{\mathsf{H}}$ then **return** r_i
  else $r_i \leftarrow \mathbb{Z}/q\mathbb{Z}$ then **return** r_i;
    $\mu_i \leftarrow \mu$; Put (μ_i, r_i) into L_1^{H}

Simulation of the Signcryption Oracle. When A_1 or A_2 makes a query m to its signcryption oracle $\mathcal{SC}$, A_{gdh} runs a signcryption oracle simulator $\mathcal{SC}$-sim, gets a result from $\mathcal{SC}$-sim and forwards a answer to A_1 or A_2. A specification for $\mathcal{SC}$-sim is given as follows.

$\mathcal{SC}$-sim$(L_2^{\mathsf{G}}, L_2^{\mathsf{H}}, y_A, y_B, m)$
  $\tau \leftarrow_R \{0,1\}^l$; $c \leftarrow \mathsf{E}_\tau(m)$
  $r \leftarrow_R \mathbb{Z}/q\mathbb{Z}$; $s \leftarrow_R \mathbb{Z}/q\mathbb{Z}$
  $\omega \leftarrow (y_A g^r)^s \bmod p$

$bind^* \leftarrow y_A||y_B$
$\omega_i \leftarrow \omega$; $\tau_i \leftarrow \tau$; $m_i \leftarrow m$; $r_i \leftarrow r$
Put $\omega_i||(?, \tau_i)$ into L_2^{G}
Put $(m_i||bind^*||?, r_i)$ into L_2^{H}
$C \leftarrow (c, r, s)$
`return` C

Simulation of the Unsigncryption Oracle. When A_1 or A_2 makes a query $C, \bar{y_A}$ (the flexible public key chosen by the A_c) to its unsigncryption oracle $\mathcal{USC}$, A_{gdh} runs a unsigncryption oracle simulator $\mathcal{USC}$-sim, gets a result from the $\mathcal{USC}$-sim and forwards a answer to A_1 or A_2. The following is a complete specification of $\mathcal{USC}$-sim.

$\mathcal{USC}$-sim$(L^{\mathsf{G}}, L^{\mathsf{H}}, X, \bar{y_A}, y_B, C)$
Parse C as (c, r, s)
$\omega \leftarrow (\bar{y_A} g^r)^s \bmod p$
`if` $\omega = X$ `return` $NULL$
$bind \leftarrow \bar{y_A}||y_B$
`if` there exists $(\kappa_i, \tau_i) \in L_1^{\mathsf{G}}$ such that $DDH_g(\omega, y_B, \kappa_i) = 1$ or
there exists $\omega_i||(?, \tau_i) \in L_2^{\mathsf{G}}$ such that $\omega = \omega_i$
then $\tau' \leftarrow \tau_i$
`else`
$\tau' \leftarrow_R \{0,1\}^l$; $\omega_i \leftarrow \omega$; $\tau_i \leftarrow \tau'$; Put $\omega_i||(?, \tau_i)$ into L_2^{G}
$m \leftarrow \mathsf{D}_{\tau'}(c)$
`if` there exists $(\mu_i, r_i) \in L_1^{\mathsf{H}}$ such that $DDH_g(\omega, y_B, \kappa_i) = 1$,
where $\mu_i = m_i||bind_i||\kappa_i$ and κ_i denotes k rightmost bits of μ_i
or there exists $\omega_i||(m_i||bind_i||?, r_i) \in L_2^{\mathsf{H}}$ such that $\omega = \omega_i$,
$m = m_i$ and $bind = bind_i$ for some r_i
then $r' \leftarrow r_i$
`else`
$\omega_i \leftarrow \omega$; $m_i \leftarrow m$; $bind_i \leftarrow bind$
$r' \leftarrow_R \mathbb{Z}/q\mathbb{Z}$; $r_i \leftarrow r'$; Put $\omega_i||(m_i||bind_i||?, r_i)$ into L_2^{H}
`if` $r = r'$ then `return` m
`else return` $NULL$

Putting It All Together. A complete specification of the adversary A_{gdh} is described as follows. Let s be state information.

