ID Based Signcryption Scheme in Standard Model

S. Sharmila Deva Selvi[1], S. Sree Vivek[1], Dhinakaran Vinayagamurthy[2] *, and C. Pandu Rangan[1]

[1] Theoretical Computer Science Lab, Department of Computer Science and Engineering,
Indian Institute of Technology, Madras, India.
{sharmila,svivek,rangan}@cse.iitm.ac.in
[2] Department of Computer Science and Engineering, College of Engineering, Guindy,
Anna University, Chennai, India.
dhinakaran2705@gmail.com

Abstract. Designing an ID based signcryption scheme in the standard model is among the most interesting and important problems in cryptography. However, all the existing systems in the ID based setting, in the standard model, do not have either the unforgeability property or the indistinguishability property or both of them. In this paper, we present the first provably secure ID based signcryption scheme in the standard model with both these properties. The unforgeability property of this scheme is based on the hardness of Computational Diffie-Hellman problem and the indistinguishability property of this scheme is based on the hardness of Decisional Bilinear Diffie-Hellman problem. Our scheme is strongly unforgeable in the strong attack mode called insider security. Moreover, our scheme possess an interesting property called public verifiability of the ciphertext. Our scheme integrates cleverly, a modified version of Waters' IBE and a suitably modified version of the ID based signature scheme in the standard model proposed by Paterson et al. However, our security reductions are more efficient. Specifically, while the security reductions for indistinguishability is similar to the bounds of Waters' scheme, the unforgeability reductions are way better than the bounds for Paterson et al.'s scheme.

Keywords: Provable Security, ID based signcryption, Strong Unforgeability, Standard Model, Public Ciphertext Verifiability, Insider Security

1 Introduction

Signcryption aims at providing the confidentiality property of encryption and authentication and non-repudiation properties of signature simultaneously with a cost significantly less than the cost of performing encryption and signature separately. This notion was introduced by Zheng [31] in 1997. The reduction in the computational and communication cost makes the scheme more practical and hence it has numerous real time applications. Fast, compact, secure, unforgeable and non-repudiated key transport, multi-cast, electronic commerce, authenticated email are some of the areas where signcryption is highly applicable.

ID based cryptography, introduced by Shamir in 1984 [22] suggests the use of user identity, such as his e-mail address or his telephone number, as his public key rather than using some arbitrary strings which requires certificates from the Certificate Authority (CA). A Private Key Generator (PKG) is a trusted entity which when given a user's identity computes the private key for the corresponding user and returns it to the user through a secure channel. This method eliminates the need for certificates, which were used in the conventional public key setting.

The first ID based signcryption scheme was proposed by Malone-Lee [18] in 2002. Many ID based signcryption schemes have been proposed since then, adopting many different strategies, thereby reducing computational cost and also reducing the ciphertext size ([5], [8], [17], [7], [2]).

But all these above schemes were proven secure in the random oracle model. Canetti et al. in 2004 [6] showed the limitations and challenges of using random oracle model. The instantiation of random oracles with real world hash functions may result in insecure schemes. So, there is a natural urge to design systems

* This material is based upon work partially supported by Summer Fellowship offered by the Department of Computer Science and Engineering, Indian Institute of Technology, Madras.

that are secure in standard model. It should be noted that the systems that are secure in standard model are in general computationally more expensive than the systems that are secure in random oracle model. We need to pay such an extra cost due to more stringent demands of standard models.

The first ID based signcryption scheme without random oracles was proposed by Yu et al. in 2009 [28] based on Waters' ID based encryption [26]. But their scheme was shown CPA insecure by Wang et al. [24], Zhang et al. [30] and Zhang [29]. Zhang [29] also showed that [28] is SUF-insecure. Meanwhile, Ren and Gu [27] proposed a Signcryption scheme based on Gentry's IBE [9] but it was shown by Wang et al. [25] that it has neither confidentiality nor existential unforgeability. An improved semantically secure scheme was proposed by Jin, Wen and Du [11] again based on Waters IBE but Li et al. [13] showed that the scheme in [11] satisfies neither IND-CCA2 property nor EUF-CMA property. Zhang [29] also proposed a new scheme. But Li et al. in 2011 [15] showed that Zhang's scheme [29] did not have IND-CPA property and they proposed a new scheme claiming it to have both IND-CCA2 and EUF-CMA properties. But the new scheme in [15] satisfies neither IND-CCA2 property nor EUF-CMA property as shown by Selvi et al. in [21]. Li et al. [14] proposed another scheme based on IBE proposed by Kiltz et al. [12] and IBS proposed by Paterson et al. [20]. But Selvi et al. [21] have also shown that there are inconsistencies in the proof of security of [14], thus concluding that all the ID based signcryption schemes proposed till now for the standard model are not provably secure. Selvi et al. [21] have also concluded that achieving a provably secure ID based signcryption scheme in the standard model through direct combination of an ID based signature scheme and an ID based encryption scheme can only be done by the Sign then Encrypt approach. However, for any Sign then Encrypt scheme, $cost\ of\ signcryption = cost\ of\ signature + cost\ of\ encryption$. But our objective of designing a signcryption scheme is to have a scheme that has $cost\ of\ signcryption < cost\ of\ signature + cost\ of\ encryption$ [31]. Hence we need to take a fresh look at the design of the signcryption protocol and arrive at an efficient customized scheme of signcryption. In the subsequent section we present one such novel scheme and formally prove its security.

Hea An, Dodis and Rabin in 2002 [1] introduced the notion of *strong unforgeability*, to avoid the problems due to malleability. If a scheme is malleable, then an adversary can produce a valid signature on a message when another valid signature on the same message is available. So, they proved the unforgeability property of their signcryption scheme using this strong notion. A signature scheme becomes non-malleable when it satisfies this property. There are several transformations available in literature to convert an EUF-CMA secure scheme to a SUF-CMA secure scheme for signature schemes. Some of the transformations available for the standard model are the transformations proposed by Boneh et al. [4], Bellare et al. [3], Teranishi et al. [23] and Huang et al. [10].

The *public ciphertext verifiability* property of a scheme is very useful in low power devices. This property allows any third party application, like firewalls, to verify the validity of the sender and ciphertext without any interaction with the receiver i.e without knowing the receiver's secret key. This will allow the application to prevent the ciphertexts, modified by an adversary, from reaching the devices. Only valid ciphertexts can reach them, thus preventing unnecessary use of their resources for decrypting the invalid ciphertexts. Here, the important property is that, the third party application while verifying should not obtain any knowledge about the message that is signcrypted. This property is provided by the signcryption scheme proposed by Chow et al. [8]. But that scheme was proven secure only in the random oracle model.

1.1 Our Contribution

In this paper we present the first provably secure ID based signcryption scheme without random oracles. Our scheme is based on the ID based signature scheme in the standard model proposed by Paterson et al. [20], which in turn is based on the PKI based signature scheme proposed by Waters [26]. We base the IND-CCIA2 property of our scheme on the hardness of the Decisional Bilinear Diffie Hellman assumption and the SUF-CMIA property of our scheme on the hardness of the Computational Diffie Hellman assumption. The property of *strong unforgeability* is present in our scheme even without using any of the transformations available to convert an existentially unforgeable scheme to a strongly unforgeable scheme in the standard model. The proposed scheme also offers insider security with respect to both confidentiality and unforgeability which ensures that the signcryption scheme is secure even when one among the sender or the receiver colludes with the adversary against the other. The scheme proposed exhibits the crucial property of public ciphertext

verifiability. Recall that all the ID based signcryption schemes in the standard model such as [28], [27], [11], [29] and [15] are completely broken and the most recent scheme proposed by Li et al. [14] has flaws in the proof. Even if the flaws in the proof of [14] are fixed, our scheme has the following advantages over [14].

- The security of our scheme is based on a harder assumption i.e DBDH, compared to the modified DBDH (mDBDH) used by [14].
- Our scheme has a tighter security reduction.
- Our scheme is more efficient than the one in [14].

1.2 Organisation

The rest of this paper is organized as follows. In section 2, preliminaries like bilinear pairing, computational assumptions, a generic ID based signcryption scheme, formal security model for ID based signcryption scheme are explained. We present our ID based signcryption scheme in section 3. We prove the confidentiality property and the strong unforgeability property of our scheme in section 4. The efficiency of our scheme is explained in section 5 and the paper is concluded in section 6.

2 Preliminaries

2.1 Bilinear Pairing

Let $\mathbb{G}$ and $\mathbb{G}_T$ be multiplicative groups of prime order p and let g be generator of $\mathbb{G}$. The bilinear map $\hat{e}$ is admissible only if it satisfies the following conditions:

- ***Bilinearity.*** For all $g_1, g_2, g_3 \in \mathbb{G}$,
  - $\hat{e}(g_1 g_2, g_3) = \hat{e}(g_1, g_3)\hat{e}(g_2, g_3)$
  - $\hat{e}(g_1, g_2 g_3) = \hat{e}(g_1, g_2)\hat{e}(g_1, g_3)$
  - $\hat{e}(g_1^a, g_2^b) = \hat{e}(g_1, g_2)^{ab}$ for all $a, b \in \mathbb{Z}_p$.
- ***Non-Degeneracy.*** For all $g_1, g_2 \in \mathbb{G}$, $\hat{e}(g_1, g_2) \neq I_{\mathbb{G}_T}$, where $I_{\mathbb{G}_T}$ is the identity element of $\mathbb{G}_T$.
- ***Computability.*** There exists an efficient algorithm to compute $\hat{e}(g_1, g_2)$ for all $g_1, g_2 \in \mathbb{G}$.

2.2 Computational Assumptions

In this section, we review the computational assumptions relevant to the protocol we propose.

Computational Diffie-Hellman Problem (CDH) Given $(g, g^a, g^b) \in \mathbb{G}^3$ for unknown $a, b \in \mathbb{Z}_p$, the CDH problem in $\mathbb{G}$ is to compute g^{ab}.

