On the Security of ID Based Signcryption Schemes

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. A signcryption scheme is secure only if it satisfies both the confidentiality and the unforgeability properties. All the ID based signcryption schemes presented in the standard model till now do not have either the confidentiality or the unforgeability or both of these properties. Cryptanalysis of some of the schemes have been proposed already. In this work, we present the security attacks on 'Secure ID based signcryption in the standard model' proposed by Li-Takagi and 'Further improvement of an identity-based signcryption scheme in the standard model' by Li et al. and the flaws in the proof of security of 'Efficient ID based signcryption in the standard model' proposed by Li et al., which are the recently proposed ID based signcryption schemes in the standard model. We also present the cryptanalysis of 'Construction of identity based signcryption schemes' proposed by Pandey-Barua and the cryptanalysis of 'Identity-Based Signcryption from Identity-Based Cryptography' proposed by Lee-Seo-Lee. These schemes present the methods of constructing an ID based signcryption scheme in the random oracle model from an ID based signature scheme and an ID based encryption scheme. Since none of the existing schemes in the standard model are found to be provably secure, we analyse the security of signcryption schemes got by directly combining an ID based signature scheme and an ID based encryption scheme in the standard model.

Keywords: Provable Security, ID-based signcryption, Cryptanalysis

1 Introduction

The aim of signcryption is to provide simultaneously, the confidentiality property of encryption and authentication and non-repudiation properties of signature, with a cost significantly lower than the cost of performing encryption and signature separately. The reduction in the computational cost makes a signcryption scheme more practical to be implemented in the areas like e-commerce and authenticated email. Zheng [25] introduced this notion in 1997.

Shamir [17] introduced the notion of ID based cryptography suggesting the use of an user's identity such as his email address or telephone number as his public key. Malone-Lee [14] proposed the first ID based signcryption scheme and he proved its security in the random oracle model. Many ID based signcryption schemes were proposed after [14] in the random oracle model including [3], [5], [13], [4], [1].

In 2009, Yu et al. [22] proposed the first ID based signcryption scheme in the standard model. But it was shown to be insecure by [18], [24] and [23]. Many such schemes [[22], [21], [6], [23]] were proposed after this, which were later shown to be insecure. The security notions claimed by various ID based signcryption schemes in the standard model and the type of cryptanalysis of those schemes that were proposed are tabulated in Table 1.

This paper is organized as follows. First, we present the cryptanalysis of *Secure identity-based signcryption in the standard model* proposed by Li et al. [12]. Then, we analyze the inconsistencies in the proof of security of *Efficient identity-based signcryption in the standard model* by Li et al. [11] and the cryptanalysis of *Further improvement of an identity-based signcryption scheme in the standard model* by Li et al. [10]. We then present the cryptanalysis of Pandey et al.'s *Construction of identity based signcryption schemes* [15] and the cryptanalysis of *Identity-Based Signcryption from Identity-Based Cryptography* proposed by Lee et al. [8]. Finally, we present our analysis on the security of signcryption schemes got by various methods of direct combination of an IBE and an IBS in the standard model.

Table 1. Existing ID based signcryption schemes in the standard model and their cryptanalysis

Scheme	**Confidentiality**	**Unforgeability**	**Cryptanalysis**	**Type of Attack**
Yu et al. [22]	IND-CCA2	SUF-CMA	Wang et al. [18], Zhang et al. [24] Zhang [23]	IND-CCA2 insecure IND-CCA2 and SUF-CMA insecure
Yanli et al. [21]	IND-CCA2	EUF-CMA	Wang et al. [19]	IND-CCA2 and EUF-CMA insecure
Jin et al. [6]	IND-CCA2	EUF-CMA	Li et al. [9]	IND-CCA2 and EUF-CMA insecure
Zhang [23]	IND-CCA2	SUF-CMA	Li et al. [12]	IND-CCA2 insecure
Li et al. [12]	IND-CCA2	EUF-CMA	Ours	IND-CCA2 and EUF-CMA insecure
Li et al. [11]	IND-CCA2	EUF-CMA	Ours	IND-CCA2 (not provably secure)
Li et al. [10]	IND-CCA2	EUF-CMA	Ours	IND-CCA2 and EUF-CMA insecure

IND-CCA2 - Indistinguishability under Adaptive Chosen Ciphertext Attack
EUF-CMA - Existential Unforgeability under Chosen Message Attack
SUF-CMA - Strong Existential Unforgeability under Chosen Message Attack

2 Cryptanalysis of Li et al.'s Scheme[12]

As mentioned in Table 1, Li et al. [12] have shown that the signcryption scheme proposed by Zhang [23] is IND-CCA2 insecure and proposed a new scheme that they claimed it to be existentially unforgeable and IND-CCA2 secure. But, here we show that [12] has neither the IND-CCA2 property nor the EUF-CMA property.

2.1 Review of Li et al.'s Scheme [12]

Setup

Given a security parameter, the PKG chooses groups $\mathbb{G}$ and $\mathbb{G}_T$ of prime order p, a generator g of $\mathbb{G}$ and a bilinear map $\hat{e}$: $\mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$. The PKG chooses a secret key $\alpha \in Z_p$ randomly and computes $g_1 = g^\alpha$ and chooses $g_2, h \in \mathbb{G}$ randomly. The PKG chooses random values $u', m' \in \mathbb{G}$ and vectors $U = (u_i), M = (m_i)$ of length n_u and n_m respectively, whose elements are chosen at random from $\mathbb{G}$. There are two hash functions defined as $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$ and $H_2 : \mathbb{G} \rightarrow \{0,1\}^{n_m}$. The PKG publishes the system parameters $params = \{\mathbb{G}, \mathbb{G}_T, \hat{e}, g, g_1, g_2, h, u', U, m', M, H_1, H_2\}$ and keeps the master secret key g_2^α to itself.

Extract

Let u be a n_u bit string representing an identity and $u[i]$ be the ith bit of u. Define $\Omega_u \subseteq \{1, 2, ..., n_u\}$ to be the set of indices i such that $u[i] = 1$. To construct the private key d_u of the identity u, PKG chooses

$r_u \in \mathbb{Z}_p$ randomly and computes

$$d_u = (d_{u1}, d_{u2}) = (g_2^{\alpha}(u' \prod_{i \in \Omega_u} u_i)^{r_u}, g^{r_u})$$

Let u_A be the n_u bit string representing Alice's identity and u_B be the n_u bit string representing Bob's identity. Let $\Omega_A \subseteq \{1, 2, ..., n_u\}$ be the set of indices i such that $u_A[i] = 1$. So, the private key of Alice is

$$d_A = (d_{A1}, d_{A2}) = (g_2^{\alpha}(u' \prod_{i \in \Omega_A} u_i)^{r_A}, g^{r_A})$$

And, the private key of Bob is

$$d_B = (d_{B1}, d_{B2}) = (g_2^{\alpha}(u' \prod_{i \in \Omega_B} u_i)^{r_B}, g^{r_B})$$

where $\Omega_B \subseteq \{1, 2, ..., n_u\}$ be the set of indices i such that $u_B[i] = 1$.

Signcrypt

To send a message $m \in \mathbb{G}_T$ to Bob, Alice follows the steps below.

- Choose $r, s \in \mathbb{Z}_p$ randomly.
- Compute $\sigma_1 = m \cdot \hat{e}(g_1, g_2)^r$.
- Compute $\sigma_2 = g^r$.
- Compute $\sigma_3 = (u' \prod_{i \in \Omega_B} u_i)^r$.
- Compute $\sigma_4 = d_{A2}$.
- Compute $t = H_1(\sigma_1, \sigma_2, \sigma_4, u' \prod_{i \in \Omega_A} u_i, u' \prod_{i \in \Omega_B} u_i)$.
- Compute $m_h = H_2(g^t h^s)$.
- Compute $\sigma_5 = d_{A1}(m' \prod_{j \in M_h} m_j)^r$, where $M_h \subseteq \{1, 2, ..., n_m\}$ denotes the set of indices j such that $m_h[j] = 1$.
- Compute $\sigma_6 = s$.

The ciphertext is $\sigma = \langle \sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6 \rangle$.

Unsigncrypt

When receiving $\sigma = \langle \sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6 \rangle$, Bob follows the steps below.

- Compute $t = H_1(\sigma_1, \sigma_2, \sigma_4, u' \prod_{i \in \Omega_A} u_i, u' \prod_{i \in \Omega_B} u_i)$.
- Compute $m_h = H_2(g^t h^{\sigma_6})$.
- Let $M_h \subseteq \{1, 2, ..., n_m\}$ denotes the set of indices j such that $m_h[j] = 1$.
- Check if the following equation holds:

$$\hat{e}(\sigma_5, g) \stackrel{?}{=} \hat{e}(g_1, g_2)\, \hat{e}(u' \prod_{i \in \Omega_A} u_i, \sigma_4)\, \hat{e}(m' \prod_{j \in M_h} m_j, \sigma_2) \tag{1}$$

  If Eq.(1) holds, return

$$m = \sigma_1 \frac{\hat{e}(d_{B2}, \sigma_3)}{\hat{e}(d_{B1}, \sigma_2)}$$

  Otherwise, the ciphertext is not valid and return $\perp$.