Adversary $A_{gdh}(k, p, q, g, X, Y)$
$r^* \leftarrow_R \mathbb{Z}/q\mathbb{Z}$; $s^* \leftarrow_R \mathbb{Z}/q\mathbb{Z}$
$y_A \leftarrow (X g^{-r^* s^*})^{\frac{1}{s^*}} \bmod p$; $y_B \leftarrow Y$; $\alpha^* \leftarrow_R \{0,1\}^l$; $bind^* \leftarrow y_A||y_B$
Run $A_1(k, \mathsf{find}, y_A, y_B)$, using G-sim, H-sim, $\mathcal{SC}$-sim and $\mathcal{USC}$-sim
to simulate answers to queries made by A_1 to its oracles
`if` A_1 queries κ to G such that G-sim$(\kappa) = NULL$
`abort` and `return` κ
`if` A_1 queries μ to H such that H-sim$(\mu) = NULL$

`abort` and `return` κ, where κ is the k rightmost bits of μ
$A_1(k, \mathsf{find}, y_A, y_B)$ outputs (m_0, m_1, s)
$b \leftarrow_R \{0,1\}$; $c^* \leftarrow \mathsf{E}_{\alpha^*}(m_b)$; $C^* \leftarrow (c^*, r^*, s^*)$
Run $A_2(k, \mathsf{guess}, m_0, m_1, C^*, y_A, y_B, s)$, using G-sim, H-sim, $\mathcal{SC}$-sim and $\mathcal{USC}$-sim to simulate answers to queries made by A_1 to its oracles
  `if` A_2 queries κ to G such that $\mathsf{G}\text{-sim}(\kappa) = NULL$
    `abort` and `return` κ
  `if` A_2 queries μ to H such that $\mathsf{H}\text{-sim}(\mu) = NULL$
    `abort` and `return` κ, where κ is the k rightmost bits of μ
$A_2(k, \mathsf{guess}, m_0, m_1, C^*, y_A, y_B, s)$ outputs b'
`return` κ^*

Analysis. Now we analyze our simulation. We consider A_c's execution in both the real attack experiment (*real*) and the GDH attack experiment (*sim*). Below, we define an event called Bad, which causes the joint distribution of A_c's view to differ in experiment *sim* from the distribution of A_c's view in experiment *real*.

For all outcomes of experiment *real* except those in the event Bad, A_c's view is distributed identically in experiments *real* and *sim*. Hence, outcomes in the complementary event $\neg\mathsf{Bad}$ of *real* have the same probability in experiment *sim*, and in particular:

$$\begin{aligned}\Pr[A_c \text{ wins} \wedge \neg\mathsf{Bad}]_{sim} &= \Pr[A_c \text{ wins} \wedge \neg\mathsf{Bad}]_{real} \\ &\geq \Pr[A_c \text{ wins}]_{real} - \Pr[\mathsf{Bad}]_{real} \\ &\geq \frac{1}{2} + \frac{1}{2}\mathbf{Succ}^{\text{fuo-ind-cca2}}_{\mathsf{SC},A_c}(k) - \Pr[\mathsf{Bad}]_{real} \qquad (1)\end{aligned}$$

Now we define an event GDHBrk in the experiment *sim* as follows:

- GDHBrk: A_c asks the Diffie-Hellman key $\kappa^* = X^{x_B} \bmod p$ to G-sim or A_c asks μ^* to H-sim such that κ^*, where κ^* is the k rightmost bits of μ^*.

Observe that if GDHBrk occurs then A_{gdh} will return the correct solution κ^* to the GDH instance that it is trying to compute. Hence, splitting $\Pr[A_c \text{ wins} \wedge \neg\mathsf{Bad}]_{sim} = \Pr[A_c \text{ wins} \wedge \neg\mathsf{Bad} \wedge \mathsf{GDHBrk}]_{sim} + \Pr[A_c \text{ wins} \wedge \neg\mathsf{Bad} \wedge \neg\mathsf{GDHBrk}]_{sim} \leq \mathbf{Succ}^{\text{invert}}_{\mathsf{GDH},A_{gdh}}(k) + \Pr[A_c \text{ wins} \wedge \neg\mathsf{Bad} \wedge \neg\mathsf{GDHBrk}]_{sim}$, and substituting in (1) we get:

