A NEW CONSTRUCTION OF IDENTITY-BASED SIGNCRYPTION WITHOUT RANDOM ORACLES

JIA FAN*

Science and Technology on Communication Security Laboratory
Chengdu, Sichuan 610000, P. R. China
fanjia0638@gmail.com

YULIANG ZHENG
University of North Carolina at Charlotte
Charlotte, North Carolina 28223, USA
yzheng@uncc.edu

XIAOHU TANG
Southwest Jiaotong University
Chengdu, Sichuan 610031, P. R. China
xhutang@swjtu.edu.cn

Received 16 March 2012
Accepted 12 June 2013
Communicated by Huaxiong Wang

Identity-based signcryption is a primitive that combines the functions of identity-based encryption and identity-based signature. In this paper, we first attack two of the existing identity-based signcryption schemes which are claimed to be provably secure without random oracles. Then we construct a new identity-based signcryption scheme and proves its security without random oracles.

Keywords: Signcryption; identity-based; provable security; attack.

1. Introduction

The concept of signcryption, introduced to the public by Zheng [19], is a primitive that combines the functions of both digital signature and public key encryption. The efficiency of signcryption is higher than sequential composition of digital signature and public key encryption. Identity-based signcryption is a specific type of signcryption, in which each user’s public key can be a string identifying this user (e.g. an e-mail address, a telephone number, etc.). This eliminates the need for certificates as used in a traditional public key infrastructure.

*The first author is supported by Innovation Fund 2012 of China Electronic Technology Group Corporation.
The first identity-based signcryption was presented by John Malone-Lee [11] in 2002. Till now, a number of identity-based signcryption schemes have been constructed [1, 5, 7, 8, 10]. While most of them are provably secure only in the random oracle model [3], which assumes that all hash functions can be regarded as random oracles. However, no real hash functions are random functions. Moreover, researchers have successfully constructed some schemes which can be proved secure in the random oracle model, but the scheme is actually not secure when random oracles are instantiated with concrete hash functions [2, 6]. Therefore, security proofs in the random oracle model only provide heuristic arguments. Designing identity-based signcryption schemes those can be proved secured without random oracles is absolutely a very interesting work.

A paper published in Eurocrypt2005 by Waters [14] presented a semantically secure identity-based encryption scheme without random oracles. Followed by this work, Paterson and Schuldt [12] constructed an identity-based signature provably secure without random oracles. Theses two schemes are of similar form (e.g. both make use of bilinear maps, the private key in both schemes are set in the same form). At first glance, it seems easy to construct an identity-based signcryption scheme provably secure without random oracles by combining these two schemes.

The first attempt to devise an identity-based signcryption provably secure without random oracles was by Yu et al. in 2009 [16]. His main idea is what we have described above, to combine the Waters identity-based encryption scheme [14] and the Paterson and Schuldt identity-based signature scheme [12]. However, this scheme was pointed out to be insecure on confidentiality [9, 13, 15, 17, 18]. Jin, Wen and Du [9] and Zhang [17] further proposed new identity-based signcryption scheme by improving the scheme of Yu et al. Both of these two improved schemes are claimed to be provably secure without random oracles.

In this paper, we will provide attacks to show that both of the two improvements [9, 17] are actually not as secure as they claimed. From these failed examples, we can see that to securely combine the Waters identity-based encryption and Paterson and Schuldt signature is actually not that easy as it first looks. Our main contribution is to present a new construction for identity-based signcryption by carefully combining the Waters scheme and the Paterson and Schuldt scheme, and strictly prove that our proposed scheme is secure under the defined security model without random oracles.

2. Preliminaries

In this section, we review the definitions of bilinear maps, collision resistant hash functions, as well as the Discrete Logarithm assumption. All these definitions will be helpful in subsequent sections when we review the Jin, Wen and Du scheme [9], the Zhang scheme [17] and describe our proposed scheme. Particularly, the definitions of collision resistant hash functions and Discrete Logarithm assumption will also be useful in Sec. 6, since the security of our proposed scheme is partially based on them.
2.1. Bilinear maps
We review bilinear maps, following the standard definition. Let $G$ and $G_T$ be two (multiplicative) cyclic groups of prime order $p$. Let $g$ be a generator of $G$. A symmetric bilinear map is a map $e : G \times G \rightarrow G_T$ with the following properties:

1. Bilinear: for all $u, v \in G$ and $a, b \in \mathbb{Z}_p$, we have $e(u^a, v^b) = e(u, v)^{ab}$.
2. Non-degenerate: for all $u, v \in G$, $e(u, v) \neq 1$.

2.2. Collision resistant hash functions
Hash functions efficiently map arbitrary length strings (usually a large, possibly variable-sized amount of data) onto elements of particular encodings (usually with a relatively small size) such as finite field elements or elliptic curve points. Collision resistant hash function is defined as follows:

Definition 1. A hash function $H$ is collision resistant, if for any adversary $A$, running in polynomial time $t$, the advantage $\epsilon_H$ is negligible in $k$, where $\epsilon_H = \Pr[A = ((M_0, M_1) : H(M_0) = H(M_1))]$, and $k$ is a security parameter that defines the size of input and output sets for this hash function.

Remark 1. Throughout this paper, when we say a function is negligible in $k$, it indicates that this function vanishes faster than the inverse of any polynomial in the same parameter $k$ when $k$ is sufficiently large.

2.3. Discrete logarithm assumption
Let $G$ be a group of prime order $p$, and $g$ be the generator for $G$, where the size of $G$ is a function of a security parameter $k$. We have the following definition for the Discrete Logarithm assumption.

Definition 2. The Discrete Logarithm assumption holds in $G$, if for any adversary $A$, given an element $Y \in G$, running in polynomial time $t$, the advantage $\epsilon_{dl}$ is negligible in $k$, where $\epsilon_{dl} = \Pr[A = (y : g^y = Y)]$.

3. Security Model of Identity-Based Signcryption
We now describe the security model for identity-based signcryption by defining the syntax and two security requirements, confidentiality and unforgeability.

3.1. Syntax
An identity-based signcryption scheme contains the following four algorithms as follows:

- **Setup($1^k$)**: Given a security parameter $1^k$, it outputs a pair of master private/public keys ($msk, mpk$). This algorithm is run by a key generation center (KGC). KGC publishes $mpk$, and keeps $msk$ secret.
five stages as follows:

- **Extract** \((mpk, msk, ID_P)\): On input \((msk, mpk)\) and an identity \(ID_P\), it outputs a private key \(sk_P\) for user \(ID_P\). This algorithm is also run by KGC. KGC sends \(sk_P\) to user \(ID_P\) in a secure way (e.g. face to face or through a secure channel).
- **Signcrypt** \((mpk, ID_S, ID_R, M, sk_S)\): On input \(mpk\), a pair of sender and receiver’s identity \((ID_S, ID_R)\), a message \(M \in \mathcal{M}\) (\(\mathcal{M}\) is the message space) and a sender’s private key \(sk_S\), it outputs a signcryption \(\sigma\). This algorithm is run by a sender \(ID_S\). \(ID_S\) sends \((\sigma, ID_S, ID_R)\) to \(ID_R\) through a public (not necessarily secure) channel.
- **Unsigncrypt** \((mpk, ID_S, ID_R, \sigma, sk_R)\): On input \((mpk, ID_S, ID_R, \sigma)\) and a receiver’s private key \(sk_R\), it outputs a message \(M\), or outputs a special symbol \(\bot\) representing that the signcryption is invalid. This algorithm is run by receiver \(ID_R\) when it receives \((\sigma, ID_S, ID_R)\) from \(ID_S\).

For consistency purpose, we require that for all \(\sigma \leftarrow \text{Signcrypt}(mpk, ID_S, ID_R, M, sk_S)\), we should have \(M = \text{Unsigncrypt}(mpk, ID_S, ID_R, \sigma, sk_R)\).

