# Cryptanalysis of Li et al.'s Identity-Based Threshold Signcryption Scheme

S. Sharmila Deva Selvi[1], S. Sree Vivek[⋆,1], Neha Jain[⋆⋆,2], and Pandu Rangan Chandrasekaran[⋆,1]

[1] {sharmila,svivek}@cse.iitm.ernet.in, prangan@iitm.ac.in
Indian Institute of Technology Madras
Theoretical Computer Science Laboratory
Department of Computer Science and Engineering
Chennai, India

[2] neo_gudiya@yahoo.co.in
VIT University
School of Computing Sciences
Vellore, India

**Abstract.** Signcryption is a cryptographic primitive that aims at providing confidentiality and authentication simultaneously. Recently in May 2008, a scheme for identity based threshold signcryption was proposed by Fagen Li and Yong Yu. They have proved the confidentiality of their scheme and have also claimed the unforgeability without providing satisfactory proof. In this paper, we show that in their signcryption scheme the secret key of the sender is exposed(total break) to the clerk during sincryption and hence insecure in the presence of malicious clerks. Further, we propose a corrected version of the scheme and formally prove its security under the existing security model for signcryption.

## 1 Introduction

Encryption and signature are the basic cryptographic tools offered by public key cryptography for achieving confidentiality and authentication. Encryption is the process of transforming information (plain text) using an algorithm (cipher) to make it unreadable for those who do not have the secret key needed to decipher (decrypt) the information. Encryption can be done using the same secret key at the sender's and the receiver's side (private key encryption) or using different keys at both sides (public key encryption). Encryption guarantees confidentiality and privacy because the encrypted text can be read only by the intended recipient. Signature

⋆ Work supported by Project No. CSE/05-06/076/DITX/CPAN on Protocols for Secure Communication and Computation sponsored by Department of Information Technology, Government of India.

⋆⋆ Work supported by IITM Internship during May - July 2008.

is the cryptographic technique where, before sending a message the sender $A$ signs it with his private key. This ensures authentication because the recipient $B$ knows that the message has been sent by $A$, and on the other hand, $A$ cannot deny having sent the message to $B$.

The concept of signcryption originates from the various applications where both confidentiality and authentication are mandatory requirements. Signcryption, introduced by Zheng in 1997[2], is a cryptographic primitive that offers confidentiality and authentication simultaneously similar to the sign-then-encrypt technique, but with lesser computational complexity and lower communication cost. After Zheng's work a number of signcryption schemes were proposed [4][13][20][14][11][10][22]. The security notion for signcryption was first formally defined in 2002 by Baek et al. in [23]. This was similar to the notion of semantic security against adaptive chosen ciphertext attack and existential unforgeability against adaptive chosen message attack. The concept of identity-based cryptosystem was introduced by Shamir [17] in 1984. The distinguishing characteristic of identity-based cryptography is the ability to use any string as a public key. In particular, this string may be the email address, telephone number, or any publicly available parameter of an individual that is unique to that individual. An identity-based cryptosystem removes the need for senders to look up the receiver's public key before sending out an encrypted message. It provides a more convenient alternative to conventional public key infrastructure.

Group oriented cryptography was introduced by Desmedt in 1987 [9]. Elaborating on this concept, Desmedt and Frankel [7] proposed a (t, n) threshold signature scheme based on the RSA system [16]. In such a (t,n) threshold signature scheme, any $t$ out of $n$ signers in the group can collaboratively sign messages on behalf of the group by sharing the signing capability. This can be visualised in the situation where a company has $n$ directors and if atleast $t$ of them agree on a decision, then only that decision is finalised. An identity-based threshold signcryption incorporates the concept of threshold cryptosystem and an identity-based system along with the basic signcryption concept.

In 2004, Duan et al. [12] proposed an identity-based threshold signcryption scheme by combining the concepts of identity-based threshold signature and signcryption together. However, in Duan et al.'s scheme [12], the master-key of the PKG is distributed to a number of other

PKGs, which creates a bottleneck on the PKGs. In 2005, Peng and Li [15] proposed an identity-based threshold signcryption scheme based on Libert and Quisquater's identity-based signcryption scheme [19]. However, Peng and Li' scheme [15] does not provide the forward security i.e., anyone who obtains the sender's private key can recover the original message of a signcrypted text. In addition, both Duan et al.'s scheme [12] and Peng and Li's scheme [15] do not consider the formal security models and security proofs. Ma et al. [21] also proposed a threshold signcryption scheme using the bilinear pairings. However, Ma et al.'s scheme [21] is not identity-based. In May 2008, another scheme was proposed by Fagen Li and Yong Yu[1]. Although the scheme is more efficient (as it requires one pairing less than the previous schemes) but it is not secure against the insider attack.

**Our contribution:** In this paper, we show that the threshold signcryption scheme of Fagen Li and Yong Yu[1] is vulnerable to the attack by the clerk(the semi trusted authorithy who combines the signatures of all the $t$ players) by demonstrating an attack which shows that if the adversary corrupts the clerk then it is able to get the secret key of the system and hence a total break of the system is possible. Further, we propose a corrected version of their scheme and prove correctness and security (confidentiality and unforgeability) under the existing security model for signcryption.

The rest of this paper proceeds as follows. In Section 2, we review the preliminaries like bilinear pairings and related computational problems, the general framework of identity-based threshold signcryption schemes, and the security models for such schemes. Next, in Section 3, we review Fagen Li's threshold signcryption scheme [1]. We present the attack on this scheme in Section 4. In Section 5, we lay out the details of our fix to the original scheme. In Section 6, we present the analysis of the corrected scheme which includes proofs of correctness, unforgeability and confidentiality of the scheme. Section 7 concludes the discussion.

## 2 Preliminaries

### 2.1 Bilinear Pairing

Let $\mathbb{G}_1$ be an additive cyclic group generated by P, with prime order q, and $\mathbb{G}_2$ be a multiplicative cyclic group of the same order $q$. A bilinear pairing is a map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_2$ with the following properties.

