A Provable Secure Identity-based Generalized Proxy Signcryption Scheme

Caixue Zhou, Yue Zhang, and Lingmin Wang
(Corresponding author: Caixue Zhou)

School of Information Science and Technology, Jiujiang University
551 Qianjin Donglu, Jiujiang 332005, China
(Email: charlesjjjx@126.com)

(Received July 8, 2017; revised and accepted Oct. 22, 2017)

Abstract

Generalized proxy signcryption (GPSC) can realize both proxy signature and proxy signcryption with only one key pair and one algorithm, which significantly improves the efficiency of a system with a large number of users, or with limited storage space, or whose functions may be changed. In this paper, we propose an identity-based GPSC scheme in the random oracle model by using bilinear pairings. Our scheme can perform public verification in proxy signcryption mode, resist proxy key exposure attacks, resist insider attacks and support self-delegation. What is more, it needs no secure channel between the original person and the proxy person. Under the adaptive chosen ciphertext, chosen identity and chosen warrant attacks, the confidentiality of our scheme can be reduced to the GBDH hard problem. Under the adaptive chosen message, chosen identity and chosen warrant attacks, the unforgeability of our scheme can be reduced to the GDH' hard problem. We compare our scheme in proxy signcryption mode with other identity-based proxy signcryption schemes that use bilinear pairings, and the results show that it is practical.

Keywords: Bilinear Pairing; Generalized Proxy Signcryption; Proxy Signature; Proxy Signcryption

1 Introduction

In the traditional public key cryptosystem [23], a user's public key is an arbitrary string. Therefore, it needs a trusted third party - certificate authority (CA) to issue a certificate to bind the public key with the user's identity. However, the cost of certificate management is considered to be very high.

The identity-based public key cryptosystem [24, 9] uses an e-mail address or a telephone number *etc.* to represent a user's public key, so there is no need for a public key certificate to bind the public key with the user's identity. In this way, the cost of public key management is greatly reduced.

Signcryption [17] can realize encryption and authentication in a single logic step in an efficient way, so it is very suitable for resource-constrained systems.

Proxy signature [4, 14, 10] allows a designated proxy signer to sign documents on behalf of the original signer when the latter was absent. When the documents must be kept secret, proxy signcryption [18, 16] can be used instead.

Proxy signature and proxy signcryption are two separate cryptographic primitives. If a person is designated by an original person to be both a proxy signer and a proxy signcrypter, he/she must use two algorithms and two key pairs to realize the two functions. If we can use only one algorithm and one key pair, that will save the storage space, simplify the key management and reduce the cost of changing system functions.

In 2016, by reference to the concept of generalized signcription [32], Zhou [31] introduced a new concept of generalized proxy signcryption, which can realize both proxy signature and proxy signcryption with only one key pair and one algorithm. The algorithm can work in two modes - proxy signature mode and proxy signcryption mode. If the receiver's identity is set to null, which is used as the input of the algorithm, the algorithm will run in proxy signature mode; else it will run in proxy signcryption mode. In the same paper, a concrete identity-based GPSC scheme in the standard model was also presented.

In this paper, we propose an identity-based GPSC scheme in the random oracle model by using bilinear pairings. Then, we prove the confidentiality of our scheme in proxy signcryption mode under the GBDH hard assumption and the unforgeability in proxy signature and proxy signcryption modes under the GDH' hard assumption. At last, we compare our scheme in proxy signcryption mode with other identity-based proxy signcryption schemes that use bilinear pairings, and the results show that our scheme is practical.

Our scheme has the following merits. First, it can be verified publicly in proxy signcryption mode. Public verification is very useful in the scenario where the firewall first verifies the validity of the ciphertexts and only allows valid ciphertexts to pass through to the receiver. It prevents the receiver from unnecessarily using their resources to decrypt the invalid ciphertexts. Second, our scheme can resist proxy key exposure attacks [20]. Proxy key is often used in a potentially hostile environment, where it can be easily exposed, but such exposure must not leak any information about the long term private key. Third, our scheme supports self-delegation. Through self-delegation, the user can avoid using the long term private key in some situations, reducing the exposure risk of long term private key. Fourth, there is no need for a secure channel between the original person and the proxy person, which reduces the cost of system implementation. Fifth, our scheme can resist insider attacks. An inside attacker refers to the original person, the proxy person or the receiver. Even with their own private keys, the inside attackers still cannot breach the security of the scheme. Sixth, our scheme is very suitable for a system whose functions may be changed. Consider the following scenario: Previously a system only had the proxy signature function. Due to some reasons, it needs to add the proxy signcryption function. If we use the traditional method, the system must be re-programmed and re-deployed, and everyone in the system will be given a new key pair for the added proxy signcryption function, which will increase the cost of key management and the storage space of the system. However, for our scheme, we only need to let it run in proxy signcrytion mode and the added function will be realized. In this case, the system does not need to be re-programmed and re-deployed, and the total number of keys remains the same. Thus, the cost of key management and the storage space are saved. Reducing the cost of key management is of great practical significance. It is just due to the costly key management that the traditional public key cryptosystem has not been widely applied.