### 3.2. Security definition for confidentiality

To define confidentiality, we first describe an attack game, called indistinguishability in identity-based signcryption under chosen ciphertext attack (IND-IBSC-CCA). This game is played between an adversary \(\mathcal{A}\) and its environment \(\Sigma\) which contains a challenger \(\mathcal{C}\) and three types of oracles, Extract Oracle \(\mathcal{O}_{ex}\), Unsigncryption Oracle \(\mathcal{O}_{usc}\) and Signcryption Oracle \(\mathcal{O}_{sc}\). Specifically, the IND-IBSC-CCA game contains five stages as follows:

- **Stage 1**: \(\mathcal{C}\) computes \((msk, mpk) \leftarrow \text{Setup}(1^k)\), gives \(mpk\) to \(\mathcal{A}\), and equips all the oracles with \((msk, mpk)\).
- **Stage 2**: \(\mathcal{A}\) is able to ask for a number of queries, each is one of the following three types:
  - **Extract Query**: \(\mathcal{A}\) submits a user identity \(ID_P\) to \(\mathcal{O}_{ex}\), which then returns an outcome of \(\text{Extract}(mpk, msk, ID_P)\) to \(\mathcal{A}\).
  - **Signcryption Query**: \(\mathcal{A}\) submits \((M, ID_S, ID_R)\) to \(\mathcal{O}_{sc}\), which then returns to \(\mathcal{A}\) with an outcome of \(\text{Signcrypt}(mpk, ID_S, ID_R, M, sk_S)\).
  - **Unsigncryption Query**: \(\mathcal{A}\) submits \((\sigma, ID_S, ID_R)\) to \(\mathcal{O}_{usc}\), which then returns the result of \(\text{Unsigncrypt}(mpk, ID_S, ID_R, \sigma, sk_R)\) to \(\mathcal{A}\).
- **Stage 3**: \(\mathcal{A}\) submits \((M_0, M_1, ID_{S'}, ID_{R'})\) to \(\mathcal{C}\), where \(M_0\) and \(M_1\) are of equal length and both in \(\mathcal{M}\). \((ID_{S'}, ID_{R'})\) is a pair of sender/receiver identities, and \(\mathcal{A}\) has not asked for an extract query on \(ID_{R'}\) at Stage 2. \(\mathcal{C}\) chooses a random bit \(\beta\), asks for an extract query on \(ID_{S'}\) to \(\mathcal{O}_{ex}\) to get \(sk_{S'}\), returns \(\sigma' \leftarrow \text{Signcrypt}(mpk, ID_{S'}, ID_{R'}, M_\beta, sk_{S'})\) to \(\mathcal{A}\).
- **Stage 4**: It is the same as Stage 2, except that \(\mathcal{A}\) is not allowed to ask for an unsigncryption query on \((\sigma', ID_{S'}, ID_{R'})\), or an extract query on \(ID_{R'}\).
- **Stage 5**: \(\mathcal{A}\) outputs a guess bit \(\beta'\). \(\mathcal{C}\) checks whether \(\beta' = \beta\). If it is, then \(\mathcal{A}\) wins the challenge.
The advantage for $A$ to win the challenge in the IND-IBSC-CCA game is defined as $\epsilon = |Pr[\beta' = \beta] - 1/2|$.

**Definition 3.** An identity-based signcryption scheme is IND-IBSC-CCA secure, if for any adversary $A$ running in time $t$, has asked for at most $q_s$ signcryption queries, at most $q_u$ unsigncryption queries and at most $q_e$ extract queries where $t$, $q_s$, $q_u$ and $q_e$ are all polynomials in $k$, the advantage $\epsilon$ is negligible in $k$.

### 3.3. Security definition for unforgeability

To define unforgeability, we first describe an attack game, called strong existential unforgeability in identity-based signcryption under chosen message attack (sEUF-IBSC-CMA). Similar as IND-IBSC-CCA, this game is also played between an adversary $A$ and its environment $\Sigma$ which contains a challenger $C$ and the oracles of $O_{ex}$, $O_{usc}$ and $O_{sc}$. sEUF-IBSC-CMA game contains three stages, where Stage 1 and Stage 2 are the same as the Stage 1 and Stage 2 in IND-IBSC-CCA game, and Stage 3 is described as follows:

- **Stage 3:** $A$ outputs $(\sigma^*, ID_{S^*}, ID_{R^*})$ to $C$. $C$ requires an extract query on $ID_{R^*}$, and runs $\text{Unsigncrypt}(mpk, \sigma^*, ID_{S^*}, ID_{R^*}, sk_{R^*})$. If it does not return $\bot$, $\sigma^*$ is not one of the results returned by $O_{sc}$ and $A$ has never required an extract query on $ID_{S^*}$, then $A$ wins the challenge.

The advantage for $A$ to win the challenge in the sEUF-IBSC-CMA game is defined as $\epsilon$ which is the probability of an event that $A$ wins the challenge.

**Definition 4.** An identity-based signcryption scheme is sEUF-IBSC-CMA secure, if for any adversary $A$ running in time $t$, has asked for at most $q_s$ signcryption queries, at most $q_u$ unsigncryption queries and at most $q_e$ extract queries where $t$, $q_s$, $q_u$ and $q_e$ are all polynomials in $k$, the advantage $\epsilon$ is negligible in $k$.

A relaxation of the sEUF-IBSC-CMA security is called existential unforgeability in identity-based signcryption under chosen message attack (EUF-IBSC-CMA). The EUF-IBSC-CMA security is defined in a similar way as we define sEUF-IBSC-CMA, except that in the EUF-IBSC-CMA attack game the adversary wins the challenge if $M^* \leftarrow \text{Unsigncrypt}(mpk, \sigma^*, ID_{S^*}, ID_{R^*}, sk_{R^*})$, $M^* \in M$, $A$ has never required a signcryption query on $(M^*, ID_{S^*}, ID_{R^*})$ and $A$ has never required an extract query on $ID_{S^*}$.

### 4. Attacks on Two Identity-Based Signcryption Schemes

In this section, we will provide attacks on two existing schemes and analyze why they fail.
4.1. Attacks on the Jin-Wen-Du scheme

Jin, Wen and Du proposed an identity-based signcryption scheme and claimed that their scheme in [9] is both IND-IBSC-CCA and EUF-IBSC-CMA secure, while we will show that it is not.

4.1.1. Review of the Jin-Wen-Du scheme

The Jin-Wen-Du scheme is described as follows:

1. Setup($1^k$): To generate a pair of system private/public key pairs, KGC chooses groups $G$ and $G_T$ of prime order $p$ such that an admissible bilinear pairing $e : G \times G \to G_T$ can be constructed and pick a generator $g$ of $G$. It chooses a bijection $\varphi : \mathcal{R} \to G_T$, where $\varphi^{-1}$ is its inverse mapping, and $\mathcal{R}$ is a subset of $\{0, 1\}^{k+l}$ with $p$ elements. It chooses a collision resistant hash function $H : \{0, 1\}^k \to \{0, 1\}^l$. It picks a secret value $\alpha \in \mathbb{Z}_p$, and computes $g_1 \leftarrow g^\alpha$. It picks random elements $g_2, u', m' \in G$ and random vectors $\vec{m} = (m_1), \vec{u} = (u_i)$ of length $l$ and $n$ respectively, whose entries are random elements from $G$. Return $(msk, mpk)$ as

$$msk \leftarrow g_2^\alpha, \ mpk \leftarrow (G, G_T, e, \varphi, \varphi^{-1}, H, g, g_1, g_2, u', m', \vec{u}, \vec{m}).$$

2. Extract($mpk, ID_P$): Let $ID_P$ be a bit string of length $n$, and let $U_P[i]$ be the $i$-th bit of $ID_P$. Define $\mathcal{B}_P \subset \{1, 2, ..., n\}$ to be the set of indices $i$ such that $U_P[i] = 1$. To generate a private key for user with identity $ID_P$, KGC randomly picks $r_P \in \mathbb{Z}_p$, then return $sk_P = (d_{P_1}, d_{P_2})$ as

$$sk_P \leftarrow \left( g_2^{u'} \prod_{i \in \mathcal{B}_P} u_i \right)^{r_P}, g^{r_P}.$$

3. Signcryption($mpk, M, ID_A, ID_B, sk_A$): To send a message $M \in \{0, 1\}^k$ to Bob with identity $ID_B$, Alice with identity $ID_A$ randomly pick $r \in \mathbb{Z}_p$ and $R \in \{0, 1\}^l$ such that $M || R \in \mathcal{R}$, then return signciphertext $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5)$ as

$$\left( e(g_1, g_2)^r \cdot \varphi(M || R), g^r, \left( u' \prod_{i \in \mathcal{B}_A} u_i \right)^r, d_{A_1} \left( m' \prod_{j \in \mathcal{I}} m_j \right)^r, d_{A_2} \right)$$

where $\mathcal{I} = \{ j \in Z : H(M)[j] \oplus R[j] = 1 \}$.