2.2 Attack on existential unforgeability

Let $\mathbb{A}$ be an adversary. On receiving the public parameters, $\mathbb{A}$ can generate a forgery by making use of the Signcrypt oracle as demonstrated below.

- Let ID_A be the identity for which $\mathbb{A}$ is going to generate the forgery.
- $\mathbb{A}$ queries the Signcrypt oracle for the signcryption of m from ID_A to ID_B ($O_{Signcrypt}(m, ID_A, ID_B)$).
- Let $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6)$ be the output of the signcrypt oracle.
- Now, $\mathbb{A}$ generates $\sigma^* = (\sigma_1^*, \sigma_2^*, \sigma_3^*, \sigma_4^*, \sigma_5^*, \sigma_6^*)$ where $\sigma_3^* \in_R \mathbb{G}$ and $\sigma_i^* = \sigma_i$, for $i = 1, 2, 4, 5, 6$.
- Here, σ^* is a valid forgery by $\mathbb{A}$ since it is a signcryption of some message m^* (not known to $\mathbb{A}$) from ID_A to ID_B which is not the output of the signcrypt oracle.

Thus, we have shown that $\mathbb{A}$ can generate a valid forgery by querying the signcrypt oracle once and hence [12] is not existentially unforgeable.

Correctness of the attack

We now show that σ^* is indeed a valid signcryption of some message m^* from ID_A to ID_B.

During the Unsigncrypt of σ^*,

- $t^* = H_1(\sigma_1^*, \sigma_2^*, \sigma_4^*, u' \prod_{i \in \Omega_A} u_i, u' \prod_{i \in \Omega_B} u_i) = H_1(\sigma_1, \sigma_2, \sigma_4, u' \prod_{i \in \Omega_A} u_i, u' \prod_{i \in \Omega_B} u_i) = t$, where t is the value generated during the execution of $O_{Signcrypt}(m, ID_A, ID_B)$.

- ${m_h}^* = H_2(g^{t^*} h^{\sigma_6^*}) = H_2(g^t h^{\sigma_6}) = m_h$, as in $O_{Signcrypt}(m, ID_A, ID_B)$.

- Therefore the test
$$\hat{e}(\sigma_5^*, g) \stackrel{?}{=} \hat{e}(g_1, g_2)\, \hat{e}(u' \prod_{i \in \Omega_A} u_i, \sigma_4^*)\, \hat{e}(m' \prod_{j \in M_h} m_j, \sigma_2^*) \tag{2}$$
is identical to the test for validity of $\sigma = \langle \sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6 \rangle$,
$$\hat{e}(\sigma_5, g) \stackrel{?}{=} \hat{e}(g_1, g_2)\, \hat{e}(u' \prod_{i \in \Omega_A} u_i, \sigma_4)\, \hat{e}(m' \prod_{j \in M_h} m_j, \sigma_2)$$
and hence equation (2) will hold true.

Thus it is clear that σ^* is a valid forgery of some message m^* from ID_A to ID_B.

2.3 Attack on Confidentiality

Let us assume $\mathbb{A}$ to be an adversary to the signcryption scheme and $\mathbb{C}$ be the challenger providing training to $\mathbb{A}$. We now show here, an attack on the confidentiality property of the signcryption scheme by using the Unsigncrypt oracle. The attack is described below.

- Let (m_0, m_1) be two equal length messages chosen by $\mathbb{A}$ and given to $\mathbb{C}$ during the challenge phase.
- Let $\sigma = \langle \sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6 \rangle$ be the challenge signcryption generated by $\mathbb{C}$ by querying the Signcrypt oracle as $O_{Signcrypt}(m_b, ID_A, ID_B)$ with $b \in_R \{0, 1\}$. This σ is given to $\mathbb{A}$ as challenge ciphertext.
- Now, $\mathbb{A}$ generates $\sigma^* = \langle \sigma_1^*, \sigma_2^*, \sigma_3^*, \sigma_4^*, \sigma_5^*, \sigma_6^* \rangle$ where $\sigma_3^* = \sigma_3\, \beta$ with $\beta \in_R \mathbb{G}$ and $\sigma_i^* = \sigma_i$ for $i = 1, 2, 4, 5, 6$.
- Now, $\mathbb{A}$ queries the Unsigncrypt oracle with σ^* as input i.e. $O_{Unsigncrypt}(\sigma^*, ID_A, ID_B)$. This query is legal since $\sigma^* \neq \sigma$ i.e the σ^* queried to the Unsigncrypt oracle is different from the challenge ciphertext σ.
- Since σ^* is valid (as shown in the correctness of the unforgeability attack), the unsigncrypt oracle returns $m^* = m_b\, \hat{e}(d_{B2}, \beta)$ to $\mathbb{A}$ as proved in Lemma 1.

– Now, $\mathbb{A}$ queries the signcrypt oracle with some message m and sender as ID_B and receiver as ID_A i.e. $O_{Signcrypt}(m, ID_B, ID_A) \rightarrow \sigma'$ to get the value d_{B2} (since by definition of signcrypt algorithm, in $\sigma' = \langle \sigma'_1, \sigma'_2, \sigma'_3, \sigma'_4, \sigma'_5, \sigma'_6 \rangle$, σ'_4 will be d_{B2}).
– Now, $\mathbb{A}$ can get m_b from m^* as $m_b = \dfrac{m^*}{\hat{e}(d_{B2}, \beta)}$. Here, $\mathbb{A}$ knows m^*, d_{B2} and β generated by $\mathbb{C}$.
– Thus, $\mathbb{A}$ can find the exact m_b of σ (the challenge ciphertext).

Lemma 1. *Let $\sigma = \langle \sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6 \rangle$ be the output of the Signcrypt algorithm in [12] for a message m, from a sender with identity ID_A to a receiver with identity ID_B. Let $\sigma^* = \langle \sigma_1^*, \sigma_2^*, \sigma_3^*, \sigma_4^*, \sigma_5^*, \sigma_6^* \rangle$ be another signcryption from the same sender ID_A to the same receiver ID_B, with $\sigma_i^* = \sigma_i$ for $i = 1, 2, 4, 5, 6$ and $\sigma_3^* = \sigma_3\, \beta$, where $\beta \in_R \mathbb{G}$. Then, σ^* is valid and the message signcrypted by σ^* is m^* where $m^* = m\, \hat{e}(d_{B2}, \beta)$.*

Proof. When σ^* is given as input to the Unsigncrypt algorithm, equation (1) will hold good since $\sigma_1^* = \sigma_1$, $\sigma_2^* = \sigma_2$, $\sigma_4^* = \sigma_4$, $\sigma_5^* = \sigma_5$ and since m_h of σ and σ^* are the same ($\because \sigma_6^* = \sigma_6$), as explained in the correctness of the attack on the existential unforgeability property of [12].

Hence, the Unsigncrypt of σ^* returns m^*, where

$$\begin{aligned}
m^* &= \sigma_1^* \, \frac{\hat{e}(d_{B2}, \sigma_3^*)}{\hat{e}(d_{B1}, \sigma_2^*)} \\
&= \sigma_1 \, \frac{\hat{e}(d_{B2}, \sigma_3\, \beta)}{\hat{e}(d_{B1}, \sigma_2)} \ (\because \sigma_1^* = \sigma_1, \sigma_2^* = \sigma_2, \sigma_3^* = \sigma_3\, \beta) \\
&= \sigma_1 \, \frac{\hat{e}(d_{B2}, \left(u' \prod_{i \in \Omega_B} u_i\right)^r \ \beta)}{\hat{e}(d_{B1}, g^r)} \\
&= \frac{\left[m \ (\hat{e}(g_1, g_2))^r\right] \left[\hat{e}(g^{r_B}, \left(u' \prod_{i \in \Omega_B} u_i\right)^r \ \beta)\right]}{\hat{e}(g_2^{\alpha} \left(u' \prod_{i \in \Omega_B} u_i\right)^{r_B}, g^r)} \\
&= \frac{m \ (\hat{e}(g_1, g_2))^r \left[\hat{e}(g^{r_B}, \left(u' \prod_{i \in \Omega_B} u_i\right)^r)\, \hat{e}(g^{r_B}, \beta)\right]}{\hat{e}(g_2, g^{\alpha})^r \, \hat{e}(\left(u' \prod_{i \in \Omega_B} u_i\right)^{r_B}, g)^r} \\
&= \frac{m \ (\hat{e}(g_1, g_2))^r \left[\hat{e}(g, \left(u' \prod_{i \in \Omega_B} u_i\right)^{r_B})^r \, \hat{e}(d_{B2}, \beta)\right]}{\hat{e}(g_2, g_1)^r \, \hat{e}(\left(u' \prod_{i \in \Omega_B} u_i\right)^{r_B}, g)^r} \\
&= m\, \hat{e}(d_{B2}, \beta)
\end{aligned}$$

Thus, σ^* is valid signcryption of m^* from a sender with identity ID_A to a receiver with identity ID_B , where $m^* = m\, \hat{e}(d_{B2}, \beta)$.

