A Secure and Efficient Identity-Based Proxy Signcryption in Cloud Data Sharing

Negalign Wake Hundera[1], Qian Mei[1], Hu Xiong[1]* and Dagmawit Mesfin Geressu[2]
[1] School of Information and Software Engineering, University of Electronic Science and Technology of China Chengdu China 610054.
[e-mail: nigaccna21@gmail.com]
[2] School of Information Communication Engineering, University of Electronic Science and Technology of China Chengdu China 610054.
*Corresponding author: Hu Xiong

Received April 21, 2019; revised July 6, 2019; accepted September 30, 2019; published January 31, 2020

Abstract

As a user in modern societies with the rapid growth of Internet environment and more complicated business flow processes in order to be effective at work and accomplish things on time when the manager of the company went for a business trip, he/she need to delegate his/her signing authorities to someone such that, the delegatee can act as a manager and sign a message on his/her behalf. In order to make the delegation process more secure and authentic, we proposed a secure and efficient identity-based proxy signcryption in cloud data sharing (SE-IDPSC-CS), which provides a secure privilege delegation mechanism for a person to delegate his/her signcryption privilege to his/her proxy agent. Our scheme allows the manager of the company to delegate his/her signcryption privilege to his/her proxy agent and the proxy agent can act as a manager and generate signcrypted messages on his/her behalf using special information called ”proxy key”. Then, the proxy agent uploads the signcrypted ciphertext to a cloud service provider (CSP) which can only be downloaded, decrypted and verified by an authorized user at any time from any place through the Internet. Finally, the security analysis and experiment result determine that the proposed scheme outperforms previous works in terms of functionalities and computational time.

Keywords: proxy signcryption, proxy credential, proxy key, proxy signature, delegator, delegate.

1. Introduction

The objective of public key cryptography (PKC) and digital signature is used to increase the privacy of the data by achieving the four important security requirements such as confidentiality, integrity, authentication, and non-repudiation. A traditional way of obtaining these four security goals is first to sign then apply the encryption algorithm on the message. Recently those most popular research areas such as cloud email systems, computer communications, a delegation of organizational power and electronic transactions need both security requirements. Because of the high computational cost and communication overhead, it's difficult to achieve all these goals simultaneously with the traditional approach. In 1997, Zheng [1] made the signcryption idea that accomplishes both confidentiality and authentication in a single reasonable step with better performance than the traditional methods. Later, many signcryption schemes have been proposed [2-9]. Some of these are proxy signcryption [6-8] which efficiently combines the idea of proxy signature with the signcryption scheme, and allows an entity to delegate its signcryption authority to a trusted agent on PKI framework. To solve the key management processes in PKI, the idea of ID-based cryptography (IBC) was developed by Shamir [10] in 1984. In ID-based cryptography, the recognized string (ASCII string) or identity such as email addresses, postal code, social security number represents an individual or organization public key, while the private key of user's is generated by PKG from their identity information. Malone-Lee [11] extended the signcryption idea to an ID-based signcryption scheme. Ever since many ID-based signatures [12-15] and signcryption [16-20] schemes have been proposed. Their key objective is to decrease the computational costs and develop efficient ID-based signcryption schemes. Now let's consider the situation when the manager of a company went for a business trip for a short or long period of time, in order to be effective at work and accomplish things on time he/she must delegate his/her signcryption authority to a proxy signcryptor who can legitimately signcrypt on behalf of him/her. So, this kind of situation must fulfill all the security requirements and the delegation process must be done in a secure and authentic way. The basic idea of our SE-IDPSC-CS scheme is as follow; the manager of the company officially delegate his/her signcryption authorization to his/her proxy agent and the proxy agent act as a manager and generate signcrypted messages on his/her behalf by using special information called " proxy key". Then the proxy agent uploads the signcrypted ciphertext to a trusted cloud service provider (CSP). Finally, the authorized user can download, recover and verify its source and validity at any time from any place through the Internet. Recently, Li and Chen [20] made ID-based proxy signcryption model, but their scheme is not secure and proxy protected, because the delegator is the only one who generates the proxy key without the knowledge of the proxy agent and they simply added the proxy key on [20] signcryption algorithm, if the original signcrypter remove the proxy key he/she will recover the message. Chen et al. [22] presented a probably secure ID-based proxy signcryption model under CDHP and BDHP assumptions. Ming et al. [23] constructed an ID-based proxy signcryption model without random oracles. Zhou [24] developed secure ID-based generalized proxy signcryption without random oracles from bilinear pairings and H Yu [25] proposed an ID-based proxy signcryption protocol with UC. But still, now all the above schemes consume high computational cost. This paper explains a new secure and efficient identity-based proxy signcryption in cloud data sharing (SE-IDPSC-CS) which is more secure and efficient than the existing schemes. The design philosophy behind our proposed scheme is as follow, the

manager of the company that is the original signcryptor officially delegate his/ her signcryption authority to proxy signcryptor, who then act as a manager and generate a signcrypted messages on his/her behalf and upload the signcrypted ciphertext to cloud service provider (CSP), it is a trusted server which supplies storage services and sends the signcrypted ciphertexts to an authorized user. Finally, an authorized user download, decrypt and confirm the source and validness of the message. We also proof the security of the scheme in terms of IND-IDPSC-CS-CCA2 and EF-IDPSC-CS-CMA under DBDH and CDH problems respectively. We organized the paper as follows. We define the preliminary work and the basic notations in section 3. The details of the system model, the overall framework and the security definition are presented in section 4. Section 5, provide the details of the construction while section 6 and section 7 describe security proof and performance analysis respectively. Finally, we conclude the paper in sections 8.

2. Related Work

A proxy signcryption [6-8] is a cryptographic algorithm that merges the idea of signcryption and algorithm with a proxy signature. In cryptography, signcryption is a cryptographic algorithm that achieves the functionality of both confidentiality and authentication in a single reasonable step with better performance than a traditional approach. The first signcryption scheme was proposed by Y. Zheng [1] later, several signcryption schemes have been proposed [2-8]. Proxy signature model, allows an entity officially delegate his/her signing right to someone so that he/she can sign a message on his/her behalf. The first proxy signature was proposed by Mambo et al. [26]. Later many proxy signature schemes [27-28] have been proposed. To solve key controlling processes in PKI, the concept of ID-based cryptography (IBC) was developed by Shamir [10] in 1984. In ID-based cryptography, the well-known string (ASCII string) or identity such as email address, postal code, social security number represents an individual or organization public key, while the secret key of the user's generated by PKG from their identity data. Later, several well-organized ID-based signatures [29-31] and ID-based schemes using pairings [32-33] have been proposed. Later many new ID-based signatures [34-35] and signcryption [18-20] schemes have been proposed. Malone-Lee [11] elaborated the signcryption idea to an ID-based signcryption scheme. Ever since many ID-based signcryption schemes have been proposed [15-20]. Recently, Li and Chen proposed an ID-based proxy signcryption scheme [21]. However, their scheme is not proxy protected and does not meet the unforgeability and forward security. In 2005, Wang and Cao [36] proposed an ID-based proxy signature and proxy signcryption scheme, which is based on [18] and is efficient than [21] in terms of computational point of view. Chen et al. [22] presented a probably secure ID-based proxy signcryption model under CDHP and BDHP assumptions. Ming et al. [23] constructed an ID-based proxy signcryption model without random oracles. Zhou [24] developed secure ID-based generalized proxy signcryption without random oracles from bilinear pairings and H Yu [25] proposed an ID-based proxy signcryption protocol with UC. Later Many schemes have been proposed for efficient and secure data accessing [37-40]. In this paper, by combining the idea of ID-based signcryption and proxy signature schemes, we proposed a new secure and efficient identity-based proxy signcryption in cloud data sharing (SE-IDPSC-CS) scheme, which is secure and efficient than the above schemes. In this scheme, after validating the identity of the delegator the proxy agent creates valid signcrypted ciphertext and uploads it to a cloud service provider (CSP). Moreover, only the authorized user can download, decrypt and confirm the source and authenticity of the message. Compared to the above schemes our scheme archived the necessary security requirements.

3. Preliminaries

In this section, we briefly define the bilinear pairings and notations.