– **Bilinearity.** For all P,Q,R $\in \mathbb{G}_1$,
$\hat{e}(P+Q,R) = \hat{e}(P,R)\hat{e}(Q,R)$
$\hat{e}(P,Q+R) = \hat{e}(P,Q)\hat{e}(P,R)$
$\hat{e}(aP,bQ) = \hat{e}(P,Q)^{ab}$
– **Non-Degeneracy**. There exist $P,Q \in \mathbb{G}_1$ such that $\hat{e}(P,Q) \neq I_{\mathbb{G}_2}$, where $I_{\mathbb{G}_2}$ is the identity element of $\mathbb{G}_2$.
– **Computability.** There exists an efficient algorithm to compute $\hat{e}(P,Q)$ for all $P,Q \in \mathbb{G}_1$.

### 2.2 Computational Assumptions

In this section, we review the computational assumptions related to bilinear maps that are relevant to the protocol we discuss.

**Bilinear Diffie Hellman Problem (BDHP)**
Given $(P,\ aP,\ bP,\ cP) \in \mathbb{G}_1^4$ for unknown $a,\ b,\ c \in \mathbb{Z}_q^*$ , the BDH problem in $\mathbb{G}_1$ is to compute $\hat{e}(P,P)^{abc}$.
The advantage of any probabilistic polynomial time algorithm $\mathcal{A}$ in solving the BDH problem in $\mathbb{G}_1$ is defined as

$$Adv_{\mathcal{A}}^{BDH} = \Pr\ [\mathcal{A}(P,aP,bP,cP) = \hat{e}(P,P)^{abc} \mid a,\ b,\ c \in \mathbb{Z}_q*]$$

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

**Decisional Bilinear Diffie-Hellman Problem (DBDHP)**
Given $(P,\ aP,\ bP,\ cP,\ \alpha) \in \mathbb{G}_1^4 \times \mathbb{G}_2$ for unknown $a,\ b,\ c \in \mathbb{Z}_q*$, the DBDH problem in $\mathbb{G}_1$ is to decide if $\alpha = \hat{e}(P,P)^{abc}$.
The advantage of any probabilistic polynomial time algorithm $\mathcal{A}$ in solving the DBDH problem in $\mathbb{G}_1$ is defined as

$$Adv_{\mathcal{A}}^{DBDH} = \mid \Pr[\mathcal{A}(P,\ aP,\ bP,\ cP,\ \hat{e}(P,P)^{abc})=1]\text{-}\Pr[\mathcal{A}P,\ aP,\ bP,\ cP,\ \alpha) = 1]|$$

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

**Computation Diffie-Hellman Problem (CDHP)**
Given $(P,\ aP,\ bP\ ) \in \mathrm{G}_1^3$ for unknown $a,\ b \in \mathbb{Z}_q^*$ , the CDH problem in $\mathbb{G}_1$ is to compute $abP$.
The advantage of any probabilistic polynomial time algorithm $\mathcal{A}$ in solving the CDH problem in $\mathbb{G}_1$ is defined as

$$Adv_{\mathcal{A}}^{CDH} = \Pr\ [\ \mathcal{A}(P,\, aP,\, bP\ ) = abP \mid a,\, b \in \mathbb{Z}_q^*]$$

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

### 2.3 Identity Based Threshold Signcryption

A generic identity-based threshold signcryption scheme with total $n$ players and $t$ threshold limit consists of the following five algorithms.

- ***Setup***: *Given a security parameter $k$, the private key generator (PKG) generates the system's public parameters $params$. Among the parameters produced by Setup is a key $P_{pub}$ that is made public. There is also corresponding master key $s$ that is kept secret.*
- ***Extract***: *Given an identity $ID$, the PKG computes the corresponding private key $S_{ID}$ and transmits it to its owner in a secure way.*
- ***Keydis***: *Given a private key $S_{ID}$ associated with an identity $ID$, the number of signcryption members $n$ and a threshold parameter $t$, this algorithm generates $n$ shares of $S_{ID}$ and provides each one to the signcryption members $M_1, M_2 \ldots, M_n$. It also generates a set of verification keys that can be used to check the validity of each shared private key. We denote the shared private keys and the matching verification keys by $\{S_i\}_{i=1,\ldots n}$ and $\{y_i\}_{i=1,\ldots,n}$, respectively. Note that each $(S_i, y_i)$ is sent to $M_i$, then $M_i$ publishes $y_i$ but keeps $S_i$ secret.*
- ***Signcrypt:*** *Give a message $m$ , the private keys of $t$ members $\{S_i\}_{i=1,\ldots t}$ in a sender group $U_A$with identity $ID_A$, a receiver's identity $ID_B$, it outputs an identity-based $(t, n)$ threshold signcryption $\sigma$ on the message $m$.*
- ***Unsigncrypt:*** *Give a ciphertext $\sigma$, the private key of the receiver $S_{ID_B}$, the identity of the sender group $ID_A$, it outputs the plain text $m$ or the symbol $\perp$ if $\sigma$ is an invalid ciphertext between the group $U_A$ and the receiver. We make the consistency constraint that if $\sigma = Signcrypt(m, \{S_i\}_{i=1,\ldots.n}, ID_B)$, then $m = Unsigncrypt(\sigma, ID_A, S_{ID_B})$.*

### 2.4 Security Model for Identity-Based Threshold Signcryption (*IDTSC*)

The notion of semantic security of public key encryption was extended to identity-based signcryption scheme by Malone-Lee in [8]. This was later modified by Sherman et al. in [10] which incorporates indistinguishability against adaptive chosen ciphertext and identity attacks (IND-IDTSC-CCA2) and existential unforgeability against adaptive chosen message and identity attacks (EUF-IDTSC). We describe below the security models for confidentiality and unforgeability given in [11], this is the strongest security notion for this problem.