4. Unsingcryption($mpk, \sigma, ID_A, ID_B, sk_B$): When Bob receives from Alice a ciphertext $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5)$, he computes $\varphi^{-1}(\sigma_1 e(d_{B_2}, \sigma_3) e(d_{B_1}, \sigma_2)^{-1}) \rightarrow M || R$, and generates $\{ j \in Z : H(M)[j] \oplus R[j] = 1 \} \rightarrow \mathcal{I}$. Return the message $M$ if the following equation holds, otherwise it returns otherwise returns $\perp$,

$$e(\sigma_4, g_2) = e(g_1, g_2) e \left( u' \prod_{i \in \mathcal{B}_A} u_i, \sigma_5 \right) e \left( m' \prod_{j \in \mathcal{I}} m_j, \sigma_2 \right).$$
A New Construction of Identity-Based Signcryption Without Random Oracles

4.1.2. Attack on IND-IBSC-CCA security of the Jin-Wen-Du scheme

The attack on IND-IBSC-CCA security is described in the following steps:

- According to the IND-IBSC-CCA attack game, the adversary $A$ will get a challenge signciphertext $\sigma^* = (\sigma_1^*, \sigma_2^*, \sigma_3^*, \sigma_4^*, \sigma_5^*)$ with sender and receiver identity $(ID_{S'}, ID_{R'})$ at Stage 3.
- Then at Stage 4, $A$ chooses a random element $r' \in Z_p$, and requires an unsigncryption query on $(\sigma, ID_{S'}, ID_{R'})$ where $\sigma = (\sigma_1^*, \sigma_2^*, \sigma_3^*, \sigma_4^* \cdot (u^* \prod_{i \in U_S} u_i) r', \sigma_5^* \cdot g r')$. The unsigncryption oracle returns to $A$ a message $M$.
- At Stage 5, $A$ checks whether $M = M_0$. If it is, then $A$ outputs $\beta' = 0$, otherwise it outputs $\beta' = 1$.

It is easy to see that at Step 2, $M_0 = \text{Unsigncrypt}(mpk, \sigma, ID_{S'}, ID_{R'}, sk_{R'})$. Then at Step 3, it is easy to see that $\beta = \beta'$. Therefore, $A$ successfully wins the challenge in IND-IBSC-CCA attack game.

4.1.3. Attack on EUF-IBSC-CMA security of the Jin-Wen-Du scheme

The attack on EUF-IBSC-CMA security is described in the following steps:

- At Stage 2, $A$ runs as follows:
  1. It requires a signcryption query on $(M, ID_S, ID_R)$ to get a signciphertext $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5)$, where it has never required an extract query on $ID_S$.
  2. It requires an extract query on $ID_R$, to get $sk_R$.
- At Stage 3, $A$ runs as follows:
  1. It runs $\text{Unsigncrypt}(mpk, \sigma, ID_S, ID_R, sk_{R'})$ to get $R$.
  2. It choose an arbitrary message $M'$ with $M \neq M'$.
  3. It finds an element $R' \in \{0, 1\}^l$ to make sure that $\mathcal{S} = \mathcal{S}'$.
  4. It computes $\sigma'_1 = \frac{\sigma_1 \cdot \phi(M' || R')}{\phi(M || R)}$.
  5. It sets $\sigma' \leftarrow (\sigma'_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5)$.
  6. It outputs $(\sigma', ID_S, ID_R)$.

It is easy to verify that $\sigma'$ is a valid signciphertext on $(M', ID_S, ID_R)$. Therefore, $A$ successfully attacks the EUF-IBSC-CMA security of the Jin-Wen-Du scheme.

4.2. Attack on the Zhang scheme

The identity-based signcryption scheme by Zhang [17] is also an improvement of the scheme by Yu et al. [16], therefore it is of similar form of the scheme in [9]. Zhang claimed that his scheme in [17] is both IND-IBSC-CCA and EUF-IBSC-CMA secure, while we will show that it is not IND-IBSC-CCA secure. We do not find an attack on the EUF-IBSC-CMA security, but we further give an attack to show that this scheme is not sEUF-IBSC-CMA secure.
4.2.1. Review of the Zhang scheme

It is described as follows:

1. Setup($1^k$): To generate a pair of system private/public key pairs, KGC picks a random element $h \in \mathbb{G}$, chooses two collision resistant hash function $H_1 : \mathbb{G}_T \times \mathbb{G}_T \rightarrow \mathbb{Z}_p^*$, $H_2 : \mathbb{G} \rightarrow \{0,1\}^l$. And it generates $\{\mathbb{G}, \mathbb{G}_T, e, g, g_1, g_2, u', m', \overline{u}, \overline{m}, \alpha\}$ as the same as in the Setup algorithm in [9]. Return $(msk, mpk)$ as

$$msk \leftarrow g_2^a, \ mpk \leftarrow (\mathbb{G}, \mathbb{G}_T, e, H_1, H_2, g, g_1, g_2, h, u', m', \overline{u}, \overline{m}).$$

2. Extract($mpk, ID_P$): To generate a private key for user with identity $ID_P$, KGC generates $sk_P$ the same way as in [9], and returns $sk_P = (d_{P_1}, d_{P_2})$ as

$$sk_P \leftarrow \left( g_2^a \prod_{i \in \mathbb{G}_T} u_i \right)^{r_{P_1}} \cdot g^{r_{P_2}}.$$

3. Signcrypt($mpk, M, ID_A, ID_B, sk_A$): To send a message $M \in \{0,1\}^k$ to Bob with identity $ID_B$, Alice with identity $ID_A$ randomly picks $r, s \in \mathbb{Z}_p$, computes $R = e(g_1, g_2)^r$, $t = H_1(M||R)$, $m'' = H_2(g^t h^s)$ and let $M' \in \{1, \ldots, l\}$ be the set of indices $j$ such that $m[j] = 1$, where $m[j]$ is the $j$-th bit of $m''$. Then it returns signcrypt $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6)$ as

$$\sigma \leftarrow \left( R \cdot M, g^f, \left( \prod_{i \in \mathbb{G}_T} u_i \right)^r, d_{A_1}, \left( \prod_{j \in M'} m_j \right)^r, d_{A_2}, s \right).$$

4. Unsigncrypt($mpk, \sigma, ID_A, ID_B, sk_B$): Receiving a ciphertext $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6)$ from Alice, Bob computes $R = e(d_{B_1}, \sigma_3)^{-1}e(d_{B_1}, \sigma_2)$, $M \leftarrow \sigma_1 \cdot R^{-1}$, $t \leftarrow H_1(M||R)$, $m'' \leftarrow H_2(g^t h^{m''})$, generates the corresponding set $M' \in \{1, \ldots, l\}$ of indices $j$ such that $m[j] = 1$ where $m[j]$ is the $j$-th bit of $m''$. It returns $M$ if

$$e(\sigma_4, g) = e(g_1, g_2)e\left( \prod_{i \in \mathbb{G}_T} u_i, \sigma_5 \right) e\left( \prod_{j \in M'} m_j, \sigma_2 \right),$$

otherwise returns $\bot$.

4.2.2. Attack on IND-IBSC-CCA security of the Zhang scheme

The attack on IND-IBSC-CCA security is described in the following steps:

- According to the IND-IBSC-CCA attack game, the adversary $A$ will get a challenge signcrypttext $\sigma^* = (\sigma_1^*, \sigma_2^*, \sigma_3^*, \sigma_4^*, \sigma_5^*, \sigma_6^*)$ with sender and receiver identity $(ID_{B'}, ID_{R'})$ at Stage 3.

- Then at Stage 4, $A$ chooses a random element $r' \in \mathbb{Z}_p$, and requires an unsign-cripttext query on $(\sigma, ID_{B'}, ID_{R'})$ where $\sigma = (\sigma_1^*, \sigma_2^*, \sigma_3^*, \sigma_4^*, (\prod_{i \in \mathbb{G}_T} u_i)^{r'}, \sigma_5^*, \sigma_6^*)$. Obviously, the unsigncripttext oracle returns to $A$ with message $M_{\beta}$. 
• At Stage 5, $A$ checks whether $M = M_0$. If it is, then $A$ outputs $\beta' = 0$, otherwise it outputs $\beta' = 1$.

It is easy to see that at Step 4, the required unsigncryption query on $\sigma$ satisfies $M_\beta = \text{Unsigncrypt}(mpk, \sigma, ID_S^*, ID_R^*, sk_R^*)$. Therefore, we have $\beta = \beta'$. That is, $A$ successfully wins the challenge in IND-IBSC-CCA attack game of the Zhang scheme.

4.2.3. Attack on sEUF-IBSC-CMA security of the Zhang scheme

We do not find an attack on the EUF-IBSC-CMA security, but we do find an attack on the sEUF-IBSC-CMA security, which is described in the following steps:

• At Stage 2, $A$ requires a signcrytption query on $M$ with a pair of sender/receiver identity $(ID_S^*, ID_R^*)$ to get a signcryptext $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6)$.

• At Stage 3, $A$ runs chooses a random element $r' \in \mathbb{Z}_p$, $\sigma^* \leftarrow (\sigma_1, \sigma_2, \sigma_3, \sigma_4 \cdot (u' \prod_{i \in U_A} u_i)^{r'}, \sigma_5 \cdot g^{r'}, \sigma_6)$, and outputs $(\sigma^*, ID_S^*, ID_R^*)$. Obviously, $\sigma^*$ is a valid signcryptext for $(M, ID_S^*, ID_R^*)$.

Obviously, the result of $\text{Unsigncrypt}(mpk, \sigma^*, ID_A^*, ID_B^*, sk_B^*)$ equals the message $M$ queried at Stage 2. Therefore, $A$ successfully attacks the sEUF-IBSC-CMA security of the Zhang scheme.