$$\begin{aligned}\mathbf{Succ}^{\text{invert}}_{\mathsf{GDH},A_{gdh}}(k) + \Pr[A_c \text{ wins} \wedge \neg\mathsf{Bad} \wedge \neg\mathsf{GDHBrk}]_{sim} \geq \\ \frac{1}{2} + \frac{1}{2}\mathbf{Succ}^{\text{fuo-ind-cca2}}_{\mathsf{SC},A_c}(k) - \Pr[\mathsf{Bad}]_{real}. \qquad (2)\end{aligned}$$

Since $\Pr[\mathsf{Bad}]_{real} = \Pr[\mathsf{Bad} \wedge (\neg\mathsf{GDHBrk} \vee \mathsf{GDHBrk})]_{real} \leq \Pr[\mathsf{Bad} \wedge \neg\mathsf{GDHBrk}]_{real} + \Pr[\mathsf{GDHBrk}]_{real} \leq \Pr[\mathsf{Bad} \wedge \neg\mathsf{GDHBrk}]_{real} + \mathbf{Succ}^{\text{invert}}_{\mathsf{GDH},A_{gdh}}(k)$, we have

$$\begin{aligned}2\mathbf{Succ}^{\text{invert}}_{\mathsf{GDH},A_{gdh}}(k) + \Pr[A_c \text{ wins} \wedge \neg\mathsf{Bad} \wedge \neg\mathsf{GDHBrk}]_{sim} \geq \\ \frac{1}{2} + \frac{1}{2}\mathbf{Succ}^{\text{fuo-ind-cca2}}_{\mathsf{SC},A_c}(k) - \Pr[\mathsf{Bad} \wedge \neg\mathsf{GDHBrk}]_{real}. \qquad (3)\end{aligned}$$

In the remaining part of the proof we upper bound two terms in (3) as follows:

$$\Pr[\mathsf{Bad} \wedge \neg\mathsf{GDHBrk}]_{real} \leq \frac{q_{sc}(q_g + q_h + 1) + q_{usc}}{2^{l_q(k)}} \tag{4}$$

and

$$\Pr[A_c \text{ wins} \wedge \neg\mathsf{Bad} \wedge \neg\mathsf{GDHBrk}]_{sim} \leq \frac{1}{2} + \frac{1}{2}\mathbf{Adv}^{\text{ind-cpa}}_{\mathsf{SC}^{\text{SYM}}}(l, t_2, 0) \tag{5}$$

The advantage bound claim of the theorem follows upon substitution of (4) and (5) in (3), and taking maximums over all GDH adversaries with the appropriate resource parameters. The running time counts can be readily checked. Hence it remains to establish the bounds (4) and (5), which will be done below.

First, to establish (4), we upper bound the probability $\Pr[\mathsf{Bad} \wedge \neg\mathsf{GDHBrk}]_{real}$ of outcomes when the view of A_c during the real attack differs from its view during the simulation. It is easy to see that the inputs to A_c are identically distributed in both *real* and *sim*. But errors can occur in simulating answers to A_c's queries to its oracles G, H, $\mathcal{SC}$ and $\mathcal{USC}$. Accordingly, we split $\mathsf{Bad} \wedge \neg\mathsf{GDHBrk} = \mathsf{GBad} \vee \mathsf{HBad} \vee \mathsf{USCBad} \vee \mathsf{SCBad}$ into a union of bad outcomes in simulating each of the oracles. We bound each as follows and then add up the bounds using the union bound.

Signcryption Oracle Simulation Error. Notice that the signcryption oracle simulator chooses r and s independently and uniformly in $\mathbb{Z}/q\mathbb{Z}$ and τ independently and uniformly in $\{0,1\}^l$ and, then (defining $\omega = (y_A g^r)^s \bmod p$ and $\kappa = \omega^{x_B} \bmod p$) forces the following input-output pairs for the random oracles G and H: $\mathsf{G}(\kappa) = \tau$ and $\mathsf{H}(m||bind(= y_A||y_B)||\kappa) = r$. Due to randomness of the random oracles, this results in the same signcryptext distribution in *sim* as in *real*, if the images of H and G at the above points have not already been fixed due to earlier queries. But outcomes in *real* when the input to H or G has already been fixed cause a simulation error in *sim*.

