An Identity Based Proxy Signcryption Scheme without Pairings

Hui Guo and Lunzhi Deng
(Corresponding author: Lunzhi Deng)

School of Mathematical Sciences, Guizhou Normal University
Guiyang 550001, China
(Email: denglunzhi@163.com)

(Received Dec. 25, 2018; Revised and Accepted Aug. 3, 2019; First Online Feb. 9, 2020)

Abstract

The identity-based cryptography avoids the storage problem of public key certificate of public key infrastructure. The signcryption mechanism completes both authentication and encryption functions with lower communication cost. The proxy signature allows the proxy signer to sign a message on the behalf of the original signer. In this paper, a new identity based proxy signcryption (IBPS) scheme without pairings is proposed, and it is proved to be secure in the random oracle model. To the best of our knowledge, our scheme is more efficient than previous ones in computation.

Keywords: Identity Based Cryptography; Proxy Signcryption; Random Oracle Model

1 Introduction

Traditional public key cryptography [11] needs a trusted certification authority (CA) to issue a certificate which links the identity and the public key of the user. Hence, the problem of certificate management arises. To solve the problem, the notion of the identity-based public key cryptography was introduced by Shamir [20] in 1984. In this cryptography, a user's public key can be arbitrary string that can identify the user, such as the e-mail address or telephone number and so on.

In the areas of computer communications and electronic transactions, one of the essential topics is how to send data in confidential and authentication way. In 1997, Zheng [28] proposed a novel cryptographic primitive, called signcryption [21] that satisfies both the functionality of digital signature and encryption in a single logical step.

The proxy signature [9, 26] is a useful tool in real life. For example, if a document is to be signed by a CEO (original signer) of the company while he/she is absent, then the document can be signed by a manager (proxy signer) designated by the CEO (original signer) [12, 17]. The proxy signature was firstly introduced by Mambo *et al.* [19] in 1996. It allows the proxy signer to sign a message on the behalf of the original signer. On the basis of the deledation type, the proxy signature is calssified into three types: Full delegation, partial delegation and delegation by warrant. Because the first two types have some drawbacks [3], most proxy signature schemes has focused on the type of the delegation with warrant.

To delegate the signcryption righs to a trusted agent, Gamage *et al.* [4] proposed a new ideal of proxy signcryption by combining the notions of proxy signature and signcryption in 1999. But their scheme does not support provable security [22]. In 2004, Li and Chen [13] proposed the first identity-based proxy signcryption scheme using bilinear pairings.

1.1 Related Work

Many researchers have been proposed variations of signcryption schemes. Arijit Karati *et al.* [10] designed a practical identity based signcryption scheme from bilinear pairing, which is based on CDH assumption and proved to be secure under standard security model. An identity-based signcryption scheme that is forward secure in a stronger sense was proposed by Madeline González Mu$\widetilde{n}$iz *et al.* [18].

Deng *et al.* [3] proposed an identity based proxy signature from RSA without pairings in the randon oracle model that admits formal proofs for unforgeability of proxy signature. He *et al.* [7] introduced an ID-based proxy signature schemes without bilinear pairings, which is secure aginst adaptive chosen message and ID attack. In 2016, Hu *et al.* [5] presented a proxy signature scheme with a formal security proof based on the CDH and BDH assumption.

Since identity-based proxy signcryption (IBPS) plays an important role in practical applications such as mobile communication and e-commerce and so on, it has attracted great attention when it was proposed, and has been studied by many scholars at home and abroad. Wu Jian [27] proposed an identity-based proxy signcryption schemes. Li and Chen [13] proposed an identity based

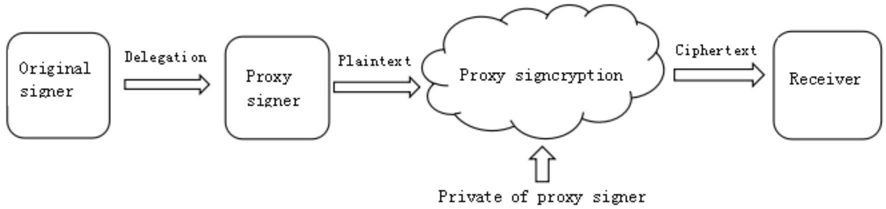

Figure 1: Process of a IBPS scheme

proxy signcryotion scheme which is based on the Libert and Quisquater's [14] identity based signcryption scheme. But Wang *et al.* [25] point that the scheme does not satisfy the strong unforgeability security in the strict sense. Saraswat [22] proposed a secure proxy signcryption scheme which provides anonymity to the proxy signer from the receiver.

Swapna *et al.* [23] introduced an efficient ID-based proxy signcryption scheme, which offers both public verifiability and forward security. Lin *et al.* [15] introduced an efficient proxy signcryption with provable CCA and CMA security. Unfortunately, Lo and Tsai [16] pointed that the scheme is not secure against the chosen warrant attack. Other schemes proposed including proxy blind signcryption [24], generalized proxy signcryption [29]– [30], certificateless proxy signcryption [2], *etc.*

1.2 Our Contributions

In this paper, we propose a new identity based proxy signcryption scheme. The main contributions of this paper are as follows:

1) The proposed scheme is proved to be secure in the random oracle model.