3 Analysis of Inconsistencies in the proof of 'Efficient Identity-Based Signcryption in the Standard Model' scheme proposed by Li et al. [11]

In 2011, Li et al. [11] have proposed a signcryption scheme in the ID based setting. This scheme is shown to be secure in the standard model. Here, we show that the proof given for the scheme in [11] has some flaws.

3.1 Review of the scheme

This section reviews Li et al.'s *Efficient Identity-Based Signcryption in the Standard Model* [11].

Setup

Given a security parameter k, the PKG chooses two multiplicative cyclic groups $\mathbb{G}$ and $\mathbb{G}_T$ of prime order p, a generator g of $\mathbb{G}$ and a bilinear map $\hat{e}$: $\mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$. The PKG chooses $\alpha, w \in \mathbb{G}$ randomly and computes $z = \hat{e}(\alpha, g)$. The PKG chooses random values $u', v' \in \mathbb{G}$ and vectors $U = (u_i), V = (v_i)$ of length n_{id} and n_m respectively, whose elements are chosen at random from $\mathbb{G}$. There are two hash functions defined as

$H_1 : \mathbb{G} \to \mathbb{Z}_p^*$ and $H_2 : \{0,1\}^* \to \{0,1\}^{n_m}$. There is a secure one time symmetric key encryption scheme $SE = (E, D)$ with key space $\kappa = \mathbb{G}_T$. There are another two hash functions defined as $H_3 : \{0,1\}^{n_{id}} \to \mathbb{G}$ and $H_4 : \{0,1\}^{n_m} \to \mathbb{G}$.

$$H_3(id) = u' \prod_{i=1}^{n_{id}} u_i^{id_i} \qquad H_4(\pi) = v' \prod_{i=1}^{n_m} v_i^{\pi_i}$$

These are the kind of functions that are used to construct IBE scheme by Waters [20], where π is the output of the hash H_2 with length n_m. The PKG publishes the system parameters

$$params = \{\mathbb{G}, \mathbb{G}_T, \hat{e}, g, w, z, u', U, v', V, H_1, H_2, H_3, H_4, SE\}$$

and keeps the master secret key α to itself.

Extract

To construct the private key sk_{id} of the identity id, PKG chooses $s \in \mathbb{Z}_p^*$ randomly and computes

$$sk_{id} = (sk_1, sk_2, sk_3) = (\alpha \cdot H_3(id)^s, g^s, w^s)$$

Let id_A be Alice's identity and id_B be Bob's identity. The private key of Alice is,

$$sk_A = (sk_{A1}, sk_{A2}, sk_{A3}) = (\alpha \cdot H_3(id_A)^{s_A}, g^{s_A}, w^{s_A})$$

The private key of Bob is

$$sk_B = (sk_{B1}, sk_{B2}, sk_{B3}) = (\alpha \cdot H_3(id_B)^{s_B}, g^{s_B}, w^{s_B})$$

Signcrypt

To send a message $m \in \mathbb{G}_T$ to Bob, Alice follows the steps below.

- Choose $r \in \mathbb{Z}_p^*$ randomly.
- Compute $c_1 = g^r$.
- Compute $t = H_1(c_1)$.
- Set $c_2 = sk_{A2}$.
- Compute $K = z^r$
- Compute $c_3 = E_K(m)$.
- Compute $c_4 = (H_3(id_B) \cdot w^t)^r$.
- Compute $\pi = H_2(c_1, c_2, c_3, c_4)$.
- Compute $c_5 = sk_{A1} H_4(\pi)^r \, c_4$

The ciphertext is $c = (c_1, c_2, c_3, c_4, c_5)$.

Unsigncrypt

When receiving $c = (c_1, c_2, c_3, c_4, c_5)$, Bob follows the steps below.

- Compute $\pi = H_2(c_1, c_2, c_3, c_4)$ and $t = H_1(c_1)$.
- Check if the following equation holds:

$$\hat{e}(c_5, g) \stackrel{?}{=} z \cdot \hat{e}(H_3(id_A), c_2) \cdot \hat{e}(H_4(\pi) \cdot H_3(id_B) \cdot w^t, c_1) \tag{3}$$

If Eq.(3) holds, compute

$$K = \frac{\hat{e}(c_1, sk_{B1} \cdot sk_{B3}^t)}{\hat{e}(c_4, sk_{B2})}$$

and message is calculated as $m = D_K(c_3)$. Otherwise, the ciphertext is not valid and return $\perp$.

3.2 Analysis of the inconsistencies in the security proof

The flaws in the proof of IND-CCA2 property are

- According to the definition of the Signcrypt protocol, the signcryption $c = (c_1, c_2, c_3, c_4, c_5)$ on any message from a sender id_A to any receiver will always have the same $c_2 = sk_{A2}$. But in the simulation of the Signcrypt oracle, a Signcrypt query will output a different $c_2 = g^{r_i}$, $r_i \in_R \mathcal{Z}_p^*$ each time when the oracle is invoked with id_A as sender, for which $J(id_A) = 0 \; mod \; \mathbb{Z}_{f_u}$, where $f_u = 4l_u$ and l_u is the length of any identity.
- Also, the signcryption $c = (c_1, c_2, c_3, c_4, c_5)$ from id_A to id_B satisfies that $(g, H_3(id_B), c_1, c_4)$ is a valid Diffie-Hellman tuple i.e it always passes the following test,

$$\hat{e}(c_4, g) \stackrel{?}{=} \hat{e}(c_1, H_3(id_B)w^t) \tag{4}$$

where $t = H_1(c_1)$.

But, the values c_1' and c_4' in the output of the Signcryption oracle $c' = (c_1', c_2', c_3', c_4', c_5')$ will be simulated as follows.

$$c_1' = sk_{B2}$$
$$c_4' = g^r . \, sk_{B1} . \, sk_{B3}^t$$

This c' fails to satisfy Eq. 4 for any sender with identity id_A and receiver with identity id_B, having $J(id_A) = 0 \; mod \; \mathbb{Z}_{f_u}$ and $J(id_B) \neq 0 \; mod \; \mathbb{Z}_{f_u}$ as shown below.

$$\begin{aligned}
\hat{e}(c_4', g) &= \hat{e}(g^r \, sk_{B1} \, sk_{B3}^t, g) \\
&= \hat{e}(g^r, g) \, \hat{e}(\alpha \, H_3(id_B)^s, g) \, \hat{e}(w^{st}, g) \\
&= \hat{e}(g^r, g) \, \hat{e}(\alpha, g) \, \hat{e}(H_3(id_B)^s, g) \, \hat{e}(w^{st}, g) \\
&= \hat{e}(g^r, g) \, \hat{e}(\alpha, g) \, \hat{e}(H_3(id_B), g^s) \, \hat{e}(w^t, g^s) \\
&= \hat{e}(g^r, g) \, \hat{e}(\alpha, g) \, \hat{e}(H_3(id_B) \, w^t, g^s) \\
\hat{e}(c_4', g) &= \hat{e}(g^r, g) \, \hat{e}(\alpha, g) \, \hat{e}(H_3(id_B) \, w^t, c_1') \\
\hat{e}(c_4', g) &\neq \hat{e}(c_1', H_3(id_B) \, w^t)
\end{aligned}$$

Here, the probability for $\alpha = g^{-r}$ is negligible.

These make the simulation imperfect i.e the simulation is different from the real protocol.

Here, for the challenge phase to succeed without aborting, $J(id_A^*) \neq 0 \; mod \; \mathbb{Z}_{f_u}$ and $J(id_B^*) = 0 \; mod \; \mathbb{Z}_{f_u}$. But, during the *Phase* 2 of training the adversary, the Signcrypt oracle will abort for all queries with receiver identity as id_B^*. And also, for the Signcrypt queries with sender identity as id_B^* and receiver identity as id_A^*, the difference in the simulation from the real protocol can be easily distinguished by the adversary.

4 Cryptanalysis of the improved Identity based Signcryption scheme in the Standard Model by Li et al. [10]

In this section, we review the scheme proposed by Li et al. [10] and then show its security weaknesses.

