

Signcryption in Hierarchical Identity Based Cryptosystem

Sherman S.M. Chow^{1*}, Tsz Hon Yuen², Lucas C.K. Hui¹, and S.M. Yiu¹

¹ Department of Computer Science
The University of Hong Kong
Hong Kong

{smchow, hui, smyiu}@cs.hku.hk

² Department of Information Engineering
The Chinese University of Hong Kong
Hong Kong

thyuen4@ie.cuhk.edu.hk

Abstract. In many situations we want to enjoy confidentiality, authenticity and non-repudiation of message simultaneously. A traditional approach to achieve this objective is to “sign-then-encrypt” the message, or we can employ special cryptographic scheme like signcryption. Two open problems about ID-based signcryption were proposed in [15]. The first one is to devise an efficient forward-secure signcryption scheme with public verifiability and public ciphertext authenticity, which is promptly closed by [10]. Another one which still remains open is to devise a hierarchical ID-based signcryption scheme that allows the user to receive signcrypted messages from sender who is under another sub-tree of the hierarchy. This paper aims at solving this problem by proposing two concrete constructions of hierarchical ID-based signcryption.

Key words: Data security, hierarchical identity-based signcryption, bilinear pairings

1 Introduction

In traditional public key infrastructure, certificates leak data and are not easily located. Strict online requirement removes offline capability, and validating policy is time-consuming and difficult to administer. Traditional PKI may not provide a good solution in many scenarios. For example, in tetherless computing architecture (TCA)[20] where two mobile hosts wanting to communicate might be disconnected from each other and also from the Internet. As exchange of public keys is impossible in this disconnected situation, ID-based cryptosystem fits in very well since the public key can be derived from the identity of another party.

In many situations we want to enjoy confidentiality, authenticity and non-repudiation of message simultaneously. A traditional approach to achieve this objective is to “sign-then-encrypt” the message, or we can employ special cryptographic scheme like signcryption. A recent direction is to merge the concept of ID-based cryptography and signcryption. Two open problems about ID-based signcryption were proposed in [15]. The first one is to devise an efficient forward-secure signcryption scheme with public verifiability and public ciphertext authenticity, which is promptly closed by [10]. Another one which still remains open is to devise a hierarchical ID-based signcryption scheme that allows the user to receive signcrypted messages from sender who is under another sub-tree of the hierarchy. This paper aims at solving this problem.

As shown by [7], the integrity checking necessary for security against adaptive adversaries can be obtained from the signature. We employ the same idea here to proposed two concrete constructions of hierarchical ID-based signcryption.

* corresponding author

1.1 Applications

Identity-based cryptography is suitable for the use of commercial organizations. The inherent key-escrow of property is indeed beneficial in commercial organizations, where the big boss has the power to monitor his/her employees' Internet communications if necessary. Hierarchical structure is common in nowadays' organizations, single trusted authority for generation of private key and authentication of users is simply impractical, all these give rise to the hierarchical ID-based cryptosystem.

Moreover, hierarchical ID-based cryptosystem is also useful in other scenarios, such as in TCA, a computing architecture with the concept of "regions", which can be viewed as a branch of the hierarchy[14].

1.2 Related Work

Malone-Lee gave the first ID-based signcryption scheme [17]. This scheme is not semantically secure as the signcrypted text produced is a concatenation of a signature by a variant of Hess's ID-based signature [13] and a ciphertext by a simplified version of Boneh and Franklin's ID-based encryption [4]. In short, the signature of the message is visible in the signcrypted message.

On the other hand, Nalla and Reddy's ID-based signcryption scheme [19] cannot provide public verifiability as well as public ciphertext authenticity since the verification can only be done with the knowledge of recipient's private key. Libert and Quisquater proposed three ID-based signcryption schemes [15]. None of them can satisfy the requirements for public verifiability and forward security at the same time.

Boyen's multipurpose ID-based signcryption scheme [5] is the first scheme that provides public verifiability and forward security and is also provably secure. However, this scheme aims at providing ciphertext unlinkability and anonymity. So, a third party cannot verify the origin of the ciphertext, thus the scheme does not satisfy the requirement of public ciphertext authenticity. We remark that Boyen's scheme is very useful in applications that require unlinkability and anonymity.

The public verifiability of the signcrypted message usually can only be checked with some ephemeral data computed by the intended recipient of the signcrypted message. The notion of verifiable pairing was introduced in [8] to ensure the non-repudiation property of the ID-based signcryption by disallowing the intended recipient to manipulate the ephemeral data.

In 2004, [18] claimed that they were the first one closing the open problem proposed by [15]; however, the open problem was indeed closed by [10] in 2003. Recently, a simple but secure ID-based signcryption scheme was proposed in [7] and another ID-based signcryption scheme was proposed in [16]. The first blind ID-based signcryption scheme with the security model was proposed in [21]. This scheme offers the option to choose between authenticated encryption and ciphertext unlinkability. Moreover, the generic group and pairing model was introduced in this paper. Notice that none of the previously mentioned schemes works with hierarchical ID-based cryptosystem.

2 Preliminaries

Before presenting our results, we give the definition of a hierarchical ID-based signcryption scheme by extending the framework in previous work (e.g. [10, 21]). We also review the definitions of groups equipped with a bilinear pairing and the related complexity assumptions.

2.1 Framework of Hierarchical ID-based Signcryption Schemes

An ID-based signcryption (IDSC) scheme consists of six algorithms: **Setup**, **Extract**, **Sign**, **Encrypt**, **Decrypt** and **Verify**. **Setup** and **Extract** are executed by the private key generators (PKGs henceforth). Based on the security level parameter, **Setup** is executed to generate the master secret and common public parameters. **Extract** is used to generate the private key for any given identity. The algorithm **Sign** is used to produce the signature of a signer on a message, it also outputs some ephemeral data; **Encrypt** takes the message, the signature, the ephemeral data and the recipient's identity to produce a signcrypted text. **Decrypt** takes the input of secret key and decrypt the message and the corresponding signature, finally **Verify** is used by any party to verify the signature of a message.

In the hierarchical ID-based signcryption (HIDSC henceforth), PKGs are arranged in a tree structure, the identities of users (and PKGs) can be represented as vectors. A vector of dimension ℓ represents an identity at depth ℓ . Each identity ID of depth ℓ is represented as an ID-tuple $ID|\ell = \{ID_1, \dots, ID_\ell\}$. The algorithms of HIDSC have similar functions to that of IDSC except that the **Extract** algorithm in HIDSC will generate the private key for a given identity which is either a normal user or a lower level PKG. The private key for identity ID of depth ℓ is denoted as $S_{ID|\ell}$ or S_{ID} if the depth of ID is not important. The functions of **Setup**, **Extract**, **Sign**, **Encrypt**, **Decrypt** and **Verify** in HIDSC are described as follows.