Definition. The advantage of any probabilistic polynomial time algorithm $\mathcal{A}$ in solving the CDH problem in $\mathbb{G}$ is defined as:

$$Adv_{\mathcal{A}}^{CDH} = Pr\left[\mathcal{A}(g, g^a, g^b) = g^{ab} \mid a, b \in \mathbb{Z}_p\right]$$

The *CDH Assumption* is that, for any probabilistic polynomial time algorithm $\mathcal{A}$, the advantage $Adv_{\mathcal{A}}^{CDH}$ is negligibly small.

Decisional Bilinear Diffie-Hellman Problem (DBDH) Given $(g, g^a, g^b, g^c, \alpha) \in \mathbb{G}^4 \times \mathbb{G}_T$ for unknown $a, b, c \in \mathbb{Z}_p$, the DBDH problem in $\mathbb{G}$ is to decide if $\alpha = \hat{e}(g, g)^{abc}$.

Definition. The advantage of any probabilistic polynomial time algorithm $\mathcal{A}$ in solving the DBDH problem in $\mathbb{G}$ is defined as:

$$Adv_{\mathcal{A}}^{DBDH} = Pr\left[\mathcal{A}(g, g^a, g^b, g^c, \hat{e}(g, g)^{abc}) = 1\right] - Pr\left[\mathcal{A}(g, g^a, g^b, g^c, \alpha) = 1\right] \mid a, b, c \in \mathbb{Z}_p\right]$$

The *DBDH Assumption* is that, for any probabilistic polynomial time algorithm $\mathcal{A}$, the advantage $Adv_{\mathcal{A}}^{DBDH}$ is negligibly small.

2.3 ID based Signcryption

A generic ID based signcryption scheme consists of the following four algorithms.

- ***Setup:*** This algorithm is run by the Private Key Generator (PKG). When given a security parameter k, this algorithm outputs public parameters *params* and a master secret key MSK. PKG keeps the corresponding MSK as its secret value.
- ***Extract:*** When given an Identity ID, the PKG runs this algorithm using the *params* and MSK and generates the private key d_u for the user. The PKG then transmits the generated private key to the corresponding user through a secure channel.
- ***Signcrypt:*** This algorithm is run by the sender. It takes as input, the public parameters *params*, the private key d_A of the sender, the identity of the receiver ID_B and the message m to be sent to the receiver. The signcryption σ is produced as output which is sent to the receiver.
- ***Unsigncrypt:*** On receiving the signcryption σ from the sender, the receiver runs this algorithm. The public parameters *params*, the identity of the sender ID_A, the private key of the receiver d_B and the signcryption σ are given as input to this algorithm. The message m is obtained as output if the signcryption is valid or $\perp$ is given as output.
  For the consistency of the signcryption algorithm, if $\sigma = Signcrypt(params, d_A, ID_B, m)$, then $m = Unsigncrypt(params, ID_A, d_B, \sigma)$.

2.4 Security model for ID based signcryption

Indistinguishability In 2002, Malone-Lee [18] proposed the first ID based signcryption scheme. He extended the semantic security of encryption schemes to signcryption schemes as *Indistinguishability of ID based signcryption under Adaptive Chosen Ciphertext Attack* (IND-IBSC-CCA2). Later, Chow et al. [8] used a stronger notion of security by allowing the adversary to adaptively choose the identities to create a forgery during the challenge phase. This is similar to the one proposed in [16]. This model was termed as *Indistinguishability of ID based signcryption under Adaptive Chosen Ciphertext and Identity Attack* (IND-IBSC-CCIA2). This is the strongest notion available in the literature for proving the Indistinguishability property of the signcryption schemes. The formal definition is given below.

A signcryption scheme is semantically secure against chosen ciphertext and identity attack (IND-IBSC-CCIA2) if no probabilistic polynomial time adversary $\mathcal{A}$ has a non-negligible advantage in the following game.

1. The challenger $\mathcal{C}$ runs the **Setup** algorithm and sends the public parameters to the adversary $\mathcal{A}$
2. **Training Phase 1:** The adversary $\mathcal{A}$ can ask a polynomially bound number of queries to the following oracles.
   - **Extract Oracle:** When $\mathcal{A}$ queries for the private key of an identity ID, the challenger $\mathcal{C}$ runs the **Extract** algorithm giving the ID and *params* as input. $\mathcal{C}$ forwards the private key d_u of ID output by the Extract algorithm to $\mathcal{A}$.
   - **Signcrypt Oracle:** $\mathcal{A}$ can ask for the signcryption on any message m from any sender identity ID_A to any receiver identity ID_B. When $\mathcal{A}$ does so, $\mathcal{C}$ runs the **Extract** algorithm for the sender identity ID_A and gets the private key d_A of ID_A. $\mathcal{C}$ then inputs $\langle m, d_A, ID_B \rangle$ into the **Signcrypt** algorithm and forwards its output σ to $\mathcal{A}$.
   - **Unsigncrypt Oracle:** $\mathcal{A}$ queries for the unsigncryption of the ciphertext σ by producing the sender identity ID_A and receiver identity ID_B. The challenger $\mathcal{C}$ runs the **Extract** algorithm to find the private key d_B of the receiver ID_B. $\mathcal{C}$ then runs the **Unsigncrypt** algorithm giving $\langle \sigma, ID_A, d_B \rangle$ as input and forwards the output m or $\perp$ to $\mathcal{A}$.

   During this phase $\mathcal{A}$ can produce its queries adaptively i.e every query can be asked dependent on the output of the previous queries.
3. **Challenge Phase** At the end of Phase 1, $\mathcal{A}$ chooses two plaintext messages m_0^*, $m_1^* \in \{0,1\}^{l_m}$, two identities i.e. sender identity ID_A^* and receiver identity ID_B^* on which it wishes to be challenged and sends them to the challenger $\mathcal{C}$. In this case, $\mathcal{A}$ should not have queried the Extract oracle for ID_B^*. $\mathcal{C}$ takes a bit b randomly from $\{0,1\}$ and runs **Signcrypt**(m_b^*,d_A^*,ID_B^*), where d_A^* is the output of **Extract**(ID_A^*). $\mathcal{C}$ sends the output σ^* to $\mathcal{A}$ as the challenge ciphertext.

4. **Training Phase 2:** The adversary $\mathcal{A}$, after receiving σ^* can ask again for polynomially bound number of queries on the above mentioned oracles adaptively in the same way as in Phase 1 except that $\mathcal{A}$ cannot ask for the **Extract**(ID_B^*) query and **Unsigncrypt** query involving $\langle \sigma^*, ID_A^*, ID_B^* \rangle$.
5. Once this Phase 2 of Training is over, $\mathcal{A}$ outputs b'. $\mathcal{A}$ wins this game if $b' = b$.

The advantage of adversary $\mathcal{A}$ in the above game is defined by $Adv(\mathcal{A}) = (2 \times Pr(b' = b) - 1)$.

The importance of this security model is that the adversary $\mathcal{A}$ can ask for the private key d_A^* of the sender whose identity is ID_A^* during Phase 2. This captures the *insider* security model, which means that $\mathcal{A}$ will not have any added advantage in the above game even when the private key of the sender is leaked.

Also, $\mathcal{A}$ is allowed to query the Signcrypt oracle with the challenge messages m_0^* or m_1^* with the sender identity as ID_A^* and receiver identity as ID_B^*.

Unforgeability Malone-Lee [18] proposed the *Existential Unforgeability of ID based signcryption under Chosen Message Attack* (EUF-IBSC-CMA). Later, Chow et al. [8] proposed a stronger notion of security called *Existential Unforgeability of ID based signcryption under Chosen Message and Identity Attack* (EUF-IBSC-CMIA), where the adversary can not only choose the message to attack adaptively but also the identities on which it is going to attack. This notion is defined by the game between challenger and adversary as given below.

An ID based signcryption scheme is said to have the property of *Existential Unforgeability under Chosen Message and Identity Attack* if no probabilistic polynomial time adversary $\mathcal{A}$ has a non-negligible advantage in the following game.

1. The challenger $\mathcal{C}$ runs the **Setup** algorithm and generates the public parameters and the Master Secret Key MSK. $\mathcal{C}$ then gives the public parameters *params* to the adversary $\mathcal{A}$
2. Now the adversary $\mathcal{A}$ can ask a polynomially bound number of queries to any of the following oracles.
   - **Extract Oracle:** When $\mathcal{A}$ queries for the private key of an identity ID, the challenger $\mathcal{C}$ runs the **Extract** algorithm giving the ID, *params* and MSK as input. $\mathcal{C}$ forwards the output d_u given by the algorithm to the adversary $\mathcal{A}$.
   - **Signcrypt Oracle:** $\mathcal{A}$ can ask for the signcryption on any message m by the sender identity ID_A for the receiver identity ID_B. In this case, $\mathcal{C}$ runs the **Extract** algorithm for the sender identity ID_A and gets the private key d_A of ID_A as output. $\mathcal{C}$ then inputs $\langle m, d_A, ID_B \rangle$ into the **Signcrypt** algorithm and forwards its output σ to $\mathcal{A}$.
   - **Unsigncrypt Oracle:** When $\mathcal{A}$ queries for the unsigncryption of the ciphertext σ by producing the sender identity ID_A and receiver identity ID_B. The challenger $\mathcal{C}$ first runs the **Extract** algorithm for finding the private key d_B of the receiver ID_B. $\mathcal{C}$ then runs the **Unsigncrypt** algorithm inputting $\langle \sigma, ID_A, d_B \rangle$ and forwards its output m to $\mathcal{A}$.

   During this phase $\mathcal{A}$ can produce its queries adaptively i.e every query is dependant on the previous queries.
3. At the end this training phase, $\mathcal{A}$ outputs the forgery $\langle \sigma^*, ID_A^*, ID_B^* \rangle$ for some message m^*. This forgery is valid when ID_A^* is not queried to the **Extract** oracle and if $\langle m^*, ID_A^*, ID_B^* \rangle$ is not already queried to the **Signcrypt** oracle.
4. $\mathcal{A}$ wins the game if σ^* is a valid forgery on the message m^* as signcrypted by the identity ID_A^* intended for the identity ID_B^*.