Let $(\tilde{c}, \tilde{r}, \tilde{s})$ denote an output of the real signcryption oracle for each single query $\tilde{m}$. Namely, $(\tilde{c}, \tilde{r}, \tilde{s}) = (\mathsf{E}_{\tilde{\tau}}(\tilde{m}), \mathsf{H}(\tilde{m}||bind||y_B^{\tilde{x}} \bmod p), \tilde{x}/(\tilde{r} + x_A) \bmod q)$, where $\tilde{\tau} = \mathsf{G}(y_B^{\tilde{x}} \bmod p)$. Now we define the following events.

- E: $y_B^{\tilde{x}} \bmod p \in [L^{\mathsf{G}}]_{in} \cup [L^{\mathsf{H}}]_{in'} \cup \{\kappa^*(= y_B^x \bmod p)\}$.

Here, $[L^{\mathsf{G}}]_{in}$ is a set of all the inputs to the random oracle G, which exists in the list L^{G}. Also, $[L^{\mathsf{H}}]_{in'}$ is a set of all rightmost k bits of the inputs which exists in the list L^{H}. Thanks to uniform distribution of $y_B^{\tilde{x}} \bmod p$ in the group, we have $\Pr[\mathsf{E}]_{real} \leq \frac{q_g + q_h + 1}{2^{l_q(k)}}$ for each signcryption oracle query.

Since there are up to q_{sc} signcryption queries, the total probability of outcomes in *real* leading to signcryption oracle simulation error is bounded as:

$$\Pr[\mathsf{SCBad}] \leq q_{sc}\Big(\frac{q_g + q_h + 1}{2^{l_q(k)}}\Big). \tag{6}$$

H-*Simulation Error.* The only event which can cause an error in simulating the random oracle H is the GDHBrk. Since $\mathsf{HBad} \subseteq \neg\mathsf{GDHBrk}$, we have $\Pr[\mathsf{HBad}] = 0$.

G-*Simulation Error.* Thanks to the fixed-input DDH oracle available to A_{gdh}, the random oracle G is perfectly simulated for any query, hence we have we have $\Pr[\mathsf{GBad}] = 0$.

Unsigncryption Oracle Simulation Error. Let USCBad be an event that unsigncryption simulation error occurs during the execution of A_c. Then we will bound $\mathsf{USCBad} \wedge \neg\mathsf{SCBad}$. Note that $(\mathsf{USCBad} \wedge \neg\mathsf{SCBad}) \subseteq \neg\mathsf{SCBad}$ and $(\mathsf{USCBad} \wedge \neg\mathsf{SCBad}) \subseteq \neg\mathsf{GDHBrk}$ since $\mathsf{USCBad} \subseteq \neg\mathsf{GDHBrk}$. Note also that the event $\mathsf{USCBad} \wedge \neg\mathsf{SCBad}$ is specified as follows.

– $\mathsf{USCBad} \wedge \neg\mathsf{SCBad}$: A_c queries signcryptext $(y_{bad}, c_{bad}, r_{bad}, s_{bad})$ to the unsigncryption oracle $\mathcal{USC}$ such that

(U.1) $\omega_{bad} = X$, where $\omega_{bad} = (y_{bad} g^{r_{bad}})^{s_{bad}} \bmod p$ and

(U.2) $\kappa^* (= X^{x_B} \bmod p) \notin L_{in}^{\mathsf{G}} \cup L_{in'}^{\mathsf{H}}$ and

(U.3) $r_{bad} = \mathsf{H}(m_{bad}||y_{bad}||y_B||\kappa^*)$ and

(U.4) $m_{bad}||y_{bad}||y_B||\kappa^* \neq m_b||y_A||y_B||\kappa^*$

We remark that if (U.1) does not occur then there is no difference between $\mathcal{USC}$ and $\mathcal{USC}$-sim. Also (U.2) must hold, otherwise SCBad or GDHBrk happens. (U.3) must occur or else both $\mathcal{USC}$ and $\mathcal{USC}$-sim reject (namely, there is no difference between $\mathcal{USC}$ and $\mathcal{USC}$-sim). Finally, we establish (U.4) in the following claim.