- **Setup**: Based on the input of an unary string 1^k where k is a security parameter, it outputs the common public parameters $params$, which include a description of a finite message space together with a description of a finite signature space; and the master secret s , which is kept secret by the private key generator.
- **Extract**: Based on the input of an arbitrary identity ID , it makes use of the master secret s (for root PKG) or $S_{ID|j-1}$ (for lower level PKGs) if ID is of depth j to output the private key $S_{ID|j}$ for ID corresponding to $params$.
- **Sign**: Based on the input (m, S_{ID}) , it outputs a signature σ and some ephemeral data r , corresponding to $params$.
- **Encrypt**: Based on the input $(m, S_A, ID_B, \sigma, r)$, it outputs a signcrypted message c .
- **Decrypt**: Based on the input (c, S_B, ID_B) , it outputs the message m and the corresponding signature σ .
- **Verify**: Based on the input (σ, m, ID) , it outputs \top for “true” or \perp for “false”, depending on whether σ is a valid signature of message m signed by ID or not, corresponding to $params$.

These algorithms must satisfy the standard consistency constraint of hierarchical ID-based signcryption, i.e. if $\{\sigma, r\} = \text{Sign}(m, S_A)$ and $C = \text{Encrypt}(S_A, ID_B, m, \sigma, r)$, we must have $\{m', ID_{A'}, \sigma'\} = \text{Decrypt}(c, S_B)$, $m = m'$, $ID_A = ID_{A'}$ and $\top = \text{Verify}(\sigma', m, ID_A)$.

2.2 Bilinear Pairing

Let (\mathbb{G}, \cdot) and (\mathbb{G}_1, \cdot) be two cyclic groups of prime order p and g be a generator of \mathbb{G} . The bilinear pairing is given as $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_1$, which satisfies the following properties:

1. *Bilinearity*: For all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}$, $\hat{e}(u^a, v^b) = \hat{e}(u, v)^{ab}$.
2. *Non-degeneracy*: $\hat{e}(g, g) \neq 1$.
3. *Computability*: There exists an efficient algorithm to compute $\hat{e}(u, v) \forall u, v \in \mathbb{G}$.

2.3 Diffie-Hellman Problems

Definition 1. *The computational Diffie-Hellman problem (CDHP) in \mathbb{G} is defined as follows: Given a 3-tuple $(g, g^a, g^b) \in \mathbb{G}^3$, compute $g^{ab} \in \mathbb{G}$. We say that the (t, ϵ) -CDH assumption holds in \mathbb{G} if no t -time algorithm has advantage at least ϵ in solving the CDHP in \mathbb{G} .*

Definition 2. *The bilinear Diffie-Hellman problem (BDHP) in \mathbb{G} is defined as follows: Given a 4-tuple $(g, g^a, g^b, g^c) \in \mathbb{G}^4$ and a pairing $\hat{e}(\cdot, \cdot)$, compute $\hat{e}(g, g)^{abc} \in \mathbb{G}_1$. We say that the (t, ϵ) -BDH assumption holds in \mathbb{G} if no t -time algorithm has advantage at least ϵ in solving the BDHP in \mathbb{G} .*

Definition 3. *The decisional bilinear Diffie-Hellman problem (DBDHP) in \mathbb{G} is defined as follows: Given a 5-tuple $(g, g^a, g^b, g^c, T) \in \mathbb{G}^4 \times \mathbb{G}_1$ and a pairing $\hat{e}(\cdot, \cdot)$, decides whether $T = \hat{e}(g, g)^{abc}$. We say that the (t, ϵ) -DBDH assumption holds in \mathbb{G} if no t -time algorithm has advantage at least ϵ in solving the DBDHP in \mathbb{G} .*

3 Security model

We present our security model for indistinguishability, existential unforgeability and ciphertext authenticity for HIDSC.

3.1 Indistinguishability

Indistinguishability for HIDSC against adaptive chosen ciphertext attack (IND-CCA2) is defined as in the following IND-CCA2 game. The adversary is allowed to query the random oracles q_H times, the key extraction oracle q_E times, the signcryption oracle q_S times and the Un-signcryption / Recover oracle q_R times. The game is defined as follows:

1. The simulator selects the public parameter and sends the parameter to the adversary.
2. The adversary is allowed to perform a polynomial number of oracle queries adaptively.
3. The adversary generates m_0, m_1, ID_A, ID_B , and sends them to the simulator. The simulator randomly chooses $b \in_R \{0, 1\}$ and delivers the challenge ciphertext c to the adversary where $\{\sigma, r\} = \mathbf{Sign}(m, S_A)$ and $C = \mathbf{Encrypt}(S_A, ID_B, m_b, \sigma, r)$.
4. The adversary can again perform a polynomial number of oracle queries adaptively.
5. The adversary tries to compute b .

The adversary wins the game if he can guess b correctly. The *advantage* of the adversary is the probability, over half, that he can compute b accurately. The oracles are defined as follows.

- **Key extraction oracle \mathcal{KEO} :** Upon the input of an identity, the key extraction oracle outputs the private key corresponding to this identity, but oracle query to \mathcal{KEO} with input ID_B is not allowed.
- **Signcryption oracle \mathcal{SO} :** Upon the input of the message m , sender ID_A , recipient ID_B , the signcryption oracle produces a valid signcryption c ; but oracle query to \mathcal{SO} with input $(m_0/m_1, ID_A, ID_B)$ is not allowed.
- **Unsigncryption oracle \mathcal{UO} :** Upon the input of the ciphertext c , sender ID_A and recipient ID_B , the unsigncryption oracle outputs the decryption result and the verification outcome, but oracle query to \mathcal{UO} for the challenge ciphertext from the simulator is not allowed.

Definition 4. (*Indistinguishability*) A hierarchical ID-based signcryption scheme is IND-CCA2 secure if no PPT adversary has a non-negligible advantage in the IND-CCA2 game.

Our security notion above is a strong one. It incorporates previous security notions including *insider-security* in [1], *indistinguishability* in [17]. Notice that if we set the adversary to send the recipient identity ID_B to the simulator before step 1 (say, in an initialization stage) in the game, the security is reduced to the indistinguishability against selective identity, adaptive chosen ciphertext attack (IND-sID-CCA2).

3.2 Existential unforgeability

Existential unforgeability against adaptive chosen message attack (EU-CMA2) for HIDSC is defined as in the following EU-CMA2 game. The adversary is allowed to query the random oracles, \mathcal{KEO} , \mathcal{SO} and \mathcal{UO} , which are defined in the above section. The game is defined as follows:

1. The simulator selects the public parameter and sends it to the adversary.
2. The adversary is allowed to perform a polynomial number of oracle queries adaptively.
3. The adversary delivers a recipient identity ID_B and a ciphertext c .

The adversary wins the game if he can produce a valid (c, ID_B) such that c can be decrypted, under the private key of ID_B , to a message m , sender identity ID_A and a signature s which passes the verification test.

Oracle query to \mathcal{KEO} with input ID_A is not allowed. No \mathcal{SO} request that resulted in a ciphertext c , the signcryption of m from ID_A to ID_B , whose decryption under the private key of ID_B is the claimed forgery (σ, m, ID_A) .

Definition 5. (*Existential Unforgeability*) A hierarchical ID-based signcryption scheme is EU-CMA2 secure if no PPT adversary has a non-negligible probability in winning the EU-CMA2 game.