2) The proposed scheme does not use pairing operation, which is more efficient than that of previous schemes [13, 16, 23, 25, 27] in computation.

2 Preliminaries

Definition 1. *Given a generator P of group G with prime order q, and a tuple $(P, aP, bP, X \in G)$ for unknown $a, b \in z_q^*$, the Decisional Diffie-Hellman problem (DDH) is to decide whether $X = abP$.*

Definition 2. *Given a generator P of group G with prime order q, and a tuple (P, aP), the Discrete Logarithm problem (DLP) is to compute a.*

2.1 Model of Identity based Proxy Signcryption

An identity based proxy signcryption scheme is composed of six polynomial time algorithms, it is defined as follows:

- Setup: Input a security parameter k, private key generator (PKG) outputs the system parameters $params$ and a master secret key msk.

- Private-Key-Extract: Input the system parameters $params$, the master secret key msk and the identity $ID_i \in \{0,1\}^*$ of a user, PKG returns a private key s_i to the user ID_i via a secure channel, and the user publish its public key R_i.

- Delegation Generate: Input the system parameters $params$, the private key s_A of original signer ID_A and a warrant w, this algorithm outputs a delegation π and sends π to the proxy signer ID_B.

- Delegation Verify: This algorithm takes as input the system parameters $params$, delegation π, and verifies whether π is a valid delegation from the original signer ID_A.

- Proxy Signcryption: Input the private key s_B of proxy signer ID_B, the receiver identity ID_C, a message m and a delegation π, this algorithm outputs a proxy signcryption ciphertext σ on behalf of the original signer ID_A.

- Proxy Unsigncryption: After receiving the ciphertext σ, the receiver ID_C decrypts the ciphertext and obtains the message m or the symbol $\perp$ if σ is a invalid ciphertext.

Definition 3. *An identity based proxy signcryption scheme is said to be indistinguishable under adaptive chosen ciphertext attacks if the polynomially bounded adversary with a negligible advantage in the following game.*

Game I. A challenger $\mathscr{C}$ and a adversary $\mathscr{A}$ play the following game.

Initialization. $\mathscr{C}$ runs the setup algorithm to generate a master secret key msk and the public system parameters $params$. $\mathscr{C}$ sends $params$ to $\mathscr{A}$ and keeps msk secret.

Phase 1. $\mathscr{A}$ makes a polynomially bounded number of adaptive queries to $\mathscr{C}$.

- Hash functions query: $\mathscr{A}$ can ask for the values of any hash functions.
- private key query: $\mathscr{A}$ chooses an identity ID_i, $\mathscr{C}$ runs the private key extraction algorithm to generate private key s_i, and sends to $\mathscr{A}$.

- Delegation query: When $\mathscr{A}$ submits the identity of original signer ID_A and a warrant w to the challenger $\mathscr{C}$, $\mathscr{C}$ responds the corresponding delegetion π to $\mathscr{A}$.
- Proxy Signcryption query: $\mathscr{A}$ chooses a message m, a receiver ID_C and the private key s_B of proxy signer ID_B, a delegetion π, and sends to $\mathscr{C}$. $\mathscr{C}$ returns the proxy signcryption ciphtext σ to $\mathscr{A}$.
- Proxy Unsigncryption query: When $\mathscr{A}$ chooses a ciphertext σ, a receiver's identity ID_C and a proxy signer ID_B, $\mathscr{C}$ outputs plaintext m generated by the proxy unsigncryption algorithm. Or $\mathscr{C}$ returns the the symbol $\perp$, if σ is an invalid proxy unsigncryption ciphertext.

Challenge. $\mathscr{A}$ sends following information to the challenger: two equal length messages m_0, m_1, a specified receiver ID_C and proxy signer ID_B, $\mathscr{C}$ takes randomly a bit $\mu \in \{0,1\}$ and computes the ciphertext σ^* on the message m_μ.

($\mathscr{A}$ should not have requested the private key for ID_C in Phase 1.)

Phase 2. $\mathscr{A}$ performs a polynomially bounded number of queries just like in phase 1, and fulfills the following restrictions:

1) $\mathscr{A}$ should not have requested the private key for ID_C.
2) $\mathscr{A}$ can not have made the proxy unsigncryption query for the ciphertext σ^*.

Response. $\mathscr{A}$ produces a bit μ' and wins the game if $\mu' = \mu$. The advantage of $\mathscr{A}$ is defined as: $Adv_{\mathscr{A}}^{IND-CLRSC}(\nu) = |2\Pr[\mu' = \mu] - 1|$.

Definition 4. *An identity based proxy signcryption scheme is said to be unforgeable under adaptive chosen message attacks if the polynomially bounded adversary with a negligible advantage in the following game.*

Game II. A challenger $\mathscr{C}$ and a adversary $\mathscr{A}$ play the following game:

Initialization, Query. Same as that in the Game I.