Generalized proxy signcryption can have many practical applications. For example, a person delegates a mobile agent to buy some goods or services on the Internet for himself/herself. For some sensitive messages, the mobile agent can use proxy signcryption, while for others it can use proxy signature. As another example, a general manager of a company delegates his/her signing/signcryption rights to his/her secretary for a period of time when he/she is on vacation. For sensitive messages, the secretary can use proxy signcryption, and for others, she can use proxy signature. Both the mobile agent and the secretary only need to keep one key pair and use one algorithm in the above two examples.

1.1 Related Works

Generalized signcryption was first introduced by Han *et al.* [8] in 2006, in which encryption, signature and signcryption share one key pair and one algorithm for the purpose of saving the storage space of keys and programs, simplifying key management and deployment of the system, and reducing the time spent in verifying the keys. Following Han *et al.*'s work, Wang *et al.* [26] pointed out some security flaws in scheme [8] and improved it, and gave a security model for generalized signcryption scheme for the first time in 2007. Lal and Kushwah [13] first gave the security model of identity-based generalized signcryption and a concrete scheme in 2008. Yu *et al.* [29] pointed out that the security model introduced in scheme [13] is not complete, and gave a new security model and a concrete provably secure scheme in 2010. In the same year, Kushwah and Lal [12] simplified the security model introduced in scheme [29], and proposed a more efficient identity-based generalized signcryption scheme. Han and Gui [7] proposed a multi-receiver generalized signcryption scheme in 2009. Ji *et al.* [11] gave for the first time a certificateless generalized signcryption scheme and security model in 2010. In the same year, Kushwah and Lal [12] pointed out that scheme [11] is insecure and proposed a new scheme. Zhou *et al.* [32] proposed a new certificateless generalized signcryption scheme which is secure against the malicious-but-passive key generation center attacks [1] in 2014. Wei *et al.* [27] proposed an identity-based generalized signcryption scheme in the standard model and applied it in big data security in 2015. In the same year, Zhou [30] pointed out that the multi-receiver generalized signcryption scheme [7] is insecure and improved it and Han *et al.* [6] proposed a generalized signcryption scheme in the attribute-based setting. Shen *et al.* [21] proposed an identity-based generalized signcryption scheme in the standard model in 2017.

The rest of the paper is organized as follows. In Section 2, we introduce the concept of bilinear pairing and some complexity assumptions. In Section 3, we describe the formal definition and security model of GPSC. In Section 4, we propose an efficient identity-based GPSC scheme in the random oracle model. In Section 5, we discuss the security and efficiency of the proposed scheme. We conclude the paper in Section 6.

2 Preliminaries

2.1 Bilinear Pairing

Let $(G_1, +)$ and $(G_2, \cdot)$ be two cyclic groups of prime order q, and g be a generator of G_1. The map $e : G_1 \times G_1 \rightarrow G_2$ is said to be an admissible bilinear pairing if the following three conditions hold.

1) Bilinearity: for all $a, b \in Z_q$, $P, Q \in G_1$, we have $e(aP, bQ) = e(P, Q)^{ab}$.

2) Non-degeneracy: $e(g, g) \neq 1_{G_2}$.

3) Computability: for all $P, Q \in G_1$, there exists an efficient algorithm to compute $e(P, Q)$.

2.2 Complexity Assumptions

1) Bilinear Diffie-Hellman (BDH) Problem: Given $(P, aP, bP, cP) \in G_1^4$ for unknown $a, b, c \in Z_q$, one must compute $e(P, P)^{abc}$. The advantage of any probabilistic polynomial time (PPT) algorithm A in solving the BDH problem in (G_1, G_2, e) is defined to be: $ADV_A^{BDH} = Pr[A(P, aP, bP, cP) = e(P, P)^{abc}, a, b, c \in Z_q]$. BDH assumption: For every PPT algorithm A, ADV_A^{BDH} is negligible.

2) Decisional Bilinear Diffie-Hellman (DBDH) Problem: Given $(P, aP, bP, cP, T) \in G_1^4 \times G_2$ for unknown $a, b, c \in Z_q$, one must decide whether $T = e(P, P)^{abc}$. The advantage of any PPT algorithm A in solving the DBDH problem in (G_1, G_2, e) is defined to be: $ADV_A^{DBDH} = Pr[A(P, aP, bP, cP, T) = 1, a, b, c \in Z_q] - Pr[A(P, aP, bP, cP, e(P, P)^{abc}) = 1, a, b, c \in Z_q]$. DBDH assumption: For every PPT algorithm A, ADV_A^{DBDH} is negligible.

3) Gap Bilinear Diffie-Hellman (GBDH) Problem [2]: Given $(P, aP, bP, cP) \in G_1^4$ for unknown $a, b, c \in Z_q$, one must compute $e(P, P)^{abc}$ with the help of a DBDH oracle. The advantage of any PPT algorithm A in solving the GBDH problem in (G_1, G_2, e) is defined to be: $ADV_A^{GBDH} = Pr[A^o(P, aP, bP, cP) = e(P, P)^{abc}, a, b, c \in Z_q]$, where o denotes a DBDH oracle. GBDH assumption: For every PPT algorithm A, ADV_A^{GBDH} is negligible.