3.1 Bilinear Pairings

Let $\mathbb{G}_1$ additive and $\mathbb{G}_2$ multiplicative cyclic groups having similar prime order q. Let $a, b \in Z_{q^*}$ and $\mathbb{G}_1$ is generated by P. A bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ has the following properties:

a. **Bilinear:** $(\mathrm{aP,bQ})=(\mathrm{P,Q})^{ab}$ for all $P, Q \in \mathbb{G}_1$, $a, b \in Z_{q^*}$.
b. **Non-degenerate:** $P, Q \in \mathbb{G}_1$ so that, $\hat{e}(P, Q) \neq 1$, where 1 is the identity of $\mathbb{G}_2$.
c. **Computability:** for all $P, Q \in \mathbb{G}_1$, $\hat{e}(P, Q)$ efficiently computable.

The revised Weil pairing and Tate pairing is acceptable applications [31]. The security of our EF-IDPSC-CS model depends on the following hard Diffie-Hellman problems. Given $\mathbb{G}_1$ additive and $\mathbb{G}_2$ multiplicative cyclic groups of the same prime order q. Let $a, b \in Z_{q^*}$ and $\mathbb{G}_1$ is generated by P, a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$:

a. *Computational Diffie-Hellman Problem (CDHP):* The CDHP in $\mathbb{G}_1$ is to calculate abP given (P, aP, bP) for $a, b \in Z_{q^*}$.
a. *Decisional Bilinear Diffie-Hellman Problem (DBDHP):* Assumed a set (P, aP, bP, cP) and an element $h \in \mathbb{G}_2$, the DBDHP to decide whether $h = \hat{e}(P, P)^{abc}$ or not. *We define the benefit of the adversary $\mathcal{C}$ against the DBDHP like this:* $Adv(\mathcal{C}) =$ $|P_{a,b,c\in Z_{q^*},h\in\mathbb{G}_2}[1 \leftarrow \mathcal{C}(aP, bP, cP, h)] - P_{a,b,c\in Z_{q^*}}[1 \leftarrow \mathcal{C}(aP, bP, cP, \hat{e}(P, P)^{abc})]|$ The DBDHP normally not harder than *CDHP*.

3.2 Notations

The notations used in this paper are listed in **Table 1**.

Table 1. Acronym and Description

No	Acronym	Description
1	Sk_{ap}	Proxy Key
2	USC	Unsigncryption
3	S_{pc}	Proxy Credential
4	PSC	Proxy Signcryption
5	PKeyGen	Proxy Key Generation
6	PKG	Private Key Generator
7	C_T	Signcryption Ciphertext
8	ID-PS	Identity-Based Proxy Signature
9	ID-PE	Identity-Based Proxy Encryption
10	ID-PSC	Identity-Based Proxy Signcryption
11	DBDH	Decision Bilinear Diffie-Hellman
12	CDH	Computation Diffie-Hellman
13	DLP	Discrete logarithm problem
14	PKI	Public key infrastructure
15	PKC	Public key cryptography
16	IBC	ID-base cryptography

4. System model, Framework and Security definition

In this section, we will define the system model, framework and security definition of the scheme.

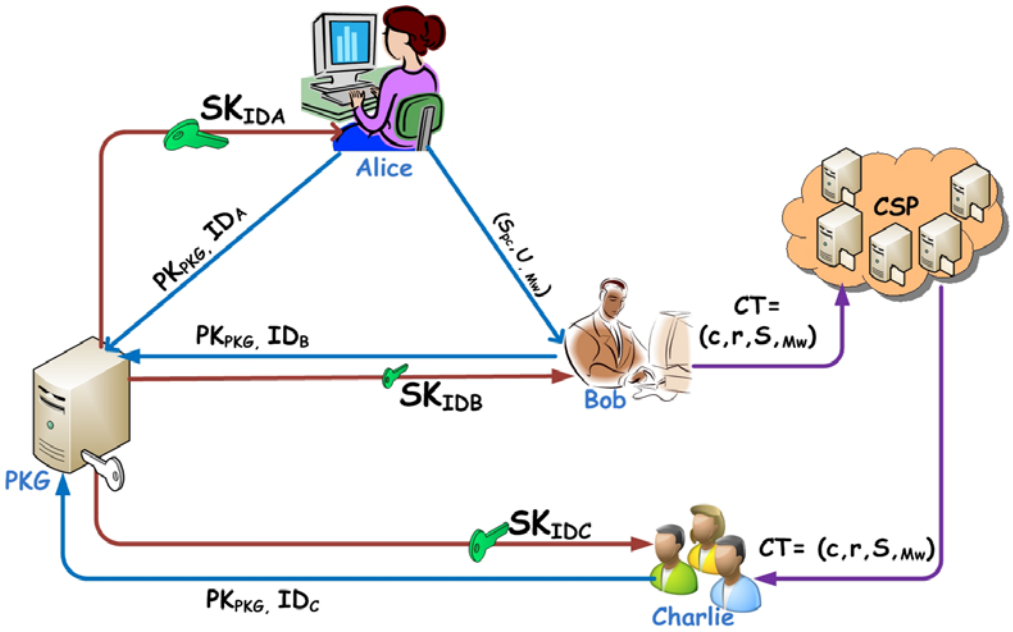

Fig. 1. Model of the SE-IDPSC-CS scheme.

4.1 System Model

According to **Fig. 1**, the architecture of SE-IDPSC-CS scheme consists of the following entities:

a. **PKG:** This is a trusted authority used to compute the user's private key from their identity information.
b. **Delegator (Alice):** This entity wants to delegate his/her signcryption authority to his/her proxy signcryptor (Bob).
c. **Proxy Signcryptor (Bob):** This is an entity that generates a signcrypted message on behalf of the delegator (Alice) by using the special information called "proxy key" and uploads it to a trusted cloud service provider (CSP).
d. **Cloud service provider (CSP):** This entity supplies the storage service and sends the signcrypted ciphertext to an authorized user.
e. **Receiver (Charlie):** An entity who download, wants to recover the message content and verify it's validity at any time, from anywhere through the Internet.

4.2 Framework of SE-IDPSC-CS scheme

Our scheme contains the following six algorithms, including the delegator (Alice) Q_{ID_A}, proxy signcryptor (Bob) Q_{ID_B}, receiver (Charlie) Q_{ID_C} and the cloud service provider (CSP).

1. **Setup:** On input the security parameter k, **PKG** output the public parameters $params$, and keep the master key s to itself.
2. **Extract:** On input, $params$, identity ID, and master secret key s returns the private key S_{ID} of ID, the **PKG** must transmit it to corresponding entities in a secure way.

Assume (Q_{ID_A}, S_{ID_A}) is delegator key pairs, (Q_{ID_B}, S_{ID_B}) proxy signcryptor key pairs and (Q_{ID_C}, S_{ID_C}) is receiver's key pairs.

3. **Proxy Credential:** This is an algorithm run by delegator that takes on input $params$, delegator private key S_{ID_A}, and warrant m_w (m_w contains the delegation time, identities of the delegator and proxy signcrypter), output a proxy credential S_{pc} and sends (S_{pc}, m_w) to a delegatee. There is a clear explanation of the delegation privileges and some common information about the delegator and proxy signcrypter in the warrant m_w such that a receiver can use it as verification information.
4. **PKeyGen:** An algorithm run by proxy signcryptor (Bob) which takes $params$, warrant m_w, proxy credential S_{pc} and delegatee private key S_{ID_B} as input. Outputs a proxy key Sk_{ap}.
5. **Proxy Signcryption:** An algorithm run by proxy signcryptor (Bob) that take on input $params$, the identity of the receiver Q_{ID_C}, the proxy key Sk_{ap}, a warrant m_w, and a message m. Output a signcryption ciphertext C_T.
6. **Unsigncryption:** On input, public parameters $params$, ciphertext C_T and S_{ID_C} of the receiver, then confirm whether the ciphertext C_T is correct, if "yes" continue and output the plaintext m, otherwise, output $\perp$.