Forge. $\mathscr{A}$ produces a tuple $\{ID_A, ID_B, \pi\}$ or $(\sigma, w, ID_A, ID_B, ID_C)$. When one of the following conditions hold, $\mathscr{A}$ wins the game.

Case 1: The final output is $\{ID_A, ID_B, \pi\}$ and it fulfills:

1) π is a valid delegation.
2) $\mathscr{A}$ should have not queried the private key of original signer ID_A.
3) π is not obtained by the delegation query.

Case 2: The final output is $(\sigma, w, ID_A, ID_B, ID_C)$ and it fulfills:

1) σ is a proxy signcryption.
2) $\mathscr{A}$ should have not queried the private key of original signer ID_A
3) The tuple $(\pi, ID_A, ID_B)$is not appear in delegation query.
4) σ is not obtained by the proxy signcryption query.

Case 3: The final output is $(\sigma, w, ID_A, ID_B, ID_C)$ and it fulfills:

1) σ is a proxy signcryption.
2) The private key of proxy signer ID_B has not been queried.
3) σ is not obtained by the proxy signcryption query.

The advantage of $\mathscr{A}$ is defined as: $Adv_{\mathscr{A}}^{UNF-IBPS}$ =Pr[$\mathscr{A}$ win].

3 Proposed Scheme

- Setup: Given the security parameter of the system k and l, PKG chooses an additive cyclic group $G = \langle P \rangle$ of prime order $q > 2^k$. Then PKG chooses four hash functions $H_1 : \{0,1\}^* \times G \to Z_q^*$, $H_2 : \{0,1\}^* \times G \times G \times G \times \{0,1\}^* \times \{0,1\}^* \to Z_q^*$, $H_3 : \{0,1\}^* \to \{0,1\}^l, H_4 : \{0,1\}^* \to Z_q^*$. The PKG randomly chooses its master secret key $x \in Z_q^*$ and computes the public key $P_{pub} = xP$. The message space is $M = \{0,1\}^l$. The PKG publishes the set of public system parameters: $params = \{G, q, P, P_{pub} = xP, H_1, H_2, H_3, H_4\}$ and keep the master key x secret.

- Private-Key-Extract: Given a user's identity $ID_i \in \{0,1\}^*$, the PKG randomly selects $r_i \in Z_q^*$ and computes $R_i = r_iP$, $d_i = H_1(ID_i, R_i)$, $s_i = r_i + d_ix$ and sends (R_i, s_i) to the user via a secure channel. The user ID_i publish his/her the public key R_i.

- Delegation Generation: The original signer ID_A selects at random $t \in Z_q^*$ and computes $T = tP$, $h = H_2(w, T, R_A, R_B, ID_A, ID_B)$, $y = t + hs_A$. Then original signer ID_A sends the delegation $\pi = (T, y, w)$ to proxy signer ID_B securely. Where w is warrant, the warrant includes the property of message to be delegated, the identity information of original signer and proxy signer, the delegation relationship between them and period of delegation, *etc.*

- Delegation Verification: On receiving the delegation $\pi = (T, y, w)$, proxy signer ID_B checks the delegation as follows:

  1) Computes: $h = H_2(w, T, R_A, R_B, ID_A, ID_B)$.

2) Checks if $yP = T + h(R_A + d_A P_{pub})$. If the equality holds, accepts π as a valid delegation. Otherwise, proxy signer ID_B rejects the delegation π.

- Proxy Signcryption: To signcrypt a message m on the behalf of the original signer ID_A for the receiver ID_C, the proxy signer ID_B proceeds as following:
  1) Randomly selects $n_1, n_2 \in Z_q^*$, computes $N_1 = n_1 P$, $N_2 = n_2 P$, $V = n_1(R_C + d_C P_{pub})$, $C = H_3(N_1, N_2, V, R_A, R_B, R_C, ID_A, ID_B, ID_C) \oplus m$;
  2) Computes: $g = H_4(m, \pi, N_1, N_2, V, R_A, R_B, R_C, ID_A, ID_B, ID_C)$, $z = y + n_2 + g s_B$;
  3) Outputs the proxy signcryption: $\sigma = \{C, N_1, N_2, z, \pi\}$.
- Proxy Unsigncryption: On receiving the ciphertext $\sigma = \{C, N_1, N_2, z, \pi\}$, the receiver ID_C decrypts the ciphertext as follows:
  1) Computes: $V = s_C N_1$, $m = C \oplus H_3(N_1, N_2, V, R_A, R_B, R_C, ID_A, ID_B, ID_C)$, $g = H_4(m, \pi, N_1, N_2, V, R_A, R_B, R_C, ID_A, ID_B, ID_C)$.
  2) Checking whether $zP = T + N_2 + h(R_A + d_A P_{pub}) + g(R_B + d_B P_{pub})$. If the equality holds, accepts m as a valid message. Otherwise, the receiver rejects the ciphertext.