The advantage of adversary $\mathcal{A}$ in the above game is defined by

$$Adv(\mathcal{A}) = Pr\left[Unsigncrypt(\sigma^*, ID_A^*, ID_B^*) \ = \ m^*\right]$$

In this security model, the importance is that the adversary can query the **Extract** oracle for the identity of the receiver ID_B^* in the above game which captures the insider security model for unforgeability. So, even when the private key of the intended receiver is leaked, the adversary $\mathcal{A}$ will not have any added advantage in producing a valid forgery in the above game. But the restriction here is that $\langle m^*, ID_A^*, ID_B^* \rangle$ should not have been queried already to the **Signcrypt** oracle. The work done by Li et al. [14] has used similar security models which provide insider security.

Strong Unforgeability Hea An et al. [1] proposed that there is no necessity for an adversary to produce forgery on a message that is not already queried. Forgery can also be produced on the message that is queried already to the Signcrypt oracle with the condition that the forged signcryption on m is not the same as the one that is output by the **Signcrypt** oracle for the same message m, with the same sender and the same receiver as the forgery. This notion is called *Strong Unforgeability*. Our new scheme satisfies the notion of *Strong Unforgeability of ID based signcryption under Chosen Message and Identity Attack* (SUF-IBSC-CMIA). This is the strongest security notion available for proving the unforgeability property of signcryption schemes. We state this notion formally as follows.

An ID based signcryption scheme is said to have the property of Strong Unforgeability under Chosen Message and Identity Attack if there is no probabilistic polynomial time adversary $\mathcal{A}$ has a non-negligible advantage in the game described as follows.

1. The challenger follows the same procedure as EUF-IBSC-CMIA game during the setup and the training phases.
2. After training is over, the adversary $\mathcal{A}$, produces $\langle \sigma^*, ID_A^*, ID_B^* \rangle$ for the message m^*, where ID_A^* is not queried to the **Extract** oracle and σ^* is not the output of the **Signcrypt** query asked by $\mathcal{A}$ with $\langle m^*, ID_A^*, ID_B^* \rangle$ as input.
3. $\mathcal{A}$ wins the game if σ^* is a valid forgery on the message m^* as signcrypted from the sender identity ID_A^* to the receiver identity ID_B^*.

The advantage of adversary $\mathcal{A}$ in the above game is defined by

$$Adv(\mathcal{A}) = Pr\left[Unsigncrypt(\sigma^*, ID_A^*, ID_B^*) \ = \ m^*\right]$$

In the above security model, $\mathcal{A}$ can produce any valid $\langle \sigma^*, ID_A^*, ID_B^* \rangle$ tuple for the message m^*, where $\langle m^*, \sigma^* \rangle$ is not the output of any **Signcrypt** query with ID_A^* and ID_B^* as the sender and receiver identities during the training phase. So, m^* may have been queried already to the Signcrypt oracle provided that σ^* is not the output of the oracle during that query with the sender and receiver identities being the same during that query and the forgery.

3 Our Scheme

Setup

Consider groups $\mathbb{G}, \mathbb{G}_T$ of prime order p whose size is determined by the security parameter k. Let g be the generator of the group $\mathbb{G}$. There exists a bilinear map defined by $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, which is efficiently computable. Now, choose $\alpha \in \mathbb{Z}_p$ randomly and compute $g_1 = g^\alpha$. Randomly pick g_2, h_2 from $\mathbb{G}$ and compute g_2^α, h_2^α. Also, choose h_1, h_3 randomly from $\mathbb{G}$. Choose u', v', m' randomly from the group $\mathbb{G}$ and also choose vectors $\mathbb{U} = (u_i)$ and $\mathbb{V} = (v_i)$ each of length n_u and $\mathbb{M} = (m_i)$ of length l, whose elements are randomly chosen from group $\mathbb{G}$. Here, n_u is the length of the identity strings that are used. Let n_m be the length of the message sent. There are four one-way, collision resistant cryptographic hash functions $H_1 : \mathbb{G}_T \times \{0,1\}^{l_\tau} \to \{0,1\}^{n_m}$, $H_2 : \{0,1\}^{|p|+n_u+l_\tau} \to \{0,1\}^l$, $H_3 : \mathbb{G} \to \mathbb{Z}_p^*$ and $H_4 : \{0,1\}^{n_m+|p|+n_u} \to \mathbb{Z}_p^*$, where l is large enough that the hash functions are collision resistant and $l_\tau \approx 40$. Note that a typical value of l could be 256 and a random bit string of length l cannot be guessed in polynomial time. The system parameter *params* is given by $\langle \mathbb{G}, \mathbb{G}_T, \hat{e}, H_1, H_2, H_3, H_4, g, g_1, g_2, h_1, h_2, h_3, u', v', m', \mathbb{U}, \mathbb{V}, \mathbb{M} \rangle$. The master secret key of the system is $\langle \alpha, g_2^\alpha, h_2^\alpha \rangle$. The following algorithms define our scheme.

Extract(u, *params*, MSK)

Let an identity of a user u be represented by ID_u which is a bit string of length n_u and let $ID_u[i]$ be the i^{th} bit of ID_u. Define $\Omega_u \subseteq \{1, 2, ..., n_u\}$ to be the set of indices i such that $ID_u[i] = 1$. The private key of a user u is constructed by choosing a random $r_u \in \mathbb{Z}_p^*$ and then computing

$$d_u = (d_S, d_{US}, d_R) = (g_2^\alpha (u' \prod_{i \in \Omega_u} u_i)^{r_u}, h_2^\alpha (v' \prod_{i \in \Omega_u} v_i)^{r_u}, g^{r_u})$$

Signcrypt(*params*, d_A, B, m)

The private key of the sender A with identity ID_A as given by the PKG is

$$d_A = (d_{S_A}, d_{US_A}, d_{R_A}) = (g_2^{\alpha}(u' \prod_{i \in \Omega_A} u_i)^{r_A}, h_2^{\alpha}(v' \prod_{i \in \Omega_A} v_i)^{r_A}, g^{r_A})$$

where $\Omega_A \subseteq \{1, 2, ..., n_u\}$ is the set of indices i such that $ID_A[i] = 1$. Now, when given a message $m \in \{0,1\}^{n_m}$ signcryption on the message is done by the sender A as follows.

- Choose $r \in \mathbb{Z}_p$ randomly and compute $\sigma_1 = g^r \in \mathbb{G}$
- Encrypt the message as $\sigma_2 = H_1(\hat{e}\,(g_1, h_2)^r, \tau) \oplus m \in \{0,1\}^{n_m}$, where $\tau \in_R \{0,1\}^{l_\tau}$
- Compute $\sigma_3 = (v' \prod_{i \in \Omega_B} v_i)^r \in \mathbb{G}$, where Ω_B is the set of vertices i such that $ID_B[i] = 1$. Here, B is the receiver of the message.
- Set $\sigma_4 = d_{R_A} \in \mathbb{G}$
- Compute $\lambda = H_3(\sigma_1)$, $\beta = H_2(\sigma_4, ID_A, \tau)$, $\rho = H_4(\sigma_2, \sigma_3, ID_B)$
- Compute $\sigma_5 = d_{S_A}(m' \prod_{j \in \overline{\beta}} m_j)^r \left(h_1^{\lambda} h_3\right)^{r\rho} \in \mathbb{G}$, where $\overline{\beta} \subseteq \{1, 2, ..., l\}$ denotes the set of indices j such that $\beta[j] = 1$

The ciphertext $\sigma = \langle \sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \tau \rangle$ is sent to the receiver.

The size of the ciphertext formed is $4|p| + n_m + l_\tau$. Note that this scheme achieves the property of strong unforgeability without using any of the transformations available to convert an existentially unforgeable scheme to a strongly unforgeable one.

Unsigncrypt(*params*, A, d_B, σ)

When the receiver B receives the ciphertext $\sigma = \langle \sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \tau \rangle$, he proceeds as follows.

- The private key d_B received from the PKG is

$$d_B = (d_{S_B}, d_{US_B}, d_{R_B}) = (g_2^{\alpha}(u' \prod_{i \in \Omega_B} u_i)^{r_B}, h_2^{\alpha}(v' \prod_{i \in \Omega_B} v_i)^{r_B}, g^{r_B})$$

- Compute $\lambda = H_3(\sigma_1)$, $\beta = H_2(\sigma_4, ID_A, \tau)$, $\rho = H_4(\sigma_2, \sigma_3, ID_B)$
- Then, using β, ρ and λ, check the validity of σ as follows

$$\hat{e}(\sigma_5, g) \stackrel{?}{=} \hat{e}(g_1, g_2)\, \hat{e}(u' \prod_{i \in \Omega_A} u_i, \sigma_4)\, \hat{e}((m' \prod_{j \in \overline{\beta}} m_j)(h_1^{\lambda} h_3)^{\rho}, \sigma_1) \tag{1}$$

  where Ω_A is the set of indices i such that $ID_A[i] = 1$ and $\overline{\beta} \subseteq \{1, 2, ..., l\}$ denotes the set of indices j such that $\beta[j] = 1$

- If σ is invalid, reject σ and halt.
- If σ is valid, compute $\hat{e}\,(g_1, h_2)^r = \dfrac{\hat{e}\,(d_{US_B}, \sigma_1)}{\hat{e}\,(d_{R_B}, \sigma_3)}$
- Obtain the message as $m = \sigma_2 \oplus H_1(\hat{e}\,(g_1, h_2)^r, \tau)$

The above verification process stated in equation (1) can be done by any user who has access to σ, because all the components used in the verification process are either the values in *params* $\langle g, g_1, g_2, u', \mathbb{U}, m', \mathbb{M}, h_1, h_3 \rangle$, components of the ciphertext $\langle \sigma_1, \sigma_4, \sigma_5 \rangle$ or components that are derived from the ciphertext $\langle \lambda, \beta, \rho \rangle$. and thus the integrity and validity of the sender and the ciphertext can be verified by anyone. This gives the property of **Public Ciphertext Verifiability** to our scheme.