The adversary is allowed to get the private key of the recipient in the adversary’s answer. This gives us an *insider-security* in [1]. Notice that if we set the adversary to send the sender identity ID_A to the simulator in Step 1 in the game, the security is reduced to the existential unforgeability against selective identity, adaptive chosen ciphertext attack (EU-sID-CMA2).

3.3 Ciphertext Authenticity

Ciphertext authenticity against adaptive chosen message attack (AUTH-CMA2) for HIDSC is defined as in the following AUTH-CMA2 game. The adversary is allowed to query the random oracles, \mathcal{KEO} , \mathcal{SO} and \mathcal{UO} , which are defined in the above section. The game is defined as follows:

1. The simulator selects the public parameter and sends the parameter to the adversary.
2. The adversary is allowed to perform a polynomial number of oracle queries adaptively.
3. The adversary delivers a recipient identity ID_B and a ciphertext c .

The adversary wins the game if he can produce a valid (c, ID_B) such that c can be decrypted, under the private key of ID_B , to a message m , sender identity ID_A and a signature s which passes the verification test.

Oracle query to \mathcal{KEO} with input ID_A and ID_B is not allowed. The adversary’s answer (c, ID_B) should not be computed by \mathcal{SO} before.

Definition 6. (*Ciphertext Authenticity*) A hierarchical ID-based signcryption scheme is AUTH-CMA2 secure if no PPT adversary has a non-negligible probability in winning the AUTH-CMA2 game.

Outsider-security is considered in this model since the adversary is not allowed to get the private key of the recipient in the adversary's answer. This model represents the attack where a signature is re-encrypted by using a public key with unknown secret key.

4 Scheme 1

4.1 Construction

Let H_1 , H_2 and H_3 be three cryptographic hash functions where $H_1 : \{0, 1\}^* \rightarrow \mathbb{G}$ and $H_2 : \{0, 1\}^* \rightarrow \mathbb{G}$, and $H_3 : \mathbb{G}_1 \rightarrow \{0, 1\}^{k_0+k_1+n}$ where k_0 is the number of bits required to represent an element of \mathbb{G} , k_1 is the maximum number of bits required to represent an identity and n is the number of bits of a message to be signcrypted. Our first construction of A hierarchical ID-based signcryption scheme is given below. The construction is based on the idea in [12].

Setup: On the input of a security parameter $k \in \mathbb{N}$, the BDH parameter generator [4] will generate \mathbb{G} , \mathbb{G}_1 , p and $\hat{e}(\cdot, \cdot)$. Then the PKG executes the following steps.

1. Select an arbitrary generator P_0 from \mathbb{G} .
2. Pick a random s_0 from \mathbb{Z}_p , which is the system's master secret key.
3. Compute $Q_0 = P_0^{s_0}$.
4. The public system parameters are

$$\text{params} = \langle \mathbb{G}, \mathbb{G}_1, \hat{e}(\cdot, \cdot), p, P_0, Q_0, H_1(\cdot), H_2(\cdot), H_3(\cdot) \rangle .$$

KeyGen: For an entity with $ID|k-1 = \{ID_1, ID_2, \dots, ID_{k-1}\}$ of depth $k-1$, it uses its secret key $S_{ID|k-1}$ to generate the secret key for a user $ID|k$ (where the first $k-1$ elements of $ID|k$ are those in $ID|k-1$) as follows.

1. Compute $P_k = H_1(ID_1, ID_2, \dots, ID_{k-1}, ID_k)$.
2. Pick random s_{k-1} from \mathbb{Z}_p .
3. Set the private key of the user to be $S_{ID|k} = S_{ID|k-1} \cdot P_k^{s_{k-1}} = \prod_{i=1}^k P_i^{s_{i-1}}$, where $S_{ID|0}$ is defined as the identity element in \mathbb{G} .
4. Send the values of $Q_i = P_0^{s_i}$ for $1 \leq i \leq k-1$ to the user.

Sign: For a user $A|k$ with secret key $S_{A|k} = \prod_{i=1}^k P_{A|i}^{s_{i-1}}$ and the points $Q_i = P_0^{s_i}$ for $1 \leq i \leq k$ to sign on a message M , he/she follows the steps below.

1. Pick a random number r from \mathbb{Z}_p^* .
2. Compute $P_M = H_2(M)$.
3. Compute $\sigma = S_{A|k} \cdot P_M^r$.
4. Return signature = $\{r, \sigma, Q_1, Q_2, \dots, Q_{k-1}, Q_k = P_0^r\}$.

Encrypt: To signcrypt the message M to user $B|l$, the steps below are used.

1. Compute $P_{B|j} = H_1(B_1, B_2, \dots, B_j)$ for $1 \leq j \leq l$.

2. Return ciphertext =

$$\{P_{B|2}^r, \dots, P_{B|l}^r, (M||\sigma||A|k) \oplus H_3(\hat{g}^r), Q_1, Q_2, \dots, Q_k\}$$

where $\hat{g} = \hat{e}(Q_0, P_{B|1}) \in \mathbb{G}_1$ and \oplus represents the bitwise XOR.

Decrypt: For user $B|l$ with secret key $S_{B|l} = \prod_{i=1}^l P_{B|i}^{s'_{i-1}}$ and the points $Q'_i = P_0^{s'_i}$ for $1 \leq i \leq l$ to decrypt the signcrypted message C , the steps below are used.

1. Let $C = \{U_2, \dots, U_l, V, Q_1, Q_2, \dots, Q_k\}$
2. Compute $V \oplus H_3\left(\frac{\hat{e}(Q_k, S_{B|l})}{\prod_{i=2}^l \hat{e}(Q'_{i-1}, U_i)}\right) = M||\sigma||A|k$.
3. Return $\{M, \sigma, H_1(ID_{A|k}), Q_1, Q_2, \dots, Q_k\}$.

Verify: For A 's signature $\{\sigma, Q_1, Q_2, \dots, Q_k\}$, everyone can do the following to verify its validity.

1. Compute $P_M = H_2(M)$.
2. Compute $P_{A|i} = H_1(A_1, A_2, \dots, A_i)$ for $1 \leq i \leq k$.
3. Return \top if $\hat{e}(P_0, \sigma) / \prod_{i=2}^k \hat{e}(Q_{i-1}, P_{A|i}) = \hat{e}(Q_0, P_{A|1}) \hat{e}(Q_k, P_M)$.

4.2 Efficiency Analysis

The signcrypted message is shortened by one \mathbb{G}_1 element, as compared with using the scheme HIDE and HIDS in [12] together. Moreover, chosen ciphertext secure HIDE requires the transformation in Section 3.2 of [12], while our scheme does not require such transformation as the integrity checking of the ciphertext is obtained from the signature.

4.3 Security analysis

Theorem 1. *Suppose that the (t, ϵ) -BDH assumption holds in \mathbb{G} , then the above scheme is $(t', q_S, q_H, q_E, q_R, \epsilon)$ -adaptive chosen ciphertext (IND-CCA2) secure for arbitrary q_S, q_H, q_E, q_R , and any $t' < t - o(t)$.*

Proof. Dealer \mathcal{D} gives (g, g^a, g^b, g^c) to Simulator \mathcal{S} and wants \mathcal{S} to compute $\hat{e}(g, g)^{abc}$. Set $P_0 = g, Q_0 = g^a$. \mathcal{S} sends the system parameter to \mathcal{A} . \mathcal{S} randomly picks μ with $1 \leq \mu \leq q_H$.