***Confidentiality :*** *A signcryption scheme is semantically secure against chosen ciphertext attack (IND-IDTSC-CCA2) if no probabilistic polynomial time adversary $\mathcal{A}$ has a non-negligible advantage in the following game.*

1. *The challenger $\mathcal{C}$ runs the Setup algorithm and sends the system public parameters to the adversary $\mathcal{A}$.*

2. *In the first phase, $\mathcal{A}$ makes polynomially bounded number of queries to the following oracles.*

***Extract Oracle :*** *$\mathcal{A}$ produces an identity $ID_i$ and queries for the secret key of user $i$. The Extract Oracle returns $S_i$ to $\mathcal{A}$.*

***Signcrypt Oracle :*** *$\mathcal{A}$ produces a message $m$, sender identity $ID_A$ and receiver identity $ID_B$. $\mathcal{C}$ computes the secret key$S_A$ from Extract$(ID_A)$ and returns to $\mathcal{A}$, the signcrypted ciphertext from Signcrypt $(m, \{S_i\}_{i=1,\ldots t}, ID_j)$.*

***Designcrypt Oracle :*** *$\mathcal{A}$ produces a sender identity $ID_A$, receiver identity $ID_B$ and a signcryption $\sigma$. The challenger $\mathcal{C}$ computes the secret key $S_B$ from Extract$(ID_B)$, returning the result of Designcrypt $(\sigma, ID_A, S_B)$ to $\mathcal{A}$. The result returned is $\perp$ if $\sigma$ is an invalid signcryption from $ID_A$ to $ID_B$.*

3. *$A$ produces two messages $m_0$ and $m_1$ of equal length from the message space $M$ and an arbitrary sender identity $ID_A$. The challenger $\mathcal{C}$ flips a coin, sampling a bit $b \leftarrow_R \{0,1\}$ and computes $\sigma^* =$ **Signcrypt**$(m_b, \{S_i\}_{i=1,\ldots,t}, ID_B)$. $\sigma^*$ is returned to $\mathcal{A}$ as challenge signcrypted ciphertext.*

4. *$\mathcal{A}$ is allowed to make polynomially bounded number of new queries as in Step 2 with the restrictions that it should not query the Designcryption Oracle for the designcryption of $\sigma^*$, the Signcryption Oracle for the signcryption of $m_0$ or $m_1$ under the sender identity $ID_A$ and the Extract Oracle for the secret keys of $ID_B$.*

5. *At the end of this game, $\mathcal{A}$ outputs a bit $b'$.$\mathcal{A}$ wins the game if $b' = b$.*

**Unforgeability:** *A signcryption scheme is existentially unforgeable under chosen message attack (EUF-IDTSC) if no probabilistic polynomial time adversary $\mathcal{A}$ has a non-negligible advantage in the following game.*

1. *The challenger $\mathcal{C}$ runs the Setup algorithm to generate the master public and private keys params and msk respectively. $\mathcal{C}$ gives system public parameters params to $\mathcal{A}$ and keeps the master private key msk secret from $\mathcal{A}$.*

2. *The adversary $\mathcal{A}$ makes polynomially bounded number of queries to the oracles as described in Step 2 of the confidentiality game.*

3. *A produces a signcrypted ciphertext $\sigma$ and wins the game if the private key of sender identity $ID_A$ was not queried in the previous step and $\perp$ is not returned by Designcrypt($\sigma, ID_A, S_B$) and $\sigma$ is not the output of a previous query to the Signcrypt Oracle with $ID_A$ as sender.*

## 3 Review of Fagen Li's Identity-Based Threshold Signcryption Scheme

In this section, we present the identity-based threshold signcryption scheme as proposed by Fagen Li and Yu . The scheme involves four roles: the PKG, a trusted dealer, a sender group $U_A = \{M_1, M_2, \ldots, M_n\}$with identity $ID_A$, and a receiver Bob with identity $ID_B$.

**Setup:** *Given a security parameter $k$, the PKG chooses groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of prime order $q$ (with $\mathbb{G}_1$ additive and $\mathbb{G}_2$ multiplicative), a generator $P$ of $\mathbb{G}_1$, a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$, a secure symmetric cipher $(E, D)$ and hash functions $H_1 : \{0,1\}^* \to \mathbb{G}_1$, $H_2 : \mathbb{G}_2 \to \{0,1\}^{n1}$ , $H3 : \{0,1\}^* \to Z_q^*$. The PKG chooses a master-key $s {\in}_R Z_q^*$ and computes $P_{pub} = sP$. The PKG publishes system parameters $(\mathbb{G}_1, \mathbb{G}_2, n_1, e, P, P_{pub}, E, D, H_1, H_2, H_3)$ 'and keeps the master-key $s$ secret.*

**Extract**: Given an identity $ID$, the PKG computes $Q_{ID} = H_1(ID)$ and the private key $S_{ID} = sQ_{ID}$. Then PKG sends the private key to its owner in a secure way.

**Keydis:** Suppose that a threshold $t$ and $n$ satisfy $1 \leq t \leq n < q$. To share the private key $S_{ID_A}$ among the group $U_A$, the trusted dealer performs the steps below.

1) Choose $F_1, F_2, ..., F_{t-1}$ uniformly at random from $\mathbb{G}_1^*$ and construct a polynomial $F(x) = S_{ID_A} + xF_1 + \ldots + x^{t-1}F_{t-1}$.

2) Compute $S_i = F(i)$ for $i = 0, \ldots, n.(S_0 = S_{ID_A})$. Send $S_i$ to member $M_i$ for $i = 1, \ldots, n$ secretly.

3) Broadcast $y_0 = \hat{e}(S_{ID_A}, P)$ and $y_j = \hat{e}(F_j, P)$ for $j = 1, ..., t-1$.

4) Each $M_i$ then checks whether his share $S_i$ is valid by computing $\hat{e}(S_i, P) = \mathit{\Pi}_{j=0}^{t-1} y_j^{i^j}$ . If $S_i$ is not valid, $M_i$ broadcasts an error and requests a valid one.

**Signcrypt:** Let $M_{1,...,} M_t$ are the $t$ members who want to cooperate to signcrypt a message $m$ on behalf of the group $U_A$.

1) Each $M_i$ chooses $x_i \in_R Z_q^*$.

-computes$R_{1i} = x_i P, R_{2i} = x_i P_{pub}$

-sends $(R_{1i}, R_{2i})$ to the clerk $C$.