Correctness of the Unsigncrypt algorithm

When the receiver B receives the ciphertext $\sigma = \langle \sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \tau \rangle$, he can calculate $\hat{e}\,(g_1, h_2)^r$ using his Unsigncrypt private key $\langle d_{US_B}, d_{R_B} \rangle$ by $\dfrac{\hat{e}\,(d_{US_B}, \sigma_1)}{\hat{e}\,(d_{R_B}, \sigma_3)}$ as shown below.

$$\frac{\hat{e}\,(d_{US_B}, \sigma_1)}{\hat{e}\,(d_{R_B}, \sigma_3)} = \frac{\hat{e}\,(h_2^{\alpha}\,(v' \prod_{i \in \Omega_B} v_i)^{r_B}, g^r)}{\hat{e}\,(g^{r_B}, (v' \prod_{i \in \Omega_B} v_i)^r)} = \frac{\hat{e}\,(h_2^{\alpha}, g^r)\,\hat{e}((v' \prod_{i \in \Omega_B} v_i)^{r_B}, g^r)}{\hat{e}\,(g^{r_B}, (v' \prod_{i \in \Omega_B} v_i)^r)} = \hat{e}\,(g_1, h_2)^r$$

The correctness of the verification procedure is shown below.

$$\begin{aligned}
\hat{e}(\sigma_5, g) &= \hat{e}(d_{S_A}(m' \textstyle\prod_{j \in \overline{\beta}} m_j)^r (h_1^{\lambda} h_3)^{r\rho}, g) \\
&= \hat{e}\,(g_2^{\alpha}\,(u' \textstyle\prod_{i \in \Omega_A} u_i)^{r_A} (m' \prod_{j \in \overline{\beta}} m_j)^r (h_1^{\lambda} h_3)^{r\rho}, g) \\
&= \hat{e}\,(g_2^{\alpha}, g)\,\hat{e}\,((u' \textstyle\prod_{i \in \Omega_A} u_i)^{r_A}, g)\,\hat{e}\,((m' \prod_{j \in \overline{\beta}} m_j)^r, g)\,\hat{e}((h_1^{\lambda} h_3)^{r\rho}, g) \\
&= \hat{e}\,(g^{\alpha}, g_2)\,\hat{e}\,(u' \textstyle\prod_{i \in \Omega_A} u_i, g^{r_A})\,\hat{e}\,(m' \prod_{j \in \overline{\beta}} m_j, g^r)\,\hat{e}((h_1^{\lambda} h_3)^{\rho}, g^r) \\
&= \hat{e}(g_1, g_2)\,\hat{e}(u' \textstyle\prod_{i \in \Omega_A} u_i, \sigma_4)\,\hat{e}((m' \prod_{j \in \overline{\beta}} m_j)(h_1^{\lambda} h_3)^{\rho}, \sigma_1)
\end{aligned}$$

where the definitions of $\overline{\beta}$ and Ω_A are as explained in the **Unsigncrypt** algorithm.

4 Security

4.1 Indistinguishability

We first prove the Indistinguishability property, *Indistinguishability of ID based signcryption under Adaptive Chosen Ciphertext and Identity Attack* (IND-IBSC-CCIA2) of our scheme with the following theorem.

Theorem 1. *If there exists an IND-IBSC-CCIA2 adversary for our scheme which can distinguish ciphertexts during the IND-IBSC-CCIA2 game explained above, with a non-negligible probability ϵ when it runs for a polynomial time t, asking at most q_E extract queries, q_S signcrypt queries and q_{US} unsigncrypt queries, then there exists another algorithm, which can solve the Decisional Bilinear Diffie-Hellman (DBDH) problem with probability ϵ' in polynomial time t', where*

$$\epsilon' \geq \frac{\epsilon}{4(q_E)(n_u + 1)}$$

$$t' \leq t + \mathcal{O}((n_u q_E + (n_u + l)(q_S + q_{US}))t_m + (q_E + q_S + q_{US})t_e + (q_S + q_{US})t_p)$$

where n_u is the length of the identity string, t_m, t_e, t_p are the time required for each multiplication, each exponentiation and each bilinear pairing respectively and l is a value large enough such that the hash functions outputting $\{0,1\}^l$ in the scheme are collision resistant.

Proof

Let us assume that a $(\epsilon, t, q_E, q_S, q_{US})$-adversary $\mathcal{A}$ for our scheme exists. We will construct another algorithm $\mathcal{B}$ from this adversary $\mathcal{A}$, who can solve the Decisional Bilinear Diffie- Hellman (DBDH) problem with a non-negligible probability ϵ' in polynomial time t'.

The algorithm $\mathcal{B}$ receives a DBDH tuple $\langle g, g^a, g^b, g^c, T \rangle \in \mathbb{G}^4 \times \mathbb{G}_T$, where g is a generator of a prime order group $\mathbb{G}$ of order p. $\mathcal{B}$ simulates a challenger for the adversary $\mathcal{A}$ to decide whether T is $\hat{e}(g,g)^{abc}$ or not. This simulation is described as follows:

Setup

The simulator $\mathcal{B}$ sets $l_u = 2(q_E)$, where q_E is the number of Extract queries. Here, the values q_S and q_{US} are not bounded because the Signcrypt and the Unsigncrypt queries do not abort when an Extract query of the sender or receiver identity, used in any Signcrypt or Unsigncrypt query, aborts. This will be evident from the explanation for the Signcrypt and the Unsigncrypt oracles. $\mathcal{B}$ then chooses an integer k_u randomly such that $0 \leq k_u \leq n_u$. For the given values of q_E, q_S, q_{US} and n_u, we assume that $l_u(n_u + 1) < p$. Then, $\mathcal{B}$ chooses $x' \in \mathbb{Z}_{l_u}$ randomly and also chooses a vector $\mathbb{X} = (x_i)$ of length n_u where the elements of $\mathbb{X}$ are chosen randomly from $\mathbb{Z}_{l_u}$. $\mathcal{B}$ chooses an integer y' randomly from $\mathbb{Z}_p$ and a vector $\mathbb{Y} = (y_i)$ of length n_u, where the elements of $\mathbb{Y}$ are also chosen randomly from $\mathbb{Z}_p$.

Here we define a pair of functions for a user with identity ID_u as follows:

$$F(u) = x' + \textstyle\sum_{i \in \Omega_u} x_i - l_u k_u \qquad\qquad J(u) = y' + \textstyle\sum_{i \in \Omega_u} y_i$$

The simulator now sets the public parameters as follows:

$$g_1 = g^a \qquad g_2 = g^d \qquad h_1 = g_1^{(\lambda^*)^{-1}} \qquad h_3 = g_1^{-1} g^\theta \qquad h_2 = g^b$$

where d and θ are chosen randomly from $\mathbb{Z}_p$ and $\lambda^* = H_3(g^c)$. The values g^a, g^b, g^c are from the DBDH tuple given to the challenger $\mathcal{C}$.

$$v' = h_2^{x' - l_u k_u} g^{y'} \qquad v_i = h_2^{x_i} g^{y_i} \qquad v' \textstyle\prod_{i \in \Omega} v_i = h_2^{F(u)} g^{J(u)}$$

Finally, $\mathcal{B}$ chooses two integers e' and f' randomly from $\mathbb{Z}_p$ and two vectors $\mathbb{E} = (e_i)$ and $\mathbb{F} = (f_i)$ of lengths n_u and n_m respectively, where the elements of $\mathbb{E}$ and $\mathbb{F}$ are chosen randomly from $\mathbb{Z}_p$.

For any identity ID_u,
$u' = g^{e'} \qquad u_i = g^{e_i} \qquad u' \prod_{i \in \Omega_u} u_i = g^{e'} g^{\sum_{i \in \Omega_u} e_i} = g^{e' + \sum_{i \in \Omega_u} e_i}$

where $\Omega_u \subseteq \{1, 2, .., n_u\}$ is the set of indices i where $ID_u[i] = 1$.

For any β got for a message m as explained in the scheme,

$$m' = g^{f'} \qquad m_i = g^{f_i} \qquad m' \textstyle\prod_{i \in \overline{\beta}} m_i = g^{f'} g^{\sum_{i \in \overline{\beta}} f_i} = g^{f' + \sum_{i \in \overline{\beta}} f_i}$$

There are four one-way, collision resistant, cryptographic hash functions $H_1 : \mathbb{G}_T \times \{0,1\}^{l_\tau} \to \{0,1\}^{n_m}$, $H_2 : \{0,1\}^{|p| + n_u + l_\tau} \to \{0,1\}^l$, $H_3 : \mathbb{G} \to \mathbb{Z}_p^*$ and $H_4 : \{0,1\}^{n_m + |p| + n_u} \to \mathbb{Z}_p^*$, where l is large enough that the hash functions are collision resistant and $l_\tau \approx 40$. Note that a typical value of l could be 256 and a random bit string of length l cannot be guessed in polynomial time.

Training Phase 1

The simulator during this phase answers to the queries from the adversary $\mathcal{A}$ as follows.

Extract Queries
The simulator $\mathcal{B}$ does not know the master secret key h_2^a. So, when the adversary $\mathcal{A}$ asks for the private key of an identity ID_u, $\mathcal{B}$ responds as follows. $\mathcal{B}$ calculates $F(u)$ for the identity ID_u. If $F(u) = 0 \bmod p$, it aborts. Otherwise, $\mathcal{B}$ randomly chooses $r_u \in \mathbb{Z}_p$ and calculates the private key as

$$d_u = (d_S, d_{US}, d_R) = \left(g_1^d \left(u' \prod_{i \in \Omega_u} u_i \right)^{r_u} g_1^{-(e' + \sum_{i \in \Omega_u} e_i)/F(u)}, g_1^{-J(u)/F(u)} \left(v' \prod_{i \in \Omega_u} v_i \right)^{r_u}, g_1^{-1/F(u)} g^{r_u} \right)$$

where $\Omega_u \subseteq \{1, 2, ..., n_u\}$ is the set of indices i such that $ID_u[i] = 1$.