4.3. Further observations of these two schemes

First, we analyze the Jin, Wen and Du scheme. It is vulnerable to our attack on IND-IBSC-CCA security for a reason that $d_{A_2}$ is not included as part of input to the hash function $H$. Therefore, if an attacker gets a valid signcryptext $\sigma$ on a message $M$, then it can reconstruct a valid signcryptext $\sigma'$ on the same $M$ by just replacing $(d_{A_1}, d_{A_2})$ with $(d_{A_1} \cdot (u' \prod_{i \in U_A} u_i)^{r}, d_{A_2} \cdot g^{r})$. This scheme is not EUF-IBSC-CMA security, since the receiver can compute $R, M$ and $S$, then it is easy to construct a valid signcryptext $\sigma'$ on $M'$ by keeping $F$ unchanged (that is choosing $H(M)[j] \oplus R[j] = H(M')[j] \oplus R'[j]$). The key improvement of Jin, Wen and Du scheme is to introduce an random element $R$ to achieve the IND-IBSC-CCA security, while from our attacks it is clear that it achieves neither this goal nor the security of EUF-IBSC-CMA. Furthermore, $F$ is a specified subset of $\{0, 1\}^{l+k}$, it is impossible to ensure that $M||R \in F$ in the Signcrypt algorithm. Therefore, we do not regard it is a good idea to make use of a random element $R$ in form of $M||R$.

As to the Zhang scheme, it is vulnerable to our attacks on both IND-IBSC-CCA security and sEUF-IBSC-CMA security for a similar reason as the Jin, Wen and Du scheme on IND-IBSC-CCA security, that is $d_{A_2}$ is not included as part of input to the hash function $H_1$.

In the following, we will propose a new scheme that is provably secure under the defined model without random oracles. During the design of our proposed scheme,
our proposed identity-based signcryption scheme is described as follows:

**Setup**

1. Choose two groups $G$ and $G_T$, where $G$ is generated by $g$, both groups are of prime order $p$, and a bilinear map $e : G \times G \rightarrow G_T$ exists.
2. Choose $\alpha \in \mathbb{Z}_p$ randomly, compute $g_1^\alpha$.
3. Choose $\{g_2, g_3, u_0, u_1, ..., u_{n_1}, v_0, v_1, ..., v_{n_2}, w_0, w_1, ..., w_{n_3}\}$ all randomly from $G$.
4. Set three vectors: $U \leftarrow (u_1, ..., u_{n_1})$, $V \leftarrow (v_1, ..., v_{n_2})$, $W \leftarrow (w_1, ..., w_{n_3})$.
5. Choose four collision resistant hash functions: $H_1 : \{0, 1\}^* \rightarrow \{0, 1\}^{n_1}$, $H_2 : \{0, 1\}^* \rightarrow \{0, 1\}^{n_2}$, $H_3 : \{0, 1\}^* \rightarrow \mathbb{Z}_p$, $H_4 : G \rightarrow \{0, 1\}^{n_3}$.
6. Return $(msk, mpk)$ as

$$msk \leftarrow \alpha, \ mpk \leftarrow \{G, G_T, e, g, g_1, g_2, g_3, u_0, v_0, w_0, U, V, W, H_1, H_2, H_3, H_4\}.$$  

**Extract**(mpk, msk, ID$_P$): To generate a private key for user ID$_P$, KGC runs:

1. Choose two random elements $r_1, r_2 \in \mathbb{Z}_p$.
2. $\tau_p \leftarrow H_1(ID_P)$, write as $(\tau_{P_1} \ldots \tau_{P_{n_1}}) \in \{0, 1\}^{n_1}$.
3. $\psi_p \leftarrow H_2(ID_P)$, write as $(\psi_{P_1} \ldots \psi_{P_{n_2}}) \in \{0, 1\}^{n_2}$.
4. Return a private key $sk_p \leftarrow (d_{P_1}, d_{P_2}, d_{P_3}, d_{P_4})$ as

$$(g^2_0, u_0^{n_1} \prod_{i=1}^{n_1} \tau_{R_1}^{r_1}, g^{r_1}, g_3^{\psi_{P_2}}, v_0^{n_2} \prod_{j=1}^{n_2} \psi_{P_j}^{r_2}, g^{r_2}).$$  

If ID$_P$ is fixed as a sender, he only needs to store $(d_{P_3}, d_{P_4})$, and if he is fixed as a receiver, he only needs to store $(d_{P_1}, d_{P_2})$.

**Signcryption**(mpk, ID$_S$, ID$_R$, M, sk$_S$): To communicate a message $M \in \mathcal{M}$ to a receiver ID$_R$, a sender ID$_S$ runs:

1. Choose two random elements $t, s \in \mathbb{Z}_p$.
2. $\tau_s \leftarrow H_1(ID_R)$, write as $(\tau_{R_1} \ldots \tau_{R_{n_1}}) \in \{0, 1\}^{n_1}$.
3. Parse $sk_S$ as $(d_{S_1}, d_{S_2}, d_{S_3}, d_{S_4})$.
4. Return $\sigma \leftarrow (\sigma_0, \sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5)$ as

$$(e(g_1, g_2)^t \cdot M, g^t, u_0^{n_1} \prod_{i=1}^{n_1} \tau_{H_i}^t, d_{S_2}, d_{S_3} (w_0^{n_3} \prod_{i=1}^{n_3} \psi_{S_i}^t), \delta).$$  

where $c \leftarrow H_4(z)$ is written as $(c_1...c_{n_3}) \in \{0,1\}^{n_3}$, $z \leftarrow g^\theta g_4^s$, $\theta \leftarrow H_3(\sigma_0, \sigma_1, \sigma_2, \sigma_3, ID_S, ID_R)$.

Unsigncryption $(mpk, ID_S, ID_R, \sigma, sk_R)$: To unsigncrypt a signcryptext $\sigma$ from a sender $ID_S$, a receiver $ID_R$ runs:

1. Parse $\sigma$ as $(\sigma_0, \sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5)$.
2. Parse $sk_R$ as $(d_{R_1}, d_{R_2}, d_{R_3}, d_{R_4})$.
3. $\psi_S \leftarrow H_2(ID_S)$, write as $(\psi_{S_1}...\psi_{S_{n_2}}) \in \{0,1\}^{n_2}$.
4. $\theta \leftarrow H_3(\sigma_0, \sigma_1, \sigma_2, \sigma_3, ID_S, ID_R)$.
5. $z \leftarrow g^\theta g_4^s$, compute $c \leftarrow H_4(z)$ and write it as $(c_1...c_{n_3}) \in \{0,1\}^{n_3}$.
6. If the following equation satisfies,

$$e(\sigma_4, g) = e(g_1, g_3) \cdot e\left(v_0 \prod_{j=1}^{n_2} \psi_{S_j}, \sigma_3\right) \cdot e\left(w_0 \prod_{i=1}^{n_3} w_i^c, \sigma_1\right)$$

then return $M \leftarrow \sigma_0 \cdot e(\sigma_2, d_{R_2}) / e(d_{R_1}, \sigma_1)$;

otherwise the signcryptext is regarded as invalid, it returns ⊥.

6. Security Analysis of Our Proposed Scheme

We will provide security proofs on aspects of both confidentiality and unforgeability. Our proposed scheme is based on a smart combination of the Waters identity-based encryption scheme [14] and the Paterson and Schudlt identity-based signature scheme [12], where both the two schemes are strictly proved to be secure under decisional-BDH assumption and computational-BDH assumption respectively.

In the following proofs, instead of deducing the security of our proposed scheme to be based on complexity problems directly, we choose to partly base it on the security of the Waters and Paterson-Schudlt scheme, which makes the whole proof more clear and concise.

6.1. Security proof on confidentiality

The confidentiality security of our proposed scheme is partially based on the semantical security of the Waters identity-based encryption scheme [14]. We will first review the description of the Waters identity-based encryption (IBE) scheme as well as its security definition on semantical security. Followed by it, we then provide a detailed security proof on the security of confidentiality.