Claim 1: $m_{bad}||y_{bad}||y_B||\kappa^* \neq m_b||y_A||y_B||\kappa^*$, i.e. the query to H by the unsigncryption oracle during unsigncryption of $(y_{bad}, c_{bad}, r_{bad}, s_{bad})$ is not the one used to create r^* in the challenge signcryptext (y_A, c^*, r^*, s^*).

proof: Suppose the contrary, i.e. that $m_{bad}||y_{bad}||y_B||\kappa = m_b||y_A||y_B||\kappa$. Then we have: (C.1) $y_{bad} = y_A$ and (C.2) $m_{bad} = \mathsf{D}_{\alpha^*}(c_{bad}) = \mathsf{D}_{\alpha^*}(c^*) = m_b$, and (C.3) $r_{bad} = r^*$ using (U.3) above. From (C.2) and the assumption that $\mathsf{D}_\alpha(\cdot)$ is one-to-one for any key α, we have (C.4) $c_{bad} = c^*$. Finally, since $\omega_{bad} = (y_{bad} g^{r_{bad}})^{s_{bad}} \bmod p = X = (y_A g^{r^*})^{s^*} \bmod p$, then using (C.1) and (C.4) and the fact that $y_A g^{r^*} \in < g >$ has order q (since $\mathrm{Ord}(g) = q$ is prime), we conclude that $s_{bad} = s^* \bmod q$ and since $(y_{bad}, c_{bad}, r_{bad}, s_{bad})$ was accepted, $s_{bad} \in \mathbb{Z}/q\mathbb{Z}$ so (C.5) $s_{bad} = s$. Combining (C.1), (C.3), (C.4), and (C.5), we arrive at the conclusion that $(y_{bad}, c_{bad}, r_{bad}, s_{bad})$ is equal to the challenge signcryptext, which is impossible since A_c is not allowed to query the challenge. □

For each signcryptext $(y_{bad}, c_{bad}, r_{bad}, s_{bad})$ queried to USC, $\Pr[(U.3)|(U.1) \wedge (U.2) \wedge (U.4)]_{real} = \frac{1}{2^{l_q(k)}}$ because $\mathsf{H}(\mu_{bad})$ is uniformly distributed in $\mathbb{Z}/q\mathbb{Z}$ and independent of r_{bad}, where $\mu_{bad} \stackrel{\text{def}}{=} m_{bad}||y_{bad}||y_B||\kappa^*$, hence $\Pr[(U.1) \wedge (U.2) \wedge (U.3) \wedge (U.4)]_{real} \leq \frac{1}{2^{l_q(k)}}$.

Since A_c makes up to q_{usc} queries to $\mathcal{USC}$ we obtain

$$\Pr[\mathsf{USCBad} \wedge \neg\mathsf{SCBad}]_{real} \leq \frac{q_{usc}}{2^{l_q(k)}}. \tag{7}$$

Adding up (6) and (7) we obtain the desired bound (4).

To complete the proof it remains to deduce the second bound (5) on the probability $\Pr[A_c \text{ wins} \wedge \neg\mathsf{Bad} \wedge \neg\mathsf{GDHBrk}]_{sim}$. We do this by constructing an adversary A'_p against the IND-CPA of the symmetric encryption scheme $\mathsf{SC}^{\mathrm{SYM}}$

used in the signcryption scheme, and show that its probability of winning the 'IND-CPA' experiment sim' is at least $\Pr[A_c \text{ wins} \wedge \neg\mathsf{Bad} \wedge \neg\mathsf{GDHBrk}]_{sim}$.

Now we construct the adversary $A'_p = (A'_1, A'_2)$ using A_{gdh} (which in turn makes use of the adversary A_c to achieve its goal). Let s' be state information. Now a specification follows.