Phase 1: Query on H_1 for input (A_1, \dots, A_k) :

- If $k = 1$, the μ -th query to H_1 with $k = 1$ is back patched to g^b . The corresponding identity is denoted as ID_b . Adds the entry $\langle ID_b, g^b \rangle$ to tape L_1 and returns g^b .
- Otherwise, randomly picks $\lambda \in \mathbb{Z}_p$; add $\langle A_1, \dots, A_k, \lambda \rangle$ to L_1 and returns g^λ .

When there is a query on H_2 for input M , randomly picks $\lambda \in \mathbb{Z}_p$; adds $\langle M, \lambda \rangle$ to L_2 and returns $(g^a)^\lambda$. Query on H_3 is handled by producing a random element from the codomain, and adding both query and answer to tape L_3 .

Key Extraction Oracle (\mathcal{KEO}): For input identity $ID = (I_1, \dots, I_k) \in \mathbb{Z}_p^k$ where $k \leq \ell$.

- If $I_1 = ID_b$, then abort the simulation.

- Otherwise, look up at the tape $L_K = \langle ID_1, \dots, ID_u, \alpha_1, \dots, \alpha_{u-1} \rangle$ which stores the previously extracted keys. Let y be the maximal value such that $(I_1, \dots, I_y) = (ID_{j_1}, \dots, ID_{j_y})$ for some tuple $\langle ID_{j_1}, \dots, ID_{j_u}, \alpha_{j_1}, \dots, \alpha_{j_{(u-1)}} \rangle \in L_K$. Then:
 - For $1 \leq i \leq y$, set $\alpha_i = \alpha_{j_i}, Q_i = g^{\alpha_i}$. Get $P_i = H_1(A_1, \dots, A_i)$ from L_1 and also get λ from $(A_1, \lambda) \in L_1$.
 - For $y < i \leq k$, query the value of P_i from H_1 . Randomly generate $\alpha_i \in \mathbb{Z}_p$.
 - Put $\langle I_1, \dots, I_k, \alpha_1, \dots, \alpha_{k-1} \rangle$ in L_K . Set the private key as $s_{ID|k} = \prod_{i=1}^k P_i^{s_i-1} = (g^a)^\lambda \cdot P_2^{\alpha_1} \dots P_k^{\alpha_{k-1}}$. Returns $s_{ID|k}$ and $Q_i = g^{\alpha_i}$ for $1 \leq i \leq k-1$.
- Note that the private key satisfies the required form.

Signcryption Oracle (\mathcal{SO}): For input message M , sender $ID_{A|k} = \{I_{A_1}, \dots, I_{A_k}\}$, and recipient $ID_{B|l} = \{I_{B_1}, \dots, I_{B_l}\}$.

- If $I_{A_1} = ID_b$, query P_M from H_2 and obtain λ_M from $\langle M, \lambda_M \rangle \in L_2$. Query $P_{A|i}$ from H_1 and obtain λ_i from $\langle I_{A_1}, \dots, I_{A_i}, \lambda_i \rangle \in L_1$, for $1 \leq i \leq k$. Randomly generate $\alpha_i \in \mathbb{Z}_p$ for $1 \leq i \leq k$. Compute $\sigma = (g^a)^{(\alpha_k \lambda_M)} \prod_{i=2}^k g^{\lambda_i \alpha_{i-1}}, Q_i = g^{\alpha_i}$ for $1 \leq i \leq k-1$, $Q_k = (g^{\alpha_k})(g^b)^{-1/\lambda_M}$. Query $P_{B|i}$ from H_1 and obtain get λ_{B_i} from $\langle I_{B_1}, \dots, I_{B_i}, \lambda_{B_i} \rangle \in L_1$, for $1 \leq i \leq l$. Compute $U_i = (g^{\alpha_k})(g^b)^{-\frac{\lambda_{B_i}}{\lambda_M}}$ for $2 \leq i \leq l$, $V = (M || \sigma || A|k) \oplus H_3(\hat{e}(g^a, (g^{\alpha_k})(g^b)^{-\frac{\lambda_{B_1}}{\lambda_M}}))$. Return the ciphertext $C = \{U_2, \dots, U_l, V, Q_1, \dots, Q_k\}$. \mathcal{S} puts $\langle ID_{A|k}, ID_{B|l}, M, C \rangle$ in L_S . It is easy to see that the signature will pass the verification test:

$$\begin{aligned}
 & \hat{e}(P_0, \sigma) / \prod_{i=2}^k \hat{e}(Q_{i-1}, P_{A|i}) \\
 &= \hat{e}(g, (g^{(a\alpha_k \lambda_M)}) \prod_{i=2}^k g^{\lambda_i \alpha_{i-1}}) / \prod_{i=2}^k \hat{e}(g^{\alpha_{i-1}}, g^{\lambda_i}) \\
 &= \hat{e}(g, g^{(a\alpha_k \lambda_M)}) \hat{e}(g, \prod_{i=2}^k g^{\lambda_i \alpha_{i-1}}) / \prod_{i=2}^k \hat{e}(g, g^{\lambda_i \alpha_{i-1}}) \\
 &= \hat{e}(g^{\alpha_k}, g^{(a\lambda_M)}) \prod_{i=2}^k \hat{e}(g, g^{\lambda_i \alpha_{i-1}}) / \prod_{i=2}^k \hat{e}(g, g^{\lambda_i \alpha_{i-1}}) \\
 &= \hat{e}(g^a, g^b) \hat{e}(g^{\alpha_k}, g^{(a\lambda_M)}) \hat{e}(g^{a\lambda_M}, (g^b)^{-1/\lambda_M}) \\
 &= \hat{e}(g^a, g^b) \hat{e}((g^{\alpha_k})(g^b)^{-1/\lambda_M}, g^{a\lambda_M}) \\
 &= \hat{e}(Q_0, P_{A|1}) \hat{e}(Q_k, P_M)
 \end{aligned}$$

- Otherwise, \mathcal{S} retrieves the private key of $ID_{A|k}$ using the same way as \mathcal{KEO} and then uses it to run signcryption and gets ciphertext c . \mathcal{S} puts $\langle ID_{A|k}, ID_{B|l}, M, C \rangle$ in L_S .

Un-signcryption / Recover Oracle (\mathcal{UO}): For input sender $ID_{A|k} = \{I_{A_1}, \dots, I_{A_k}\}$, recipient $ID_{B|l} = \{I_{B_1}, \dots, I_{B_l}\}$ and ciphertext $C = \{U_2, \dots, U_l, V, Q_1, \dots, Q_k\}$.