6.1.1. Overview of Waters IBE

Waters IBE as well as its security definition [14] are reviewed as follows. All the undefined variables and primitives are computed or chosen the same way as in our proposed signcryption scheme in Sec. 5.
\begin{itemize}
\item Setup(1^k): To generate a pair of master private and public key pair, KGC computes \( msk_w \leftarrow g_2^r \); \( mpk_w \leftarrow \{ G, G_T, e, g, g_1, g_2, u_0, U, H_1 \} \). Return \((msk_w, \ mpk_w)\).
\item Extract(\(mpk_w, msk_w, ID_P\)): To generate a private key for user \( ID_P \), KGC computes \( dw_{P_1} \leftarrow d_{P_1}; dw_{P_2} \leftarrow d_{P_2}\). Return \( skw_P \leftarrow (dw_{P_1}, dw_{P_2})\).
\item Encrypt(\(mpk, ID_R, M\)): To send a message \( M \in M \) (\( M \) is the message space) to a receiver \( ID_R \), a sender computes \( (\sigma_{w_0}, \sigma_{w_1}, \sigma_{w_2}) \leftarrow (\sigma_0, \sigma_1, \sigma_2)\). Return \( \sigma_w \leftarrow (\sigma_{w_0}, \sigma_{w_1}, \sigma_{w_2})\).
\item Decrypt(\(mpk, ID_R, \sigma_w, skw_R\)): On receiving a ciphertext \( \sigma_w \), a receiver \( ID_R \) computes \( M \leftarrow \frac{\sigma_{w_0} e(\sigma_{w_2}, dw_{P_2})}{e(dw_{P_1}, \sigma_{w_1})} \). Return \( M\).
\end{itemize}

The attack game for semantical security of the above IBE contains five Stages. At Stage 1, an adversary \( A \) is given \( mpk_w \). At Stage 2, \( A \) has access to a number of extract queries on various identities to get the corresponding private keys. At Stage 3, \( A \) submits two equal length messages \((M_0, M_1)\) and a receiver’s identity \( ID^* \), and then gets a challenge ciphertext \( \sigma^*_w \), which is an encryption of \( m_\beta \) (\( \beta \) is a random bit) on \( ID^* \). Stage 4 is mostly the same as Stage 2, except that \( A \) is not allowed to ask an extract query on \( ID^* \). At Stage 5, \( A \) outputs a guess bit \( \beta' \). If \( \beta = \beta' \), then \( A \) wins the challenge. The advantage for \( A \) to win the challenge in this game is defined as \( \epsilon = |Pr[\beta = \beta'] - 1/2|\).

**Definition 5.** If for any adversary \( A \) in the above attack game, running in time \( t \), has asked for at most \( q_e \) extract queries where \( t, q_e \) are both polynomials in \( k \), the advantage \( \epsilon \) is negligible in \( k \), then the Waters identity-based encryption scheme is semantically secure.

The Waters identity-based encryption scheme has been proved to be semantically secure in [14].

6.1.2. Detailed proof on IND-IBSC-CCA security

**Theorem 1.** Our proposed identity-based signcryption scheme is IND-IBSC-CCA secure, assuming that the Waters identity-based encryption scheme [14] is semantically secure, the hash functions of \( H_3 \) and \( H_4 \) are collision resistant, and the Discrete Logarithm assumption holds in \( G \). Specifically, for an adversary runs in time \( t \), makes at most \( q_e \) extract queries, \( q_s \) signcryption queries, \( q_a \) unsigncryption queries, the advantage satisfies the following condition:

\[
\epsilon \leq \frac{\epsilon_{enc}}{1 - q_a (\epsilon_{H_3} + \epsilon_{H_4} + \epsilon_{dl} + 1/p + 1/p^4)}
\]

where \( \epsilon_{enc}, \epsilon_{H_3}, \epsilon_{H_4}, \epsilon_{dl} \) represent the advantage of attacking semantical security of the Waters encryption scheme which runs in time \( t + O(q_e + q_s + q_a) \) and asks for at most \( q_e \) extract queries, finding a collision for \( H_3 \) in time \( t \), finding a collision for \( H_4 \) in time \( t \), finding a solution for the Discrete Logarithm problem in \( G \) in time \( t \) respectively.
Proof of Theorem 1. In the IND-IBSC-CCA game, we use a simulator $S$ to simulate the adversary $A$’s environment. That is, $S$ simulates the behavior of the challenger $C$ as well as the oracles $O_{ex}$, $O_{usc}$, $O_{sc}$. $S$ is also an adversary in the attack game of semantical security for the Waters identity-based encryption scheme.

Specifically, $S$ simulates the IND-IBSC-CCA game as follows. Note that as an adversary in the attack game for encryption, $S$ is first given $mpk_w$.

**Stage 1:** $S$ runs the following steps to simulate the challenger $C$:

1. Parse $mpk_w$ as $\{G, \mathbb{G}_T, e, g, g_1, g_2, u_0, U, H_1\}$.
2. Choose a random element $\mu \in \mathbb{Z}_p$, compute $g_3 \leftarrow g^{\mu}$.
3. Choose a random element $y \in \mathbb{Z}_p$, compute $g_4 \leftarrow g^{y}$.
4. Choose random elements $\delta_0, \delta_1, ..., \delta_{n_2} \in \mathbb{Z}_p$, from $j = 0$ to $n_2$ compute $v_j \leftarrow g^{\delta_j}$, set $V \leftarrow (v_1...v_{n_2})$.
5. Choose random elements $k_1, ..., k_{n_3} \in \mathbb{Z}_p$, and from $i = 1$ to $n_3$ compute $w_i \leftarrow g_2^{k_i}$, set $W \leftarrow (w_1...w_{n_3})$.
6. Choose random elements $\rho^*, \lambda \in \mathbb{Z}_p$, compute $c^* \leftarrow H_2(g^{\rho^*})$, write $c^*$ as $(c_1^*, ..., c_{n_3}^*) \in \{0, 1\}^{n_3}$.
7. Compute $\tau^* \leftarrow \sum_{i=1}^{n_3} k_i c_i^* \mod p$, $w_0 \leftarrow g_2^{-\tau^*} g^\lambda$.
8. Generate $H_2, H_3, H_4$ according to the $Setup$ algorithm.
9. Return $mpk \leftarrow \{G, \mathbb{G}_T, e, g, g_1, g_2, g_3, g_4, u_0, v_0, w_0, U, V, W, H_1, H_2, H_3, H_4\}$ to $A$.

**Analysis of Stage 1:** It is obvious that the distribution of $(msk, mpk)$ are the same as it is according to the $Setup$ algorithm. Therefore, we claim that the $S$ simulates perfectly at Stage 1.

**Stage 2:** In this stage $S$ simulates all the three types of oracles. Each type of queries is simulated as follows:

- **Extract Query:** When $A$ submits an identity $ID_P$ to $S$, $S$ runs the following steps:
  1. Require an extract query in the encryption attack game to get $skw_P = (dw_{P_1}, dw_{P_2})$;
  2. $(d_{P_1}, d_{P_2}) \leftarrow (dw_{P_1}, dw_{P_2})$;
  3. Choose a random element $r_2 \in \mathbb{Z}_p$;
  4. $\psi_P \leftarrow H_2(ID_P)$, write as $(\psi_{P_1}...\psi_{P_{n_2}}) \in \{0, 1\}^{n_2}$;
  5. $d_{P_3} \leftarrow g_1^{\mu} (v_0 \prod_{j=1}^{n_2} \psi_{P_j})^{r_2}$;
  6. $d_{P_4} \leftarrow g^{x^2}$;
  7. Return $sk_P \leftarrow \{d_{P_1}, d_{P_2}, d_{P_3}, d_{P_4}\}$.

- **Signcryption Query:** When $A$ submits $(M, ID_S, ID_R)$ to $S$, $S$ runs the following steps:
(1) Run Step 3 to Step 6 of dealing with Extract query on \( ID_S \) to get \( d_{S_3} \) and \( d_{S_1} \).

(2) Run Signcryption algorithm to get a signcryption text \( \sigma \);

(3) Return \( \sigma \).

- Unsigncryption Query: When \( A \) submits \( (\sigma, ID_S, ID_R) \) to \( S \), \( S \) runs the following steps:

  (1) Run Unsigncryption algorithm to check whether \( \sigma \) is valid. If it is not, return \( \perp \).

  (2) If \( \sum_{i=1}^{n_3} k_{ci} \mod p = \tau^* \), abort the game.

  (3) Choose a random element \( r_1' \in \mathbb{Z}_{p^*} \)

  (4) Compute \((sk'_{R_1}, sk'_{R_2}) \leftarrow (g_1^{\sum_{i=1}^{n_3} k_{ci} - \tau'} (w_0 \prod_{i=1}^{n_3} w_i^{c_i}) r_1', g_1^{\sum_{i=1}^{n_3} k_{ci} - \tau'} g^{r_1'})\) .

  (5) \( \sigma'_4 \leftarrow \delta_{\sigma_4, \sigma_{sk}_{R_2}} \left( g_0^{\sum_{i=1}^{n_3} k_{ci} - \tau'} 1 \right) \).

  (6) Return \( M \leftarrow \delta_{\sigma_0, \sigma_{sk}_{R_1}} \).

Analysis of Stage 2: (1) Analysis of Extract queries: According to the sematical attack game of Waters identity-based encryption scheme, we can see that the distribution of \((dw_{P_1}, dw_{P_2})\) is the same as \((d_{P_1}, d_{P_2})\) computed according to the Extract algorithm of the sigcryption scheme. Since \( g_3 = g^\mu \), then \( d_{P_3} = g_3^{\alpha} (v_0 \prod_{j=1}^{n_2} v_j^{\psi P_j})^\tau \).

For the randomness of \( r_2 \), the distribution of \((d_{P_3}, d_{P_1})\) computed by \( S \) is the same as that computed according to the Extract algorithm of the sigcryption scheme.