Adversary $A'_1(l, \mathsf{find})$
Find k corresponding to l
$r^* \leftarrow_R \mathbb{Z}/q\mathbb{Z}$; $s^* \leftarrow_R \mathbb{Z}/q\mathbb{Z}$; $x \leftarrow_R \mathbb{Z}/q\mathbb{Z}$; $X \leftarrow g^x \bmod p$
$y_A \leftarrow (Xg^{-r^*s^*})^{\frac{1}{s^*}} \bmod p$; $x_B \leftarrow_R \mathbb{Z}/q\mathbb{Z}$; $y_B \leftarrow g^{x_B} \bmod p$
Run $A_1(k, \mathsf{find}, y_A, y_B)$, using G-sim, H-sim, $\mathcal{SC}$-sim and $\mathcal{USC}$-sim
to simulate answers to queries made by A_1 to its oracles
`if` A_1 queries κ to G such that $\mathsf{G}\text{-sim}(\kappa) = NULL$
`abort` and `return` κ
`if` A_1 queries μ to H such that $\mathsf{H}\text{-sim}(\mu) = NULL$
`abort` and `return` κ where κ is the k rightmost bits of μ
$A_1(k, \mathsf{find}, y_A, y_B)$ outputs (m_0, m_1, s)
$s' \leftarrow s||k||r^*||s^*||y_A||y_B||L^{\mathsf{G}}||L^{\mathsf{H}}$
`return` (m_0, m_1, s')

Outside the view of A'_p, a random bit $b \in \{0,1\}$ and a random key $\alpha \in \{0,1\}^l$ are chosen and $c^* = \mathsf{E}_\alpha(m_b)$ is computed. Then (m_0, m_1, c^*, s') is provided as an input to A'_2.

Adversary $A'_2(l, \mathsf{guess}, m_0, m_1, c, s')$
Retrieve $s||k||r^*||s^*||y_A||y_B||L^{\mathsf{G}}||L^{\mathsf{H}}$ from s'
$C^* \leftarrow c^*||r^*||s^*$
Run $A_2(k, \mathsf{guess}, m_0, m_1, C^*, y_A, y_B, C^*, s)$, using G-sim, H-sim,
$\mathcal{SC}$-sim and $\mathcal{USC}$-sim to simulate answers to
queries made by A_2 to its oracles
`if` A_2 queries κ to G such that $\mathsf{G}\text{-sim}(\kappa) = NULL$
`abort` and `return` κ
`if` A_2 queries μ to H such that $\mathsf{H}\text{-sim}(\mu) = NULL$
`abort` and `return` κ where κ is the k rightmost bits of μ
$A_2(k, \mathsf{guess}, (m_0, m_1), C^*, y_A, y_B, s)$ outputs b'
`return` b'

Now observe the following properties of A'_p: (P.1) A'_p makes no queries to its symmetric encryption oracle. (P.2) If event A_c wins $\wedge \neg\mathsf{Bad} \wedge \neg\mathsf{GDHBrk}$ occurs, then A_c's view is identical in both sim and sim' (P.3) If A_c wins $\wedge\neg\mathsf{Bad} \wedge \neg\mathsf{GDHBrk}$ occurs in sim' then A'_p wins.

Combining (P.1), (P.2) and (P.3) we get the desired bound $\Pr[\mathsf{A_c} \text{ wins} \wedge \neg\mathsf{Bad} \wedge \neg\mathsf{SDHBrk}]_{sim} \leq \frac{1}{2} + \frac{1}{2}\mathbf{Succ}^{\text{ind-cpa}}_{\mathsf{SC}^{\text{SYM}}, A'_p}(l) \leq \frac{1}{2} + \frac{1}{2}\mathbf{Adv}^{\text{ind-cpa}}_{\mathsf{SC}^{\text{SYM}}}(l, t_2, 0)$, which establishes (5) and completes the proof. □

4.2 Unforgeability of Signcryption

In this section, we state our results on unforgeability of signcryption. Due to lack of space, all the proofs for showing signcryption SC is existentially unforgeable against adaptive chosen message attack [12] are omitted in this version of the paper. The basic idea of proofs is to use the ID reduction technique [15].