- For the case $ID_{B|l} = ID_b$, \mathcal{S} finds if $\langle ID_{A|k}, ID_{B|l}, M, C \rangle$ is in L_S . If so, returns M . Otherwise, \mathcal{S} searches for all combinations $\langle M, \sigma \rangle$ such that $\langle M, h_2 \rangle \in L_2$, $\langle g', h_3 \rangle \in L_3$, for some h_2, h_3 , under the constraints that $h_3 \oplus V = M || \sigma || A|k$, and $\hat{e}(g, \sigma) =$

$\hat{e}(g^a, P_{A|1})\hat{e}(Q_k, h_2)\prod_{i=2}^k \hat{e}(Q_{i-1}, P_{A|i})$. \mathcal{S} simply picks one of the valid message M from the above and return it as answer. If no such tuple is found, the oracle signals that the ciphertext is invalid.

- For other cases, \mathcal{S} retrieves the private key of $ID_{B|l}$ using the same way as \mathcal{KEO} and then uses it to decrypt and verify.

Witness Extraction: As in the IND-CCA2 game, at some point \mathcal{A} chooses plaintext m_0, m_1 , and sender $ID_{A|k}$ on which he wishes to be challenged. \mathcal{S} retrieves the private key of $ID_{A|k}$ and Q_1, \dots, Q_k using the same way as \mathcal{KEO} . \mathcal{S} queries $P_{B|i}$ from H_1 and obtain get λ_{B_i} from $\langle I_{B_1}, \dots, I_{B_i}, \lambda_{B_i} \rangle \in L_1$, for $2 \leq i \leq l$. \mathcal{S} randomly picks $V \in \{0, 1\}^{k_0+k_1+n}$ and responds with challenge ciphertext $C = \{(g^c)^{\lambda_{B_2}}, \dots, (g^c)^{\lambda_{B_l}}, V, Q_1, \dots, Q_{k-1}, g^c\}$. All further queries by \mathcal{A} are processed adaptively as in the oracles above, with no private key extraction of $ID_{B|l}$. Finally, \mathcal{A} returns its final guess b' . \mathcal{S} ignores the answer from \mathcal{A} , randomly picks an entry $\langle g', h_3 \rangle$ in L_3 , and returns g' as the solution to the BDH problem.

If the recipient identity is $ID_{B|l} = \{I_{B_1}, \dots, I_{B_l}\}$ with $I_{B_1} = ID_b$, to recognize the challenge ciphertext is incorrect, \mathcal{A} needs to query random oracle H_3 with $g' = \hat{e}(Q_0, P_{B|1})^c = \hat{e}(g, g)^{abc}$. It will leave an entry $\langle g', h_3 \rangle$ on L_3 , from which \mathcal{S} can extract $g' = \hat{e}(g, g)^{abc}$. \square

Theorem 2. *Suppose that the (t, ϵ) -CDH assumption holds in \mathbb{G} , then the above scheme is $(t', q_S, q_H, q_E, q_R, \epsilon)$ -adaptive chosen message (EU-CMA2) secure for arbitrary q_S, q_H, q_E, q_R , and any $t' < t - o(t)$, $\epsilon' > \frac{\epsilon}{e^{2q_S q_E}}$.*

Proof. Dealer \mathcal{D} gives (g, g^a, g^b) to Simulator \mathcal{S} and wants \mathcal{S} to compute g^{ab} . Set $g_1 = g^a, g_2 = g^b$. The initialization, setup and the simulation of oracles are similar to those in the proof in Theorem 1. The difference is that probabilistic simulations are used in the simulation of two hash oracles: the one for hashing the identity ($H_1(\cdot)$) and the one for hashing the message ($H_2(\cdot)$).

Queries on oracle H_1 for identity (A_1, \dots, A_k) : If $k = 1$, \mathcal{S} embeds part of the challenge g^b in the answer of many H_1 queries [11]. \mathcal{S} picks $\lambda \in_R \mathbb{F}_q^*$ and repeats the process until λ is not in the list L_1 . \mathcal{S} then flips a coin $W_1 \in \{0, 1\}$ that yields 0 with probability ζ_1 and 1 with probability $1 - \zeta_1$. (ζ_1 will be determined in the probability analysis shortly afterward.) If $W_1 = 0$ then the hash value $H_1(A_1)$ is defined as g^λ ; else if $W_1 = 1$ then returns $H_1(A_1) = (g^b)^\lambda$. In either case, \mathcal{S} stores $\langle A_1, \lambda, W_1 \rangle$ in the list L_1 .

On the other hand, if $k > 1$, \mathcal{S} performs the simulation as that in the proof of Theorem 1.

Queries on oracle H_2 for message m : In this case, \mathcal{S} embeds the remaining part of the challenge g^a in the answer of many H_2 queries. \mathcal{S} picks $\beta \in_R \mathbb{F}_q^*$ and repeats the process until β is not in the list L_2 . \mathcal{S} then flips a coin $W_2 \in \{0, 1\}$ that yields 0 with probability ζ_2 and 1 with probability $1 - \zeta_2$. (ζ_2 will be determined later.) If $W_2 = 0$ then the hash value $H_2(m)$ is defined as $(g^a)^\beta$; else if $W_2 = 1$ then returns $H_2(m) = g^\beta$. In either case, \mathcal{S} stores $\langle m, \beta, W_2 \rangle$ in the list L_2 .

Witness Extraction: After such probabilistic behaviour is introduced to the simulation, \mathcal{S} will fail for the \mathcal{KEO} query of (A_1, \dots, A_k) if $W_1 = 1$ is found in the corresponding entry of A_1 in L_1 . The \mathcal{SO} query for the signcryption of message m done by (A_1, \dots, A_k) will fail too when $W_1 = 1$ and $W_2 = 1$ are found in the corresponding entry of A_1 in L_1 and m in L_2 respectively.

At the end of the game, \mathcal{A} returns a forgery $C = \{U_2, \dots, U_l, V, Q_1, Q_2, \dots, Q_k\}$ which is the signcryption of message m done by (A_1, \dots, A_k) . \mathcal{S} cannot solve the CDH problem

if the forgery is not related to the problem instance at all, i.e. when $W_1 = 0$ is found in the corresponding entry of A_1 in the list L_1 and $W_1 = 1$ and $W_2 = 0$ are found in the corresponding entry of A_1 in L_1 and m in L_2 respectively.

For successful cases, \mathcal{S} gets the forged signature $\{\sigma, Q_1, Q_2, \dots, Q_k\}$ by the decryption of the signcrypted text. Suppose that λ_i is the corresponding entry of $P|i$ in the list L_1 and β is the corresponding entry of P_M in the list L_2 , since $\sigma = \prod_{i=1}^k (P_i^{s_{i-1}}) \cdot P_M^r = g^{ab} \cdot \prod_{i=2}^k (P_i^{s_{i-1}}) \cdot P_M^r$ \mathcal{C} can compute the solution of the CDH problem by $\sigma / \prod_{i=2}^k (Q_{i-1}^{\lambda_i}) \cdot Q_k^\beta$.

Probability Analysis: The probability that \mathcal{S} answers to all private key extraction queries is $\zeta_1^{q_E}$. \mathcal{S} can answer all signcrypt queries for users $H_1(A_1, \dots, A_k)$ where $\langle A_1, \lambda, 0 \rangle$ is in the list L_1 . So the worst case for \mathcal{S} to answer all signcrypt queries correctly happens when all signcrypt requests are for users $H_1(A_1, \dots, A_k)$ where $\langle A_1, \lambda, 1 \rangle$ is in the list L_1 . For these class of users, \mathcal{S} can still signcrypt given the message is M where $\langle M, \beta, 0 \rangle$ can be found in the list L_2 . So the probability for \mathcal{S} to successfully answer all signcrypt requests is $\zeta_2^{q_S}$.