4.1 Review of the scheme

Setup

Given a security parameter, the PKG chooses groups $\mathbb{G}$ and $\mathbb{G}_T$ of prime order p, a generator g of $\mathbb{G}$, and a bilinear map $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$. The PKG chooses $\alpha, \mu, v \in \mathbb{G}$ randomly and computes $z = \hat{e}(\alpha, g)$. Additionally, the PKG chooses random values $u_0, m_0 \in \mathbb{G}$ and vectors $U = (u_i), M = (m_i)$ of length n_u and n_m, respectively, whose elements are chosen at random from $\mathbb{G}$. We also need a hash function $H_1 : \mathbb{G} \rightarrow \mathcal{Z}_p^*$ and a secure one-time symmetric key encryption scheme (E, D) with key space $\kappa = \mathbb{G}_T$. The PKG publishes system parameters $params = \{\mathbb{G}, \mathbb{G}_T, \hat{e}, g, \mu, v, z, u_0, U, m_0, M, H_1, E, D\}$ and keeps the master secret α to itself.

Extract

Let u be a n_u bit string representing an identity and $u[i]$ be the ith bit of u. Define $\Omega_u \subseteq \{1, ..., n_u\}$ to be the set of indices i such that $u[i] = 1$. To construct the private key d_u of the identity u. The PKG chooses $r_u \in \mathbb{Z}_p$ randomly and computes

$$d_u = (d_{u1}, d_{u2}) = (g_2^{\alpha}(u_0 \prod_{i \in \Omega_u} u_i)^{r_u}, g^{r_u})$$

Let u_A be the n_u bit string representing Alice's identity and u_B be the n_u bit string representing Bob's identity. Let $\Omega_A \subseteq \{1, 2, ..., n_u\}$ be the set of indices i such that $u_A[i] = 1$. So, the private key of Alice is

$$d_A = (d_{A1}, d_{A2}) = (g_2^{\alpha}(u_0 \prod_{i \in \Omega_A} u_i)^{r_A}, g^{r_A})$$

And, the private key of Bob is

$$d_B = (d_{B1}, d_{B2}) = (g_2^{\alpha}(u_0 \prod_{i \in \Omega_B} u_i)^{r_B}, g^{r_B})$$

where $\Omega_B \subseteq \{1, 2, ..., n_u\}$ be the set of indices i such that $u_B[i] = 1$.

Signcrypt

To send a message to Bob, Alice follows the following steps. Let $M \subseteq \{1, ..., n_m\}$ is the set of indices j such that $m[j] = 1$, where $m[j]$ is the jth bit of m.

- Choose $r \in \mathcal{Z}_p^*$ randomly and compute $\sigma_1 = g^r$
- Compute $t = H_1(\sigma_1)$
- Compute $\sigma_2 = (u_0 \prod_{i \in \Omega_B} u_i)^r$
- Compute $\sigma_3 = (\mu^t v)^r$
- Compute $V = d_{A1}(m_0 \prod_{j \in M} m_j)^r$
- Compute $X = d_{A2}$
- Compute $K = z^r$
- Compute $\sigma_4 = E_K(V||X||m)$

The ciphertext is $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4)$.

Unsigncrypt

When receiving $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4)$, Bob follows the following steps.

- Compute
$$K = \frac{\hat{e}(d_{B1}, \sigma_1)}{\hat{e}(d_{B2}, \sigma_2)} = \frac{\hat{e}(\alpha(u_0 \prod_{i \in \Omega_B} u_i)^{r_B}, \sigma_1)}{\hat{e}(g^{r_B}, \sigma_2)}$$
- Compute $V||X||m = D_K(\sigma_4)$
- Verify if the following equation holds.
$$\hat{e}(V, g) \stackrel{?}{=} z\, \hat{e}(u_0 \prod_{i \in \Omega_A} u_i, X)\, \hat{e}(m_0 \prod_{j \in M} m_j, \sigma_1)$$

If the above equation holds, then Bob accepts the message. Otherwise Bob returns $\perp$.

4.2 Flaws in the scheme

1. The component σ_3 of the ciphertext σ is not verified of its consistency in the Unsigncrypt phase. In this case, an adversary in the EUF-CMA game, can produce a valid forgery through the following steps.
   - The adversary makes a Signcrypt query for any $\langle ID_A, ID_B, m\rangle$ tuple and gets $\sigma' = (\sigma'_1, \sigma'_2, \sigma'_3, \sigma'_4)$ as the output of the query, which is a valid signcryption on m by ID_A for ID_B.
   - Now, the adversary can produce a valid forgery $\sigma^* = (\sigma^*_1, \sigma^*_2, \sigma^*_3, \sigma^*_4)$, where σ^*_3 is randomly chosen from $\mathcal{Z}^*_p - \{\sigma'_3\}$ and $\sigma^*_1 = \sigma'_1, \sigma^*_2 = \sigma'_2, \sigma^*_4 = \sigma'_4$.
   - This σ^* is a valid signcryption on m by ID_A and with ID_B as the receiver.

   Also, an adversary in the IND-CCA2 game can distinguish the challenge ciphertext whether it is the signcryption of m^*_0 or m^*_1 through the following steps (which are similar to the steps above).
   - The adversary after getting the challenge ciphertext $\sigma^* = (\sigma^*_1, \sigma^*_2, \sigma^*_3, \sigma^*_4)$, queries the Unsigncrypt oracle with $\langle \sigma', ID^*_A, ID^*_B\rangle$ as input, where $\sigma' = (\sigma'_1, \sigma'_2, \sigma'_3, \sigma'_4)$, with σ'_3 randomly chosen from $\mathcal{Z}^*_p - \{\sigma^*_3\}$ and $\sigma'_1 = \sigma^*_1, \sigma'_2 = \sigma^*_2, \sigma'_4 = \sigma^*_4$.
   - The output of the Unsigncrypt oracle for this query reveals m^*_β to the adversary, from which it can output $\beta \in \{0, 1\}$ successfully with probability 1.
2. Now, we consider the security of the scheme [10] including the following verification step in the Unsigncrypt oracle.
$$\hat{e}(\sigma_3, g) \stackrel{?}{=} \hat{e}(\mu^{t'} v, \sigma_1) \tag{5}$$
   where $t' = H_1(\sigma_1)$.
   When this verification step is included, the scheme becomes secure against the security game proposed in Step 1. But the simulation of the Signcrypt oracle provided by [10] becomes inconsistent with respect to this verification step, making the scheme provably insecure.
3. When we analyze this scheme further, we find out that even if proper signcrypt and unsigncrypt oracles were provided, the security of this scheme is susceptible to the EUF-CMA security game, which is described below.
   - The adversary $\mathcal{A}$ queries the tuple $\langle ID_A, ID_B, m\rangle$ to the signcrypt oracle and gets $\sigma' = (\sigma'_1, \sigma'_2, \sigma'_3, \sigma'_4)$ as output.
   - $\mathcal{A}$ also queries the extract oracle for the secret keys of ID_B, ID_C. These two extract queries are legal, since the forgery to be produced by $\mathcal{A}$ is with ID_A as sender. And hence, $\mathcal{A}$ can query for the secret key of any identity other than ID_A.
   - Now, having the secret key of ID_B, $\mathcal{A}$ can find the key K used in the one-time symmetric key encryption algorithm, by following the first step of the Unsigncrypt algorithm of [10].
   - By using this K, $\mathcal{A}$ can find the components V, X, m obtained during the generation of σ' by performing $D_K(\sigma'_4)$.
   - $\mathcal{A}$ can now generate a valid signcryption on m by ID_A with ID_C as the receiver by encrypting V, X, m with a different key $K' = \dfrac{\hat{e}(d_{C1}, \sigma_1)}{\hat{e}(d_{C2}, \sigma_2)}$.
   - Thus, $\mathcal{A}$ outputs a valid forgery $\sigma^* = (\sigma^*_1, \sigma^*_2, \sigma^*_3, \sigma^*_4)$, where $\sigma^*_1 = \sigma'_1, \sigma^*_2 = \sigma'_2, \sigma^*_3 = \sigma'_3, \sigma^*_4 = E_{K'}(V||X||m)$.
4. The real-world insecurity of the scheme which is captured by the above security game is explained below. In this scheme, ID_B and ID_C can collude to convert a valid signcryption from ID_A to ID_B to a valid signcryption from ID_A to ID_C without solving any hard problem.
   - ID_A creates a signcryption $\sigma = (\sigma_1, \sigma_2, \sigma_3, \sigma_4)$ on message m and sends it to the intended receiver ID_B.
   - ID_B finds the key K used in the one-time symmetric key encryption scheme E, during the generation of σ, by performing the first step of the unsigncrypt algorithm of [10].
   - Now, using K, ID_B can get V, X, m that are obtained during the generation of σ by performing $D_K(\sigma_4)$.
   - When ID_B passes these values $\langle V, X, m\rangle$ to ID_C, ID_C can compute $K' = \dfrac{\hat{e}(d_{C1}, \sigma_1)}{\hat{e}(d_{C2}, \sigma_2)}$.
   - Then $\sigma'_4 = E_{K'}(V||X||m)$ can be computed by ID_C.
   - Now $\sigma' = (\sigma'_1, \sigma'_2, \sigma'_3, \sigma'_4)$ is a valid signcryption on m by ID_A with ID_C as the intended receiver, where $\sigma'_1 = \sigma_1, \sigma'_2 = \sigma_2, \sigma'_3 = \sigma_3$.