2) The clerk $C$( one among the $t$ cooperating players) computes,

-$R_1 = \sum_{i=1}^{t} R_{1i}, R_2 = \sum_{i=1}^{t} R_{2i}$

-$\tau = \hat{e}(R_2, Q_{ID_B})$

-$k = H_2(\tau), c = E_k(m)$, and $h = H_3(m, R_1, k)$.

3) Then the clerk $C$ sends $h$ to $M_i \ fori = 1, ..., t$.

4) Each $M_i$ computes the partial signature $W_i = x_i P_{pub} + h\eta_i S_i$ and sends it to the clerk $C$, where $\eta_i = \prod_{j=1;j\neq i}^{t} -j(i-j)^{-1} modq$

5) Clerk $C$ verifies the correctness of partial signatures by checking if the following equation holds

$\hat{e}(P, W_i) = \hat{e}(R_{1i}, P_{pub})(\prod_{j=0}^{t-1} y_j^{i^j})^{h\eta_i}$

If all partial signatures are verified to be legal, the clerk $C$ computes $W = \sum_{i=1}^{t} W_i$otherwise rejects it and requests a valid one.

6) The final threshold signcryption is $\sigma = (c, R_1, W)$.

**Unsigncrypt:** When receiving $\sigma$, Bob follows the steps below.

1) Compute $\tau = \hat{e}(R_1, S_{ID_B})$ and $k = H_2(\tau)$.

2) Recover $m = D_k(c)$.

3) Compute $h = H_3(m, R_1, k)$ and accept $\sigma$ if and only if the following equation holds:

$\hat{e}(P, W) = \hat{e}(P_{pub}, R_1 + hQ_{ID_A})$

## 4 Attack on the scheme

The scheme described above [1] is insecure from the point of view of attack by the clerk. The clerk is the semi trusted body in the scheme. He combines all the partial signatures to generate the final signature for the message. If the clerk becomes corrupt, the secret key of the system is revealed and hence a total break of the system occurs. We describe how the attack proceeds in this section.

we know that,

$W = \sum_{i=1}^{t} W_i$

$W = \sum_{i=1}^{t}(x_i P_{pub} + h\eta_i S_i)$

$W = \sum_{i=1}^{t}(x_i P_{pub}) + \sum_{i=1}^{t}(h\eta_i S_i)$

$W = R_2 + hS_A$

The clerk has the value of $(R_2, W, h)$, hence the secret key of $A$, $S_A$is exposed as shown below:

$$\frac{W - R_2}{h} = S_A$$

## 5 The Improved Scheme

In this section, we propose an improved version of the Fagen Li's scheme, which we formally prove to be secure. The setup and key generation algorithms of our scheme are similar to that of Li's scheme. The modification has been made in the signcryption algorithm such that the system is secure against the clerk or any other insider. The details of the scheme are as follows:

The scheme involves four roles: the PKG, a trusted dealer, a sender group $U_A = \{M_{1,} M_2, \ldots, M_n\}$with identity $ID_A$, and a receiver Bob with identity $ID_B$.

**Setup:** Given a security parameter $k$, the PKG chooses groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of prime order $q$ (with $\mathbb{G}_1$ additive and $\mathbb{G}_2$ multiplicative), a generator $P$ of $\mathbb{G}_1$, a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$, a secure symmetric cipher $(E, D)$ and hash functions $H_1 : \{0,1\}^* \to \mathbb{G}_1$, $H_2 : \mathbb{G}_2 \to \{0,1\}^{n1}$ , $H3 : \{0,1\}^* \to Z_q^*$. The PKG chooses a master-key $s \in_R \mathrm{Z}_q^*$ and computes $P_{pub} = sP$. The PKG publishes system parameters $(\mathbb{G}_1, \mathbb{G}_2, n_1, e, P, P_{pub}, E, D, H_1, H_2 f, H_3)$ and keeps the master-key $s$ secret.

**Extract**: Given an identity $ID$, the PKG computes $Q_{ID} = H_1(ID)$ and the private key $S_{ID} = sQ_{ID}$. Then PKG sends the private key to its owner in a secure way.

**Keydis:** Suppose that a threshold $t$ and $n$ satisfy 1≤t≤n < q. To share the private key $S_{ID_A}$ among the group $U_A$, the trusted dealer performs the steps below.

1) Choose $F_1, F_2, ..., F_{t-1}$ uniformly at random from $\mathbb{G}_1^*$ and construct a polynomial $F(x) = S_{ID_A} + xF_1 + \ldots + x^{t-1}F_{t-1}$.

2) Compute $S_i = F(i)$ for $i = 0, \ldots, n.(S_0 = S_{ID_A})$. Send $S_i$ to member $M_i$ for $i = 1, \ldots, n$ secretly.

3) Broadcast $y_0 = \hat{e}(S_{ID_A}, P)$ and $y_j = \hat{e}(F_j, P)$ for $j = 1, ..., t-1$.

4) Each $M_i$ then checks whether his share $S_i$ is valid by computing $\hat{e}(S_i, P) = \mathit{\Pi}_{j=0}^{t-1} y_j^{ij}$. If $S_i$ is not valid, $M_i$ broadcasts an error and requests a valid one.

**Signcrypt:** Let $M_{1,...,} M_t$ are the $t$ members who want to cooperate to signcrypt a message $m$ on behalf of the group $U_A$.

1) Each $M_i$ chooses $x_i \in_R Z_q^*$.

-computes$R_{1i} = x_i P, R_{2i} = x_i P_{pub}, \tau_i = \hat{e}(R_{2i}, Q_{ID_B})$

-sends $(R_{1i}, \tau_i)$ to the clerk $C$.

2) The clerk $C$ (one among the $t$ cooperating players) computes,

-$R_1 = \sum_{i=1}^{t} R_{1i}$

-$\tau = \prod_{i=1}^{t} \tau_i$

-$k = H_2(\tau), c = E_k(m)$, and $h = H_3(m, R_1, k)$.

3) Then the clerk $C$ sends $h$ to $M_i$ $fori = 1, ..., t$.