4.3 Security Definition

We use a security game to describe the confidentiality and unforgeability of the message, here $\mathcal{C}$ is a challenger and $\mathcal{A}$ is an adversary respectively. We define two security models for these notions as follows:

Definition 1. *We say that a SE-IDPSC-CS scheme is said to achieve the security requirement of IND-SE-IDPSC-CS-CCA2 if no polynomial-time adversaries who have a non-negligible advantage win in the following game:*

Initial: $\mathcal{C}$ runs **Setup** algorithm to get $params$ and s. Then send $params$ to $\mathcal{A}$ and keeps s to itself.

Phase 1: $\mathcal{A}$ adaptively performs several kinds of queries; each query may be dependent on the outcome of the previous queries:

Extract query *QExtract(**ID**):* $\mathcal{A}$ chooses an identity ID. $\mathcal{C}$ runs $\mathbf{Extract}(ID)$ and S_{ID} to $\mathcal{A}$.

a. **Proxy Credential query** *QPC*$(params, S_{ID_i}, m_w)$*:* $\mathcal{A}$ issues a proxy credential request with respect to the delegatee. $\mathcal{C}$ returns a warrant m_w and proxy credential S_{pc}.
b. **PKeyGen query** *QPKeyGen*(S_{ID_j}, S_{pc})**:** $\mathcal{A}$ selects two identities ID_i, ID_j, for a given identity ID_i and ID_j, $\mathcal{C}$ first runs the *QPC* query to get S_{pc}. Then $\mathcal{C}$ runs **PKeyGen**(S_{pc}, S_{ID_j}) and returns Sk_{ap} to $\mathcal{A}$.
c. **Proxy Signcryption query** *QPSC*$(m, S_{ID_j}, ID_u, Sk_{ap})$**:** $\mathcal{A}$ selects a message m and three identities ID_i, ID_j and ID_C. $\mathcal{C}$ first, run **Extract** and **Proxy Credential** to get the private keys of S_{ID_i}, S_{ID_j} and the proxy credential S_{pc}, then run **PKeyGen**(S_{pc}, S_{ID_j}) to get Sk_{ap}. Finally, $\mathcal{C}$ it runs **PSC** $(m, S_{ID_j}, ID_C, Sk_{ap})$ and sends the result C_T to $\mathcal{A}$.
d. **Unsigncryption query** *QUS*(C_T, ID_i, ID_j, ID_C)*:* $\mathcal{A}$ chooses a ciphertext C_T and three identities ID_i, ID_j and ID_C. $\mathcal{C}$ first, run an **Extract** algorithm to get the S_{ID_C}.

Then $\mathcal{C}$ runs **USC**$(C_T, ID_i, ID_j, S_{ID_C})$ and sends the output to $\mathcal{A}$. This output can be the symbol $\perp$ if C_T is an invalid ciphertext.

Challenge: $\mathcal{A}$ choose two plaintext $M_0, M_1 \in M$ and two identities ID_B, ID_C on which he wishes to be challenged. He cannot have asked the private key corresponding to neither ID_B nor ID_C in the first stage. $\mathcal{C}$ chooses a random bit $b \in \{0,1\}$ and computes $\mathcal{C}$= signcryption (M_b, S_{ID_B}, ID_C) that is sent to $\mathcal{A}$.

Phase 2: $\mathcal{A}$ perform again a polynomial limited number of requests like in *Phase 1*. Except for the **Extract query** on ID_B nor ID_C and the plaintext corresponding to $\mathcal{C}$.

Guess: $\mathcal{A}$ creates a guess a bit b' and wins the game if $b' = b$. We define $\mathcal{A}$'s advantage as $Adv(\mathcal{A}) = |Pr[b' = b] - \frac{1}{2}|$.

Definition 2. *A SE-IDPSC-CS scheme is said to achieve the security requirement of EF-SE-IDPSC-CS-CMA if no polynomially time adversaries who have a non-negligible advantage in the following game:*

Initial: $\mathcal{C}$ runs **Setup** algorithm to get $params$ and s. Then sends $params$ to $\mathcal{F}$.

The adversary $\mathcal{F}$ performs a polynomial limited number of requests just like in the gam IND-SE-IDPSC-CS-CCA2.

Finally, $\mathcal{F}$ produces a new triple (C_T, ID_B, ID_C), where the secret key of ID_B was not asked in the 2nd phase and $\mathcal{F}$ wins the game if the output of **Unsigncryption** (C_T, S_{ID_B}, ID_C) is not $\perp$ a symbol. The advantage of $\mathcal{F}$'s is simply its probability of a win.

5. Construction

In this section, we briefly describe the six algorithms of our SE-IDPSC-CS scheme.

1. **Setup** (k): on input the security parameters k, the PKG choose two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of prime order q, a generator P of $\mathbb{G}_1$ a bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ and hash functions $H_1 : \{0,1\}^* \to \mathbb{G}_1$, $H_2 : \{0,1\}^* \to Z_{q^*}$, $H_3 : \mathbb{G}_2 \to \{0,1\}^n$, $H_4 : \{0,1\}^n \times \mathbb{G}_2 \to Z_{q^*}$. The PKG randomly chooses $s \in Z_{q^*}$ as a master key and calculates $P_{pub} = sP$. It also chooses a secure symmetric cipher(E, D). Then PKG publishes the system public parameters as $params$: $\{\mathbb{G}_1, \mathbb{G}_2, n, \hat{e}, P, P_{pub}, H_1, H_2, H_3, H_4, E, D\}$ and keeps the master key s secret. Where n is the bit length of a message.
2. **Extract** (ID) **:** on input an identity ID, the PKG computes $Q_{ID} = H_1(ID)$ and $S_{ID} = sQ_{ID}$, as the public and private keys of the user's respectively and transmit the private key $S_{ID} = sQ_{ID}$ to its owner in a secure way.
3. **Proxy Credential** $(params, S_{ID_A}, m_w)$**:** On input $params$, a delegator private key S_{ID_A}, and a warrant m_w. Then, delegator generates a proxy credential S_{pc} as follows:

$$x \in Z_{q^*}$$
$$U = xP$$
$$z = H_2(m_w, U)$$
$$S_{pc} = zS_{ID_A} + xP_{pub}$$

sends (m_w, U, S_{pc}) to a proxy signcryptor. There is a clear explanation of the delegation privileges and some common information about original and proxy signcrypter in the m_w which helps the receiver for verification.

4. **PKeyGen** $(m_w, U, S_{pc}, S_{ID_B})$**:** Upon receiving (m_w, U, S_{pc}), the proxy signcryptor confirms the validity of the received proxy credential by computing:

$$\hat{e}(P, S_{pc}) = \hat{e}(P_{pub}, zQ_{ID_A} + U) \tag{1}$$

Here, we show the verification process for Equ (1):

$$\begin{aligned}\hat{e}(P, S_{pc}) &= \hat{e}(P_{pub}, zQ_{ID_A} + U)\\ &= \hat{e}(P, zsQ_{ID_A} + xsP)\\ &= \hat{e}(P, zS_{ID_A} + xP_{pub})\\ &= \hat{e}(P, S_{pc})\end{aligned}$$

If Equ (1) is true, the proxy signcryptor computes the proxy key as follows

$$Sk_{ap} = zS_{ID_B} + S_{pc}$$

keep Sk_{ap} to itself and later it will be used to signcrypt message on behalf of the delegator.

5. **Proxy Signcryption** $(params, m, Q_{ID_C}, Sk_{ap}, S_{ID_B}, m_w)$**:** When the proxy signcryptor wants to signcrypt a message $m \in \{0,1\}^n$, on behalf of the delegator he/she first chooses $x^{'} \in Z_{q^*}$ and then computes

$$\begin{aligned}Q_{ID_C} &= H_1(ID_C)\\ k_1 &= \hat{e}(P, P_{pub})^{x^{'}}\\ k_2 &= H_3(\hat{e}(P_{pub}, Q_{ID_C})^{x^{'}})\\ c &= E_{k_2}(m)\\ r &= H_4(c, k_1)\\ S &= x^{'}P_{pub} - (rS_{ID_B} + Sk_{ap})\\ C_T &= (c, r, S, m_w)\end{aligned}$$

where $x^{'}$ is the random number, E_{k_2} is the encryption function with the private key k_2. Then, the proxy signcryptor uploads the ciphertext $C_T = (c, r, S, m_w, U)$ to the cloud service provider (CSP).