Thus any two users (receivers) can collude to transform a signcryption on a message by a sender for one receiver to a valid signcryption for another receiver on the same message, without the knowledge of the sender or the sender's secret key.

Thus, we show that the scheme proposed in [10] is insecure.

5 Cryptanalysis of Pandey et al.'s signcryption scheme [15]

Here, we review *Construction of ID based signcryption schemes* proposed by Pandey et al. [15].

Setup($SecParam$)

Let $H_1 : \{0,1\}^* \rightarrow \{0,1\}^{l_1}$, $H_2 : \{0,1\}^* \rightarrow \{0,1\}^{l_2}$, $H_3 : \{0,1\}^* \rightarrow \{0,1\}^{l_1}$ be secure hash functions. The public parameters, $Params$, consist of $(Params_{IBE}, Params_{IBS}, H_1, H_2, H_3)$ and the master secret key msk is (msk_{IBE}, msk_{IBS}).

Key Generation(ID)

Let $sk_{IBE} \leftarrow KeyGen_{IBE}(ID)$ and $sk_{IBS} \leftarrow KeyGen_{IBS}(ID)$. The private key corresponding to identity ID will be (sk_{IBE}, sk_{IBS}).

Signcryption(m, ID_{Rec}, ID_{Sen}, $sk_{ID_{Sen}}$, $Params$)

1. Choose r randomly from $\mathcal{R}$.
2. Let $c' \leftarrow ENC.S_{IBE}(r,\ ID_{Rec},\ Params_{IBE})$.
3. Compute $h_1 = H_1(r, c', ID_{Sen})$.
4. Compute $h_2 = H_2(m, c', h_1, ID_{Rec}, ID_{Sen})$.
5. Compute $c = H_3(h_1, ID_{Sen}) \oplus m$.
6. Let $sk_{ID_{Sen}} = (sk_{ID_{Sen}}IBE,\ sk_{ID_{Rec}}IBS)$ and let $(m, s) \leftarrow SIG.S_{IBS}(m, sk_{ID_{Sen}}IBS, Params_{IBS})$.
7. Compute $d = h_2 \oplus s$.

The cipher-text will be $C \equiv (c', c, d)$.

Here, S_{IBE} is an identity based encryption that is IND-CCA2 secure and S_{IBS} is an identity based signature that is existentially unforgeable against chosen message attacks.

Designcryption(C, ID_{Rec}, ID_{Sen}, $Params$)

1. Let $sk_{ID_{Rec}} = (sk_{ID_{Rec}IBE}, sk_{ID_{Rec}IBS})$. Let $r' \leftarrow DEC.S_{IBE}(c', sk_{ID_{Rec}IBE}, Params_{IBE})$.
2. Compute $h_1' = H_1(r', c', ID_{Sen})$.
3. Compute $m' = H_3(h_1', ID_{Sen}) \oplus c$.
4. Compute $h_2' = H_2(m', c', h_1', ID_{Sen}, ID_{Rec})$.
5. Compute $s' = h_2' \oplus d$.
6. Let $x \leftarrow VER.S_{IBS}(m', s', ID_{Sen}, Params_{IBS})$.
   – If above step is correctly verified, then $VER.S_{IBS}(., ..., .)$ returns m', else $\perp$.
7. Return x.

5.1 Attack on Unforgeability

In this section we show that the scheme proposed in [15] does not provide the unforgeability property. During the unforgeability game, the adversary $\mathbb{A}$ can generate a valid forgery (which is a signcryption of message m with sender as ID_A and receiver as ID_B) by making use of the Signcryption and Key Generation oracles as shown below.

- Let ID_A, ID_B, ID_D be three identities.
- $\mathbb{A}$ queries the private key of ID_D to the Key Generation oracle. This query is legal since ID_D is the identity of neither the sender nor the receiver, involved in the forgery which is going to be produced.
- $\mathbb{A}$ also queries the signcryption of a message m from ID_A to ID_D to the Signcryption oracle.
- Let C be the signcryption of m output by the Signcryption oracle.
- Now, $\mathbb{A}$ designcrypts $C \equiv (c'_1, c_1, d_1)$, since it knows the private key of ID_D, by performing the following steps.
  1. $r_1 \leftarrow DEC.S_{IBE}(c'_1, sk_{ID_B IBE}, Params_{IBE})$.
  2. $h_1 = H_1(r_1, c'_1, ID_A)$.
  3. $m = H_3(h_1, ID_A) \oplus c_1$.
  4. $h_2 = H_2(m, c'_1, h_1, ID_A, ID_D)$.
  5. $s_1 = h_2 \oplus d_1$.
- We now show how $\mathbb{A}$ can produce the signcryption of m from ID_A to ID_B, without knowing the secret key of ID_A. This will prove the ability of $\mathbb{A}$ to produce forgery.
- The only step where the secret key of ID_A is involved in generating the signcryption is the Step 6 of the Signcryption algorithm where one should compute $(m, s) \leftarrow SIG.S_{IBS}(m, sk_{ID_A} IBS, Params_{IBS})$.
- However, the value of s_1 obtained in Step 5 of the Designcryption of $\langle C, ID_A, ID_D\rangle$ shown above is precisely the value of $SIG.S_{IBS}(m, sk_{ID_A} IBS, Params_{IBS})$.
- That is, $s^* = s_1 = SIG.S_{IBS}(m, sk_{ID_A} IBS, Params_{IBS})$.
- Thus, s^* can be obtained by $\mathbb{A}$ without even knowing the secret key of ID_A.
- Now $\mathbb{A}$ has no problems in executing the steps of Signcryption(m, ID_B, ID_A, sk_{ID_A}, $Params$). Specifically,
  1. $\overline{r}$ is chosen randomly from $\mathcal{R}$
  2. $c'_2 \leftarrow ENC.S_{IBE}(\overline{r}, ID_B, params)$
  3. $h^*_1 = H_1(\overline{r}, c'_2, ID_A)$
  4. $h^*_2 = H_2(m, c'_2, h^*_1, ID_B, ID_A)$
  5. $c_2 = m \oplus H_3(h^*_1, ID_A)$
  6. $d_2 = s^* \oplus h^*_2 = s_1 \oplus h^*_2$
- Now, $\mathbb{A}$ submits $C^* \equiv (c'_2, c_2, d_2)$, which is a valid forgery of message m from ID_A to ID_B.

Thus, [15] is not outsider secure since a valid forgery is produced without involving the private key of the sender or the receiver.

6 Cryptanalysis of Lee et al.'s signcryption scheme [8]

Lee et al. [8] improved Pandey et al.'s signcryption scheme [15] and claimed to achieve additional security notions, ciphertext anonymity and ciphertext authentication along with the message confidentiality and signature non-repudiation properties claimed by [15]. But here we show that the signcryption scheme proposed by Lee et al. [8] is not even IND-CPA secure.

6.1 Review of the scheme

Notation

Let $H_1 : \{0,1\}^{l_1} \to \{0,1\}^{l_2}$ and $H_2 : \{0,1\}^{l_3} \to \{0,1\}^{l_4}$ are hash functions. We assume that e_1 is the bit-length of outputs of E_{IBE}, let e_2 be the bit-length of an identity and let l_4 be the bit-length of signature s where $l_1 = l_2 + e_1 + e_2$ and $l_3 = 2l_2 + e_2$. Moreover, we assume that $param_{IBE} \bigcap param_{IBS} = \Phi$.

Construction

This IBSC scheme is based on the ordinary IBE and IBS schemes.

- Setup(1^k) $\to$ (msk_{IBE}, $param_{IBE}$, msk_{IBS}, $param_{IBS}$, H_1, H_2): Given a security parameter 1^k, output (msk_{IBE}, $param_{IBE}$, msk_{IBS}, $param_{IBS}$, H_1, H_2) where the master secret key is msk = (msk_{IBE}, msk_{IBS}), the global parameter is $param$ = ($param_{IBE}$, $param_{IBS}$, H_1, H_2).