4) Each $M_i$ computes the partial signature $W_i = x_i P_{pub} + h\eta_i S_i$ and sends it to the clerk $C$, where

$\eta_i = \prod_{j=1;j\neq i}^{t} -j(i-j)^{-1} mod q$

5) Clerk $C$ verifies the correctness of partial signatures by checking if the following equation holds

$\hat{e}(P, W_i) = \hat{e}(R_{1i}, P_{pub})(\prod_{j=0}^{t-1} y_j^{ij})^{h\eta_i}$

If all partial signatures are verified to be legal, the clerk $C$ computes $W = \sum_{i=1}^{t} W_i$ otherwise rejects it and requests a valid one.

6) The final threshold signcryption is $\sigma = (c, R_1, W)$.

**Unsigncrypt:** When receiving $\sigma$, Bob follows the steps below.

1) Compute $\tau = \hat{e}(R_1, S_{ID_B})$ and $k = H_2(\tau)$.

2) Recover $m = D_k(c)$.

3) Compute $h = H_3(m, R_1, k)$ and accept $\sigma$ if and only if the following equation holds:

$\hat{e}(P, W) = \hat{e}(P_{pub}, R_1 + hQ_{ID_A})$

## 6 Analysis of the scheme

### A. Correctness Proof:

The correctness can be easily verified by the following equations.

$\hat{e}(R_1, S_{ID_B}) = \hat{e}(\sum_{i=1}^{t} R_{1i}, S_{ID_B})$
$=\hat{e}(\sum_{i=1}^{t}(x_i P_{pub}), Q_{ID_B})$
$=\hat{e}(\sum_{i=1}^{t} R_{2i}, Q_{ID_B})$
$=\hat{e}(R_{21}, Q_{ID_B}).\hat{e}(R_{22}, Q_{ID_B}) \ldots \hat{e}(R_{2t}, Q_{ID_B})$.
$=\prod_{i=1}^{t} \tau_i$

and,

$\hat{e}(P, W) = \hat{e}(P, \sum_{i=1}^{t} W_i)$
$= \hat{e}(P, \sum_{i=1}^{t}(x_i P_{pub} + h\eta_i S_i))$
$= \hat{e}(P, \sum(x_i P_{pub}) + \sum_{i=1}^{t}(h\eta_i S_i))$
$= \hat{e}(P, \sum_{i=1}^{t}(x_i P_{pub}) + hS_{ID_A})$
$= \hat{e}(P, \sum_{i=1}^{t}(x_i P) + hQ_{ID_A}) = \hat{e}(P_{pub}, R_1 + hQ_{ID_A})$

### B. Security Analysis

### 1. Unforgeability Proof:

*Theorem* : Our identity based threshold signcryption scheme is secure against any EUF-IDTSC adversary $\mathcal{A}$ under the random oracle model if CDHP is hard in $G_1$.

The challenger $\mathcal{C}$ receives an instance $(P, aP, bP\ )$ of the $CDH\ problem$. His goal is to determine $abP$ . Suppose there exists an EUF-IDTSC adversary $\mathcal{A}$ for our proposed scheme. We show that $\mathcal{C}$ can use $\mathcal{A}$ to solve the CDH problem. $\mathcal{C}$ will set the random oracles $\mathcal{O}_{H1}$ , $\mathcal{O}_{H2}$ ,$\mathcal{O}_{H3}$ , $\mathcal{O}_{extract}$, $\mathcal{O}_{signcrypt}$, $\mathcal{O}_{unsigncrypt}$. The answers to the oracles $\mathcal{O}_{H1}$ , $\mathcal{O}_{H2}$ and $\mathcal{O}_{H3}$ are randomly selected, therefore, to maintain consistency, $\mathcal{C}$ will maintain three lists $L_1$, $L_2, L_3$. We assume that $\mathcal{A}$ will ask for $H_1(ID)$ before $ID$ is used in any key *extraction*, *signcryption*, and *unsigncryption* queries. First, the adversary $\mathcal{A}$ outputs the identity $ID_A$ of the sender whose signcryption he claims to be able to forge. Then, the challenger $\mathcal{C}$ gives $\mathcal{A}$ the system parameters *params*, consisting of $P, P_{pub} = aP$ . The descriptions of the oracles is as follows:

**Oracle** $\mathcal{O}_{H1}$ ($\mathrm{ID}_i$) **:** $\mathcal{C}$ checks if there exists a tuple $(ID_i, b_e)$ in $L_1$. If such a tuple exists, $\mathcal{C}$ answers with $b_e$. Otherwise, $\mathcal{C}$ does the following.

1. If $ID_i = ID_A$, answer by giving $bP$.

2. If $ID_i \neq ID_A$, choose a new $b \in \mathrm{Z}_q^*$. Add the tuple $(ID_i, b)$ to $L_1$ and return $b$.

**Oracle** $\mathcal{O}_{H2}(\tau_e)$: $\mathcal{C}$ checks if there exists a tuple $(\tau_e, k_e)$ in $L_2$. If such a tuple exists, $\mathcal{C}$ answers with $k_e$. Otherwise, $\mathcal{C}$ does the following.

1. Choose a new $k \in_R Z_q^*$.such that no tuple $(\textbf{.}, k)$ exists in $L_2$.

2. Add the tuple $(\tau_e, k)$ to $L_2$ and return $k$.

**Oracle** $\mathcal{O}_{H3}(m_e, \mathrm{R}_{1e}, \mathrm{k}_e)$: $\mathcal{C}$ checks if there exists a tuple $(m_e, R_{1e}, k_e, h_e)$ in $L_3$. If such a tuple exists, $\mathcal{C}$ answers with $h_e$. Else $\mathcal{C}$ performs:

1. Choose a new $h \in_R Z_q^*$.

2. Add the tuple $(m_e, R_{1e}, k_e, h)$ to $L_3$ and return $h$.

**Extract** ( $ID_e$)**:**

1. if $ID_e = ID_A$ return $\perp$.