The correctness of this equation is shown as follows:

$$\begin{aligned}
d_S &= g_1^d \left(u' \prod_{i\in\Omega_u} u_i\right)^{r_u} g_1^{-(e'+\sum_{i\in\Omega_u} e_i)/F(u)} \\
&= g^{ad}\, g^{(e'+\sum_{i\in\Omega_u} e_i)r_u} g^{-a(e'+\sum_{i\in\Omega_u} e_i)/F(u)} \\
&= g_2^a \left(u' \prod_{i\in\Omega_u} u_i\right)^{r_u - a/F(u)} \\
d_{US} &= g_1^{-J(u)/F(u)} \left(v' \prod_{i\in\Omega_u} v_i\right)^{r_u} \\
&= h_2^a\, h_2^{-a}\, g_1^{-J(u)/F(u)} (h_2^{F(u)}\, g^{J(u)})^{r_u} \\
&= h_2^a\, h_2^{-aF(u)/F(u)}\, g^{-aJ(u)/F(u)} (h_2^{F(u)}\, g^{J(u)})^{r_u} \\
&= h_2^a \left(h_2^{F(u)}\, g^{J(u)}\right)^{-a/F(u)} (h_2^{F(u)}\, g^{J(u)})^{r_u} \\
&= h_2^a \left(v' \prod_{i\in\Omega_u} v_i\right)^{r_u - a/F(u)}
\end{aligned}$$

Here, we can write $\overline{r}_u = r_u - a/F(u)$. Thus the private key generated by the simulator can be written as

$$d_u = (d_S, d_{US}, d_R) = (g_2^a\, (u' \prod_{i\in\Omega_u} u_i)^{\overline{r}_u}, h_2^a\, (v' \prod_{i\in\Omega_u} v_i)^{\overline{r}_u}, g^{\overline{r}_u})$$

which is a valid and mathematically consistent private key for the identity ID_u queried by $\mathcal{A}$.

Signcrypt Queries
When $\mathcal{A}$ queries the Signcrypt oracle for signcryption of a message m by the user with identity ID_A as sender and the user with identity ID_B as the intended receiver, $\mathcal{B}$ simulates a valid ciphertext as follows.

$\sigma_1 = g^r$, where $r \in \mathbb{Z}_p$ is randomly chosen by $\mathcal{B}$

$\sigma_2 = H_1(\hat{e}(g_1, h_2)^r, \tau) \oplus m$, where $\tau \in_R \{0,1\}^{l_\tau}$

$\sigma_3 = \left(v' \prod_{i\in\Omega_B} v_i\right)^r$

$\sigma_4 = g^{r_A}$, where r_A is the randomness stored for ID_A in the list l_r. Otherwise choose $r_A \in \mathbb{Z}_p$ randomly and store it in l_r. Note that l_r is the list that stores $\langle ID_u, r_u \rangle$ tuples.

$\beta = H_2(\sigma_4, ID_A, \tau)$, $\rho = H_4(\sigma_2, \sigma_3, ID_B)$ and $\lambda = H_3(\sigma_1)$

$$\sigma_5 = g_1^d \left(u' \prod_{i\in\Omega_A} u_i\right)^{r_A} \left(m' \prod_{j\in\overline{\beta}} m_j\right)^r (h_1^\lambda h_3)^{r\rho} = g_2^a \left(u' \prod_{i\in\Omega_A} u_i\right)^{r_A} \left(m' \prod_{j\in\overline{\beta}} m_j\right)^r (h_1^\lambda h_3)^{r\rho}$$

where $\overline{\beta} \subseteq \{1, 2, ..., l\}$ denotes the set of indices j such that $\beta[j] = 1$. The equation above for σ_5 is correct because $g_1^d = (g^a)^d = (g^d)^a = g_2^a$.

The ciphertext $\sigma = \langle \sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \tau \rangle$ is sent to the adversary $\mathcal{A}$. Here, the Signcrypt queries never abort and they do not need an Extract query for the sender identity ID_A within them.

Unsigncrypt Queries
When $\mathcal{A}$ queries $\langle \sigma, ID_A, ID_B \rangle$ i.e the unsigncryption of the ciphertext σ which was signcrypted by the sender ID_A for the intended receiver ID_B, to the Unsigncrypt oracle simulated by $\mathcal{B}$, it proceeds as follows. $\mathcal{B}$ does an Extract query for the receiver identity ID_B. If the Extract query does not abort, $\mathcal{B}$ receives the

private key for ID_B as output from the Extract oracle as

$$d_B = (d_{S_B}, d_{US_B}, d_{R_B}) = (g_2^a\,(u' \prod_{i \in \Omega_B} u_i)^{r_B}, h_2^a\,(v' \prod_{i \in \Omega_B} v_i)^{r_B}, g^{r_B})$$

The simulator uses the private keys d_{US_B}, d_{R_B} got from the extract oracle to unsigncrypt the ciphertext σ using the Unsigncrypt algorithm as given in the Scheme.

If the extract query aborts i.e if $F(B) = 0 \ mod \ p$, the simulator proceeds as follows. The simulator calculates Δ as given below.

$$\Delta = \frac{\sigma_5}{g_1^d\,\sigma_4^{(e' + \sum_{i \in \Omega_A} e_i)}\,\sigma_1^{(f' + \sum_{i \in \overline{\beta}} f_i)}\,\sigma_1^{\theta\rho}} = \frac{\sigma_5}{d_{S_A}(u' \prod_{i \in \Omega_A} u_i)^{r_A}(m' \prod_{j \in \overline{\beta}} m_j)^r \sigma_1^{\theta\rho}} = \frac{(h_1^{\lambda} h_3)^{r\rho}}{\sigma_1^{\theta\rho}} = (g_1^{(\lambda/\lambda^*)-1})^{r\rho}$$

Then, we calculate $\Delta^* = \Delta^{(((\lambda/\lambda^*)-1)\rho)^{-1}} = g_1^r$, where $\lambda = H_3(\sigma_1)$.

Now, we can obtain the message as follows.
$m = \sigma_2 \oplus H_1(\hat{e}(\Delta^*, h_2), \tau) = \sigma_2 \oplus H_1(\hat{e}(g_1^r, h_2), \tau) = \sigma_2 \oplus H_1(\hat{e}(g_1, h_2)^r, \tau)$

This message can be returned if the verification in Eq.(1) is satisfied. Thus, the Unsigncrypt queries never abort even if the Extract queries for the corresponding receiver identities abort.

Challenge Phase

The adversary $\mathcal{A}$ can adaptively ask polynomially bound number of these Extract, Signcrypt and Unsigncrypt queries to $\mathcal{B}$. When $\mathcal{A}$ decides that training is enough, it produces two messages m_0^* and m_1^* along with the sender identity ID_A^* and receiver identity ID_B^* adaptively and sends them to the challenger. The challenger randomly chooses $\gamma \in \{0, 1\}$ and then simulates the challenge ciphertext as follows.

$\sigma_1^* = g^c$

$\sigma_2^* = H_1(T, \tau^*) \oplus m_\gamma^*$, where g^c and T are taken by $\mathcal{B}$ from the DBDH tuple given and $\tau^* \in_R \{0,1\}^{l_\tau}$.

$\sigma_3^* = \left(v' \prod_{i \in \Omega_B} v_i\right)^c = (g^c)^{J(B^*)}$, where $F(B^*) = 0 \ mod \ p$

$\sigma_4^* = g^{r_A}$, where $r_A \in \mathbb{Z}_p$ is randomly chosen

$\beta^* = H_2(\sigma_4^*, ID_A^*, \tau^*)$, $\rho^* = H_4(\sigma_2^*, \sigma_3^*, ID_B^*)$ and $\lambda^* = H_3(\sigma_1^*)$

$\sigma_5^* = g_2^a\ \left(u' \prod_{i \in \Omega_A} u_i\right)^{r_A}\ \left(m' \prod_{j \in \overline{\beta^*}} m_j\right)^c \left(h_1^{\lambda^*} h_3\right)^{c\rho^*} = g_1^d\ \left(u' \prod_{i \in \Omega_A} u_i\right)^{r_A}\ (g^c)^{f' + \sum_{j \in \overline{\beta^*}} f_i}\,(g^c)^{\theta\rho^*}$

where $\overline{\beta^*} \subseteq \{1, 2, ..., l\}$ denotes the set of indices j such that $\beta^*[j] = 1$.

Note that, the simulator will be able to successfully simulate the challenge ciphertext without aborting, as explained above, only if $F(B^*) = 0 \ mod \ p$. The simulator aborts if $F(B^*) \neq 0 \ mod \ p$ as it will not be able to simulate the component σ_3 when $F(B^*) \neq 0 \ mod \ p$. The ciphertext $\sigma^* = \langle \sigma_1^*, \sigma_2^*, \sigma_3^*, \sigma_4^*, \sigma_5^*, \tau^* \rangle$ is sent to the adversary $\mathcal{A}$.

Here, if the simulator $\mathcal{B}$ was given a valid DBDH tuple i.e. if $T = \hat{e}(g, g)^{abc}$, then the challenge ciphertext $\sigma^* = \langle \sigma_1^*, \sigma_2^*, \sigma_3^*, \sigma_4^*, \sigma_5^*, \tau^* \rangle$, which is sent to the adversary $\mathcal{A}$, is a valid signcryption on the message m_γ^* by the sender with identity ID_A^* for the receiver with identity ID_B^*.

Otherwise, if T is a random element in $\mathbb{G}_T$, then challenge ciphertext is indistinguishable. So, in this case the simulator will give no information about the choice of γ that it made.

Training Phase II

In this phase, the simulator answers to the queries from the adversary $\mathcal{A}$ in the same way as it did in the Training Phase I. Here, $\mathcal{A}$ cannot ask for the Unsigncrypt query of the challenge ciphertext σ^* with sender identity as ID_A^* and receiver identity as ID_B^* and the Extract query for the receiver identity ID_B^*.