6. **Unsigncryption** $(params, S_{ID_C}, C_T)$**:** When Charlie wants the data, he can download the signcrypted ciphertext $C_T = (c, r, S, m_w)$ from a cloud service provider (CSP) and perform the following tasks:

$$\begin{aligned}Q_{ID_A} &= H_1(ID_A)\\ Q_{ID_B} &= H_1(ID_B)\\ k_1' &= \hat{e}(P, S)\hat{e}(P_{pub}, Q_{ID_B})^{z+r}\hat{e}(P_{pub}, zQ_{ID_A} + U)\\ k_2' &= H_3(\hat{e}(S, Q_{ID_C})\hat{e}(Q_{ID_B}, S_{ID_C})^{z+r}\hat{e}(zQ_{ID_A} + U, S_{ID_C}))\end{aligned}$$

then receives $m = E_{k_2'}(c)$ and accepts C_T iff $r = H_4(c, k_1')$. Otherwise, returns an error symbol $\perp$.

Proof of correctness
The following equations give the correctness of our proposed scheme:

$$\begin{aligned}
k_1 &= \hat{e}(P,S)\hat{e}(P_{pub},Q_{ID_B})^{z+r}\hat{e}(P_{pub},zQ_{ID_A}+U)\\
&= \hat{e}(P,S)\hat{e}(P_{pub},Q_{ID_B})^{z+r}\hat{e}(P,zS_{ID_A}+xP_{pub})\\
&= \hat{e}(P,S)\hat{e}(P_{pub},Q_{ID_B})^{z+r}\hat{e}(P,S_{pc})\\
&= \hat{e}(P,S)\hat{e}(P,rS_{ID_B})\hat{e}(P,zS_{ID_B})\hat{e}(P,S_{pc})\\
&= \hat{e}(P,x^{'}P_{pub}-rS_{ID_B}-Sk_{ap})\hat{e}(P,rS_{ID_B})\hat{e}(P,SK_{ap})\\
&= \hat{e}(P,P_{pub})^{x'}\\
&= k_1
\end{aligned}$$

$$\begin{aligned}
k_2 &= H_3(\hat{e}(S,Q_{ID_C})\hat{e}(Q_{ID_B},S_{ID_C})^{z+r}\hat{e}(zQ_{ID_A}+U,S_{ID_C}))\\
&= H_3(\hat{e}(S,Q_{ID_C})\hat{e}(rS_{ID_B},Q_{ID_C})\\
&.\hat{e}(zS_{ID_B}+zS_{ID_A}+xP_{pub},Q_{ID_C}))\\
&= H_3(\hat{e}(S,Q_{ID_C})\hat{e}(rS_{ID_B},Q_{ID_C})\hat{e}(Sk_{ap},Q_{ID_C}))\\
&= H_3(\hat{e}(x^{'}P_{pub}-rS_{ID_B}-Sk_{ap},Q_{ID_C})\\
&\hat{e}(rS_{ID_B}+Sk_{ap},Q_{ID_C}))\\
&= H_3(\hat{e}(P_{pub},Q_C)^{x'})\\
&= k_2
\end{aligned}$$

6. Security proof

In this section, we prove that the proposed scheme fulfills *IND-SE-IDPSC-CS-CCA2* and *EUF- SE-IDPSC-CS-CMA* security by the following Theorems 1 and 2, respectively.
Theorem 1. (Proof of *IND-SE-IDPSC-CS-CCA2): The proposed scheme in this paper secure against IND-SE-IDPSC-CS-CCA2, if no polynomial-time adversaries who have a non-negligible advantage* $\mathcal{A}$ *which can* $(\varepsilon^{'},t^{'})$ *break DBDHP where,*

$$\varepsilon^{'} \geq 2(\varepsilon - q_u/2^{k-1})/q_{H_1}^4$$

$$t^{'} \approx t + O(q_{pk}+q_s+q_u)t_\lambda$$

where t_λ is time to calculate one pairing operation.

Proof: Assume $\mathcal{A}$ can $(t,q_{H_1},q_{H_2},q_{H_3},q_{H_4},q_e,q_{pk},q_s,q_u,\varepsilon)$- break the SE-IDPSC-CS scheme with significant advantage ε under adaptive CCA2 after running t (time) and requesting at most q_{H_i} random oracle for $(i=1 \text{ to } 4)$, q_e Extract query, q_{pk} PKeyGen query, q_s proxy signcryption query, and q_u unsigncryption query. Then we can build another algorithm $\mathcal{C}$ that $(t^{'},\varepsilon^{'})$-breaks the DBDHP by taking $\mathcal{A}$ as a subroutine.
Assume $\mathcal{C}$ obtains a random instance (P,aP,bP,cP,h) of the DBDHP and the objective of $\mathcal{C}$ is to obtain $h=\hat{e}(P,P)^{abc}$ or not.

Initial: In this proof, $\mathcal{C}$ pretends $\mathcal{A}$'s challenger in the IND-SE-IDPSC-CS-CCA2 game.
Phase 1: $\mathcal{C}$ maintains five empty lists L_1, L_2, L_3, L_4 and L_{ap} which used to simulate the hash oracles H_1, H_2, H_3 and H_4 respectively and L_{ap} is an empty list which is used to describe the **Extract** and **PKeyGen query** Oracles and L_4 will be used to simulate the proxy signcryption query.
At the beginning of the game, $\mathcal{C}$ gives $\mathcal{A}$ $params$ with $P_{pub} = cP$. Then, $\mathcal{C}$ chooses two distinct random numbers $i, j \in \{1, ..., q_{H_1}\}$. $\mathcal{A}$ requests a number of H_1 queries on identities of his/her choice. At i^{th} H_1 request, $\mathcal{C}$ answers $H_1(ID_i) = aP$. At j^{th} H_1 request, $\mathcal{C}$ answers $H_1(ID_j) = bP$. Since, aP and bP belong to DBDHP, $\mathcal{A}$'s understanding will not be changed by these variations. Hence, the private keys S_{ID_i} and S_{ID_j} are acP and bcP respectively. Thus the solution $\hat{e}(P, P)^{abc}$ of the DBDHP is given by $\hat{e}(Q_{ID_i}, S_{ID_j}) = (S_{ID_i}, Q_{ID_j})$. For requests $H_1(ID_e)$ with $e \neq i, j, \mathcal{C}$ chooses $b_e \leftarrow Z_{q^*}$, puts the pair (ID_e, b_e) in list L_1 and answers $H_1(ID_e) = b_e P$.
Let see how $\mathcal{A}$ issues the other kinds of request.

1. H_2 **requests:** $\mathcal{A}$ sends a query to H_2 oracle $\mathcal{C}$ checks L_2 for a matched entry. If similar entry is found in L_2, $\mathcal{C}$ returns z. Otherwise, $\mathcal{C}$ randomly chooses $z \in Z_{q^*}$ and stores the entry (m_w, U, z) into L_2.
2. H_3 **requests:** on $H_3(g_e)$ request, $\mathcal{C}$ searches (g_e, R_e) in the list L_3. If similar entry is found in L_3, $\mathcal{C}$ replies R_e otherwise $\mathcal{C}$ answer $\mathcal{A}$, $R_e \leftarrow \{0,1\}^n$ such that, no entry $(., R)$ exists in L_3 and puts the pair (g_e, R) into L_3.
3. H_4 **requests:** for $H_4(c_e, k_e)$ query, $\mathcal{C}$ first checks L_4 does not contain a tuple (c_e, k_e, r_e). If (c_e, k_e, r_e) is found, $\mathcal{C}$ answers r_e, otherwise $\mathcal{C}$ choose $r \leftarrow Z_{q^*}$ and give it the query as an answer and place (c_e, k_e, r) into L_4.
4. **Extract query:** when $\mathcal{A}$ asks Extract(ID_A), then $\mathcal{C}$ fails and stops, if $ID_A = ID_i$ or $ID_A = ID_j$. If $ID_A \neq ID_i, ID_j$ then L_1 must contain a pair (ID_A, d) for some d (It shows $\mathcal{C}$ previously answered $H_1(ID_A) = dP$ H_1 query on ID_A). Then $\mathcal{C}$ computes $dP_{pub} = cdP$ as ID_A private key and returned to $\mathcal{A}$.
5. **PKeyGen query:** Let us assume $\mathcal{A}$ has made the H_2 and extraction query before the proxy key query. When $\mathcal{A}$ makes a proxy key query of the tuple (ID_A, ID_B, m_w) for the proxy signcryptor, $\mathcal{C}$ first check the L_2 list for a matching entry. Otherwise, $\mathcal{C}$ randomly choose $x_a \in Z_{q^*}$ to compute.