2. If$ID_e \neq ID_A$, recover the tuple $(ID_e, b_e)$ from $L_1$ and return $(b_e P_{pub})$ as the secret key.

**Signcrypt** $(m, ID_s, ID_B)$:

1. if $ID_s \neq ID_A$, $\mathcal{C}$ computes the private key $\mathrm{S}_{ID_S}$ corresponding to $ID_S$ by running the key extraction query algorithm. Then $\mathcal{C}$ runs *Keydis*

to output $n$ shared private keys $\{S_i\}_{i=1,...,t}$. Finally, $\mathcal{C}$ answers the query by a call to $Signcrypt(m, \{S_i\}_{i=1,...,t}, Q_{ID_B})$.

2. else, $\mathcal{C}$ chooses $x, h \in_R Z_q^*$. and computes $R_1 = xP - hQ_{ID_A}, W = xP_{pub}$, and $\tau = \hat{e}(R_1, S_{ID_B})$ ($\mathcal{C}$ could obtain $S_{ID_B}$ from the key extraction algorithm because $ID_B \neq ID_A$). $\mathcal{C}$ runs the $H_2$simulation algorithm to find $k = H_2(\tau)$ and computes $c = E_k(m)$. $\mathcal{C}$ then checks if $L_3$ already contains a tuple $(m, R_1, k, h^{'})$ with $h' \neq h$. In this case, $\mathcal{C}$ repeats the process with another random pair $(x, h)$ until finding a tuple $(m, R_1, k, h)$ whose first three elements do not appear in a tuple of the list $L_3$. Such a tuple, $(m, R_1, k, h)$ is then entered in $L_3$. $(C, R_1, W)$ is hence a valid signcryption according to the oracle.

**Unsigncryption queries:**

For a unsigncryption query on a ciphertext $\sigma' = (C', R_1', W')$ between a sender group with identity $ID_A$ and a receiver with identity $ID_B$. We have the following two cases to consider.

1. If $ID_A = ID_B$ .$\mathcal{C}$ always answers $\mathcal{A}$ that $\sigma'$ is invalid.

2. If $ID_B \neq ID_A$ . $\mathcal{C}$ computes $\tau' = \hat{e}(R_1', S_{ID_B})$. $\mathcal{C}$ then runs the $H_2$simulation algorithm to obtain $k' = H_2(\tau')$ and computes $m' = D_{k'}(c)$. Finally, $\mathcal{C}$ runs the $H_3$ simulation algorithm to obtain $h' = H_3(m', R_1', k')$ and checks if $\hat{e}(P, W') = \hat{e}(P_{pub}, R_1' + h'Q_{ID_A})$ holds. If the above equation does not hold, $\mathcal{C}$ rejects the ciphertext. Otherwise $\mathcal{C}$ returns $m$'.

Eventually $\mathcal{A}$ outputs a forged signcryption $\sigma' = (C, R_1, W)$ on some message$m'$ from the sender $ID_A$ to receiver $ID_B$. Challenger $\mathcal{C}$ designcrypts the ciphertext $\sigma'$ with identity $ID_B$ to get the 'signature' $W$ of $ID_A$, if $\sigma'$ is a valid signcrypted ciphertext from $ID_A$to $ID_B$on message $m'$. Now, $\mathcal{C}$ applies the oracle replay technique to produce two valid signcrypted ciphertexts $\sigma_1 = (C, R_1, W_1)$, and $\sigma_2 = (C, R_1, W_2)$for the same message $m$. $\mathcal{C}$ designcrypts 1 and 2 to obtain signatures $W_1 = (R_2 + h_1 S_A)$ and $W_2 = (R_2 + h_2 S_A)$. Now we can apply standard arguments for the outputs of the forking lemma since both $W_1$ and $W_2$ are valid signatures for the same message $m$ and same random tape of the adversary. Finally, $\mathcal{C}$ obtains the solution to the CDH instance as

$$W_1 = (R_2 + h_1 S_A)$$
$$W_2 = (R_2 + h_2 S_A)$$

---

$$W_1 - W_2 = (h_1 - h_2) S_A$$

hence, $(W_1 - W_2)(h_1 - h_2)^{-1} = aQ_A = abP$

So, we can see that the challenger $\mathcal{C}$ has the same advantage in solving the $CDH$ problem as the adversary $\mathcal{A}$ has in forging a valid signcrypted ciphertext. So, if there exists an adversary who can forge a valid signcrypted ciphertext with non-negligible advantage, that means there exists an algorithm to solve the CDH problem with non-negligible advantage. Since this is not possible, no adversary can forge a valid signcrypted ciphertext with non-negligible advantage. Hence, the scheme is secure against any EUF-IDTSC attack.

**2. Confidentiality Proof:**
*Theorem* :Our identity based threshold signcryption scheme is secure against any IND-IDTSC-CCA2 adversary $\mathcal{A}$ under the random oracle model if $DBDHP$ is hard in $\mathbb{G}_1$.

The challenger $\mathcal{C}$ receives an instance $(P, aP, bP, cP, h)$ of the DBDH problem. His goal is to decide whether $h = \hat{e}(P, P)^{abc}$ or not. Suppose there exists an IND-IDTSC-CCA2 adversary $\mathcal{A}$ for the proposed scheme. We show that $\mathcal{C}$ can use $\mathcal{A}$ to solve the DBDH problem. $\mathcal{C}$ will set the random oracles $\mathcal{O}_{H1}$ , $\mathcal{O}_{H2}$ ,$\mathcal{O}_{H3}$ , $\mathcal{O}_{extract}$, $\mathcal{O}_{signcrypt}$, $\mathcal{O}_{unsigncrypt}$. The answers to the oracles $\mathcal{O}_{H1}$ , $\mathcal{O}_{H2}$ and $\mathcal{O}_{H3}$ are randomly selected, therefore, to maintain consistency, $\mathcal{C}$ will maintain three lists$L_1$, $L_2, L_3$. We assume that $\mathcal{A}$ will ask for $H_1(ID)$ before $ID$ is used in any *extraction*, *signcryption*, and *unsigncryption* queries First, the adversary $\mathcal{A}$ outputs the identity $ID_A$ of the sender whose signcryption he claims to be able to forge. Then, the challenger $\mathcal{C}$ gives $\mathcal{A}$ the system parameters *params*, consisting of $P, P_{pub} = aP$ . The descriptions of the oracles is as follows:

**Oracle** $\mathcal{O}_{H1}$ (ID$_i$) **:** $\mathcal{C}$ checks if there exists a tuple $(ID_i, b_e)$ in $L_1$. If such a tuple exists, $\mathcal{C}$ answers with $b_e$. Otherwise, $\mathcal{C}$ does the following.