– KeyGen(msk, ID) $\rightarrow SK_{ID}$: Given master secret key msk and an identity ID, output the private key SK_{ID} for ID. The private key is computed as follows:
  1. $SK_{IBE.ID} \leftarrow KeyGen_{IBE}(ID)$;
  2. $SK_{IBS.ID} \leftarrow KeyGen_{IBS}(ID)$;
  3. $SK_{ID} = (SK_{IBE.ID},\ SK_{IBS.ID})$.
– Signcryption(m, ID_S, ID_R, SK_{ID_S}) $\rightarrow C$: Given a message m, a sender's identity ID_S, a recipient's identity ID_R, and a sender's private key SK_{ID_S}, output a ciphertext $C = (c_1, c_2, d)$. The computation is as follows:
  1. $r \leftarrow \{0,1\}^{l_2}$;
  2. $c_1 \leftarrow E_{IBE}((r||ID_S), ID_R)$;
  3. $t_1 \leftarrow H_1(r||c_1||ID_S)$;
  4. $t_2 \leftarrow H_2(m||t_1||ID_R)$;
  5. $c_2 = t_1 \oplus m$;
  6. $s \leftarrow S_{IBS}((m||ID_R), SK_{IBS.ID_S})$;
  7. $d = t_2 \oplus s$.
– Designcryption(C, ID_R, SK_{ID_R}) $\rightarrow m$ or $\perp$: Given a ciphertext C, a recipient's identity ID_R, and a recipient's private key SK_{ID_R}, output a message m or $\perp$ indicating an error. The computation is as follows:
  1. $r||ID_S = D_{IBE}(c_1, SK_{IBE.ID_R})$;
  2. $t_1 \leftarrow H_1(r||c_1||ID_S)$;
  3. $m = t_1 \oplus c_2$;
  4. $t_2 \leftarrow H_2(m||t_1||ID_R)$;
  5. $s = t_2 \oplus d$;
  6. m or $\perp = V_{IBS}((m||ID_R), s, ID_S)$.

6.2 Attack on message confidentiality

The authors have claimed that the scheme proposed in [8] is IND-IBSC-CCA secure. But here we show that it is IND-IBSC-CPA insecure as follows.

– During the IND-CPA game, the adversary $\mathcal{A}$ randomly chooses two messages, say m_0^* and m_1^*, sender identity ID_S^* and receiver identity ID_R^* and gives them to the challenger.
– The challenger randomly chooses $\beta \in_R \{0,1\}$ and gives $C^* \leftarrow$ Signcryption(m_β, ID_S^*, ID_R^*) to $\mathcal{A}$.
– Now, $\mathcal{A}$ can find out whether $C^* = (c_1^*, c_2^*, d^*)$ is a valid signcryption of m_0^* or m_1^* as shown below.
  - $\mathcal{A}$ initially guesses m_β to be m_0^* and hence calculates $t_1^* = c_2^* \oplus m_0^*$. Thus, the value of t_1^* is got by $\mathcal{A}$ without the knowledge of the secret key of the receiver $SK_{IBE.ID_R}$.
  - Still assuming that $m_\beta = m_0^*$, $\mathcal{A}$ calculates t_2^* as $t_2^* = H_2(m_0^*||t_1^*||ID_R^*)$.
  - Now, s^* can be got by $\mathcal{A}$ as $s^* = d^* \oplus t_2^*$.
  - The guess that $m_\beta = m_0^*$ can be validated by the verification algorithm of the underlying signature scheme IBS i.e $V_{IBS}((m_0^*, ||ID_R^*), s^*, ID_S^*)$.
  - If $V_{IBS}((m_0^*, ||ID_R^*), s^*, ID_S^*)$ returns $Valid$, then C^* is a valid signcryption on m_0^* and hence the guess made by $\mathcal{A}$ is correct i.e $m_\beta = m_0^*$.
    Otherwise, $m_\beta = m_1^*$, since C^* is a valid signcryption of either m_0^* or m_1^*.

Thus, the adversary $\mathcal{A}$ can always distinguish whether the challenge signcryption C^* is a valid signcryption on m_0^* or m_1^*, proving that the signcryption scheme in [8] is not even IND-IBSC-CPA secure.

6.3 Absence of Ciphertext anonymity

Lee et al. [8] have also claimed that the signcryption scheme proposed by them has the property of ciphertext anonymity. But during ANON-IBSC-CCA game defined by [8], the adversary can always distinguish the sender and receiver identities as shown below.

1. After training phase 1, during the ANON-IBSC-CCA game, the adversary $\mathcal{A}$ produces a message m^* along with two distinct sender identities (ID_{S0}, ID_{S1}) and two distinct receiver identities (ID_{R0}, ID_{R1}) to the challenger.

2. The challenger now chooses two bits $a, b \in_R \{0,1\}$ and computes the challenge ciphertext $C^* = \langle c_1^*, c_2^*, d^* \rangle$ which is a signcryption on the message m^* with the sender identity as ID_{Sa} and receiver identity as ID_{Rb} and returns C^* to $\mathcal{A}$.
3. Now, $\mathcal{A}$ can obtain t_1^* that would have been obtained during the generation of the challenge ciphertext C^* as $t_1^* = c_2^* \oplus m^*$.
4. Having obtained the values of t_1^* and m^*, $\mathcal{A}$ guesses the receiver identity ID_{Rb} to be ID_{R0} and calculates $t_2' = H_2(m^*||t_1^*||ID_{R0})$.
5. $\mathcal{A}$ then calculates $s' = d^* \oplus t_2'$. Note that the values t_2' and s' got here are based on the guess that the receiver identity is ID_{R0}. Hence, only if $b = 0$, t_2^* would be t_2' and s^* would be s'.
6. The adversary $\mathcal{A}$ can now verify its guess regarding the receiver identity and identify the sender identity from the following steps.
   - If $V_{IBS}((m||ID_{R0}), s', ID_{S0})$ returns m^*, then the sender identity is ID_{S0} and the receiver identity is ID_{R0} i.e $a = 0$ and $b = 0$ respectively.
   - Else,
     - If $V_{IBS}((m||ID_{R0}), s', ID_{S1})$ returns m^*, then the sender identity is ID_{S1} and the receiver identity is ID_{R0} i.e $a = 1$ and $b = 0$. respectively.
     - Else, $\mathcal{A}$ outputs the receiver identity to be ID_{B1} i.e $b = 1$. In order to find the sender identity, $\mathcal{A}$ repeats this process from Step 4, with the receiver identity as ID_{R1}. Now, the t_2' and s' got here are respectively equal to t_2^* and s^* got during the generation of the challenge ciphertext C^*. So,
       * If $V_{IBS}((m||ID_{R1}), s', ID_{S0})$ returns m^*, then the sender identity is ID_{S0} i.e $a = 0$.
       * Else, the sender identity is ID_{S1} i.e $a = 1$.

Thus, the identities of the sender and the receiver can always be distinguished by the adversary during the ANON-IBSC-CCA game, after the challenge ciphertext is given to it, refuting the claim of the authors of [8] that the signcryption scheme proposed in [8] provides ciphertext anonymity.

6.4 Attack on unforgeability

In the AUTH-IBSC-CMA security game, the adversary $\mathcal{A}$ can produce a valid forgery C^* on message m with sender and receiver identities as ID_S^* and ID_R^* respectively, as follows.

- Initially, $\mathcal{A}$ obtains a valid signature s on the message m by ID_S^* with ID_R^* as the intended receiver, through the following steps.
  - $\mathcal{A}$ makes a signcryption query to the challenger with $\langle m, ID_S^*, ID_R^* \rangle$ as input. The challenger returns the output of the signcryption oracle $C = \langle c_1, c_2, d \rangle$ to $\mathcal{A}$.
  - From this valid signcryption C on m got from the signcryption query mentioned above, $\mathcal{A}$ can find the value of t_1 that is obtained during the execution of this signcryption query as $t_1 = c_2 \oplus m$.
  - $\mathcal{A}$ then can calculate the value of t_2 obtained during the same signcryption query as $t_2 \leftarrow H_2(m||t_1||ID_R^*)$.
  - Now, $\mathcal{A}$ can obtain s as $s = d \oplus t_2$.
  - Thus the adversary $\mathcal{A}$ could obtain s which is a valid signature on the message m intended for the receiver ID_R^*, without the knowledge of the secret key of the sender involved in generating the signature i.e $SK_{IBS.ID_S^*}$.
- Now, with the value of s, $\mathcal{A}$ can produce another valid signcryption on the message m with the sender and receiver identities as ID_S^* and ID_R^* respectively as shown below.
- $\mathcal{A}$ randomly chooses $r \leftarrow \{0,1\}^{l_2}$ and calculates $c_1^* \leftarrow E_{IBE}((r||ID_S^*), ID_R^*)$.
- $\mathcal{A}$ then calculates t_1^* and t_2^* as $t_1^* \leftarrow H_1(r||c_1||ID_S^*)$ and $t_2^* \leftarrow H_2(m||t_1^*||ID_R^*)$.
- Now, $\mathcal{A}$ computes $c_2^* = t_1^* \oplus m$ and $d^* = t_2^* \oplus s$, where s is the valid signature got from the steps explained above.
- The tuple $\langle c_1^*, c_2^*, d^* \rangle$ is output as forgery by the adversary $\mathcal{A}$.