$$x_a \in Z_{q^*}$$
$$U_a = x_a P$$
$$z = H_2(m_w, U_a)$$
$$S_{pc} = zS_{ID_A} + x_a P_{pub}$$

If $\hat{e}(P, S_{pc}) = \hat{e}(P_{pub}, zQ_{ID_A} + U_a)$ holds $\mathcal{C}$ compute the proxy key Sk_{ap} using the equality $Sk_{ap} = zS_{ID_B} + S_{pc}$ and return Sk_{ap} to the adversary. Finally, $\mathcal{C}$ stores (Sk_{ap}, z, m_w) to L_{ap}.
6. **Proxy Signcryption query:** Assume $\mathcal{A}$ has made many **Hash**, **Extract, PKeyGen** queries before q_s query. $\mathcal{A}$ request a query of $(M, ID_B, ID_C, m_w, SK_{ap})$ to **Proxy Signcryption query**. $\mathcal{C}$ considers the following two cases as a reply.

Case 1: If ID_B and ID_C are not the identities ID_i and ID_j. In the case $ID_B \neq ID_i, ID_j$, $\mathcal{C}$ run an **Extract** algorithm to calculate S_{ID_B} of ID_B and then can

simply run **Signcryption** (m, S_{ID_B}, Q_{ID_C}) and send C_T to $\mathcal{A}$.

Case 2: If $ID_B = ID_i$ or $ID_B = ID_j$ and $ID_C \neq ID_i, ID_j$, $\mathcal{C}$ has to simulate the execution of the **Proxy Signcryption query** as follows. $\mathcal{C}$ chooses $r, z \in Z_{q^*}$ and $S \in \mathbb{G}_1^*$. Then, computes:

$$k^{'} = \hat{e}(P,S)\hat{e}(P_{pub}, Q_{ID_B})^{z+r}\hat{e}(P_{pub}, zQ_{ID_A} + U) \ and$$
$$\tau = \hat{e}(S, Q_{ID_C})\hat{e}(Q_{ID_B}, S_{ID_C})^{z+r}\hat{e}(zQ_{ID_A} + U, S_{ID_C})$$

where S_{ID_C} is the private key of ID_C. $\mathcal{C}$ runs H_3 algorithm to find $k_2 = H_3(\tau)$ and calculates $c = E_{k_2}(M)$. $\mathcal{C}$ checks if L_4 contains $(c, k^{'}, r^{'})$ with $r^{'} \neq r$. $\mathcal{C}$ repeats checking with another pair (r, S) until he found $(c, k^{'}, r)$. Before obtaining an allowable $(k^{'}, r, S)$, $\mathcal{C}$ must repeat the checking at most $2q_R$ times. At each attempt, $\mathcal{C}$ must calculate four pairings $\hat{e}$. Once $(k^{'}, r, S)$ is found, $\mathcal{C}$ puts $(c, k^{'}, r)$ into L_4 before returning (c, r, S) which will look to be effective from $\mathcal{A}$'s point of view.

If ID_B and ID_C are the identities of ID_i and ID_j, $\mathcal{C}$ signcrypt M like this. $\mathcal{C}$ chooses $r^* \in Z_{q^*}$ and $S^* \in \mathbb{G}_1$ computes $k_1^{'} = \hat{e}(P, S^*)\hat{e}(P_{pub}, Q_{ID_b})^{r^*} = \hat{e}(P, S^*)\hat{e}(cP, bP)^{r^*}$ and chooses random $\tau^* \in \mathbb{G}_2$ and $k_2^{'} \in \{0,1\}^n$ such that no entry $(., k_2^{'})$ is in L_2 and computes $c^* = E_{k_2^{'}}(M)$. $\mathcal{C}$ then confirms if L_3 contains $(c^*, k_1^{'}, r^*)$ such that $r^{'} \neq r^*$. If not, $\mathcal{C}$ puts the $(c^*, k_1^{'}, r^*)$ into L_4 and $(\tau^*, k_2^{'})$ into L_3. In the opposite case, $\mathcal{C}$ chooses another random pair (r^*, S^*) and repeats the above process until $\mathcal{C}$ finds a tuple $(c^*, k_1^{'}, r^*)$. Once $\mathcal{C}$ has admissible elements (r^*, S^*), $\mathcal{C}$ gives the ciphertext $C_T^* = (c^*, r^*, S^*)$ to $\mathcal{A}$. $\mathcal{A}$ will never know that C_T^* is not a valid ciphertext of M for ID_i and ID_j since $\mathcal{C}$ will not ask unsigncrypt of C_T^*.

Unsigncrypt query: Once $\mathcal{A}$ observes $C_T^* = (c^*, r^*, S^*)$ for ID_B and ID_C, $\mathcal{A}$ needs to ask $\mathcal{C}$ for unsigncryption C_T^*. In this case $\mathcal{C}$ informs $\mathcal{A}$ that C_T^* is invalid: if $\mathcal{A}$ before requesting the hash value $H_3(c^{'}, \hat{e}(P,S)\hat{e}(P_{pub}, Q_{ID_B})^{z+r}\hat{e}(P_{pub}, zQ_{ID_A} + U)$, $\mathcal{C}$ has a probability of at most $1/2^k$ to answer $r^{'}$. This will fails if L_4 contains $(c^{'}, \hat{e}(P,S)\hat{e}(P_{pub}, Q_{ID_B})^{z+r}$ $\hat{e}(P_{pub}, zQ_{ID_A} + U), r^{'})$. When getting an **Unsigncrypt query** for $C_T^* = (c^*, r^*, S^*)$ identities ID_B and ID_C that is not ID_i and ID_j, $\mathcal{C}$ first calculates $k_1^{'} = \hat{e}(P, S^{'})\hat{e}(P_{pub}, Q_{ID_B})^{z+r^{'}}\ \hat{e}(P_{pub}, zQ_{ID_A} + U)$ and checks if the list L_4 contains the tuple $(c^{'}, k_1^{'}, r^{'})$. If it is not found, $\mathcal{C}$ reject the ciphertext. Otherwise, $\mathcal{C}$ can recover $r^{'}$ and calculate $\tau^{'} = \hat{e}(S^{'}, Q_{ID_C})\hat{e}(Q_{ID_B}, S_{ID_C})^{z+r^{'}}\hat{e}(zQ_{ID_A} + U, S_{ID_C})$. $\mathcal{C}$ then searches for $H_3(\tau^{'})$ queries in the list L_3. If $H_3(\tau^{'})$ query is not found in L_3, $\mathcal{C}$ takes $(\tau, k_2^{'}) \in \mathbb{G}_2 \times \{0,1\}^n$ such that no $(., k_2^{'})$ already exists in L_3 and inserts $(\tau, k_2^{'})$ into L_3. $\mathcal{C}$ finally uses $k_2^{'}$ to find $m^{'} = D_{k_2^{'}}(c^{'})$ and returns $m^{'}$. Here we can easily observe that the probability of refusing a valid ciphertext does not exceed $qu/2^k$. Definitely, for a query on identities ID_B and ID_C that is not ID_i and ID_j, if $\mathcal{A}$ later requests $H_3(c^{'}, \hat{e}(P,S^{'})\hat{e}(P_{pub}, Q_{ID_B})^{z+r^{'}}\hat{e}(P_{pub}, zQ_{ID_A} + U))$, there is a probability of at most $1/2^k$ that $\mathcal{C}$ answers $r^{'}$. After a polynomial limited number of queries $\mathcal{A}$ chooses (ID_i, ID_j) which he needs to be challenged with a probability at least $(1/2^{qH_1})$. Notice that, if $\mathcal{A}$ actually selects to be challenged on ID_i and ID_j, then in the 2nd stage $\mathcal{A}$ cannot ask the private key of ID_i and ID_j. If $\mathcal{A}$ does not select ID_i and ID_j as target identities, then $\mathcal{C}$ fails. When $\mathcal{A}$