1. If $ID_i = ID_A$, answer by giving $bP$.

2. If $ID_i \neq ID_A$, choose a new $b \in Z_q^*$. Add the tuple $(ID_i, b)$ to $L_1$ and return $b$.

**Oracle** $\mathcal{O}_{H2}(\tau_e)$: $\mathcal{C}$ checks if there exists a tuple $(\tau_e, k_e)$ in $L_2$. If such a tuple exists, $\mathcal{C}$ answers with $k_e$. Otherwise, $\mathcal{C}$ does the following.

1. Choose a new $k \in_R Z_q^*$.such that no tuple $(., k)$ exists in $L_2$.

2. Add the tuple $(\tau_e, k)$ to $L_2$ and return $k$.

**Oracle** $\mathcal{O}_{H3}(m_e,\mathrm{R}_{1e}, \mathrm{k}_e)$: $\mathcal{C}$ checks if there exists a tuple $(m_e, R_{1e}, k_e, h_e)$ in $L_3$. If such a tuple exists, $\mathcal{C}$ answers with $h_e$. Else $\mathcal{C}$ performs:

1. Choose a new $h \in_R Z_q^*$.
2. Add the tuple $(m_e, R_{1e}, k_e, h)$ to $L_3$ and return $h$.

**Extract** ( $ID_e$)**:**

1. if $ID_e = ID_A$ return $\perp$.
2. If$ID_e \neq ID_A$, recover the tuple $(ID_e, b_e)$ from $L_1$ and return $(b_e P_{pub})$ as the secret key.

**Signcrypt** $(m, ID_s, ID_B)$:

1. if $ID_s \neq ID_A$, $\mathcal{C}$ computes the private key $\mathrm{S}_{ID_S}$ corresponding to $ID_S$ by running the key extraction query algorithm. Then $\mathcal{C}$ runs Keydis to output $n$ shared private keys $\{S_i\}_{i=1,\ldots,t}$. Finally, $\mathcal{C}$ answers the query by a call to $Signcrypt(m, \{S_i\}_{i=1,\ldots,t}, Q_{ID_B})$.

2. else, $\mathcal{C}$ chooses $x, h \in_R Z_q^*$. and computes $R_1 = xP - hQ_{ID_A}, W = xP_{pub}$, and $\tau = \hat{e}(R_1, S_{ID_B})$ ($\mathcal{C}$ could obtain $S_{ID_B}$ from the key extraction algorithm because $ID_B \neq ID_A$). $\mathcal{C}$ runs the $H_2$simulation algorithm to find $k = H_2(\tau)$ and computes $c = E_k(m)$. $\mathcal{C}$ then checks if $L_3$ already contains a tuple $(m, R_1, k, h^{'})$ with $h' \neq h$. In this case, $\mathcal{C}$ repeats the process with another random pair $(x, h)$ until finding a tuple $(m, R_1, k, h)$ whose first three elements do not appear in a tuple of the list $L_3$. Such a tuple, $(m, R_1, k, h)$ is then entered in $L_3$. $(C, R_1, W)$ is hence a valid signcryption according to the oracle.

**Unsigncryption queries:**

For a unsigncryption query on a ciphertext $\sigma' =(C', R_1', W')$ between a sender group with identity $ID_A$ and a receiver with identity $ID_B$. We have the following two cases to consider.

1. If $ID_A = ID_B$ .$\mathcal{C}$ always answers $\mathcal{A}$ that $\sigma'$ is invalid.
2. If $ID_B \neq ID_A$ . $\mathcal{C}$ computes $\tau'= \hat{e}(R_1', S_{ID_B})$. $\mathcal{C}$ then runs the $H_2$simulation algorithm to obtain $k' = H_2(\tau')$ and computes $m' = D_{k'}(c)$. Finally, $\mathcal{C}$ runs the $H_3$ simulation algorithm to obtain $h' = H_3(m', R_1', k')$ and checks if $\hat{e}(P, W') = \hat{e}(P_{pub}, R_1' + h'Q_{ID_A})$ holds. If the above equation does not hold, $\mathcal{C}$ rejects the ciphertext. Otherwise $\mathcal{C}$ returns $m$'.

After the first stage, $\mathcal{A}$ picks a pair of identities on which he wishes to be challenged on $(ID_i, ID_j)$. Note that If $\mathcal{A}$ queried the identity of $ID_A$,it would have failed in the first step itself. Then $\mathcal{A}$ outputs two plaintexts $\mathrm{m}_0$ and $\mathrm{m}_1$. $\mathcal{C}$ chooses $b \in_R \{0, 1\}$ and signcrypts $m_b$. To do so, he sets

$R_1^*= cP$, obtains $k^* = H_2(h)$(where $h$ is $\mathcal{C}$ candidate for the $DBDH$ problem) from the $H_2$ simulation algorithm, and computes $c_b = E_{k'}(m_b)$. Then $\mathcal{C}$ chooses $W^*\in\mathbb{G}_1^*$ and sends the ciphertext $\sigma^*= (c_b, R_1^*, W^*)$ to $\mathcal{A}$. $\mathcal{A}$ then performs a second series of queries which is treated in the same way as the first one. At the end of the simulation, he produces a bit $b_0$ for which he believes the relation $\sigma^*= Signcrypt(m_{b'}, S_{ii=1,\dots,t}, ID_j)$ holds. At this moment, if $b = b'$, $\mathcal{C}$ outputs $h = \hat{e}(R_1^*, S_{ID_j}) = \hat{e}(cP, abP) = \hat{e}(P, P)^{abc}$as a solution of the DBDH problem, otherwise $\mathcal{C}$ stops and outputs "failure".