The strength of our scheme is that the adversary can again query the Signcrypt Oracle for the signcryption of either of the challenge ciphertexts m_0^* or m_1^* with the sender identity as ID_A^* and receiver identity as ID_B^*, during this phase. $\mathcal{A}$ can also query the Extract oracle for the sender identity ID_A^*, which makes our scheme insider secure.

Guess Phase

When the adversary $\mathcal{A}$ decides the training is enough, $\mathcal{A}$ outputs its guess γ' of γ.

If the guess $\gamma' = \gamma$, then the simulator outputs that T in the given DBDH tuple is valid i.e $T = \hat{e}(g,g)^{abc}$. Otherwise $\mathcal{B}$ outputs that $\langle g, g^a, g^b, g^c, T\rangle$ is not valid DBDH tuple.

Thus, $\mathcal{B}$ simulates a challenger for the adversary $\mathcal{A}$ and solves the DBDH problem with a probability ϵ' from the forgery produced by $\mathcal{A}$. This concludes the description of the simulation.

Analysis

In this section, we analyse the probability ϵ' with which the simulator will be able to solve the hard problem DBDH, given that the adversary is able to produce a valid forgery with a non-negligible probability ϵ. The simulation done is completed without aborting if in all the Extract queries, $F(u) \neq 0 \ mod \ l_u$(since $l_u(n_u+1) < p$, $F(u) \neq 0 \ mod \ l_u \implies F(u) \neq 0 \ mod \ p$), for the identity ID_u during the Training phase and if $F(u^*) = 0 \ mod \ p$ during the Challenge phase. Thus, l_u set as $2(q_E)$, sets a bound on the number of Extract queries to be asked by $\mathcal{A}$, whereas no bound is needed for the number of Signcrypt and Unsigncrypt queries because the signcryption and the unsigncryption oracles are always simulated without aborting. Let us assume the events A_i, A^* as follows.

$$A_i : F(u) \neq 0 \ mod \ l_u \ \ ; \quad A^* : F(u^*) = 0 \ mod \ p$$

Thus, from the analysis done above probability for the simulation not aborting is

$$Pr[\neg Abort] = Pr[\bigwedge_{i=1}^{q_E} A_i \wedge A^*]$$

Since $l_u(n_u+1) < p$, $F(u) = 0 \ mod \ p$ implies $F(u) = 0 \ mod \ l_u$. For the event A^* to occur, $x' + \sum_{i\in\Omega} x_i = 0 \ mod \ l_u$, hence $F(u^*) = 0 \ mod \ l_u$. And also, there should be a unique value of k_u, where $0 \leq k_u \leq n_u$, such that $F(u^*) = 0 \ mod \ p$. Here, k_u is randomly chosen. So, the probability for the event A^* to occur is

$$\begin{aligned} Pr[A^*] &= Pr[F(u^*) = 0 \ mod \ p \wedge F(u^*) = 0 \ mod \ l_u] \\ &= Pr[F(u^*) = 0 \ mod \ l_u] \, Pr[F(u^*) = 0 \ mod \ p | F(u^*) = 0 \ mod \ l_u] \\ &= \frac{1}{l_u} \frac{1}{n_u + 1} \end{aligned}$$

Since, the adversary cannot produce forgery on a message for which an Extract query is asked, the events A_i and A^* are independent. Also, the events A_i and A_j i.e $F(u_i) = 0 \, mod \, p$ and $F(u_j) = 0 \, mod \, p$ are independent. So,

$$Pr[\bigwedge_{i=1}^{q_E} A_i] = \prod_{i=1}^{q_E} Pr[A_i]$$

Thus, the probability for not aborting becomes,

$$Pr[\bigwedge_{i=1}^{q_E} A_i \wedge A^*] = Pr[\bigwedge_{i=1}^{q_E} A_i]\, Pr[A^*]$$
$$\geq \left(1 - \frac{1}{l_u}\right)^{q_E} \frac{1}{l_u} \frac{1}{n_u + 1}$$
$$\geq \left(1 - \frac{q_E}{l_u}\right) \frac{1}{l_u} \frac{1}{n_u + 1}$$
$$Pr[\bigwedge_{i=1}^{q_E} A_i \wedge A^*] \geq \frac{1}{4(q_E)(n_u + 1)} \; (\because l_u = 2q_E)$$

Thus, the probability for the challenger to produce a valid forgery ϵ' is defined as

$$\epsilon' \geq \frac{\epsilon}{4(q_E)(n_u + 1)}$$

The tightness of the security reduction in the proof of CCIA2 security of our scheme is of the order of the tightness of security reduction of Waters' scheme [26].

4.2 Unforgeability

We now prove the unforgeability property, *Strong Unforgeability under Chosen Message and Identity Attack*(SUF-CMIA) of our scheme with the following theorem.

Theorem 2. *If there exists an SUF-CMIA adversary for our scheme who can create valid ciphertexts during the SUF-CMIA game explained above, with a non-negligible probability ϵ when it runs for a polynomial time t, asking at most q_E extract queries, q_S signcrypt queries and q_{US} unsigncrypt queries, then there exists another algorithm, who can solve the Computational Diffie-Hellman (CDH) problem with probability ϵ' in polynomial time t', where*

$$\epsilon' \geq \frac{\epsilon}{4\kappa q_E(n_u + 1)(n_m + 1)}$$

$$t' \leq t + \mathcal{O}((n_u q_E + (n_u + l)(q_S + q_{US}))t_m + (q_E + q_S + q_{US})t_e + (q_S + q_{US})t_p)$$

where n_u is the length of the identity string and n_m is the length of the message, κ is the security parameter, t_m, t_e, t_p are the time required for each multiplication, each exponentiation and each bilinear pairing respectively and l is a value large enough such that the hash functions outputting $\{0,1\}^l$ in the scheme are collision resistant.

Proof

Let us assume that a $(\epsilon, t, q_E, q_S, q_{US})$-adversary $\mathcal{A}$ for our scheme exists. We will construct another algorithm $\mathcal{B}$ from this adversary $\mathcal{A}$, who can solve the Computational Diffie- Hellman (CDH) problem with a non-negligible probability ϵ' in polynomial time t'.

The algorithm $\mathcal{B}$ receives a CDH tuple $\langle g, g^a, g^b \rangle$, where g is a generator of prime order group $\mathbb{G}$ of order p. $\mathcal{B}$ simulates a challenger for the adversary $\mathcal{A}$ to calculate g^{ab} from the tuple given. This simulation is described as follows:

Setup

The simulator $\mathcal{B}$ sets $l_u = 2(q_E)$ and $l_m = \kappa$, where κ is the security parameter. Here, the values of q_S and q_{US} are not included while calculating l_u because the Signcrypt queries do not abort when the Extract query within them aborts and the Unsigncrypt queries do need an Extract query within them. $\mathcal{B}$ then chooses two integers k_u, k_m randomly such that $0 \leq k_u \leq n_u$ and $0 \leq k_m \leq n_m$. For the given values of q_E, q_S, q_{US}, n_u and n_m, we assume that $l_u(n_u + 1) < p$ and $l_m(n_m + 1) < p$. Then, $\mathcal{B}$ chooses the elements $x' \in \mathbb{Z}_{l_u}$ and

$z' \in \mathbb{Z}_{l_m}$ randomly and also chooses two vectors $\mathbb{X} = (x_i)$ of length n_u and $\mathbb{Z} = (z_i)$ of length l where the elements of $\mathbb{X}$ are chosen randomly from $\mathbb{Z}_{l_u}$ and the elements of $\mathbb{Z}$ are chosen randomly from $\mathbb{Z}_{l_m}$, with l large enough so that the hash functions are collision resistant. $\mathcal{B}$ also chooses two integers y' and w' randomly from $\mathbb{Z}_p$ and two vectors $\mathbb{Y} = (y_i)$ of length n_u and $\mathbb{W} = (w_i)$ of length l, where the elements of $\mathbb{Y}$ and $\mathbb{W}$ are chosen randomly from $\mathbb{Z}_p$.

Here we define two pairs of functions for a user with identity ID_u and for a value $\beta \in \{0,1\}^l$ as follows:

$F(u) = x' + \sum_{i \in \Omega_u} x_i - l_u k_u$ $\qquad$ $J(u) = y' + \sum_{i \in \Omega_u} y_i$

$K(\beta) = z' + \sum_{j \in \overline{\beta}} z_j - l_m k_m$ $\qquad$ $L(\beta) = w' + \sum_{j \in \overline{\beta}} w_i$

The simulator now sets the public parameters as follows:

$g_1 = g^a \qquad g_2 = g^b \qquad h_1 = g^\theta \qquad h_3 = g^{\theta'} \qquad h_2 = g^d$

where d, θ and θ' are chosen randomly from $\mathbb{Z}_p$. Here, g^a and g^b are taken by $\mathcal{B}$ from the CDH tuple.

$u' = g^{x' - l_u k_u} g^{y'} \qquad u_i = g_2^{x_i} g^{y_i} \qquad u' \prod_{i \in \Omega_u} u_i = g_2^{F(u)} g^{J(u)}$

$m' = g^{z' - l_m k_m} g^{w'} \qquad m_i = g_2^{z_i} g^{w_i} \qquad m' \prod_{i \in \overline{\beta}} m_i = g_2^{K(\beta)} g^{L(\beta)}$

Finally, $\mathcal{B}$ chooses an integer e' randomly from $\mathbb{Z}_p$ and a vector $\mathbb{E} = (e_i)$ of length n_u, where the elements of $\mathbb{E}$ are chosen randomly from $\mathbb{Z}_p$.

$v' = g^{e'} \qquad v_i = g^{e_i} \qquad v' \prod_{i \in \Omega_u} v_i = g^{e'} g^{\sum_{i \in \Omega_u} e_i} = g^{e' + \sum_{i \in \Omega_u} e_i}$

There are four one-way, collision resistant, cryptographic hash functions defined as $H_1 : \mathbb{G}_T \times \{0,1\}^{l_\tau} \to \{0,1\}^{n_m}$, $H_2 : \{0,1\}^{|p| + n_u + l_\tau} \to \{0,1\}^l$, $H_3 : \mathbb{G} \to \mathbb{Z}_p^*$ and $H_4 : \{0,1\}^{n_m + |p| + n_u} \to \mathbb{Z}_p^*$, where $l_\tau \approx 40$.