produces two plaintexts m_0 and m_1, $\mathcal{C}$ selects a random bit $b \in \{0,1\}$ and signcrypt m_b. To do so $\mathcal{C}$ chooses $r^* \leftarrow Z_{q^*}$ and $S^* \leftarrow \mathbb{G}_1$.
$\mathcal{C}$ calculates $k_1^{'} = \hat{e}(P,S^*)\hat{e}(P_{pub},Q_{ID_b})^{r^*} = \hat{e}(P,S^*)\hat{e}(cP,bP)^{r^*}$, $\tau^* = \hat{e}(S^*,Q_{ID_C})^{h^{r^*}}$ (where h is $\mathcal{C}$'s candidate for DBDHP) to obtain $k_2^{'} = H_3(\tau^*)$ and $c_b = E_{k_2^{'}}(m_b)$. $\mathcal{C}$ then verifies if L_4 already contains an entry $(c_b, k_1^{'}, r^{'})$ such that $r^{'} \neq r^*$. If not, $\mathcal{C}$ puts the tuple $(c_b, k_1^{'}, r^*)$ into L_4. In the reverse case, $\mathcal{C}$ chooses another random pair (r^*, S^*) and repeats the procedure until finding a tuple $(c_b, k_1^{'}, r^*)$. Once $\mathcal{C}$ has admissible elements (r^*, S^*), $\mathcal{C}$ just have to send the ciphertext $\sigma = (c_b, r^*, S^*)$ to $\mathcal{A}$.
$\mathcal{A}$ then runs the second queries which are similar to the above one, finally $\mathcal{A}$ produces a bit $b^{'}$ for the relation $C_T = Signcrypt\ (m_{b'}, S_{ID_i}, ID_j)$ holds. At this moment, if $b = b^{'}$, $\mathcal{C}$ answers 1 because his candidate h permitted him to create σ that seemed to $\mathcal{A}$ as a valid signcrypted message of m_b. If $b \neq b^{'}$, $\mathcal{C}$ then answers 0. We must have to calculate $\mathcal{C}$'s probability of success. We saw that $\mathcal{C}$ fails if $\mathcal{A}$ asks the private key of either ID_i or ID_j in the first phase, and also we know that there are (2^{qH_1}) ways to select ID_i, ID_j among those (2^{qH_1}) pairs of identities, none of them are the subject of an **Extract** query from $\mathcal{A}$. Then, with a probability greater than $1/(2^{qH_1})$, $\mathcal{A}$ will not ask $Extract\ (ID_i)$ and $Extract\ (ID_i)$. Additionally, with a probability exactly $1/(2^{qH_1})$, $\mathcal{A}$ selects to be challenged on the pair ID_i, ID_j and this must allow $\mathcal{C}$ solving his decisional problem if $\mathcal{A}$ wins the IND-SE-IDPSC-CS-CCA2 game.

Finally,

$$
\begin{aligned}
&p_1 = P[b^{'} = b \mid C_T = Signcrypt(m_b, S_{ID_i}, ID_j)] \\
&= \frac{\varepsilon + 1}{2} - \frac{qU}{2^k}, and \\
&p_0 = P[b^{'} = i \mid h \in \mathbb{G}_2] = \frac{1}{2}, for\ i = 0, 1, we\ then\ have \\
&Adv(\mathcal{C}) = |P_{a,b,c} \in Z_{q^*}[1 \leftarrow \mathcal{C}(aP, bP, cP, \hat{e}(P,P)^{abc})] - \\
&P_{a,b,c} \in Z_{q^*}, h \in \mathbb{G}_2[1 \leftarrow \mathcal{C}(aP, bP, cP, h)]| \\
&= \frac{|p_1 - p_0|}{(2^{qH_1})^2} = \frac{\varepsilon - qu/2^{k-1}}{2(2^{qH^1})^2} > \frac{2(\varepsilon - qu/2^{k-1})}{qH_1^4}.
\end{aligned}
$$

We note that the denominator is qH_1^4 rather than qH_1^2
Theorem 2. *(Proof of EUF-SE-IDPSC-CS-CMA): The proposed scheme in this paper secure against EUF-SE-IDPSC-CS-CMA security if no probabilistic polynomial-time adversary $\mathcal{F}$ with a non-negligible advantage that can $(\varepsilon^{'}, t^{'})$ break CDHP where,*

$$\varepsilon \geq 10(q_s + 1)(q_s + q_{H_3})q_{H_1}/(2^k - 1)$$

$$t^{'} \leq 120686 q_{H_1} q_{H_3} \frac{t + O((q_{pk} + q_s + q_u q_{H_3})t_\lambda)}{\hat{e}(1 - 1/2^k)}$$

where t_λ is time to calculate one pairing operation.

Proof: Assume $\mathcal{A}$ can $(t, q_{H_1}, q_{H_2}, q_{H_3}, q_{H_4}, q_e, q_{pk}, q_s, \varepsilon)$-break our *SE-IDPSC-CS* scheme with non-negligible advantage ε under the adaptive CMA after running in t (time) and asking at most q_{H_i} random oracle for $(i = 1 \text{ to } 4)$, q_e extraction query, q_{pk} PKeyGen query, q_s proxy signcryption query. Then we can build another algorithm $\mathcal{C}$ that $(t^{'}, \varepsilon^{'})$-breaks the CDHP by taking $\mathcal{F}$ as a subroutine. Assume the algorithm $\mathcal{C}$ takes as the input (P, aP, bP) of the CDHP, and the objective of $\mathcal{C}$ is to computeabP.

As proof in [18], this theorem using the famous forking lemma [31]. $\mathcal{C}$ simulates $\mathcal{F}$'s a challenger in the *EUF-SE-IDPSC-CS-CMA* game. $\mathcal{F}$ adaptively asks queries as described in the *EUF-SE-IDPSC-CS-CMA* game. We define the procedure as follows.

Initial: $\mathcal{C}$ runs **Setup** algorithm with k and gives $\mathcal{F}$ $P_{pub} = bP$.

Attack: $\mathcal{C}$ simulates$\mathcal{F}$'s a challenger in the *EUF-SE-IDPSC-CS-CMA* game and answers $\mathcal{F}$'s queries according to the procedures in **Theorem 1**. In addition, we need to set the challenge identity $ID_A = ID_l$ in H_1 queries.

Forgery: $\mathcal{F}$ outputs a triple (ID_A, ID_B, C_T), where $C_T = (X, y, \tau)$ (note that $\mathcal{C}$ can extract message-signature pair from C_T since it identifies S_{ID_B} of ID_B because of irreflexivity assumption). Since the dividing lemma is only suitable for identity-less chosen message attacks, we need to merge the message m and the sender identity ID_A into a "general" forged message (ID_A, m). From the dividing lemma, if $\mathcal{F}$ is an effective forger in the above game, we can build a Las Vegas machine $\mathcal{F}^{'}$that outputs $((ID_A, m), h, Z)$ and $((ID_A, m), h^*, Z^*)$ with $h \neq h^*$ and the same commitment. To solve the given CDHP based on the machine $\mathcal{F}^{'}$ derived from $\mathcal{F}$, we build $\mathcal{C}$ as follows.

1) $\mathcal{C}$ obtains two different signatures $((ID_A, m), h, Z)$ and $((ID_A, m), h^*, Z^*)$ by running $\mathcal{F}^{'}$.
2) $\mathcal{C}$ computes $abP = (h - h^*)^{-1}(Z - Z^*)$.
3) $\mathcal{C}$ computes abP as the solution to the CDH problem.