Theorem 2. *If the signcryption scheme* SC *is forged with* q_s, q_g *and* q_h *queries to the signcryption oracle* $\mathcal{SC}$ *and the random oracles* G *and* H, *respectively, within execution time* t, *then the discrete logarithm of the sender's public key* $y_A = g^{x_A} \bmod p$ *can be found with the following bound.*

$$\mathbf{Adv}^{\mathrm{cma}}_{\mathsf{SC}}(k, t, q_g, q_h, q_{sc}) \leq 2q_h\big(\mathbf{Adv}^{\mathrm{search}}_{\mathsf{DLP}}(k, t^*)\big)^{\frac{1}{2}} + \frac{1}{2^{l_q(k)}}$$

where execution time $t^* = O(t + time_{sc} + time_v + time_c)$. *Note that* $time_{sc}$ *is the simulation time of* q_{sc} *signcryptexts* $time_v$ *and* $time_c$ *and denote the time for verification in* IDSC *which is an identification scheme derived from* SC *and the calculation time of* $x_A \bmod q$, *respectively.*

5 Conclusions

We have proved the confidentiality of Zheng's original signcryption scheme with respect to a strong well-defined security notion similar to the well known 'IND-CCA2' notion defined for standard public-key encryption schemes. Our confidentiality notion is even stronger than the direct adaptation of 'IND-CCA2' to the setting of signcryption, since we allow the attacker to query the signcryption oracle, as well as the unsigncryption oracle. We have also proved the unforgeability of signcryption in a strong sense, namely existential unforgeability against adaptive chosen message attack. Currently we are working on strengthening our confidentiality result even further by allowing the attacker to have 'flexible access' to the signcryption oracle, i.e., the ability to specify an arbitrary recipient's public key in signcryption queries. We will call this new model *Flexible Signcryption Oracle (FSO)*-model. We are also working on extending results presented in this paper to prove the security of various other signcryption schemes proposed in [21] and [25]. We leave technical details for the on-going work to future papers.

Acknowledgement

The authors would like to thank anonymous referees for their helpful comments. The first two authors would also like to thank Dr Jan Newmarch from Monash University for his support and encouragement.

References

1. J. An : *Authenticated Encryption in the Public-Key Setting: Security Notions and Analyses*, available at `http://eprint.iacr.org/`.