Training Phase

The simulator during this phase answers to the queries from the adversary $\mathcal{A}$ as follows.

Extract Queries
The simulator $\mathcal{B}$ does not know the master secret key g_2^a. So, when the adversary $\mathcal{A}$ asks for the private key of an identity ID_u, $\mathcal{B}$ responds as follows. $\mathcal{B}$ calculates $F(u)$ for the identity ID_u. If $F(u) = 0 \bmod p$, the simulator aborts. Otherwise, $\mathcal{B}$ randomly chooses $r_u \in \mathbb{Z}_p$ and calculates the private key as

$$d_u = (d_S, d_{US}, d_R) = \left(g_1^{-J(u)/F(u)} \left(u' \prod_{i \in \Omega_u} u_i \right)^{r_u}, g_1^d \left(v' \prod_{i \in \Omega_u} v_i \right)^{r_u} g_1^{-(e' + \sum_{i \in \Omega_u} e_i)/F(u)}, g_1^{-1/F(u)} g^{r_u} \right)$$

where $\Omega_u \subseteq \{1, 2, ..., n_u\}$ is the set of indices i such that $ID_u[i] = 1$.

Signcrypt Queries
When $\mathcal{A}$ queries the Signcrypt oracle for signcryption of message m by the identity ID_A and with ID_B as the intended receiver, $\mathcal{B}$ simulates a valid ciphertext as follows.

$\mathcal{B}$ asks for the Extract query of the sender identity ID_A. If $F(A) \neq 0 \ mod \ l_u$, $\mathcal{B}$ follows the Signcrypt algorithm using the private key d_A for identity ID_A got from the Extract algorithm to signcrypt the given message m as done by identity ID_A and with ID_B as the intended receiver.

Else if $F(A) = 0 \ mod \ p$, $\mathcal{B}$ proceeds as follows.

1. Choose $r \in \mathbb{Z}_p$ randomly and calculate the following parameters.

2. $\sigma_4 = g^{r_A}$, where r_A is taken from the list l_r corresponding to ID_A. If there is no entry for ID_A in l_r, then choose $r_A \in_R \mathbb{Z}_p$ and store $\langle ID_A, r_A \rangle$ in l_r.

3. $\beta = H_2(\sigma_4, ID_A, \tau)$, where $\tau \in_R \{0,1\}^{l_\tau}$

4. If $K(\beta) = 0 \ mod \ p$, repeat the simulation process from Step 3 by choosing a different $\tau \in_R \{0,1\}^{l_\tau}$, else continue.

5. $\sigma_1 = g^r g_1^{-1/K(\beta)} = g^{\overline{r}}$ and $\lambda = H_3(\sigma_1)$

6. $\sigma_2 = H_1(\hat{e}(g_1, h_2)^r \, \hat{e}(g_1, g_1)^{-d/K(\beta)}, \tau) \oplus m = H_1(\hat{e}(g_1, h_2)^r \, \hat{e}(g_1, g^d)^{-a/K(\beta)}, \tau) \oplus m = H_1(\hat{e}(g_1, h_2)^{\overline{r}}, \tau) \oplus m$

7. $\sigma_3 = \left(v' \prod_{i \in \Omega_B} v_i\right)^r \ g_1^{-(e' + \sum_{i \in \Omega_B} e_i)/K(\beta)} = \left(g^{(e' + \sum_{i \in \Omega_B} e_i)}\right)^r \ g_1^{-(e' + \sum_{i \in \Omega_B} e_i)/K(\beta)} = \left(v' \prod_{i \in \Omega_B} v_i\right)^{\overline{r}}$

8. $\rho = H_4(\sigma_2, \sigma_3, ID_B)$
9. $\sigma_5 = \left(u' \prod_{i \in \Omega_A} u_i\right)^{r_A} \ \left(m' \prod_{j \in \overline{\beta}} m_j\right)^r \ g_1^{-L(\beta)/K(\beta)} \ \left(h_1^\lambda h_3\right)^{r\rho} \ g_1^{-(\theta\lambda + \theta')\rho/K(\beta)}$

   $= \left(u' \prod_{i \in \Omega_A} u_i\right)^{r_A} \ g_2^a \ \left(m' \prod_{j \in \overline{\beta}} m_j\right)^{r - a/K(\beta)} \ \left(h_1^\lambda h_3\right)^{(r - a/K(\beta))\rho}$

   $= g_2^a \ \left(u' \prod_{i \in \Omega_A} u_i\right)^{r_A} \ \left(m' \prod_{j \in \overline{\beta}} m_j\right)^{\overline{r}} \ \left(h_1^\lambda h_3\right)^{\overline{r}\rho}$

   where $\overline{\beta} \subseteq \{1, 2, ..., l\}$ denotes the set of indices j such that $\beta[j] = 1$. The above steps in the calculation of σ_5 take place similar to the simulation of d_S in the Extract algorithm.

10. The ciphertext $\sigma = \langle \sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \tau \rangle$ is sent to the adversary $\mathcal{A}$.

Here, since β is chosen in such a way that $K(\beta) \neq 0 \ mod \ p$, this simulation of the signcrypted ciphertext σ never aborts. Thus, Signcrypt queries never abort.

Unsigncrypt Queries
When $\mathcal{A}$ queries $\langle \sigma, ID_A, ID_B \rangle$ i.e the unsigncryption of the ciphertext σ which was signcrypted by the sender ID_A for the intended receiver ID_B, to the challenger simulated by $\mathcal{B}$, it proceeds as follows.

$\mathcal{B}$ can directly calculate $\hat{e}(g_1, h_2)^r$ as $\hat{e}(g_1, h_2)^r = \hat{e}(g_1, g)^{dr} = \hat{e}(g_1, g^r)^d = \hat{e}(g_1, \sigma_1)^d$

The calculation of $\hat{e}(g_1, h_2)^r$ can also be done as per the Unsigncrypt algorithm first by simulating $\langle d_{US_B}, d_{R_B} \rangle$ for the receiver identity ID_B as

$\langle d_{US_B}, d_{R_B} \rangle = \langle g_1^d \, (v' \prod_{i \in \Omega_B} v_i)^{r_B} = h_2^a \, (v' \prod_{i \in \Omega_B} v_i)^{r_B}, g^{r_B} \rangle$

where r_B is randomly chosen from $\mathbb{Z}_p$ and then $\hat{e} \, (g_1, h_2)^r = \dfrac{\hat{e} \, (d_{US_B}, \sigma_1)}{\hat{e} \, (d_{R_B}, \sigma_3)}$

Then, in both these cases, m is calculated as $m = \sigma_2 \oplus H_1(\hat{e}(g_1, h_2)^r, \tau)$

Verification is done as mentioned in the Unsigncrypt algorithm and m is returned if this verification test is passed, else $\perp$ is returned. Note that, the Unsigncrypt queries never get aborted for a valid ciphertext.

Forgery Phase

The adversary $\mathcal{A}$ can ask polynomially bound number of these queries to $\mathcal{B}$ adaptively. When the adversary feels that the training is enough, it produces a valid forgery $\sigma^* = \langle \sigma_1^*, \sigma_2^*, \sigma_3^*, \sigma_4^*, \sigma_5^*, \tau^* \rangle$, where σ^* is a valid

signcrypted ciphertext for the message m^* by the identity ID_A^* for the identity ID_B^* who is the intended receiver. Here, note that as per the security model for SUF-CMIA, the $\langle \sigma^*, m^*, ID_A^*, ID_B^* \rangle$ given by the adversary $\mathcal{A}$ is valid only when $\langle \sigma^*, m^* \rangle$ is not the output of any Signcrypt Query with ID_A^* as the sender identity and with ID_B^* as the intended receiver. But, there is no constraint on the adversary that the forgery should be on a message m^* that was not a part of any Signcrypt Query with the sender and receiver identities as ID_A^* and ID_B^* respectively. $\mathcal{A}$ can also adaptively choose the sender identity ID_A^* and receiver identity ID_B^*, irrespective of the Signcrypt queries during the Training phase. $\mathcal{A}$ sends $\langle \sigma^*, m^*, ID_A^*, ID_B^* \rangle$ to $\mathcal{B}$.

When the simulator receives the tuple $\langle \sigma^*, m^*, ID_A^*, ID_B^* \rangle$, it calculates the solution to the given instance of the CDH problem as follows.

$$\frac{\sigma_5^*}{{\sigma_4^*}^{J(A^*)}\ {\sigma_1^*}^{L(\beta^*)}\ \sigma_1^{(\theta\lambda^*+\theta')\rho}} = \frac{g_2^a (g_2^{F(A^*)}\ g^{J(A^*)})^{r_A}\ (g_2^{K(\beta^*)}\ g^{L(\beta^*)})^r\ (h_1^{\lambda} h_3)^{r\rho})}{(g^{r_A})^{J(A^*)}\ (g^r)^{L(\beta^*)} (g^r)^{(\theta\lambda^*+\theta')\rho}} = g_2^a = g^{ab}$$

where $\lambda^* = H_3(\sigma_1^*)$, $\rho = H_4(\sigma_2^*, \sigma_3^*, ID_B^*)$ and $\beta^* = H_2(\sigma_4^*, ID_A^*, \tau^*)$

This can happen only when where $F(A^*) = 0\ mod\ p$ and $K(\beta^*) = 0\ mod\ p$. The simulator aborts otherwise.

Thus, the simulator solves the given instance of the CDH problem with probability ϵ', from the forgery produced by the adversary $\mathcal{A}$, by simulating a challenger for $\mathcal{A}$.