4 Analysis of Proposed Scheme

4.1 Correctness Analysis

$$\begin{aligned} V &= n_1(R_C + d_C P_{pub}) \\ &= n_1(r_C P + d_C x P) \\ &= (r_C + d_C x) n_1 P = s_C N_1; \\ yP &= (t + h s_A) P \\ &= tP + h s_A P \\ &= T + h(r_A + d_A x) P \\ &= T + h(r_A P + d_A x P) \\ &= T + h(r_A P + d_A P_{pub}) \\ &= T + h(R_A + d_A P_{pub}); \end{aligned}$$

$$\begin{aligned} zP &= (y + n_2 + g s_B) P \\ &= yP + n_2 P + g s_B P \\ &= T + h(R_A + d_A P_{pub}) + N_2 + g(r_B + d_B x) P \\ &= T + h(R_A + d_A P_{pub}) + N_2 + g(r_B P + d_B x P) \\ &= T + h(R_A + d_A P_{pub}) + N_2 + g(R_B + d_B P_{pub}). \end{aligned}$$

4.2 Security Analysis

Theorem 1. *In random oracle model, the scheme is indistinguishable against the adversary $\mathscr{A}$ if the DDH is hard.*

Proof. Assume that the challenger $\mathscr{C}$ receives a random instance (P, aP, bP, X) of the DDH, the goal of $\mathscr{C}$ is to determine whether $X = abP$ or not. $\mathscr{C}$ runs $\mathscr{A}$ as a subroutine and plays the role of the challenger in the Game I. ☐

Initialization. $\mathscr{C}$ runs the setup algorithm to generate system parameters. Then $\mathscr{C}$ sends the system parameters $params = \{G, q, P, P_{pub} = xP, H_1, H_2, H_3, H_4\}$ to $\mathscr{A}$.

Queries. Without losing generality, assuming that each query is different. $\mathscr{A}$ will ask for $H_1(ID_i)$ before the identity ID_i is used any other queries. $\mathscr{C}$ will maintain some lists to store the queries and answers, all of the lists are initially empty.

- H_1 queries: $\mathscr{C}$ maintains the list L_1 of tuple (ID_i, R_i, d_i). When $H_1(ID_i, R_i)$ is queried by $\mathscr{A}$, $\mathscr{C}$ selects at random $d_i \in Z_q^*$ and sets $H_1(ID_i, R_i) = d_i$, and adds (ID_i, R_i, d_i) to list L_1.
- H_2 queries: $\mathscr{C}$ maintains the list L_2 of tuple (β, h). When $H_2(\beta)$ is queried by $\mathscr{A}$, $\mathscr{C}$ selects at random $h \in Z_q^*$, sets $H_2(\beta) = h$ and adds (β, h) to list L_2.
- H_3 queries: $\mathscr{C}$ maintains the list L_3 of tuple (U, α). When $H_3(U)$ is queried by $\mathscr{A}$, $\mathscr{C}$ selects at random $\alpha \in \{0,1\}^l$, sets $H_3(U) = \alpha$ and adds (U, α) to list L_3.
- H_4 queries: $\mathscr{C}$ maintains the list L_4 of tuple (β', h'). When $H_4(\beta')$ is queried by $\mathscr{A}$, $\mathscr{C}$ selects at random $h' \in Z_q^*$, sets $H_4(\beta') = h'$ and adds (β', h') to list L_4.
- User public key queries: $\mathscr{C}$ maintains the list L_U of tuple (ID_i, R_i). When $\mathscr{A}$ makes this query, $\mathscr{C}$ answers the query as follows:

  At the j^{th} query, $\mathscr{C}$ sets $R_j = aP$. For $i \neq j$, $\mathscr{C}$ selects at random $r_i \in Z_q^*$ and sets $R_i = r_i P$, the query and the respond will be stored in the list L_U.
- private key queries: $\mathscr{C}$ maintains the list L_K of tuple (ID_i, R_i, d_i). When $\mathscr{A}$ makes this query, $\mathscr{C}$ answers the query as follows:
  If $ID_i = ID^*$, $\mathscr{C}$ fails and stops. Otherwise $\mathscr{C}$ finds the tuple (ID_i, R_i, d_i) in list L_1, responds with $s_i = r_i + x d_i$ and adds (ID_i, R_i) to list L_D.
- Proxy Delegation queries: $\mathscr{C}$ answers the query as follows:

  If $ID_A \neq ID^*$, $\mathscr{C}$ give a delegation π by calling the proxy delegation algorithm to answer $\mathscr{A}$. Otherwise, $\mathscr{C}$ does as follows.
  1) Randomly chooses $y, h \in Z_q^*$, computes: $T = yP - h(R_A + d_A P_{pub})$;
  2) Stores the relation: $h = H_2(w, T, R_A, R_B, ID_A, ID_B)$ and adds to the list L_1. If collision occurs, repeats the steps (1)-(2).

3) Outputs the delegation: $\pi = (T, y, w)$.