Therefore, we claim that the distribution of \((d_{P_1}, d_{P_2}, d_{P_3}, d_{P_1})\) is the same as that computed by the Extract algorithm of the sigcryption scheme. In other words, \( S \) simulates the extract oracle perfectly.

(2) Analysis of Signcryption queries: It is easy to see that \( S \) simulates the signcryption oracle perfectly.

(3) Analysis of Unsigncryption queries: Define \( r''_1 = \sum_{i=1}^{n_3} k_{ci} - \tau^* + r'_1 \), then we have

\[
\begin{align*}
    sk'_{R_1} &= g_1^{\sum_{i=1}^{n_3} k_{ci} - \tau'} \left( w_0 \prod_{i=1}^{n_3} w_i^{c_i} \right) r'_1 \\
    &= g_1^{\sum_{i=1}^{n_3} k_{ci} - \tau'} \left( w_0 \prod_{i=1}^{n_3} w_i^{c_i} \right) r''_1 + g_1^{\sum_{i=1}^{n_3} k_{ci} - \tau^*} \left( w_0 \prod_{i=1}^{n_3} w_i^{c_i} \right) r'_1 \\
    &= g_1^{\sum_{i=1}^{n_3} k_{ci} - \tau^*} \left( w_0 \prod_{i=1}^{n_3} w_i^{c_i} \right) \left( g_2^\alpha g_1^{\sum_{i=1}^{n_3} k_{ci} - \tau^*} \right) \left( w_0 \prod_{i=1}^{n_3} w_i^{c_i} \right) r''_1
\end{align*}
\]
A New Construction of Identity-Based Signcryption Without Random Oracles

\[\begin{align*}
&= g_2^{\alpha} \left( w_0 \prod_{i=1}^{n_3} w_{c_i}^i \right)^{r_1''} \\
&= g_1^{\sum_{i=1}^{n_3} k_{c_i} - \tau^*} g^{r_1''} = g^{\sum_{i=1}^{n_3} k_{c_i} - \tau^* + r_1''} = g^{r'}. \\
\text{From } (sk''_{R_1}, sk''_{R_2}), \text{ we can compute}
\end{align*}\]

\[\begin{align*}
\sigma_4' &= \frac{\sigma_4}{g_1^\mu \cdot g_3^{\sum_{j=1}^{n_2} \delta_j \psi_j}} \\
&= g_3^\alpha \cdot g^{r_2 (\delta_0 + \sum_{j=1}^{n_2} \delta_j \psi_j)} \\
&= g_3^\alpha \cdot g^{r_2 (\delta_0 + \sum_{j=1}^{n_2} \delta_j \psi_j)} \\
&= \left( w_0 \prod_{i=1}^{n_3} w_{c_i}^i \right)^{t}. \\
\end{align*}\]

Finally, we have

\[\begin{align*}
\frac{\sigma_0 \cdot e(\sigma_4', sk''_{R_2})}{e(sk''_{R_1}, \sigma_1)} &= \frac{e(g_1, g_2)^t \cdot M \cdot e \left( \left( w_0 \prod_{i=1}^{n_3} w_{c_i}^i \right)^{t}, g^{r''} \right)}{e(g_1, g_2)^t \cdot M \cdot e \left( \left( w_0 \prod_{i=1}^{n_3} w_{c_i}^i \right)^{t}, g^{r''} \right)} \\
&= \frac{e(g_1, g_2)^t \cdot M \cdot e \left( \left( w_0 \prod_{i=1}^{n_3} w_{c_i}^i \right)^{t}, g^{r''} \right)}{e(g_1, g_2)^t \cdot M \cdot e \left( \left( w_0 \prod_{i=1}^{n_3} w_{c_i}^i \right)^{t}, g^{r''} \right)} \\
&= M. \\
\end{align*}\]

Now, it is clear that \( S \) simulates the unsigncryption oracle perfectly if \( \sum_{i=1}^{n_3} k_{c_i} \neq \tau^* \).

**Stage 3:** When \( A \) submits \((M_0, M_1, ID_{S*}, ID_{R*})\) to \( S \), \( S \) runs the following steps:

1. Forward \((M_0, M_1, ID_{R*})\) to the challenger in the attack game for encryption to get a challenge \( \sigma''_w = (\sigma''_{w_0}, \sigma''_{w_1}, \sigma''_{w_2}) \);
2. \((\sigma'_0, \sigma'_1, \sigma'_2) \leftarrow (\sigma''_{w_0}, \sigma''_{w_1}, \sigma''_{w_2})\);
3. Choose a random element \( r''_2 \in \mathbb{Z}_p \);
one of the following cases happens:

(2) \( z \) is uniformly at random and are hidden from the adversary’s view. Else if \( c \) the probability that

\[ \sum \text{attack game for the encryption scheme.} \]

is almost perfect except in the unsigncryption query when the signcryp-

Stage 5: When \( A \) outputs a guess bit \( \beta’ \). \( S \) forwards it to the challenger of the attack game for the encryption scheme.

Now we analyze the errors during \( S \)’s simulation. From the above analysis, the simulation is almost perfect except in the unsigncryption query when the signcrypt-text is valid and \( \sum_{i=1}^{n_3} k_i c_i = \tau^* \). For each unsigncryption query, if \( c \neq c^* \), then the probability that \( \sum_{i=1}^{n_3} k_i c_i = \tau^* \) is 1/p, since all the values of \( k_i \) are chosen uniformly at random and are hidden from the adversary’s view. Else if \( c = c^* \), then one of the following cases happens:

Stage 4: At this Stage \( S \) simulates the oracles the same way as at Stage 2.

Stage 5: When \( A \) outputs a guess bit \( \beta’ \). \( S \) forwards it to the challenger of the attack game for the encryption scheme.

Now we analyze the errors during \( S \)’s simulation. From the above analysis, the simulation is almost perfect except in the unsigncryption query when the signcrypt-text is valid and \( \sum_{i=1}^{n_3} k_i c_i = \tau^* \). For each unsigncryption query, if \( c \neq c^* \), then the probability that \( \sum_{i=1}^{n_3} k_i c_i = \tau^* \) is 1/p, since all the values of \( k_i \) are chosen uniformly at random and are hidden from the adversary’s view. Else if \( c = c^* \), then one of the following cases happens:

(1) \( z \neq z^* \): In this case, the adversary finds a collision for \( H_4 \);
(2) \( z = z^* \) and \( \sigma^*_5 \neq \sigma_5 \): In this case, the adversary finds a solution for the Discrete Logarithm problem on \( g_4 \) by computing \( \log g_4 \leftarrow \frac{g - \theta}{\sigma^*_5 - \sigma_5} \).
(3) \( z = z^* \) and \( \sigma^*_5 = \sigma_5 \) and \( (\sigma_0, \sigma_1, \sigma_2, \sigma_3, ID_{S'}, ID_{R'}) \neq (\sigma^*_0, \sigma^*_1, \sigma^*_2, \sigma^*_3, ID_{S'}, ID_{R'}) \):
   In this case \( \theta = \theta^* \), the adversary finds a collision for \( H_3 \);
(4) \( z = z^* \) and \( \sigma^*_5 = \sigma_5 \) and \( (\sigma_0, \sigma_1, \sigma_2, \sigma_3, ID_{S'}, ID_{R'}) = (\sigma^*_0, \sigma^*_1, \sigma^*_2, \sigma^*_3, ID_{S'}, ID_{R'}) \):
   By the Signcrypt algorithm, it is easy to verify that \( \sigma_4 = \sigma^*_4 \). Therefore, \( \sigma = \sigma^* \) and \( (ID_{S'}, ID_{R'}) = (ID_{S'}, ID_{R'}) \). According to the game rule, \( A \) is not allowed to ask such an unsigncryption query at Stage 4. And at Stage 2, the probability that \( A \) generate a signcrypttext \( \sigma = \sigma^* \) is at most \( 1/p^3 \) (which means \( A \) chooses the same \( t, s \) and \( r^*_2 \)).

Therefore, for each unsigncryption query, the probability that \( S \) makes mistakes is at most \( \epsilon_H + \epsilon_H + \epsilon_{dl} + 1/p + 1/p^3 \). During the whole simulation, the probability that \( S \) makes mistakes is at most \( q_u (\epsilon_H + \epsilon_H + \epsilon_{dl} + 1/p + 1/p^3) \).