Analysis

Here, we analyse the probability ϵ' with which the simulator will be able to solve the instance of the CDH problem given to the challenger, given that the adversary is able to produce a valid forgery with a non-negligible probability ϵ. The simulation done is completed without aborting if for all the Extract queries of identity ID_u, $F(u) \neq 0\ mod\ l_u$ and if $F(u^*) = 0\ mod\ p$ and $K(\beta^*) = 0\ mod\ p$ during the Forgery phase. Let us assume the events A_i, A^* and B^* as follows.

$$A_i : F(u) \neq 0\ mod\ l_u\ ;\ \ A^* : F(u^*) = 0\ mod\ p\ ;\ \ B^* : K(\beta^*) = 0\ mod\ p$$

Thus, from the analysis done above, the probability for the simulation not aborting is

$$Pr[\neg Abort] = Pr[\bigwedge_{i=1}^{q_E} A_i \wedge A^* \wedge B^*]$$

The estimation of $Pr[\neg Abort]$ is similar to the one done after the IND-CCIA2 game. The probability for the challenger to produce a valid forgery ϵ' is defined as

$$\epsilon' \geq \frac{\epsilon}{4\kappa q_E (n_u + 1)(n_m + 1)} \tag{2}$$

Thus, the probability for the challenger to solve the instance of CDH problem in the unforgeability game is $\Theta(2^{\kappa})$ times more for our scheme than [20]. This makes the security reduction of our scheme tighter than [20].

5 Efficiency

The Signcrypt algorithm of our scheme performs one bilinear pairing operation while calculating $\hat{e}(g_1, h_2)^r$. But note that $\hat{e}(g_1, h_2)$ can be precomputed before the protocol begins since both g_1 and h_2 are public parameters and they are same for all runs of the protocol. The algorithm also performs 5 exponentiations (4 of elements of group $\mathbb{G}$ and one of element of $\mathbb{G}_T$). The unsigncrypt algorithm performs 6 bilinear pairing operations of which $\hat{e}(g_1, g_2)$ can be precomputed and one exponentiation of an element of group $\mathbb{G}$. Note

that the calculation of $(h_1^{\lambda} h_3)^{\rho}$ involves only one exponentiation according to the well known "square and multiply" technique explained in [19]. When the number of computations performed by our scheme and the scheme in Li et al. [14] are compared (excluding the precomputed values), our scheme performs one exponentiation less than [14] with same number of bilinear pairings.

Since none of the ID based signcryption schemes without random oracles are provably secure in the literature, we will compare the efficiency of our scheme with the ID based signcryption scheme π that was conceptually formatted in [21] obtained by the 'Sign then Encrypt' approach. Note that π is the most efficient signcryption scheme that can be got by the direct combination of IBE and IBS schemes, since [20] and [12] are the most efficient IBS and IBE schemes with SUF-CMA and IND-CCA2 properties respectively in the standard model.

Table 1. Computational Complexity of π and Ours

Scheme	**Secret key size**	**Ciphertext size**	**#pairings** Signcrypt, Unsigncrypt	**#exponentiations** Signcrypt, Unsigncrypt
π (Direct combination)	$5\|p\|$	$2\|p\| + n_m$	0(+1), 5(+1)	8, 3
Ours	$3\|p\|$	$4\|p\| + n_m + l_\tau$	0(+1), 5(+1)	5, 1

The numbers shown in the brackets indicate the values that can be precomputed before the algorithm begins (and they remain same for all runs of the protocol)

6 Conclusion

We have presented the first secure ID based signcryption scheme and proven its security in the standard model. This scheme satisfies the strongest notions of security available for the signcryption schemes. Moreover, it has additional interesting properties such as public ciphertext verifiability which is very useful in the context of firewalls and spam filters. The security reduction is also tighter compared to many other schemes in the standard model. There is a trade-off in this scheme between the size of public parameters and the tightness to the underlying hard assumption. In our scheme we have included some extra parameters namely a unsigncryption key to increase the probability to a much larger value so that the security of our scheme is more tight to the underlying hard problem much more than the existing signcryption schemes. An interesting and potential future direction will be designing a more efficient protocol with reduced public parameters, key size and reduced ciphertext size.

Acknowledgements: We sincerely thank Prof. Qiong Huang for shepherding and for pointing us a subtle inconsistency in the proof. We also thank the anonymous reviewers of the ProvSec 2012 program committee for their insightful reviews.

References

1. Jee Hea An, Yevgeniy Dodis, and Tal Rabin. On the security of joint signature and encryption. In *EUROCRYPT*, pages 83–107, 2002.
2. Paulo S. L. M. Barreto, Benoît Libert, Noel McCullagh, and Jean-Jacques Quisquater. Efficient and provably-secure identity-based signatures and signcryption from bilinear maps. In *ASIACRYPT*, pages 515–532, 2005.
3. Mihir Bellare and Sarah Shoup. Two-tier signatures, strongly unforgeable signatures, and fiat-shamir without random oracles. In *Public Key Cryptography*, pages 201–216, 2007.

4. Dan Boneh, Emily Shen, and Brent Waters. Strongly unforgeable signatures based on computational diffie-hellman. In *Public Key Cryptography*, pages 229–240, 2006.
5. Xavier Boyen. Multipurpose identity-based signcryption (a swiss army knife for identity-based cryptography). In *CRYPTO*, pages 383–399, 2003.
6. Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *J. ACM*, 51(4):557–594, 2004.
7. Liqun Chen and John Malone-Lee. Improved identity-based signcryption. In *Public Key Cryptography*, pages 362–379, 2005.
8. Sherman S. M. Chow, Siu-Ming Yiu, Lucas Chi Kwong Hui, and K. P. Chow. Efficient forward and provably secure id-based signcryption scheme with public verifiability and public ciphertext authenticity. In *ICISC*, pages 352–369, 2003.
9. Craig Gentry. Practical identity-based encryption without random oracles. In *EUROCRYPT*, pages 445–464, 2006.
10. Qiong Huang, Duncan Wong, Jin Li, and Yi-Ming Zhao. Generic transformation from weakly to strongly unforgeable signatures. *Journal of Computer Science and Technology*, 23:240–252, 2008. 10.1007/s11390-008-9126-y.
11. Zhengping Jin, Qiaoyan Wen, and Hongzhen Du. An improved semantically-secure identity-based signcryption scheme in the standard model. *Computers & Electrical Engineering*, 36(3):545–552, 2010.
12. Eike Kiltz and Yevgeniy Vahlis. Cca2 secure ibe: Standard model efficiency through authenticated symmetric encryption. In *CT-RSA*, pages 221–238, 2008.
13. Fagen Li, Yongjian Liao, and Zhiguang Qin. Analysis of an identity-based signcryption scheme in the standard model. *IEICE Transactions*, 94-A(1):268–269, 2011.
14. Fagen Li, Fahad Bin Muhaya, Mingwu Zhang, and Tsuyoshi Takagi. Efficient identity-based signcryption in the standard model. In *ProvSec*, pages 120–137, 2011.
15. Fagen Li and Tsuyoshi Takagi. Secure identity-based signcryption in the standard model. *Mathematical and Computer Modelling*, 2011. http://www.sciencedirect.com/science/article/pii/S0895717711003840.
16. Benoit Libert and Jean-Jacques Quisquater. New identity based signcryption schemes from pairings. In *IEEE Information Theory Workshop 2003*, pages 155–158, 1 2003. extended version.
17. Benoît Libert and Jean-Jacques Quisquater. Efficient signcryption with key privacy from gap diffie-hellman groups. In *Public Key Cryptography*, pages 187–200, 2004.
18. John Malone-Lee. Identity-based signcryption. Cryptology ePrint Archive, Report 2002/098, 2002. http://eprint.iacr.org/.
19. Wenbo Mao. *Modern Cryptography: Theory and Practice*. Prentice Hall Professional Technical Reference, 2003.
20. Kenneth G. Paterson and Jacob C. N. Schuldt. Efficient identity-based signatures secure in the standard model. In *ACISP*, pages 207–222, 2006.
21. S. Sharmila Deva Selvi, S. Sree Vivek, Dhinakaran Vinayagamurthy, and C. Pandu Rangan. On the security of id based signcryption schemes. Cryptology ePrint Archive, Report 2011/664, 2011. http://eprint.iacr.org/.
22. Adi Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO*, pages 47–53, 1984.
23. Isamu Teranishi, Takuro Oyama, and Wakaha Ogata. General conversion for obtaining strongly existentially unforgeable signatures. In *INDOCRYPT*, pages 191–205, 2006.
24. Xing Wang and Hai feng Qian. Attacks against two identity-based signcryption schemes. In *Second International Conference on Networks Security Wireless Communications and Trusted Computing (NSWCTC)*, volume 1, pages 24 –27, april 2010.
25. Xu An Wang, Weidong Zhong, and Haining Luo. Cryptanalysis of efficient identity based signature/signcryption schemes in the standard model. In *Intelligence Information Processing and Trusted Computing (IPTC), 2010 International Symposium on*, pages 622 –625, oct. 2010.
26. Brent Waters. Efficient identity-based encryption without random oracles. In *EUROCRYPT*, pages 114–127, 2005.
27. Ren Yanli and Gu Dawu. Efficient identity based signature/signcryption scheme in the standard model. In *The First International Symposium on Data, Privacy, and E-Commerce, 2007. ISDPE 2007.*, pages 133 –137, 2007.
28. Yong Yu, Bo Yang, Ying Sun, and Shenglin Zhu. Identity based signcryption scheme without random oracles. *Computer Standards & Interfaces*, 31(1):56–62, 2009.
29. Bo Zhang. Cryptanalysis of an identity based signcryption scheme without random oracles. *Journal of Computational Information Systems*, 6(6):1923–1931, 2010.
30. Mingwu Zhang, Pengcheng Li, Bo Yang, Hao Wang, and Tsuyoshi Takagi. Towards confidentiality of id-based signcryption schemes under without random oracle model. In Hsinchun Chen, Michael Chau, Shu-hsing Li, Shalini Urs, Srinath Srinivasa, and G. Wang, editors, *Intelligence and Security Informatics*, volume 6122 of *Lecture Notes in Computer Science*, pages 98–104. Springer Berlin / Heidelberg.
31. Yuliang Zheng. Digital signcryption or how to achieve cost(signature & encryption) << cost(signature) + cost(encryption). In *CRYPTO*, pages 165–179, 1997.