- Proxy Signcryption queries: When $\mathscr{A}$ selects a message m, proxy signer ID_B and receiver ID_C, $\mathscr{C}$ returns a proxy signcryption as follows:
  If $ID_B \neq ID^*$, $\mathscr{C}$ give a proxy signcryption σ by calling the the proxy signcryption algorithm to answer $\mathscr{A}$. Otherwise, $\mathscr{C}$ does the following steps:

  1) Randomly selects $n_1, n_2, g \in Z_q*$, computes: $N_1 = n_1P$, $N_2 = n_2P - g(R_B + d_BP_{pub})$, $V = n_1(R_C + d_CP_{pub})$, $C = H_3(N_1, N_2, V, R_A, R_B, R_C, ID_A, ID_B, ID_C) \oplus m$;
  2) Computes: $z = y + n_2$;
  3) Stores the relations: g=$H_4(m, w, N_1$,$V, N_2, R_A, R_B, R_C, ID_A, ID_B, ID_C)$. If collision occurs, repeats Steps (1)-(3);
  4) Outputs the proxy signcryption:

$$\sigma^* = \{C, N_1, N_2, z, \pi\}.$$

- Proxy Unsigncryption queries: If $ID_C \neq ID^*$, $\mathscr{C}$ give a message m by calling the proxy unsigncryption algorithm. Otherwise, $\mathscr{C}$ notifies that σ is an invaild ciphertext.

Challenge. $\mathscr{A}$ chooses two equal length messages m_0, m_1, a specified receiver ID_C, and proxy signer ID_B. If $ID_C \neq ID^*$, $\mathscr{C}$ fails and stops. Otherwise, $\mathscr{C}$ picks $\mu \in \{0, 1\}$, and computes ciphertext σ^* on the message M_μ as follows:

1) Randomly selects $b, n_2 \in Z_q^*$, computes: $N_1 = bP$, $N_2 = n_2P$, $V = X + d_Cx \cdot N_1$, $C = H_3(N_1$, N_2, V, R_A, R_B, R_C, $ID_A, ID_B, ID_C) \oplus m$;
2) Computes: $g = H_4(m, \pi, N_1,\ V, N_2, R_A, R_B$, R_C, ID_A, ID_B, $ID_C)$, $z = y + n_2 + gs_B$;
3) Outputs the proxy signcryption ciphertext:

$$\sigma = \{C, N_1, N_2, z, \pi\}.$$

Phase 2. $\mathscr{A}$ makes a polynomially bounded number of queries just like Phase 1. (but $\mathscr{A}$ should not have queried the private key for ID_C and requested the plaintext corresponding to the ciphertext σ^*).

Response. $\mathscr{A}$ outputs $\mu' \in \{0, 1\}$. If $\mu' \doteq \mu$, $\mathscr{C}$ outputs 1. Otherwise, $\mathscr{C}$ outputs 0. If $X = abP$, σ^* is a valid ciphertext. Then $\mathscr{A}$ can distinguishes μ with the advantage ε. So $\Pr[\mathscr{C} \longrightarrow 1|X = abP] = \Pr[\mu^{'} \doteq \mu|X = abP] = \frac{1}{2} + \varepsilon$.

If $X \neq abP$, when $\mu = 0$ or $\mu = 1$, each part of the ciphertext has the same probability distribution, so $\mathscr{A}$ has no advantage in distinguishing μ. So $\Pr[\mathscr{C} \longrightarrow 1|X \neq abP\ ]= \Pr[\mu^{'} \doteq \mu|X \neq abP] = \frac{1}{2}$.

Probability. Let $q_{H_i}(i = 1, 2, 3, 4)$, q_U, q_K, q_D and q_S be the number of $H_i(i = 1, 2, 3, 4)$ queries, public key queries, private key queries, delegating queries and proxy signcryption queries, respectively.

We denotes some events as follows:

- π_1: $\mathscr{C}$ does not fail in private key queries;
- π_2: $\mathscr{C}$ does not fail in proxy unsigncryption queries;
- π_3: $\mathscr{C}$ does not fail in challenge stage.

It is easy to get following results:

$$\begin{aligned}
Pr[\pi_1] &= 1 - \frac{q_K}{q_U}, \\
Pr[\pi_2] &= 1 - \frac{1}{2^k}, \\
Pr[\pi_3] &= \frac{1}{q_U - q_K}. \\
\Pr[\mathscr{C}\ success] &= \Pr[\pi_1 \wedge \pi_2 \wedge \pi_3] \\
&= \Pr[\pi_1] \cdot \Pr[\pi_2] \cdot \Pr[\pi_3] \\
&= (1 - \frac{q_K}{q_U}) \cdot (1 - \frac{1}{2^k}) \cdot \frac{1}{q_U - q_K} \\
&\approx \frac{1}{q_U}
\end{aligned}$$

Therefore, if $\mathscr{A}$ can succeed with the probability ε, then $\mathscr{C}$ can solve the DDH with probability $\frac{\varepsilon}{q_U}$.

Theorem 2. *In random oracle model, the scheme is unforgeable against adversary $\mathscr{A}$ if the DLP is hard.*

Proof. Assume that the challenger $\mathscr{C}$ receives a random instance (P, aP) of the DLP. the goal of $\mathscr{C}$ is to compute the value of a. $\mathscr{C}$ will run $\mathscr{A}$ as a subroutine and play the role of challenger in the Game II. □

Initialization, Query. Same as that in the Game II.