Finally, the probability that \mathcal{A} makes a forged signature for user $H_1(A_1, \dots, A_k)$ where $\langle A_1, \lambda, 1 \rangle$ is in the list L_1 is $1 - \zeta_1$ and the probability that \mathcal{A} makes a forged signature on message M where $\langle M, \beta, 1 \rangle$ is in the list L_2 is $1 - \zeta_2$.

Hence the probability for \mathcal{S} to solve CDH problem successfully is $f_{q_E}(\zeta_1) f_{q_S}(\zeta_2)$ where $f_x(\zeta) = \zeta^x(1-\zeta)$. Simple differentiation shows that $f_x(\zeta)$ is maximized when $\zeta = 1 - (x+1)^{-1}$, and the corresponding probability is $\frac{1}{x}(1 - \frac{1}{x+1})^{x+1}$. So the maximum probability for \mathcal{S} to solve CDH problem successfully is

$$\frac{1}{q_S q_E} \left(1 - \frac{1}{q_S + 1}\right)^{q_S + 1} \left(1 - \frac{1}{q_E + 1}\right)^{q_E + 1}$$

For large q_S and q_E , this probability is approximately equal to $1/e^2 q_S q_E$. □

5 Scheme 2

5.1 Construction

Let H be a cryptographic hash function where $H : \{0, 1\}^* \rightarrow \mathbb{Z}_p$. We use $H(\cdot)$ to hash the string representing the identity into an element in \mathbb{Z}_p^k , the same hash function will be used in the signing algorithm too. Similar to [3], H is not necessarily a full domain hash function. Our second construction of HIDSC, based on the ideas in [9] and [3], is given below.

Setup: On the input of a security parameter $k \in \mathbb{N}$, the BDH parameter generator [4] will generate $\mathbb{G}, \mathbb{G}_1, p$ and $\hat{e}(\cdot, \cdot)$. Then the PKG executes the following steps.

1. Select α from \mathbb{Z}_p^* , h_1, h_2, \dots, h_ℓ from \mathbb{G} and two generators g, g_2 from \mathbb{G}^* , where ℓ is the number of levels of the hierarchy that our scheme supports.
2. The public parameters are: $\{g, g_1 = g^\alpha, g_2, h_1, h_2, \dots, h_\ell, \hat{e}(g_1, g_2)\}$.
3. The master secret key is $d_{ID|0} = g_2^\alpha$.

KeyGen: For a user $ID|k-1 = \{I_1, I_2, \dots, I_{k-1}\}$ of depth $k-1$, he/she uses his/her secret key $d_{ID|k-1}$ to generate the secret key for a user $ID|k$ (where the first $k-1$ elements of $ID|k$ are those in $ID|k-1$) as follows.

1. Pick random r_k from \mathbb{Z}_p .

2. $d_{ID|k} = \{d_0 F_k(ID_k)^{r_k}, d_1, \dots, d_{k-1}, g^{r_k}\}$, where $F_k(x)$ is defined as $g_1^x h_k$.

Sign: For a user $ID|k$ with secret key $\{g_2^\alpha \prod_{j=1}^k F_j(ID_j)^{r_j}, g^{r_1}, \dots, g^{r_k}\}$ to sign on a message m , he/she follows the steps below.

1. Pick a random number s from \mathbb{Z}_p^* .
2. Compute $h = H(m, \hat{e}(g_1, g_2)^s)$.
3. Repeat Steps 1-3 in case the unlikely event $s + h = 0$ occurs.
4. For $j = \{1, 2, \dots, k\}$, compute $y_j = d_j^{s+h}$.
5. Compute $z = d_0^{s+h}$.
6. Return signature = $\{s, y_1, y_2, \dots, y_k, z\}$.

Encrypt: To signcrypt a message $M \in \mathbb{G}_1$ under the public key $ID|l = \{I_1, I_2, \dots, I_l\}$, the ciphertext is

$$\{\hat{e}(g_1, g_2)^s \cdot M, g^s, F_1(I_1)^s, F_2(I_2)^s, \dots, F_l(I_l)^s, y_1, y_2, \dots, y_k, z\}.$$

Decrypt: For a user $ID'|l$ with secret key $\{g_2^\alpha \prod_{j=1}^l F_j(ID'_j)^{r'_j}, g^{r'_1}, \dots, g^{r'_l}\}$ to decrypt the signcrypted text $\{A, B, C_1, \dots, C_l, y_1, y_2, \dots, y_k, z\}$, he/she follows the steps below.

1. Compute $\sigma = \hat{e}(g_1, g_2)^s$ by $\frac{\hat{e}(B, d'_0)}{\prod_{j=1}^l \hat{e}(C_j, d'_j)}$.
2. Obtain the message M by $A \cdot \sigma^{-1}$.

Verify: For $ID|k = \{I_1, I_2, \dots, I_k\}$'s signature $\{\sigma, y_1, y_2, \dots, y_k, z\}$, everyone can do the following to verify its validity.

1. Compute $h = H(m, \sigma)$.
2. Return \top if $\hat{e}(g, z) = \sigma \cdot \hat{e}(g_1, g_2^h \prod_{j=1}^k y_j^{ID_j}) \prod_{j=1}^k \hat{e}(y_j, h_j)$, \perp otherwise.

5.2 Efficiency Analysis

The signcrypted message is shortened by one \mathbb{G}_1 element, as compared with using the scheme in [9] and [3] together. Moreover, chosen ciphertext secure HIDE requires the transformation in Section 4 of [6], while our scheme does not require such transformation as the integrity checking of the ciphertext is obtained from the signature.

5.3 Security Analysis

Theorem 3. *Suppose that the (t, ϵ) -Decision BDH assumption holds in \mathbb{G} , then the above scheme is $(t', q_S, q_H, q_E, q_R, \epsilon)$ -selective identity, adaptive chosen ciphertext (IND-sID-CCA2) secure for arbitrary q_S, q_H, q_E, q_R , and any $t' < t - o(t)$.*

Proof. Dealer \mathcal{D} gives (g, g^a, g^b, g^c, T) to Simulator \mathcal{S} and wants \mathcal{S} to output 1 if $T = \hat{e}(g, g)^{abc}$ or output 0 otherwise. Set $g_1 = g^a, g_2 = g^b, g_3 = g^c$.

Initialization: Adversary \mathcal{A} sends an identity $ID^* = (I_1^*, \dots, I_k^*) \in Z_p^k$ of depth $k \leq \ell$ that it intends to attack to \mathcal{S} .

Setup: \mathcal{S} randomly picks $\alpha_1, \dots, \alpha_\ell \in \mathbb{Z}_p$ and defines $h_j = g_1^{-I_j^*} g^{\alpha_j} \in \mathbb{G}$ for $j = 1, \dots, \ell$. \mathcal{S} sends the system parameter $(g, g_1, g_2, h_1, \dots, h_\ell, \hat{e}(g_1, g_2))$ to \mathcal{A} .