Note that, $C^* = \langle c_1^*, c_2^*, d^* \rangle$ is not the output of any signcryption oracle. However, C^* is signcryption on m from ID_S^* to ID_R^*. That is, from a valid signcryption on m from ID_S^* to ID_R^*, we are able to generate another valid signcryption on m from ID_S^* to ID_R^*. This is similar to the attack of strong unforgeability in the signature scheme. As the definition of unforgeability in [8] does not prevent this scenario, our attack becomes a valid one.

7 Security of Direct Combination of IBE and IBS

Now, all the ID based signcryption schemes proposed in the standard model are not provably secure. This has motivated us to analyse the security of getting a provably secure scheme by the direct combination of an ID based signature scheme and an ID based encryption scheme both in the standard model.

The design of a strongly unforgeable IND-CCA2 secure signcryption scheme can be attempted by combining a strongly unforgeable ID based signature scheme and an IND-CCA2 secure ID based encryption scheme based on three approaches.

- Sign then Encrypt
- Encrypt then Sign
- Sign and Encrypt (done in parallel)

Insider security is an important property of signcryption schemes, which ensures that a scheme offers confidentiality even if the private key of the sender is compromised. Similarly the unforgeability property is preserved even if the private key of the receiver is compromised. This is the strongest notion of security for signcryption primitive.

Let $Sign_{sk_A}(m, ID_A) \rightarrow \langle \sigma, m \rangle$ and $Verify_{ID_A}(\sigma, m) \rightarrow Valid/Invalid$ form a strongly unforgeable signature scheme and let $Encrypt_{ID_B}(m) \rightarrow C$ and $Decrypt_{sk_B}(C, ID_B) \rightarrow m/\perp$ form an IND-CCA2 secure encryption scheme, where ID_A and ID_B are the identities of sender and receiver respectively.

7.1 Encrypt then sign approach

Signcrypt$(m, ID_A, ID_B) = (Encrypt_{ID_B}(m||ID_A) \rightarrow C,\ Sign_{sk_A}(C||ID_B, ID_A) \rightarrow \langle \sigma, C \rangle)$.
The tuple $\Delta = \langle \sigma, C \rangle$ is the output of the resulting Signcrypt algorithm.

Unsigncrypt$(\Delta, ID_A, ID_B) = (Verify_{ID_A}(\sigma, C, ID_B) \rightarrow Valid/Invalid,$
if $Valid$ perform $(Decrypt_{sk_B}(C, ID_B) \rightarrow m/\perp))$.
The $m/\perp$ obtained is the output of the Unsigncrypt algorithm.

Security

During the proof of the IND-CCA2 property of the above signcryption scheme, the adversary $\mathcal{A}$ sends $\langle m_0, m_1 \rangle$ to the challenger $\mathcal{C}$ after phase 1 of training.
Then, $\mathcal{C}$ randomly chooses $\beta \in_R \{0, 1\}$ and performs Signcrypt$(m^*_\beta, ID_A, ID_B) = \Delta^*$ and sends Δ^* to $\mathcal{A}$. Note that $\mathcal{A}$ has access to the secret key sk_A of the sender for the scheme to satisfy insider security. Now, the adversary finds whether β is 0 or 1 as follows.

- $\mathcal{A}$ generates $\Delta' = \langle C^*, \sigma' \rangle$, where $\sigma' = Sign_{sk_A}(C^*||ID_B, ID_A)$. Note that $\sigma' \neq \sigma^*$ with high probability since Sign is a probabilistic algorithm.
- Now, Δ is a valid signcryption and can be queried to the Unsigncrypt oracle. This query is legal since $\Delta' \neq \Delta$.
- The unsigncryption query returns m^*_β to the adversary.

Thus, the adversary directly gets message that is signcrypted in the challenge ciphertext without having any knowledge about the secret key of the receiver, making the signcryption scheme IND-CCA2 insecure.

7.2 Sign and Encrypt approach

In this approach the Sign and Encrypt algorithms are run simultaneously to produce the signcryption of the message. So, no parameters are shared between these two algorithms.

Signcrypt$(m, ID_A, ID_B) = (Sign_{sk_A}(m||ID_B, ID_A) \rightarrow \sigma,\ Encrypt_{ID_B}(m||ID_A) \rightarrow C)$.
The tuple $\Delta = \langle \sigma, C \rangle$ is the output of the resulting Signcrypt algorithm.

Unsigncrypt$(\Delta, ID_A, ID_B) = (Decrypt_{sk_B}(C, ID_B) \rightarrow m/\perp$, and if m is returned, $Verify_{ID_A}(m, ID_B) \rightarrow Valid/Invalid)$.
The m obtained is the output of the Unsigncrypt algorithm, if $Valid$ is returned.

Security

In the proof of the IND-CCA2 property of the above signcryption scheme, after phase 1 of training, the adversary is given the challenge signcryption Δ^*, where Δ^* is the signcryption of m_β^*. Here, the adversary can always differentiate between m_0^* and m_1^* as follows.

- The adversary takes the component σ^*.
- It then performs $Verify(\sigma^*, m_0^*, ID_A, ID_B)$.
- If the above step returns $Valid$, then $\beta = 0$, otherwise $\beta = 1$.

This makes the signcryption scheme IND-CCA2 insecure.

In the unforgeability game, the adversary produces a forgery as follows.

- The adversary queries $\langle m, ID_A, ID_B \rangle$ to the Signcrypt oracle and gets $\Delta = \langle \sigma, C \rangle$.
- Now, it runs $Encrypt_{ID_B}(m)$ again, where m is a message that has already been queried to the Signcrypt oracle.
- Being a randomized algorithm, $Encrypt_{ID_B}(m)$ produces a different encryption C' on the same message and intended for the same receiver ID_B.
- This C' when combined with σ from the output of the Signcrypt query, gives another valid signcryption $\Delta' = \langle \sigma, C' \rangle$ on the message m.

This Δ' is a valid forgery since $\Delta' \neq \Delta$, making the signcryption scheme SUF-CMA insecure.

Thus, the resulting signcryption scheme is not outsider secure, since the secret key of the sender(receiver) is not involved in the IND-CCA2(SUF-CMA) game.

7.3 Sign then encrypt approach

Signcrypt$(m, ID_A, ID_B) = (Sign_{sk_A}(m||ID_B, ID_A) \rightarrow \sigma$, then $Encrypt_{ID_B}(\sigma||m||ID_A) \rightarrow C)$.
The tuple $\Delta = \langle C \rangle$ is the output of the resulting Signcrypt algorithm.

Unsigncrypt$(\Delta, ID_A, ID_B) = (Decrypt_{sk_B}(C, ID_B) \rightarrow \langle \sigma||m \rangle/\perp$, if $\perp$ is not returned, $Verify_{ID_A}(\sigma, m, ID_B) \rightarrow Valid/Invalid)$.
The m obtained is the output of the Unsigncrypt algorithm, if $Valid$ is returned.

Security

The above signcryption scheme is strongly unforgeable with insider security, because even if the secret key of the receiver sk_B is compromised, $\mathcal{A}$ cannot generate another valid signcryption of m due to the strong unforgeability property of the underlying signature scheme $Sign_{sk_A}(m||ID_B, ID_A)$.

Also, the confidentiality of the resulting signcryption scheme Signcrypt$(m, ID_A, ID_B) \rightarrow \Delta$ can be reduced directly to the confidentiality of the underlying encryption scheme, since the output $\Delta = \langle C \rangle$ is just the output of $Encrypt_{ID_B}(\sigma||m||ID_A)$. Since the encryption scheme used is IND-CCA2 secure, the signcryption scheme is also IND-CCA2 secure.

In Table 2, we present the level of security of the signcryption schemes that can be achieved by the three methods mentioned above.

Table 2. Security of signcryption schemes got by the direct combination of IBE and IBS

Approach	**Confidentiality** (IND-CCA2)	**Unforgeability** (SUF-CMA)
Sign then Encrypt	Yes	Yes
Encrypt then Sign	No	Yes
Sign and Encrypt	No	No

7.4 Scheme obtained by Direct Combination

From Table 2, we can find that a signcryption scheme obtained by a direct combination of a signature and an encryption scheme is SUF-CMA and IND-CCA2 secure only when a strongly unforgeable signature scheme and an IND-CCA2 secure encryption scheme are combined by Sign then Encrypt approach. The most efficient ID based signature scheme without random oracles is the one proposed by Paterson et al. [16]. But it is only EUF-CMA secure. To make it strongly unforgeable, we apply the transformation suggested by Boneh et al. [2]. And we take the IND-CCA2 secure ID based encryption scheme in the standard model proposed by Kiltz-Vahlis [7]. Using these as basic building blocks, let us conceptually formulate a scheme which we refer as scheme π. The scheme π is obtained by combining these schemes in a direct way as follows.