From the dividing lemma and the lemma on the affiliation among the chosen identity attack and the given-identity attack [35], if $\mathcal{F}$ flourishes in time t with probability $\varepsilon \geq 10(q_s + 1)(q_s + q_{H_3})q_{H_1}/(2^k - 1)$, then $\mathcal{C}$ can solve the given CDHP in the expected time

$$t^{'} \leq 120686 q_{H_1} q_{H_3} \frac{t + O((q_{pk} + q_s + q_u q_{H_3})t_\lambda)}{\hat{e}(1 - 1/2^k)}$$

Table 2. Shows the comparison of computational cost and security

Schemes	PKeyGen	Proxy Signcryption	Unsigncryption	Total	Security	
					IND-CCA2	EUF-CMA
[22]	3M + 2P + 1E	3M + 2P + 2E	1M + 5P + 1E	7M + 9P + 4E	✓	✓
[23]	2M + 2P + 1E	1M + 2P + 2E	4E + 6P	3M + 10P + 7E	✓	✓
[25]	2M + 2P + 1E	3M + 1P + 1E	1M + 4P + 1E	6M + 7P + 3E	✓	✓
Ours	2M + 2P	2M	2M + 4P	6M + 6P	✓	✓

7. Performance analysis

Following the idea of [20-21], the pairing $\hat{e}(P, P_{pub})$ is pre-computed and when users frequently talk with each other all $\hat{e}(P_{pub}, Q_{ID_A})$, $\hat{e}(P_{pub}, Q_{ID_B})$, $\hat{e}(P_{pub}, Q_{ID_C})$, $\hat{e}(Q_{ID_A}, S_{ID_C})$ and $\hat{e}(Q_{ID_B}, S_{ID_C})$ can be pre-computed by the sender and receiver once for all.

7.1 Comparison

In this section, we compare the performance of our scheme with related schemes such as

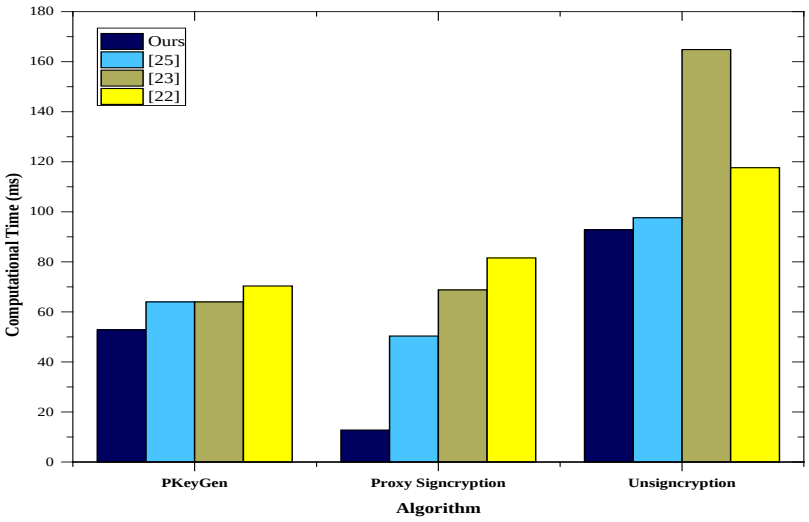

Fig. 2. Computational time of our scheme

Chen et al. [22], Ming et al. [23] and H Yu [25] interims of efficiency and security. We denote M one scalar multiplication operation in $\mathbb{G}_1$, E exponentiation computation and P the pairing operation in $\mathbb{G}_2$. In **Table 2**, a symbol $\surd$ means that all the schemes satisfy the related security requirement. Our experiment was implemented on the hardware platform of ASUS Z-Book with an Intel (R) Core™ i3-6100U CPU 2.3GHz and 4 GB memory running 64-bit Windows 10 operating system. According to Cao [38], time spent on one scalar multiplication operation in $\mathbb{G}_1$ is approximately 6.38 ms, one exponentiation computation in $\mathbb{G}_2$ and paring are approximately 11.20ms and 20.01ms respectively. From the comparison result shown in **Table 2**, one can see that the computation cost of our SE-IDPSC-CS scheme is lower than other schemes and **Fig. 2** clearly shows that our SE-IDPSC-CS scheme is much more efficient than the present schemes. It is known from **Table 2** that all the schemes satisfy the IND-CCA2 and EUF-CMA security requirements.

8. Conclusion

In this paper, we explain a new secure and efficient identity-based proxy signcryption in cloud data sharing (SE-IDPSC-CS) which is secure and efficient than the current schemes. The idea behind our proposed scheme is as follow, the manager of the company that is the original signcryptor officially delegate his/ her signcryption authority to the proxy signcryptor, then the proxy signcrypter act as a manager and generate a signcrypted messages on his/her behalf and upload the signcrypted ciphertext to cloud service provider (CSP) it is a trusted server which supplies storage services and sends the signcrypted ciphertexts to an authorized users. Finally, an authorized user download, decrypt and confirm the source and validity of the message. We also prove the security of the scheme in terms of IND-SE-IDPSC-CS-CCA2 and EF-SE-IDPSC-CS-CMA under DBDH and CDH problems respectively. Finally, we will work to design an outsourced ID-based proxy signcryption scheme in cloud data sharing as our future work to reduce the signing computational overload both in the delegate and user side.

Acknowledgments

This work was supported in part by the 13th Five-Year Plan of National Cryptography Development Fund for Cryptographic Theory of China under Grant MMJJ20170204, in part by the Fundamental Research Funds for the Central Universities under Grant ZYGX2016J091, the Guangxi Colleges and Universities Key Laboratory of Cloud Computing and Complex Systems, and in part by the Natural Science Foundation of China under Grants U1401257, 61472064 and 61602096.

References

[1] Y. Zheng, "Digital Signcryption or how to achieve cost (signature & encryption) << cost (signature)+ cost (encryption)," in *Proc. of Advances in Cryptology — CRYPTO '97*, pp 165-179, 1997. Article (CrossRef Link)

[2] F. Li, B. Liu, and J. Hong, "An efficient signcryption for data access control in cloud computing," *Computing*, vol. 99, no. 5, pp. 465-479, 2017. Article (CrossRef Link).

[3] R.-J. Hwang, C.-H. Lai, and F.-F. Su, "An efficient signcryption scheme with forward secrecy based on elliptic curve," *Applied Mathematics and computation*, vol. 167, no. 2, pp. 870-881, 2005. Article (CrossRef Link).

[4] H. Y. Jung, D. H. Lee, J. I. Lim, and K. S. Chang, “Signcryption schemes with forward secrecy,” in *Proc. of WISA2001, Springer-Verlag*, pp. 4303–475, 2001.

[5] Y. Zheng and H. Imai, "How to construct efficient signcryption schemes on elliptic curves," *Information Processing Letters*, vol. 68, no. 5, pp. 227-233, 1998. Article (CrossRef Link).

[6] C. Gamage, J. Leiwo, and Y. Zheng, “An efficient scheme for secure message transmission using proxy-signcryption,” in *Proc. of the 22nd Australasian Computer Science Conference, Springer*, pp. 420–431, 1999.

[7] C. Zhou, Z. Zhao, W. Zhou, and Y. Mei, "Certificateless key-insulated generalized signcryption scheme without bilinear pairings," *Security and Communication Networks*, vol. 2017, 17 pages, 2017. Article (CrossRef Link).

[8] V. Saraswat, R. A. Sahu, and A. K. Awasthi, "A secure anonymous proxy signcryption scheme," *Journal of Mathematical Cryptology*, vol. 11, no. 2, pp. 63-84, 2017. Article (CrossRef Link).

[9] P. Pandiaraja, P. Vijayakumar, V. Vijayakumar, and R. Seshadhri, “Computation efficient attribute based broadcast group key management for secure document access in public cloud.” *J. Inf. Sci. Eng.*, vol. 33, no. 3, pp. 695–712, 2017. Article (CrossRef Link).