Phase 1: Query on H for input (m, σ) :

- If $(m, \sigma, h) \in L$ for some h , return h .
- Otherwise, randomly picks $h \in \mathbb{Z}_p$; add (m, σ, h) to L and returns h .

Key Extraction Oracle (\mathcal{KEO}): For input identity $ID = (I_1, \dots, I_u) \in \mathbb{Z}_p^u$ where $u \leq \ell$.

- If $ID = ID^*$ or ID is a prefix of ID^* , then abort the simulation.
- Otherwise, let j be the smallest index such that $I_j \neq I_j^*$. \mathcal{S} firstly derives a private key for identity (I_1, \dots, I_j) from which it then construct a private key for ID . \mathcal{S} randomly picks $r_1, \dots, r_j \in \mathbb{Z}_p$ and sets:

$$d_0 = g_2^{\frac{-\alpha_j}{I_j - I_j^*}} \prod_{v=1}^j F_v(I_v)^{r_v}, d_1 = g^{r_1}, \dots, d_{j-1} = g^{r_{j-1}}, d_j = g_2^{\frac{-1}{I_j - I_j^*}} g^{r_j}$$

We now show that (d_0, d_1, \dots, d_j) is a valid random private key for (I_1, I_2, \dots, I_j) . Let $\tilde{r}_j = r_j - b/(I_j - I_j^*)$ and then we have:

$$g_2^{\frac{-\alpha_j}{I_j - I_j^*}} F_j(I_j)^{r_j} = g_2^{\frac{-\alpha_j}{I_j - I_j^*}} (g_1^{I_j - I_j^*} g^{\alpha_j})^{r_j} = g_2^a (g_1^{I_j - I_j^*} g^{\alpha_j})^{r_j - \frac{b}{I_j - I_j^*}} = g_2^a F_j(I_j)^{\tilde{r}_j}$$

So the private key satisfies the required form.

Signcryption Oracle (\mathcal{SO}): For input message m , sender $ID_{A|k} = \{I_{A1}, \dots, I_{Ak}\}$, and recipient $ID_{B|l} = \{I_{B1}, \dots, I_{Bl}\}$.

- If $ID_{A|k}$ equals ID^* or a prefix of ID^* , then \mathcal{S} randomly chooses $h \in \mathbb{Z}_p$, and computes $\sigma = \hat{e}(g_1, g_2)^{-h}$. Then \mathcal{S} randomly picks $r_1, \dots, r_k \in \mathbb{Z}_p$, computes $y_v = g_2^{r_v}$ for $1 \leq v \leq k$ and $z = \prod_{v=1}^k g_2^{r_v \alpha_v}$. Then \mathcal{S} adds the tuple (m, σ, h) to L to force the random oracle $H(m, \sigma) = h$. Finally, \mathcal{S} returns the ciphertext $C = \{\sigma \cdot m, g^{-h}, F_1(I_{B1})^{-h}, F_2(I_{B2})^{-h}, \dots, F_l(I_{Bl})^{-h}, y_1, y_2, \dots, y_k, z\}$. \mathcal{S} puts $(ID_{A|k}, ID_{B|l}, m, C, -h, h)$ in L_S .
- Otherwise, \mathcal{S} retrieves the private key of $ID_{A|k}$ using the same way as \mathcal{KEO} and then uses it to run signcryption and gets ciphertext c . \mathcal{S} puts $(ID_{A|k}, ID_{B|l}, m, C, s, h)$ in L_S .

Un-signcryption / Recover Oracle (\mathcal{UO}): For input sender $ID_{A|k} = \{I_{A1}, \dots, I_{Ak}\}$, recipient $ID_{B|l} = \{I_{B1}, \dots, I_{Bl}\}$ and ciphertext $C = \{A, B, C_1, \dots, C_l, y_1, \dots, y_k, z\}$.

- For the case $ID_{B|l} = ID^*$, \mathcal{S} finds if $(ID_{A|k}, ID_{B|l}, m, C, s, h)$ is in L_S . If so, returns m . Otherwise, \mathcal{S} searches for a valid m in all entries $\langle m, \sigma, h \rangle \in L$, under the constraints that $\sigma \cdot m = A$, $\sigma = \hat{e}(g, z) / (\hat{e}(g_1, g_2^h \prod_{j=1}^k y_j^{ID_{A|j}}) \prod_{j=1}^l \hat{e}(y_j, h_j))$ and $\hat{e}(B, F_j(I_{Bj})) = \hat{e}(g, C_j)$ for $1 \leq j \leq l$. \mathcal{S} simply picks a message in one of the valid m in the above and return it as the answer. If no such tuple is found, the oracle signals that the ciphertext is invalid.
- For other cases, \mathcal{S} retrieves the private key of $ID_{A|k}$ using the same way as \mathcal{KEO} and then uses it to decrypt and verify.

Witness Extraction: As in the IND-sID-CCA2 game, at some point \mathcal{A} chooses plaintext m_0, m_1 , and sender $ID_{A|k}$ on which he wishes to be challenged. \mathcal{S} picks a random bit $b \in \{0, 1\}$ and responds with challenge ciphertext $C = \{T \cdot M_b, g_3, g_3^{\alpha_1}, \dots, g_3^{\alpha_l}, y_1, \dots, y_k, z\}$,

where (y_1, \dots, y_k, z) is a valid signature from $ID_{A|k}$. All further queries by \mathcal{A} are processed adaptively as in the oracles above. Finally, \mathcal{A} returns its final guess b' . If $b = b'$, then \mathcal{S} outputs 1 meaning $T = \hat{e}(g, g)^{abc}$. Otherwise it outputs 0 meaning $T \neq \hat{e}(g, g)^{abc}$.

If the recipient identity is ID^* , then the value of $\hat{e}(g_1, g_2)^s$ is equal to $\hat{e}(g^a, g^b)^c = \hat{e}(g, g)^{abc}$. If \mathcal{A} has the advantage ϵ to guess b correctly, then \mathcal{S} has the advantage ϵ to solve the DBDHP. \square

Theorem 4. *Suppose that the (t, ϵ) -CDH assumption holds in \mathbb{G} , then the above scheme is $(t', q_S, q_H, q_E, q_R, \epsilon')$ -selective identity, adaptive chosen message (EU-sID-CMA) secure for arbitrary q_S, q_H, q_E, q_R , and any $t' < t - o(t)$, $\epsilon' > \epsilon \cdot (1 - \frac{q_S(q_H+q_S)}{q})$.*

Proof. Dealer \mathcal{D} gives (g, g^a, g^b) to Simulator \mathcal{S} and wants \mathcal{S} to compute g^{ab} . Set $g_1 = g^a, g_2 = g^b$. The initialization, setup and the simulation of oracles are the same as the proof of Theorem 3. At the end of the game, \mathcal{A} returns a forgery $C = \{A, B, C_1, \dots, C_l, y_1, \dots, y_k, z\}$ using h from H query. By forking lemma, we rewind \mathcal{A} to the time when the H query was issued and get $C' = \{A', B', C'_1, \dots, C'_l, y'_1, \dots, y'_k, z'\}$ using h' from H query. We can get $d_j = (y_j/y'_j)^{(h-h')^{-1}}$ for $1 \leq j \leq k$. Then we can calculate $d_0 = (z/z')^{(h-h')^{-1}}$. Finally we can get $g_2^a = d_0 / \prod_{j=1}^k d_j^{\alpha_j}$ which is the solution to the CDH problem. \square