Signcrypt

Setup

Choose groups $\mathbb{G}$ and $\mathbb{G}_T$ of prime order p. Let g be the generator of group $\mathbb{G}$. And the bilinear pairing $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is admissible. Choose $g_2, g_u, g_\alpha \in_R \mathbb{G}$ and compute $z = \hat{e}(g, g_\alpha)$. Also pick $\alpha \in_R \mathbb{Z}_p^*$. Compute $g_1 = g^\alpha$. Then pick elements $u', m' \in_R \mathbb{G}$ and vectors $\mathbb{U} = (u_i), \mathbb{M} = (m_i)$ of length n_u and n_m respectively. The elements of these vectors are randomly picked from $\mathbb{G}$. There are two cryptographic hash functions $H_1 : \{0,1\}^{n_m+n_u+2|p|} \rightarrow \mathbb{Z}_p^*$ and $H_2 : \mathbb{G} \rightarrow \{0,1\}^l$, where l is large enough that the hash functions are collision resistant. Let $TCR : \mathbb{G} \rightarrow \mathbb{Z}_p^*$ be a target collision-resistant hash function and $SE = (E, D)$ be a symmetric encryption scheme with key-space $\kappa = \mathbb{G}_T$.

Extract

For an identity u represented by a string of bits of length n_u, define $\Omega_u \subseteq \{1, ..., n_u\}$ as the set of indices i for which $u[i] = 1$. The secret key of a user is

$$\langle (d_{u1}, d_{u2}), (d_{u3}, d_{u4}, d_{u5}) \rangle = \langle (g_2^\alpha (u' \prod_{i \in \Omega_u} u_i)^{r_u}, g^{r_u}), (g_\alpha (u' \prod_{i \in \Omega_u} u_i)^s, g^{-s}, g_u^s) \rangle$$

where $s, r_u \leftarrow_R \mathbb{Z}_p^*$. (d_{u1}, d_{u2}) form the signing key and (d_{u3}, d_{u4}, d_{u5}) form the decryption key.

Sign

The signing key of the sender ID_A is

$$\langle d_{u1}, d_{u2} \rangle = \langle g_2^\alpha (u' \prod_{i \in \Omega_u} u_i)^{r_u}, g^{r_u} \rangle$$

$\sigma_1 = d_{A2}$

$\sigma_2 = g^{r_m}$, where $r_m \in_R \mathbb{Z}_p^*$

$h_t = H_1(M||ID_B||\sigma_1||\sigma_2)$, where M is the message for which signcryption is produced and ID_B is the identity of the receiver of the ciphertext

$\beta = H_2(g^{h_t} h^{r_s})$, where $r_s \in \mathbb{Z}_p^*$

$\sigma_3 = d_{A1}(m' \prod_{j \in \overline{\beta}} m_j)^{r_m}$, where $\overline{\beta} \subseteq \{1, 2, ..., l\}$ be the set of indices i such that $\beta[i] = 1$

The signature is $\sigma' = \langle \sigma_1, \sigma_2, \sigma_3, r_s \rangle$.

Encrypt

This algorithm receives $\langle \sigma', M, ID_A, ID_B \rangle$ from the Sign algorithm.

$\sigma_4 = g^r$, where $r \in_R \mathbb{Z}_p^*$

$t = TCR(\sigma_4); \sigma_5 = ((u' \prod_{i \in \Omega_B} u_i) g_u^t)^r$, where $\Omega_B \subseteq \{1, ..., n_B\}$ is the set of indices i with $ID_B[i] = 1$

$K = z^r; \sigma_6 = E_K(M||\sigma'||ID_A)$

The signcryption produced is $\sigma = \langle \sigma_4, \sigma_5, \sigma_6 \rangle$.

Unsigncrypt

On receiving $\sigma = \langle \sigma_4, \sigma_5, \sigma_6 \rangle$, the Unsigncrypt algorithm performs the following two algorithms.

Decrypt

The decryption key for the receiver ID_B is

$$\langle d_{u3}, d_{u4}, d_{u5} \rangle = \langle g_\alpha (u' \prod_{i \in \Omega_u} u_i)^s, g^{-s}, g_u^s \rangle$$

$t = TCR(\sigma_4)$

$K = \hat{e}(\sigma_4, d_{B3}.d_{B5}^t)\, \hat{e}(\sigma_5, d_{B4})$

$(M||\sigma'||ID_A) = D_K(\sigma_6)$

Return M if σ' satisfies the following $Verify$ algorithm.

Verify

Parsing $\sigma' = \langle \sigma_1, \sigma_2, \sigma_3, r_s \rangle$, calculate $h_t = H(M||ID_A||\sigma_1||\sigma_2)$ and then $\beta = g^{h_t} h^{r_s}$. Then the following check is performed.

$$\hat{e}(\sigma_3, g) \stackrel{?}{=} \hat{e}(g_1, g_2)\, \hat{e}(u' \prod_{i \in \Omega_A} u_i, \sigma_1)\, \hat{e}(m' \prod_{j \in \overline{\beta}} m_j, \sigma_2)$$

where $\Omega_A \subseteq \{1, ..., n_A\}$ is the set of indices i with $ID_A[i] = 1$.

Efficiency

The signcryption scheme proposed in the previous section π is strongly unforgeable and IND-CCA2 secure. π performs computations as described in Table 3.

Table 3. Computational Complexity of π

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

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)

We may refer π as a scheme obtained by naive or straightforward combination of an encryption scheme and a signature scheme because the secret key of π is nothing but component-wise concatenation of the secret key of the schemes in [16] and [7] and the Sign/Encrypt and Decrypt/Verify algorithms are independent and sequential.

8 Conclusion

Thus, all the ID based signcryption schemes proposed in the standard model are not provably secure. And, by analyzing the various types of direct combination, we conclude that a strongly unforgeable, IND-CCA2 secure ID based signcryption scheme in the standard model can be obtained through direct combination of an IBE and an IBS only by the Sign then Encrypt approach. But the scheme obtained through this approach is not efficient. Other than the approaches used for direct combination, obtaining an efficient provably secure ID based signcryption scheme in the standard model still remains an open problem.

References

1. 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.
2. Dan Boneh, Emily Shen, and Brent Waters. Strongly unforgeable signatures based on computational diffie-hellman. In *Public Key Cryptography*, pages 229–240, 2006.
3. Xavier Boyen. Multipurpose identity-based signcryption (a swiss army knife for identity-based cryptography). In *CRYPTO*, pages 383–399, 2003.
4. Liqun Chen and John Malone-Lee. Improved identity-based signcryption. In *Public Key Cryptography*, pages 362–379, 2005.
5. 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.
6. 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.
7. Eike Kiltz and Yevgeniy Vahlis. Cca2 secure ibe: Standard model efficiency through authenticated symmetric encryption. In *CT-RSA*, pages 221–238, 2008.
8. Woomyo Lee, Jae Woo Seo, and Pil Joong Lee. Identity-based signcryption from identity-based cryptography. In *Proceedings of the 12th international workshop on Information security applications*, WISA'11, pages 70–83. Springer-Verlag, 2011.
9. 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.
10. Fagen Li, Yongjian Liao, Zhiguang Qin, and Tsuyoshi Takagi. Further improvement of an identity-based signcryption scheme in the standard model. *Comput. Electr. Eng.*, 38(2):413–421, March 2012.
11. Fagen Li, Fahad Bin Muhaya, Mingwu Zhang, and Tsuyoshi Takagi. Efficient identity-based signcryption in the standard model. In *ProvSec*, pages 120–137, 2011.
12. 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.
13. Benoît Libert and Jean-Jacques Quisquater. Efficient signcryption with key privacy from gap diffie-hellman groups. In *Public Key Cryptography*, pages 187–200, 2004.

14. John Malone-Lee. Identity-based signcryption. Cryptology ePrint Archive, Report 2002/098, 2002. http://eprint.iacr.org/.
15. Sumit Kumar Pandey and Rana Barua. Construction of identity based signcryption schemes. In *Proceedings of the 11th international workshop on Information security applications*, WISA'10, pages 1–14. Springer-Verlag, 2011.
16. Kenneth G. Paterson and Jacob C. N. Schuldt. Efficient identity-based signatures secure in the standard model. In *ACISP*, pages 207–222, 2006.
17. Adi Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO*, pages 47–53, 1984.
18. 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.
19. 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.
20. Brent Waters. Efficient identity-based encryption without random oracles. In *EUROCRYPT*, pages 114–127, 2005.
21. 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.
22. Yong Yu, Bo Yang, Ying Sun, and Shenglin Zhu. Identity based signcryption scheme without random oracles. *Computer Standards* & *Interfaces*, 31(1):56–62, 2009.
23. Bo Zhang. Cryptanalysis of an identity based signcryption scheme without random oracles. *Journal of Computational Information Systems*, 6(6):1923–1931, 2010.
24. 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.
25. Yuliang Zheng. Digital signcryption or how to achieve cost(signature & encryption) << cost(signature) + cost(encryption). In *Proceedings of the 17th Annual International Cryptology Conference on Advances in Cryptology*, pages 165–179, London, UK, 1997. Springer-Verlag.