[10] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proc. of Workshop on the theory and application of cryptographic techniques, Springer*, pp. 47-53, 1984. Article (CrossRef Link).
[11] J. Malone-Lee, "Identity-based signcryption." *IACR Cryptology ePrint Archive*, vol. 2002, p. 98, 2002.
[12] J. Xie, Y.-p. Hu, J.-t. Gao, and W. Gao, "Efficient identity-based signature over ntru lattice," *Frontiers of Information Technology & Electronic Engineering*, vol. 17, no. 2, pp. 135-142, 2016. Article (CrossRef Link).
[13] Z. Qin, C. Yuan, Y. Wang, and H. Xiong, "On the security of two identity-based signature schemes based on pairings," *Information Processing Letters*, vol. 116, no. 6, pp. 416-418, 2016. Article (CrossRef Link).
[14] X. Hu, H. Xu, J. Wang, W. Tan, and Y. Yang, "A generic construction of identity-based proxy signature scheme in the standard model," *International Journal of Information and Computer Security*, vol. 11, no. 1, pp. 83-100, 2019. Article (CrossRef Link).
[15] P. S. Barreto, B. Libert, N. McCullagh, and J.-J. Quisquater, "Efficient and provably-secure identity-based signatures and signcryption from bilinear maps," in *Proc. of International conference on the theory and application of cryptology and information security, Springer*, pp. 515-532, 2005. Article (CrossRef Link).
[16] A. Karati, S. H. Islam, G. Biswas, M. Z. A. Bhuiyan, P. Vijayakumar, and M. Karuppiah, "Provably secure identity-based signcryption scheme for crowdsourced industrial Internet of things environments," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2904-2914, 2018. Article (CrossRef Link).
[17] X. Zhang, C. Xu, and J. Xue, "Efficient multi-receiver identity-based signcryption from lattice assumption," *International Journal of Electronic Security and Digital Forensics*, vol. 10, no. 1, pp. 20-38, 2018. Article (CrossRef Link).
[18] L. Chen and J. Malone-Lee, "Improved identity-based signcryption," in *Proc. of International Workshop on Public Key Cryptography, Springer*, pp. 362-379, 2005. Article (CrossRef Link).
[19] S. S. Chow, S.-M. Yiu, L. C. Hui, and K. Chow, "Efficient forward and provably secure id-based signcryption scheme with public verifiability and public ciphertext authenticity," in *Proc. of International Conference on Information Security and Cryptology, Springer*, pp. 352-369, 2003. Article (CrossRef Link).
[20] B. Libert and J.-J. Quisquater, "A new identity based signcryption scheme from pairings," in *Proc. of Information Theory Workshop*, pp. 155–158, 2003. Article (CrossRef Link).
[21] X. Li and K. Chen, "Identity based proxy-signcryption scheme from pairings," in *Proc. of Services Computing, 2004.(SCC 2004). Proceedings. 2004 IEEE International Conference on. IEEE, 2004*, pp. 494–497, 2004. Article (CrossRef Link)
[22] S.-X. Chen, S.-X. Zhou, X.-F. Yao, and F.-W. Li, "Efficient identity-based proxy signcryption scheme," *Application Research of Computers*, vol. 7, p. 084, 2011.
[23] Y. Ming, J. Feng, and J. Hu Q, "Secure identity-based proxy signcryption scheme in standard model," *Journal of Computer Applications*, vol. 34, no. 10, pp. 2834–2839, 2014.
[24] C.-X. Zhou, "Identity-based generalized proxy signcryption scheme," *Information Technology and Control*, vol. 45, no. 1, pp. 13-26, 2016. Article (CrossRef Link).
[25] H. Yu, Z. Wang, J. Li, and X. Gao, "Identity-based proxy signcryption protocol with universal composability," *Security and Communication Networks*, vol. 2018, 11 pages, 2018. Article (CrossRef Link).
[26] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signatures: Delegation of the power to sign messages," *IEICE transactions on fundamentals of electronics, communications and computer sciences*, vol. 79, no. 9, pp. 1338–1354, 1996.
[27] S. Kim, S. Park, and D. Won, "Proxy signatures, revisited," in *Proc. of International Conference on Information and Communications Security, Springer*, pp. 223-232, 1997. Article (CrossRef Link).
[28] B. Lee, H. Kim, and K. Kim, "Strong proxy signature and its applications," *Proceedings of SCIS*, vol. 2001, pp. 603–608, 2001.

[29] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *Proc. of Advances in Cryptology CRYPTO86, Springer*, pp. 186-194, 1986. Article (CrossRef Link).
[30] U. Feige, A. Fiat, and A. Shamir, "Zero-knowledge proofs of identity," *Journal of cryptology*, vol. 1, no. 2, pp. 77-94, 1988. Article (CrossRef Link).
[31] F. Zhang and K. Kim, "Id-based blind signature and ring signature from pairings," in *Proc. of International Conference on the Theory and Application of Cryptology and Information Security, Springer*, pp. 533-547, 2002. Article (CrossRef Link).
[32] K. G. Paterson, "Id-based signatures from pairings on elliptic curves," *Electronics Letters*, vol. 38, no. 18, pp. 1025-1026, 2002. Article (CrossRef Link).
[33] N. P. Smart, "Identity-based authenticated key agreement protocol based on weil pairing," *Electronics letters*, vol. 38, no. 13, pp. 630-632, 2002. Article (CrossRef Link).
[34] M. C. Gorantla, R. Gangishetti, and A. Saxena, "A survey on id-based cryptographic primitives." *IACR Cryptology ePrint Archive*, vol. 2005, p. 94, 2005.
[35] J. C. Choon and J. H. Cheon, "An identity-based signature from gap diffie-hellman groups," in *Proc. of International workshop on public key cryptography, Springer*, pp. 18-30, 2003. Article (CrossRef Link).
[36] Q. Wang and Z. Cao, "Efficient id-based proxy signature and proxy signcryption form bilinear pairings," in *Proc. of International Conference on Computational and Information Science, Springer*, pp. 167-172, 2005. Article (CrossRef Link).
[37] D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures," *Journal of Cryptology*, vol. 13, no. 3, pp. 361-396, 2000. Article (CrossRef Link).
[38] X. Cao, W. Kou, and X. Du, "A pairing-free identity-based authenticated key agreement protocol with minimal message exchanges," *Information Sciences*, vol. 180, no. 15, pp. 2895-2903, 2010. Article (CrossRef Link).
[39] S. Namasudra, P. Roy, B. Balusamy, and P. Vijayakumar, "Data accessing based on the popularity value for cloud computing," in *Proc. of 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), IEEE*, pp. 1-6, 2017. Article (CrossRef Link).
[40] S. Namasudra, P. Roy, P. Vijayakumar, S. Audithan, and B. Balusamy, "Time efficient secure dna based access control model for cloud computing environment," *Future Generation Computer Systems*, vol. 73, pp. 90-105, 2017. Article (CrossRef Link).
[41] P. Vijayakumar, S. M. Ganesh, L. J. Deborah, S. H. Islam, M. M. Hassan, A. Alelaiwi, and G. Fortino, "Mgpv: A novel and efficient scheme for secure data sharing among mobile users in the public cloud," *Future Generation Computer Systems*, vol. 95, pp. 560-569, 2019. Article (CrossRef Link).
[42] H.Xiong, Y.Zhao, L.Peng, H.Zhang, and K.H.Yeh, "Partially policy-hidden attribute-based broadcast encryption with secure delegation in edge computing," *Future Generation Computer Systems*, vol.97, pp. 453-461, 2019. Article (CrossRef Link).

Negalign Wake Hundera is a Ph.D. Candidate at the University of Electronic Science and Technology China (UESTC) since 2017. He obtained his MSc degree in Computer Science and technology from the University of Electronic Science and Technology China (UESTC), Chengdu, China, in 2016 and a Bachelor degree in Information Technology from Jimma University, Jimma, Ethiopia in 2009. His research interest includes network Security, cryptographic protocols, information security, and cloud computing.

Qian Mei is currently pursuing her Ph.D. degree from the School of Information and Software Engineering, University of Electronic Science and Technology of China. She received her B.S. degree from Jiangxi University of Science and Technology in 2017. Her research interests include certificateless public key cryptography and privacy-preserving.

Hu Xiong received the Ph.D. degree from the University of Electronic Science and Technology of China (UESTC) in 2009. He is currently a full Professor at UESTC. His research interests include cryptography and ad hoc network security.

Dagmawit Mesfin Geresu is currently pursuing her bachelor degree in Electronic Information Engineering from the School of Information and Communication Engineering, University of Electronic Science and Technology of China (UESTC).