So, we can see that the challenger $\mathcal{C}$ has the same advantage in solving the DBDH problem as the adversary $\mathcal{A}$ has in distinguishing a valid signcrypted ciphertext from a random string. So, if there exists an adversary who can succeed in such a CCA2 attack with nonnegligible advantage, that means there exists an algorithm to solve the DBDH problem with non-negligible advantage. Since this is not possible, no adversary can distinguish a valid signcrypted ciphertext from a random string with non-negligible advantage. Hence the scheme is secure against any IND-IDTSC-CCA2 attack.

## 7 Conclusions

In this paper, we have studied an existing identity-based threshold signcryption scheme by Fagen Li [1]. They have proved the confidentiality of their scheme, but the unforgeability proof given by them is based on the underlying scheme's security which loses it's validity in the new scheme. We have shown a possible attack on their scheme where the clerk can obtain the secret key of the sender. Hence a total break of the system is possible. We have also proposed an improved scheme and we have proved its security formally in the existing security model for identity-based threshold signcryption schemes. We leave it as an open problem to investigate for more efficient schemes for identity-based threshold signcryption.

## References


1. Fagen Li , Yong Yu, "An efficient and Provably Secure ID- Based Threshold Signcryption Scheme" In ICCCAS 2008.
2. Zheng Y.: Digital signcryption or How to achieve cost(signature & Encrytpion) cost(signature) + cost(encryption). In: Kaliski Jr., B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 165179. Springer, Heidelberg 1997.
3. Petersen H., Michels M.: Cryptanalysis and improvement of signcryption schemes. In: IEE proceedings- Computers and Digital Techniques 1998.
4. Bao F., Deng R.H.: A signcryption scheme with signature directly verifiable by public key. In: Imai, H., Zheng, Y. (eds.) PKC 1998. LNCS, vol. 1431, pp. 5559. Springer, Heidelberg 1998.

5. J.H. An, Y. Dodis, and T. Rabin, "On the security of joint signature and encryption", In Proc. Advances in Cryptology-EUROCRYPT 2002, LNCS 2332, pp. 83–107, Springer-Verlag, 2002.
6. J. Baek, R. Steinfeld, and Y. Zheng, "Formal proofs for the security of signcryption", In Proc. Public Key Cryptography-PKC 2002, LNCS 2274, pp. 80–98, Springer-Verlag, 2002
7. Y. Desmedt and Y. Frankel, "Shared generation of authenticators and signatures", In Proc. Advances in Cryptography-CRYPTO'91, LNCS 576, pp. 457–469, Springer-Verlag, 1991.
8. Malone-Lee J: Identity based signcryption. In: Cryptology ePrint Archive. Report 2002/098, 2002.
9. Y. Desmedt, "Society and group oriented cryptography: a new concept", In Proc. Advances in Cryptography-CRYPTO'87, LNCS 293, pp. 120– 127, Springer-Verlag, 1987.
10. Chow S.S.M., Yiu S.M., Hui L.C.K., Chow K.P.: Efficient forward and provably secure ID-based signcryption scheme with public verifiability and public ciphertext authenticity. In: Lim, J.-I., Lee, D.-H. (eds.) ICISC 2003. LNCS, vol. 2971, pp. 352369. Springer, Heidelberg 2004.
11. Boyen X.: Multipurpose identity based signcryption: a swiss army knife for identity based cryptography. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 383399. Springer, Heidelberg 2003.
12. S. Duan, Z. Cao, and R. Lu, "Robust ID-based threshold signcryption scheme from pairings", In Proc. 2004 International Conference on Information security, pp. 33–37, Shanghai, China, 2004.
13. Mu Y., Varadharajan V.: Distributed signcryption. In Roy, B., Okamoto, E. (eds.) INDOCRYPT 2000. LNCS, vol. 1977, pp. 155-164. Springer, Heidelberg (2000)
14. Yang G., Wong D.S., Deng X.: Analysis and improvement of a signcryption scheme with key privacy. In:Zhou, J., Lopez, J., Deng, R.H., Bao, F. (eds.) ISC 2005. LNCS, vol. 3650, pp. 218-232. Springer, Heidelberg (2005).
15. C. Peng and X. Li, "An identity-based threshold signcryption scheme with semantic security", In Proc. Computational Intelligence and Security-CIS 2005, LNAI 3802, pp. 173–179, Springer-Verlag, 2005.
16. R.L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems", Communications of the ACM, Vol. 21, No. 2, pp. 120–126, 1978.
17. Adi Shamir: Identity-Based Cryptosystems and Signature Schemes. In: CRYPTO 1984, Lecture Notes in Computer Science, pp. 47-53, 1984.
18. Malone-Lee J., Mao M: Two birds one stone: signcryption using RSA. In: Joye, M. (ed.) CT-RSA 2003. LNCS, vol. 2612, pp. 211226. Springer, Heidelberg 2003.
19. B. Libert and J.J. Quisquater, "A new identity based signcryption schemes from pairings", In Proc. 2003 IEEE information theory workshop, pp. 155–158, Paris, France, 2003.
20. Steinfeld R., Zheng Y.: A signcryption scheme based on integer factorization. In:Okamoto, E., Pieprzyk, J.P., Seberry, J. (eds.) ISW 2000. LNCS, vol. 1975, pp. 308-322. Springer, Heidelberg (2000)
21. C. Ma, K. Chen, D. Zheng, and S. Liu, "Efficient and proactive threshold signcryption", In Proc. Information Security Conference-ISC 2005, LNCS 3650, pp. 233–243, Springer-Verlag, 2005.
22. Libert B., Quisquater J.-J.: Efficient signcryption with key privacy from gap Diffie-Hellman groups. In: Bao, F., Deng, R., Zhou, J. (eds.) PKC 2004. LNCS, vol. 2947, pp. 187-200. Springer, Heidelberg (2004).

23. Baek J., Steinfeld R., Zheng Y.: Formal proofs for the security of signcryption.. In:Public Key Cryptography - PKC 2002, volume 2274 of Lecture Notes in Computer Science, pages 80-98. Springer-Verlag, 2002