Forge. $\mathscr{A}$ outputs a tuple $\{\pi = \{T, y, w\}, ID_A\}$ or $\{\sigma = (C, N_1, N_2, z, \pi), ID_A, ID_B, ID_C\}$. There are three situations to consider:

Case 1. The final output is $\{\pi = \{T, y, w\}, ID_A\}$ and the output fulfills the demande of Case 1 as defined in the game.

Solve DLP. Using the forking lemma for generic signature scheme [1], after replays $\mathscr{A}$ with the same random tape except the λ^{th} result returned by H_2 query of the forged message, $\mathscr{C}$ gets two valid proxy signcryptions: $\{T, y, w\}$ and $\{T, y', w\}$. Where $h = H_2(w, T, R_A, R_B, ID_A, ID_B)$, $h' = H_2'(w, T, R_A, R_B, ID_A, ID_B)$, $h \neq h'$. If $ID_A = ID^*$, $\mathscr{C}$ solves DLP by computing: $a = (h' - h)^{-1}(y' - y) - d_Ax$.

Probability. Let $q_{H_i}(i = 1, 2, 3, 4)$, q_U, q_K, q_D and q_S be the number of $H_i(i = 1, 2, 3, 4)$ queries, public key queries, private key queries, delegating queries and proxy signcryption queries, respectively.

We denote some events as follows: π_1: $\mathscr{C}$ does not fail during the queries; π_2: $\mathscr{C}$ does not fail in proxy unsigncryption queries. π_3: $ID_A = ID^*$.

It is easy to get following results:

$$\begin{aligned}
\Pr[\pi_1] &= \frac{q_U - q_K}{q_U}, \\
\Pr[\pi_2|\pi_1] &= 1 - \frac{1}{2^k}, \\
\Pr[\pi_3] &= \frac{1}{q_U - q_K}. \\
\Pr[\mathscr{C}\ success] &= \Pr[\pi_1 \wedge \pi_2 \wedge \pi_3] \\
&= \Pr[\pi_1] \cdot \Pr[\pi_2|\pi_1] \cdot \Pr[\pi_3] \\
&= \frac{q_U - q_K}{q_U} \cdot (1 - \frac{1}{2^k}) \cdot \frac{1}{q_U - q_K} \\
&\approx \frac{1}{q_U}
\end{aligned}$$

Therefore, if $\mathscr{A}$ can succeed with the probability ε, then $\mathscr{C}$ can solve DLP with the probability $\frac{\varepsilon}{q_U}$.

Case 2. The final output is $\{\sigma = (C,\ N_1, N_2,\ z, \pi),\ ID_A, ID_B, ID_C\}$ and the output fulfills the demand of Case 2 as defined in Game II.

Solve DLP. Using the forking lemma for generic signature Scheme [1], after replays $\mathscr{A}$ with the same random tape except the result returned by H_2 query of the forged message, $\mathscr{C}$ gets two valid proxy signcryptions: $\{C, N_1,\ N_2, z, \pi = (T, y, w)\}$ and $\{C, N_1, N_2, z,\ \pi' = (T, y, w)\}$. Where $h = H_2(w,\ T, R_A, R_B,\ ID_A, ID_B)$, $h' = H_2'(w, T, R_A, R_B, ID_A, ID_B)$, $h \neq h'$. $g = g' = H_4(m, \pi, N_1, V, N_2, R_A, R_B, R_C, ID_A, ID_B, ID_C)$. If $ID_A = ID^*$, $\mathscr{C}$ solves DLP by computing: $a = (h' - h)^{-1}(y' - y) - d_A x$.

Probability. Probability of success is same as the probability in Case 1.

Case 3. The final output is $\{\sigma = (C, N_1, N_2, z, w), ID_A, ID_B, ID_C\}$ and the output fulfills the demand of Case 3 as defined in Game II.

Solve DLP. Using the forking lemma for generic signature Scheme [1], after replays $\mathscr{A}$ with the same random tape except the result returned by H_4 query of the forged message, $\mathscr{C}$ gets two valid proxy signcryptions: $\{C, N_1,\ N_2,\ z,\ \pi\}$ and $\{C,\ N_1,\ N_2,\ z', \pi\}$. Where $g = H_4(m,\pi, N_1, V, N_2, R_A, R_B, R_C, ID_A, ID_B, ID_C)$, $g' = H_4'(m, \pi, N_1, V, N_2, R_A, R_B, R_C, ID_A, ID_B, ID_C)$, $g \neq g'$. If $ID_c = ID^*$, $\mathscr{C}$ solves DLP by computing: $a = (g' - g)^{-1}(z' - z) - d_B x$.

Probability. Probability of success is same as the probability in Case 1.

5 Efficiency and Comparison

By using a famous encryption library (MIRACL) on a mobile device (Samsung Galaxy S5 with a Quad-core 2.45G processor, 2G bytes memory and the Google Android 4.4.2 operating system), He *et al.* [7] obtained the running time for cryptographic operations. The running time are listed in Table 1.