From the simulation, it is easy to see that if the simulation is perfect and \( A \) wins the challenge, then \( S \) also wins the challenge in the attack game for the encryption scheme. Therefore, we have

\[ \epsilon_{enc} \geq \epsilon \cdot (1 - q_u (\epsilon_H + \epsilon_H + \epsilon_{dl} + 1/p + 1/p^3)) \]

The running time for \( S \) in the attack game for encryption is \( (t + O(q_e + q_s + q_u)) \), which is sum of \( A \)’s running time \( t \) and \( S \)’s simulation time \( O(q_e + q_s + q_u) \).
Then we get our conclusion that
\[ \epsilon \leq \epsilon_{\text{enc}} - q_u(\epsilon_{H_3} + \epsilon_{H_4} + 1/p + 1/p^3). \]

6.2. Security proof for unforgeability

The unforgeability security of our proposed scheme is partially based on the existential unforgeability against chosen message attack (EUF-CMA) security of the Paterson and Schuldt identity-based signature (IBS) scheme [12]. We will first review the description of the Paterson and Schuldt identity-based signature scheme as well as its security definition on EUF-CMA security. Followed by it, we then provide a detailed security proof on the security of unforgeability.

6.2.1. Overview of Paterson and Schuldt IBS

The Paterson and Schuldt IBS as well as its security definition [12] are reviewed as follows. All the undefined variables and primitives are computed or chosen the same way as in our proposed signcryption scheme.

- **Setup\( (1^k)\):** To generate a pair of master private and public key, KGC computes \( msk_s \leftarrow g_3^\alpha; \) \( mpk_s \leftarrow \{G, G_T, e, g, g_1, v_0, w_0, V, W, H_2, H_4\}. \) Return \( (msk_s, mpk_s)\).

- **Extract\( (mpk_s, msk_s, ID_P)\):** To generate a private key for a user \( ID_P\), KGC computes \( ds_{P_1} \leftarrow d_{P_1}; ds_{P_2} \leftarrow d_{P_2}. \) Return \( sks_P \leftarrow (ds_{P_1}, ds_{P_2})\).

- **Sign\( (mpk_s, sks_S, ID_S, z)\):** To sign on a message \( z\), a signer \( ID_S\) computes \( (\sigma_{s_0}, \sigma_{s_1}, \sigma_{s_2}) \leftarrow (\sigma_1, \sigma_3, \sigma_4). \) Return \( \sigma_s \leftarrow (\sigma_{s_0}, \sigma_{s_1}, \sigma_{s_2})\).

- **Verify\( (mpk, ID_S, z, \sigma_s, sks_S)\):** To check whether \( \sigma_s\) is a valid signature on message \( z\) originated from \( ID_S\), a verifier check whether \( e(\sigma_{s_2}, g) = e(g_1, g_3) \cdot e(v_0 \prod_{i=1}^n v_i^{\sigma_{s_1}}, \sigma_{s_1}) \cdot e(w_0 \prod_{i=1}^n w_i^{\sigma_{s_1}}, \sigma_{s_0}). \) If it is, then return \( \top\), otherwise return \( \bot\).

The EUF-CMA attack game for the Paterson and Schuldt IBS contains three stages. At Stage 1, an adversary \( A\) is given \( mpk_s\). At Stage 2, the adversary has access to a number of extract queries on various identity \( ID\) and signature queries on various message \( z\) and identity \( ID\). At Stage 3, \( A\) outputs \((z^*, \sigma^*_s)\). If \( Verify(mpk, z^*, ID_S, \sigma_s, sks_S) = \top\) and the adversary has never required a signature on \( z^*\), then \( A\) wins the challenge. \( A\)'s advantage in winning the challenge is defined as \( \epsilon\) which is the probability that \( A\) wins the challenge.

**Definition 6.** If for any adversary \( A\) in the EUF-CMA game, running in time \( t\), has asked for at most \( q_e\) extract queries, \( q_s\) signature queries, where \( t, q_e, q_s\) are all polynomials in \( k\), the advantage \( \epsilon\) is negligible in \( k\), then the Paterson and Schuldt identity-based signature is EUF-CMA secure.
The Paterson and Schuld identity-based signature scheme has been proved to be EUF-CMA secure in [12].

6.2.2. Detailed proof on sEUF-IBSC-CMA security

**Theorem 2.** The identity-based signcryption scheme is sEUF-IBSC-CMA secure, assuming that the Paterson and Schuld identity-based signature scheme [12] is existential unforgeable under chosen message attack, the hash function $H_3$ is collision resistant, and the Discrete Logarithm assumption holds in $\mathbb{G}$.

Specifically, for an adversary runs in time $t$, makes at most $q_e$ extract queries, $q_s$ signcryption queries, $q_a$ unsigncryption queries, with advantage $\epsilon$, there exists an adversary runs in time $t$, asks at most $q_e$ extract queries, $q_s$ signature queries and with advantage $\epsilon/3$ to output a successful forgery for the signature scheme, or an adversary runs in time $t$ with advantage $\epsilon/3$ to find a collision in $H_3$, or an adversary runs in time $t$ with advantage $\epsilon/3$ to solve the Discrete problem in $\mathbb{G}$.

Proof of Theorem 2. In the sEUF-IBSC-CMA game, the adversary $A$’s goal is to forge a valid signcryption $\sigma^* = (\sigma_0^*, \sigma_1^*, \sigma_2^*, \sigma_3^*, \sigma_4^*)$ where $\sigma^* \neq \sigma^{(i)}$. Throughout this proof, the variables with superscript $(i)$ denote the variables computed in the $i$-th signcryption oracle. And the variables with superscript * denote the variables computed at Stage 3. According to the result of $A$’s forgery, we divide it into four types as follows:

- **Type I:** $z^* = z^{(i)}$ (for all $i$ form 1 to $q_a$),
- **Type II:** $z^* = z^{(i)}$ and $\sigma_5^* \neq \sigma_5^{(i)}$ for some $i \in \{1, ..., q_s\}$,
- **Type III:** $z^* = z^{(i)}$, $\sigma_5^* = \sigma_5^{(i)}$ and $(\sigma_0^{(i)}, \sigma_1^{(i)}, \sigma_2^{(i)}, \sigma_3^{(i)}, ID_S^{(i)}, ID_R^{(i)}) \neq (\sigma_0^*, \sigma_1^*, \sigma_2^*, \sigma_3^*, ID_S^*, ID_R^*)$ for some $i \in \{1, ..., q_s\}$,
- **Type IV:** $z^* = z^{(i)}$, $\sigma_5^* = \sigma_5^{(i)}$ and $(\sigma_0^{(i)}, \sigma_1^{(i)}, \sigma_2^{(i)}, \sigma_3^{(i)}, ID_S^{(i)}, ID_R^{(i)}) = (\sigma_0^*, \sigma_1^*, \sigma_2^*, \sigma_3^*, ID_S^*, ID_R^*)$ for some $i \in \{1, ..., q_s\}$.

We will show that a successful type I forgery will lead to a successful attack for the above identity-based signature scheme, a successful type II forgery will lead to a solution for the Discrete Logarithm assumption in $\mathbb{G}$, a successful type III forgery will lead to a break for the collision-resistant hash function $H_3$, and the type IV forgery is always not successful since in this case $\sigma_4^* = \sigma_4^{(i)}$ according to the Signcrypt algorithm, then $\sigma^* = \sigma^{(i)}$, which is not allowed according to the game rule.

Before this attack, the simulator $S$ flips a random coin to guess which kind of successful forgery $A$ will output, then sets up the master public key and performs appropriately, and all our simulations are perfect.

**Type I Forgery:** In the sEUF-IBSC-CMA game, we use a simulator $S$ to simulate the adversary $A$’s environment. That is, $S$ simulates the behavior of the challenger $C$ as well as the oracles $O_{ex}$, $O_{usc}$, $O_{ac}$. $S$ is also an adversary in the EUF-CMA
attack game for the above identity-based signature scheme. Specifically, $S$ simulates the sEUF-IBSC-CCA game as follows. Note that as an adversary in the attack game for the above identity-based signature scheme. Specifically,

- **Stage 1:** $S$ runs the following steps to simulate the challenger $C$:
  1. Parse $mpk_s$ as $\{G, \mathbb{G}_T, e, g, g_1, g_2, g_3, g_4, v_0, v_0, w_0, W, H_2, H_3, H_4\}$.
  2. Choose a random element $\mu \in \mathbb{Z}_p$, compute $g_2 \leftarrow g^{\mu}$.
  3. Choose a random element $y \in \mathbb{Z}_p$, compute $g_4 \leftarrow g^y$.
  4. Choose random elements $\delta_0, \delta_1, \ldots, \delta_n$ from $\mathbb{Z}_p$, compute $u_0 \leftarrow g^\delta_0$, $u_1 \leftarrow g^\delta_1$, $\ldots$, $u_n \leftarrow g^\delta_n$, set $U = (u_1, \ldots, u_n)$.
  5. Generate $H_1, H_2, H_3$ according to the $Setup$ algorithm.
  6. Return $mpk \leftarrow \{G, \mathbb{G}_T, e, g, g_1, g_2, g_3, g_4, v_0, v_0, w_0, U, V, W, H_2, H_3, H_4\}$ to $A$.