Let us consider the possibility for \mathcal{SO} to fail. The only possibility for introducing an error is in defining $H(m, \sigma)$ which is already defined. Since σ takes its value uniformly at random in \mathbb{G}_1 , the chance for the occurrence of one of these events is at most $(q_H + q_S)/q$ for each query. Therefore over the whole simulation, the chance of an error is at most $q_S(q_H + q_S)/q$. Hence \mathcal{S} succeeds with probability at least $\epsilon \cdot (1 - \frac{q_S(q_H+q_S)}{q})$. \square

6 Conclusion

Two concrete constructions of hierarchical identity based signcryption are proposed, which closed the open problem proposed by [15]. Our schemes are provably secure under the random oracle model [2]. Moreover, our schemes does not require transformation which is necessary for the case of hierarchical identity based encryption as the integrity checking of the ciphertext is obtained from the signature. We believe that hierarchical identity based signcryption schemes are useful in nowadays commercial organization and also in new network architecture such as tetherless computing architecture. Future research directions include further improvement on the efficiency of hierarchical identity based signcryption schemes and achieving other security requirements such as public ciphertext authenticity ([10, 15]) or ciphertext anonymity ([5]).

References

1. Jee Hea An, Yevgeniy Dodis, and Tal Rabin. On the Security of Joint Signature and Encryption. In Lars R. Knudsen, editor, *Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings*, volume 2332 of *Lecture Notes in Computer Science*, pages 83–107. Springer-Verlag Heidelberg, 2002.
2. Mihir Bellare and Phillip Rogaway. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In *1st ACM Conference on Computer and Communications Security*, pages 62–73, 1993.
3. Dan Boneh and Xavier Boyen. Efficient Selective-ID Secure Identity-Based Encryption Without Random Oracles. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, volume 3027 of *Lecture Notes in Computer Science*, pages 223–238. Springer, 2004.

4. Dan Boneh and Matt Franklin. Identity-Based Encryption from the Weil Pairing. In Joe Kilian, editor, *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23, 2001, Proceedings*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer-Verlag Heidelberg, 2001.
5. Xavier Boyen. Multipurpose Identity-Based Signcryption : A Swiss Army Knife for Identity-Based Cryptography. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, volume 2729 of *Lecture Notes in Computer Science*, pages 382–398. Springer, 2003.
6. Ran Canetti, Shai Halevi, and Jonathan Katz. Chosen-Ciphertext Security from Identity-Based Encryption. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, volume 3027 of *Lecture Notes in Computer Science*, pages 207–222. Springer, 2004.
7. Liqun Chen and John Malone-Lee. Improved Identity-Based Signcryption. Cryptology ePrint Archive, Report 2004/114, 2004. Available at <http://eprint.iacr.org>.
8. Sherman S.M. Chow. Verifiable Pairing and Its Applications. In *5th Workshop on Information Security Applications (WISA 2004)*, Lecture Notes in Computer Science, Jeju Island, Korea, August 2004. Springer-Verlag. To Appear.
9. Sherman S.M. Chow, Lucas C.K. Hui, S.M. Yiu, and K.P. Chow. Secure Hierarchical Identity Based Signature and its Application. In *6th International Conference on Information and Communications Security (ICICS04)*, Lecture Notes in Computer Science, Malaga, Spain, October 2004. Springer-Verlag. To Appear.
10. Sherman S.M. Chow, S.M. Yiu, Lucas C.K. Hui, and K.P. Chow. Efficient Forward and Provably Secure ID-Based Signcryption Scheme with Public Verifiability and Public Ciphertext Authenticity. In Jong In Lim and Dong Hoon Lee, editors, *Information Security and Cryptology - ICISC 2003, 6th International Conference Seoul, Korea, November 27-28, 2003, Revised Papers*, volume 2971 of *Lecture Notes in Computer Science*, pages 352–369. Springer, 2003.
11. Jean-Sébastien Coron. On the Exact Security of Full Domain Hash. In Mihir Bellare, editor, *Advances in Cryptology - CRYPTO 2000, 20th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2000, Proceedings*, volume 1880 of *Lecture Notes in Computer Science*, pages 229–235. Springer, 2000.
12. Craig Gentry and Alice Silverberg. Hierarchical ID-Based Cryptography. In Yuliang Zheng, editor, *Advances in Cryptology - ASIACRYPT 2002, 8th International Conference on the Theory and Application of Cryptology and Information Security, Queenstown, New Zealand, December 1-5, 2002, Proceedings*, volume 2501 of *Lecture Notes in Computer Science*, pages 548–566. Springer, 2002. Available at <http://eprint.iacr.org>.
13. Florian Hess. Efficient Identity Based Signature Schemes based on Pairings. In Kaisa Nyberg and Howard M. Heys, editors, *Selected Areas in Cryptography, 9th Annual International Workshop, SAC 2002, St. John's, Newfoundland, Canada, August 15-16, 2002. Revised Papers*, volume 2595 of *Lecture Notes in Computer Science*, pages 310–324. Springer, 2003.
14. Berkeley. Intel Research. Identity Based Cryptosystem for Secure Delay Tolerant Networking.
15. Benoît Libert and Jean-Jacques Quisquater. New Identity Based Signcryption Schemes from Pairings. In *IEEE Information Theory Workshop*, pages 155–158, 2003. Full Version Available at <http://eprint.iacr.org>.
16. Benoît Libert and Jean-Jacques Quisquater. The Exact Security of an Identity Based Signature and its Applications. Cryptology ePrint Archive, Report 2004/102, 2004. Available at <http://eprint.iacr.org>.
17. John Malone-Lee. Identity Based Signcryption. Cryptology ePrint Archive, Report 2002/098, 2002. Available at <http://eprint.iacr.org>.
18. Noel McCullagh and Paulo S. L. M. Barreto. Efficient and Forward-Secure Identity-Based Signcryption. Cryptology ePrint Archive, Report 2004/117, 2004. Available at <http://eprint.iacr.org>.
19. Divya Nalla and K.C. Reddy. Signcryption scheme for Identity-Based Cryptosystems. Cryptology ePrint Archive, Report 2003/066, 2003. Available at <http://eprint.iacr.org>.
20. Aaditeshwar Seth, Patrick Darragh, and Srinivasan Keshav. A Generalized Architecture for Tetherless Computing in Disconnected Networks. Manuscript.
21. Tsz Hon Yuen and Victor K. Wei. Fast and Proven Secure Blind Identity-Based Signcryption from Pairings. In *Topics in Cryptology - CT-RSA 2005, The Cryptographers' Track at the RSA Conference 2005 (CTRSA05)*, Lecture Notes in Computer Science, San Francisco, CA, USA, February 2005, Springer. To Appear.