For the IBPS scheme based on biliner pairing, to achieve the 1024 bits RSA level security, a Tate pairing $G_1 \times G_1 \longrightarrow G_2$ defined over the supersigingular elliptic curve E/F_p: $y^2 = x^3 + x$ was used, where both q and p are 160 bits and 512 bits, respectively. To achieve the same level of scurity, for the IBPS scheme based on the non-singular elliptic curve cryptography, they used an additive group with the prime order q, which is defined on a non-sigular elliptic curve over the finite field F_p, where both p and q are 160 bits. We define some notations as follows:

P: A pairing operation.

M_{G_1}: A scalar multiplication operation in G_1.

M_G: A scalar multiplication operation in G.

E_{G_2}: A exponentiation operation in G_2.

We use a simple method to evaluate the computaion efficiency of the different schemes. For example, the scheme [25] needs 13 pairing operations, 4 scalar multiplication operation in G_1, 7 exponentiation operations in G_2. Therefore, the resulting operation time is $13 \times 32.713 + 4 \times 13.405 + 7 \times 2.249 = 494.632$.

According to the above ways, the resulting operation time of other shemes [13,16,23,25,27] is shown in Table 2.

Table 1: Cryptographic operation time (in milliseconds)

P	M_{G_1}	M_G	E_{G_2}
32.713	13.405	3.335	2.249

6 Conclusion

Although several good results have been achieved in speeding up the computation of bilinear pairing function in recent years. The pairing operation is still relatively expensive and the relative computation cost of the pairing is approximately twenty times higher than that of scalar multiplication over elliptic curve group. So it is still quite significant to design cryptography scheme with less pairing operation. In order to save the running time, in the letter, we constructe an identity based proxy signcryption without bilinear pairings. With the running time being saved greatly, as far as my knowledge is concerned, our scheme is more effective than the previous related schemes in computation.

Table 2: Comparison of several IBPS schemes

Schemes	Delegate	D-Verify	Proxy signcryption	P-unsigncryption	Time
Wu [27]	$2M_{G_1}$	2P+M_{G_1}	$P+2M_{G_1}+E_{G_2}$	$2P+E_{G_2}$	235.088
Wang [25]	$3M_{G_1}$	$3P+E_{G_2}$	$2P+M_{G_1}+2E_{G_2}$	$8P+4E_{G_2}$	494.632
Swapna [23]	$2M_{G_1}$	$2P+M_{G_1}$	$P+2M_{G_1}+E_{G_2}$	$3P+2M_{G_1}$	292.362
Lo [16]	M_{G_1}	$2M_{G_1}$	$P+4M_{G_1}$	$3P+5M_{G_1}$	291.712
Li [13]	$3M_{G_1}$	$3P+E_{G_2}$	$2P+2M_{G_1}+2E_{G_2}$	$8P+4E_{G_2}$	508.037
Our scheme	M_G	$3M_G$	$4M_G$	$6M_G$	46.69

Acknowledgments

The authors are grateful to the anonymous referees for their helpful comments and suggestions. The research is supported by the National Natural Science Foundation of China under Grants 61562012, the Innovation Group Major Research Projects of Department of Education of Guizhou Province under Grant No. KY[2016]026.

References

[1] M. Bellare and G. Neven, "Multi-signatures in the plain public key model and a general forking lemma," in *Proceedings of 13th ACM Conference on Computer and Communications Security (CCS'06)*, pp. 390–399, 2006.

[2] T. Bhatia and A. K. Verma, "Cryptanalysis and improvement of certificateless proxy signcryption scheme for e-prescription system in mobile cloud computing," *Annals of Telecommunications*, vol. 72, no. 9–10, pp. 563–576, 2017.

[3] L. Deng, H. Huang and Y. Qu, "Identity based proxy signature from RSA without pairings," *International Journal of Network Security* , vol. 19, no. 2, pp. 229–235, 2017.

[4] C. Gamage, J. Leiwo and Y. Zheng, "An efficient scheme for secure message transmission using proxy-signcryption," in *Proceeding of 22nd Australasian Computer Science Conference (ACSC'99)*, pp. 420–431, 1999.

[5] X. Hu, W. Tan, H. Xu and J. Wang, "Short and provably secure designated verifier proxy signature scheme," *IET Information Security* , vol. 10, no. 2, pp. 69–79, 2013.

[6] D. He, H. Wang, L. Wang, J. Shen and X. Yang, "Efficient certificateless anonymous multi-receiver encryption scheme for mobile devices," *Soft Computing*, vol. 21, no. 22, pp. 6801–6810, 2017.

[7] D. He, J. Chen and J. Hu, "An ID-based proxy signature scheme without bilinear pairings," *Annual Telecommunications*, vol. 66, no. 11–12, pp. 657–662, 2011.

[8] Y. Huang and J. Yang, "A novel identity-based signcryption scheme in the standard model," *Information*, vol. 8, no. 2, pp. 58, 2017.