- **Stage 2:** In this stage $S$ simulates all the three types of oracles. Each type of queries is simulated as follows:
  - **Extract Query:** When $A$ submits an identity $ID_P$ to $S$, $S$ runs the following steps:
    1. Require an extract query in the EUF-CMA attack game to get $sk_{s_P} = (ds_{P_1}, ds_{P_2})$.
    2. $(d_{P_1}, d_{P_2}) \leftarrow (ds_{P_1}, ds_{P_2})$.
    3. Choose a random element $r_1 \in \mathbb{Z}_p$.
    4. $\tau_P \leftarrow H_1(ID_P)$, write as $(\tau_{P_1}, \ldots, \tau_{P_n}) \in \{0, 1\}^n$.
    5. $d_{P_1} \leftarrow g_2^{u_0} (u_1 \prod_{i=1}^n u_i^{r_i})^{r_1}$.
    6. $d_{P_2} \leftarrow g^{r_1}$.
    7. Return $sk_{P} = \{d_{P_1}, d_{P_2}, d_{P_3}, d_{P_4}\}$.
  - **Signcryption Query:** When $A$ submits $(M, ID_S, ID_R)$ to $S$, $S$ runs the following steps:
    1. Choose a random element $\phi \in \mathbb{Z}_p$.
    2. $z \leftarrow g^{\phi}$.
    3. $S$ requires a signature query on $(z, ID_S)$ to get a signature $\sigma_S = (\sigma_{s_0}, \sigma_{s_1}, \sigma_{s_2})$.
    4. $(\sigma_1, \sigma_3, \sigma_4) \leftarrow (\sigma_{s_0}, \sigma_{s_1}, \sigma_{s_2})$.
    5. $\sigma_0 \leftarrow e(g_1, \sigma_0^u) \cdot M$.
    6. $\tau_R \leftarrow H_1(ID_R)$, write as $(\tau_{R_1}, \ldots, \tau_{R_n}) \in \{0, 1\}^n$.
    7. $\sigma_2 \leftarrow \sigma_1^{\frac{\delta_2}{\delta_1}} \prod_{i=1}^n \delta_i^{\tau_{R_i}}$.
    8. $\theta \leftarrow H_3(\sigma_0, \sigma_1, \sigma_2, \sigma_3, ID_S, ID_R)$.
    9. $\sigma_3 \leftarrow (\phi - \theta)/y$.
    10. Return $\sigma = (\sigma_0, \sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5)$.
  - **Unsigncryption Query:** When $A$ submits $(\sigma, ID_S, ID_R)$ to $S$, $S$ runs the following steps:
(1) Run Step 3 to Step 6 of dealing with the Extract query on $ID_R$ to get $d_{R_1}$ and $d_{R_2}$.

(2) Run the Unsigncrypt algorithm, and return its result.

- **Stage 3:** When $A$ outputs $(\sigma^*, ID_{S^*}, ID_{R^*})$ to $S$, $S$ runs Step 1 to Step 4 of the Unsigncrypt algorithm to get $z^*$, then outputs $(z^*, \sigma^*_s, ID_{S^*})$ where $\sigma^*_s = (\sigma^*_1, \sigma^*_2, \sigma^*_3)$ in the EUF-CMA game as its forgery.

Now we can see that if $A$ finally makes a successful forgery, then $S$ also makes a valid forgery for the identity-based signature scheme.

**Type II Forgery:** In the sEUF-IBSC-CMA game, let $A$ be a type II adversary and $S$ be a simulator which simulates the adversary $A$’s environment. Besides, $S$ is given a random element $g_4 \in \mathbb{G}$, and $S$ is aimed to compute $y \in \mathbb{Z}_p$ where $g^y = g_4$.

$S$ simulates the game as a normal challenger in the definition except that in the Setup system step, he sets $g_4 \leftarrow g_4'$. Finally, if $A$ outputs a successful type II forgery that $z^* = z^{(i)}$ and $\sigma^*_5 \neq \sigma^*_5^{(i)}$ for some $i \in \{1, ..., q_s\}$, then $S$ can computes $y \leftarrow (\theta^* - \theta^{(i)})/((\sigma^*_5) - \sigma^*_5^{(i)})$.

**Type III Forgery:** In the sEUF-IBSC-CMA game, let $A$ be a type III adversary for the signcryption scheme, and $S$ be a simulator which simulates the adversary’s environment. Besides, $S$ is aimed to find a collision for $H_3$.

In this case, $S$ simulates the game as a normal challenger in the definition. Finally, if $A$ outputs a successful type III forgery that $z^* = z^{(i)}$, $\sigma^*_5 = \sigma^*_5^{(i)}$ and $(\sigma^*_0, \sigma^*_1^{(i)}, \sigma^*_2^{(i)}, \sigma^*_3^{(i)}, ID_{S^{(i)}}, ID_{R^{(i)}}) \neq (\sigma^*_0, \sigma^*_1, \sigma^*_2, \sigma^*_3, ID_{S}, ID_{R})$ for some $i \in \{1, ..., q_s\}$, then $S$ finds a collision for hash function $H_3$, since in this case $\theta^* = \theta^{(i)}$.

7. More Discussions

Our identity-based signcryption scheme smartly combines the Waters IBE and a variation of Paterson and Schult IBS. Recall that the signature in Paterson and Schult IBS is $(g^t, d_{P_1}, d_{P_2}(w_0 \prod_{i=1}^{n} w_i^{c_i}))(c)$ with $c \leftarrow H_4(M_s)$. The original scheme only satisfies weak unforgeability. To achieve strong unforgeability, we apply a general transfer method proposed by Boneh, Shen and Waters [4]. The signature in the resulted scheme is $(g^t, d_{P_1}, d_{P_2}(w_0 \prod_{i=1}^{n} w_i^{c_i}))(c)$ with $c \leftarrow H_4(g^{M_s}, g_3^t)$.

Table 1 and Table 2 compare our proposed identity-based signcryption scheme with the Jin, Wen and Du scheme [9], the Zhang scheme [17] as well as the traditional Encrypt-then-Sign (E-t-S) and Sign-then-Encrypt (S-t-E) combination by making use of Waters IBE and the variation of Paterson and Schult IBS. Table 1 focuses on efficiency, while Table 2 focuses on security (including IND-IBSC-CCA, EUF-IBSC-CMA as well as sEUF-IBSC-CMA) and properties (including public verifiability and forward security). Public verifiability means the validity of signcrypted text can...
Table 1. Comparison on efficiency.

<table>
<thead>
<tr>
<th></th>
<th>Signcryptext Size</th>
<th>Signcryption Cost</th>
<th>Unsigncryption Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>[9]</td>
<td>$</td>
<td>G_T</td>
<td>+ 4</td>
</tr>
<tr>
<td>[17]</td>
<td>$</td>
<td>G_T</td>
<td>+ 4</td>
</tr>
<tr>
<td>Our</td>
<td>$</td>
<td>G_T</td>
<td>+ 4</td>
</tr>
<tr>
<td>E-t-S</td>
<td>$</td>
<td>G_T</td>
<td>+ 5</td>
</tr>
<tr>
<td>S-t-E</td>
<td>$</td>
<td>G_T</td>
<td>+ 5</td>
</tr>
</tbody>
</table>

$|*|$ means the length of elements in group $*$. $\text{pairing}$, $\exp$ and $H_w$ means the computation time of doing pairing, modular exponentiation and Waters-hash once respectively. Waters-hash is $H_w(W, c, n_3) = w_0 \prod_{x=1}^{n_3} w_x^c$.

Table 2. Comparison on securities and properties.

<table>
<thead>
<tr>
<th></th>
<th>IND-IBSC -CCA</th>
<th>EUF-IBSC -CMA</th>
<th>sEUF-IBSC -CMA</th>
<th>Forward Security</th>
<th>Public Verifiability</th>
</tr>
</thead>
<tbody>
<tr>
<td>[9]</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>?</td>
<td>No</td>
</tr>
<tr>
<td>[17]</td>
<td>No</td>
<td>?</td>
<td>No</td>
<td>?</td>
<td>No</td>
</tr>
<tr>
<td>Our</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>E-t-S</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>S-t-E</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>?</td>
<td>No</td>
</tr>
</tbody>
</table>

"?" means we are not sure.

be verified only by public information. And forward security in signcryption means even if the sender's private key is exposed, an attacker without the knowledge of the receiver's private key still cannot recover the message signcrypted to the receiver before. According to Libert and Quisquater's point of view [10], to design a signcryption scheme satisfying both forward security and public verifiability is not an easy work.

From Table 1 we can see, the efficiency of our scheme is comparable to the Zhang scheme and it is more efficient than the S-t-E as well as the E-t-S construction. From Table 2 we can see our scheme satisfies all the listed security requirements and properties, while the other constructions cannot.

8. Conclusions

In this paper, we first find attacks on two identity-based signcryption schemes which are claimed to be provably secure without random oracles. After studying the failure of these two schemes, we further propose a new construction on identity-based signcryption and prove that it is secure without random oracles.
References