2. M. Bellare, A. Desai, E. Jokipii and P. Rogaway: *A Concrete Security Treament of Symmetric Encryption*, Proceedings of FOCS '97, IEEE Computer Society Press, 1997, pages 394–403.
3. M. Bellare, A. Desai, D. Pointcheval and P. Rogaway: *Relations Among Notions of Security for Public-Key Encryption Schemes*, Advances in Cryptology - Proceedings of CRYPTO '98, Vol. 1462 of LNCS, Springer-Verlag 1998, pages 26–45.
4. M. Bellare and C. Namprepre: *Authenticated Encryption: Relations among notions and analysis of the generic composition paradigm*, Advances in Cryptology - Proceedings of ASIACRYPT 2000, Vol. 1976 of LNCS, Springer-Verlag 2000, pages 531–545.
5. M. Bellare and P. Rogaway: *Optimal asymmetric encryption*, Advances in Cryptology -Proceedings of Eurocrypt '94, Vol. 950 of LNCS, Springer-Verlag 1994, pages 92–111.
6. M. Bellare and P. Rogaway: *Random Oracles are Practical: A Paradigm for Designing Efficient Protocols*, Proceedings of First ACM Conference on Computer and Communications Security 1993, pages 62–73.
7. R. Cramer and V. Shoup: *A Practical Public Key Cryptosystem Provably Secure against Adaptive Chosen Ciphertext Attack*, Advances in Cryptology - Proceedings of CRYPTO '98, Vol. 1462 of LNCS, Springer-Verlag 1998, pages 13–25.
8. T. ElGamal: *A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms*, IEEE Trans. Information Theory, 31, 1985, pages 469–472.
9. A. Frier, P. Karlton and P. Kocher: *The SSL 3.0 Protocol*, Netscape Communications Corp., 1996, available at `http://home.netscape.com/eng/ssl3/ssl.toc.html`.
10. E. Fujisaki and T. Okamoto: *How to Enhance the Security of Public-Key Encryption at Minimum Cost*, Proceedings of Public Key Cryptography '99 (PKC '99), Vol. 1666 of LNCS, Springer-Verlag 1999, pages 53–68.
11. S. Goldwasser and S. Micali: *Probabilistic Encryption*, Journal of Computer and System Sciences, Vol. 28, 1984, pages 270–299.
12. S. Goldwasser, S. Micali and R. Rivest: *A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks*, SIAM Journal on Computing, 17, 2, 1988, pages 281–308.
13. S. Kent and R. Atkinson: *IP Encapsulating Security Payload (ESP)*, RFC 2406, 1998.
14. H. Krawczyk: *The Order Of Encryption And Authentication For Protecting Communications (Or: How Secure Is SSL?)*, Advances in Cryptology - Proceedings of CRYPTO 2001, Vol. 2139 of LNCS, Springer-Verlag 2001, pages 310–331.
15. K. Ohta and T. Okamoto: *On Concrete Security Treatment of Signatures Derived from Identification*, Advances in Cryptology - Proceedings of CRYPTO '98, Vol. 1462 of LNCS, Springer-Verlag 1998, pages 354–369.
16. T. Okamoto and D. Pointcheval: *The Gap-Problems: A New Class of Problems for the Security of Cryptographic Schemes*, Proceedings of Public Key Cryptography 2001 (PKC 2001), Vol. 1992 of LNCS, Springer-Verlag 2001, pages 104–118.
17. D. Pointcheval: *Chosen-Ciphertext Security for Any One-Way Cryptosystem*, Proceedings of Public Key Cryptography 2000 (PKC 2000), Vol. 1751 of LNCS, Springer-Verlag 2000, pages 129–146.
18. D. Pointcheval and J. Stern: *Security Arguments for Digital Signatures and Blind Signatures*, Jornal of Cryptology, Vol. 13-Number 3, Springer-Verlag 2000, pages 361-396.

19. C. P. Schnorr: *Efficient Identification and Signatures for Smart Cards*, Advances in Cryptology - Proceedings of CRYPTO '89, Vol. 435 of LNCS, Springer-Verlag 1990, pages 235–251.
20. C. P. Schnorr and M. Jakobsson: *Security of Signed ElGamal Encryption* , Advances in Cryptology - Proceedings of ASIACRYPT 2000, Vol. 1976 of LNCS, Springer-Verlag 2000, pages 73–89.
21. R. Steinfeld and Y. Zheng: *A Signcryption Scheme Based on Integer Factorization*, Proceedings of Information Security Workshop 2000 (ISW2000), Vol. 1975 of LNCS, Springer-Verlag 2000, pages 308–322.
22. Y. Tsiounis and M. Yung: *On the Security of ElGamal-Based Encryption*, Proceedings of Public Key Cryptography '98 (PKC '98), Vol. 1431 of LNCS, Springer-Verlag 1998, pages 117–134.
23. Y. Zheng: *Digital Signcryption or How to Achieve Cost (Signature & Encryption) ≪ Cost (Signature) + Cost (Encryption)*, Advances in Cryptology - Proceedings CRYPTO '97, Vol. 1294 of LNCS, Springer-Verlag 1997, pages 165–179.
24. Y. Zheng: *Digital Signcryption or How to Achieve Cost (Signature & Encryption) ≪ Cost (Signature) + Cost (Encryption)*, full version, available at `http://www.pscit.monash.edu.au/ yuliang/pubs/`.
25. Y. Zheng: *Identification, Signature and Signcryption Using High Order Residues Modulo an RSA Composite*, Proceedings of Public Key Cryptography 2001 (PKC 2001), Vol. 1992 of LNCS, Springer-Verlag 2001, pages 48–63.
26. Y. Zheng and J. Seberry: *Immunizing public key cryptosystems against chosen ciphertext attacks*, the Special Issue on Secure Communications, IEEE Journal on Selected Areas in Communications, Vol. 11, No. 5, 1993, pages 715–724.