[9] M. S. Hwang, S. F. Tzeng, C. S. Tsai, "Generalization of proxy signature based on elliptic curves", *Computer Standards & Interfaces*, vol. 26, no. 2, pp. 73–84, 2004.

[10] A. Karati and G. P. Biswas, "A practical identity based signcryption scheme from bilinear pairing," *Advances in Computing*, pp. 832–836, 2016.

[11] A. V. N. Krishna, A. H. Nareyana, K. M. Vani, "Window method based cubic spline curve public key cryptography," *International Journal of Electronics and Information Engineering*, vol. 4, no. 2, pp. 94–102, 2016.

[12] L. H. Li, S. F. Tzeng, M. S. Hwang, "Generalization of proxy signature based on discrete logarithms", *Computers & Security*, vol. 22, no. 3, pp. 245–255, 2003.

[13] X. Li and K. Chen, "Identity based proxy signcryption scheme from pairings," in *Proceedings of the IEEE International Conference on Services Computing (SCC'04)*, pp. 494–497, 2004.

[14] B. Libert and J. Quisquater, "A new identity based signcryption schemes from pairings," in *Proceedings of the IEEE International Theory Workshop*, pp. 155–158, 2003.

[15] H. Lin, T. Wu, S. Huang and Y. S. Yeh, "Efficient proxy signcryption schemes with provable CCA and CMA security," *Computers and Mathmatics with Applications*, vol. 60, no. 7, pp. 1850–1858, 2010.

[16] N. Lo and J. Tsai, "A provably secure proxy signcryption scheme using bilinear pairings," *Journal of Applied Mathematics* , vol. 2014, pp. 10, 2014. (http://dx.doi.org/10.1155/2014/454393)

[17] E. J. L. Lu, M. S. Hwang, and C. J. Huang, "A new proxy signature scheme with revocation", *Applied Mathematics and Computation*, vol. 161, no. 3, PP. 799-806, Feb. 2005.

[18] M. G. Mu$\tilde{n}$iz and P. Laud, "Strong forward security in identity-based signcryption," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 16, no. 4–5, pp. 235–258, 2013.

[19] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signature: Delegtion of power to sign messages," *IEICE Trasactions on Fundamentals*, vol. E79-A, no. 9, pp. 1138–1353, 1996.

[20] A. Shamir, "Identity-based cryptosystem and signature scheme," in *Advances in Cryptology*, pp. 47–53, 1985.

[21] S. Shan, "An efficient certificateless signcryption scheme without random oracles," *International Journal of Electronics and Information Engineering*, vol. 11, no. 1, pp. 9–15, 2019.

[22] V. Saraswat, R. A. Sahu and A. k. Awasthi, "A secure anonymous proxy signcryption scheme," *Jouranl of Mathematical Cryptology*, vol. 11, no. 2, pp. 63–84, 2017.

[23] G. Swapna, P. V. S. S. N. Gopal, T. Gowri and P. V. Reddy, "An efficient ID-based proxy signcryption scheme," *International Jouranl of Information and Network Security*, vol. 3, no. 1, pp. 200–206, 2012.

[24] S. Ullah, M. Junaid, F. Habib, Sana, *et al.*," A novel proxy blind signcryption scheme based on hyper elliptic curve," in *Proceedings of the 12th International Conference on Natural Computation Fuzzy Systems and Knowledge Discovery*, pp. 1964–1968, 2016.

[25] M. Wang, H. Li and Z. Liu, "Efficient identity based proxy-signcryption schemes with forward security and public verifiability," *Networking and Mobile Computing*, pp. 982–991, 2005.

[26] F. Wang, C. C. Chang, C. L. Lin, and S. C. Chang, "Secure and efficient identity-based proxy multisignature using cubic residues," *International Journal of Network Security*, vol. 18, no. 1, pp. 90–98, 2016.

[27] J. Wu, "Identity-based proxy signcryption schemes," *Applied Mechanics and Materials*, vol. 380-384, pp. 2605–2608, 2013.

[28] Y. Zheng, "Digitial signcryption or how to achieve cost (Signature and encryption) cost (Signature) + Cost (Encryption)," *Advances in Cryptology*, pp. 165–179, 1997.

[29] C. Zhou, "Identity based generalized proxy signcryption scheme," *Information Technology and Control*, vol. 45, no. 1, pp. 13–26, 2016.

[30] C. Zhou, "A provable secure identity-based generalized proxy signcryption scheme," *International Journal of Network Security*, vol. 20, no. 6, pp. 1183–1193, 2018.

Biography

Hui Guo received her B.S. from Guizhou Normal University, Guiyang, PR China, in 2016; She is now a master student in the School of Mathematical Sciences, Guizhou Normal University, Guiyang, PR China. Her recent interest include cryptography and information safety.

Lunzhi Deng received his B.S. from Guizhou Normal University, Guiyang, PR China, in 2002; M.S. from Guizhou Normal University, Guiyang, PR China, in 2008; and Ph.D. from Xiamen, PR China, in 2012. He is now a professor in the School of Mathematical Sciences, Guizhou Normal University, Guiyang, PR China. His recent interest include algebra and information safety.