4) Computational Diffie-Hellman (CDH) Problem: Given $(P, aP, bP) \in G_1^3$ for unknown $a, b \in Z_q$, one must compute abP. The advantage of any PPT algorithm A in solving the CDH problem in G_1 is defined to be: $ADV_A^{CDH} = Pr[A(P, aP, bP) = abP, a, b \in Z_q]$. CDH assumption: For every PPT algorithm A, ADV_A^{CDH} is negligible.

5) Gap Diffie-Hellman (GDH') Problem [2]: Given $(P, aP, bP) \in G_1^3$ for unknown $a, b \in Z_q$, one must compute abP with the help of a DBDH oracle. The advantage of any PPT algorithm A in solving the GDH' problem in (G_1, G_2, e) is defined to be: $ADV_A^{GDH'} = Pr[A^o(P, aP, bP) = abP, a, b \in Z_q]$, where o denotes a DBDH oracle. GDH' assumption: For every PPT algorithm A, $ADV_A^{GDH'}$ is negligible.

3 Formal Definition and Security Model of Identity-based Generalized Proxy Signcryption

3.1 Formal Definition

An identity-based GPSC scheme consists of the following six algorithms, involving the original person ID_A, the proxy person ID_P and the receiver ID_R (ID_R may be null):

1) $Setup(1^k)$: Given a security parameter 1^k, the private key generator (PKG) generates a master private key s and a common parameter $params$. $params$ are public to all. PKG keeps the private key s secret.

2) $Extraction(params, s, ID_i)$: On input $params$, the PKG uses s to generate a private key D_i for user ID_i, and then he/she sends it to the user securely.

3) $Delegation(params, D_A, m_w)$: On input $params$, an original person's private key D_A and a warrant m_w (which includes the delegation period, the identities of original person and proxy person and the types of delegated messages, *etc.*), the original person outputs a delegation σ and sends $\{\sigma, m_w\}$ to the proxy person ID_P.

4) $ProxyKey-generation(params, m_w, \sigma, D_P)$: On input $params$, a warrant m_w, a delegation σ and a proxy person's private key D_P, the proxy person produces a proxy key SK_P.

5) $GPSC(params, m, m_w, SK_P, ID_R)$: This algorithm has two modes: proxy signature mode and proxy signcryption mode.

   Proxy-signature mode: If the input to the receiver's identity ID_R is null, it will run in this mode. Other inputs are the message m, the $params$, the warrant m_w and the proxy key SK_P. The proxy signer produces a signature σ_P.

   Proxy-signcryption mode: If the input to the receiver's identity ID_R is not null, it will run in this mode. Other inputs are the message m, the $params$, the warrant m_w and the proxy key SK_P. The proxy signcrypter produces a ciphertext σ_P.

6) UN-GPSC: This algorithm also has two modes: proxy signature verification mode and proxy unsigncryption mode.

   Proxy-signature verification mode ($params$, m_w, σ_P, ID_R): If the input to the receiver's identity ID_R is null, it will run in this mode. Any person can verify the validity of the proxy signature σ_P. If it is correct, the proxy signature will be accepted.

   Proxy-unsigncrytion mode ($params$, m_w, σ_P, ID_R, D_R): If the input to the receiver's identity ID_R is not null, it will run in this mode. The receiver ID_R uses his/her private key D_R to recover the message m or an invalid symbol $\perp$.

For consistency, we require if $\sigma_P = GPSC(params,\ m,\ m_w,\ SK_P,\ ID_R)$, then $UN - GPSC(params,\ m_w,\ \sigma_P)$ = true when ID_R is null or $UN - GPSC(params,\ m_w,\ \sigma_P,\ ID_R,\ D_R) = m$ when ID_R is not null.

3.2 Security Model of Identity-based Generalized Proxy Signcryption

When the scheme run in proxy signcryption mode, it has confidentiality security. The following security model [31] considers proxy key exposure attacks, insider attacks and self-delegation.

Definition 1. *(confidentiality, proxy signcryption mode) An identity-based GPSC scheme is semantically secure against the adaptive chosen ciphertext, chosen identity and chosen warrant attacks (IND-IB-GPSC-CCA for short) in proxy signcryption mode if no PPT adversary A has a non-negligible advantage in the following game:*

Setup: *The challenger C runs the setup algorithm to generate a master private key s and a common parameter params. C gives params to A and keeps s secret.*

Phase 1: *A can make the following polynomially bounded number of queries.*

1) *Extraction queries: A produces an identity ID_i. C runs the extraction algorithm to produce a D_i and returns it to A.*

2) *Delegation queries: A produces a warrant m_w, a proxy identity ID_P and an original identity ID_A. C runs the delegation algorithm to produce a σ and returns (σ, m_w) to A. Here the delegation may be a self-delegation, i.e., the identity ID_P may be equal to the identity ID_A.*

3) *ProxyKey queries: A produces a proxy identity ID_P. C runs the proxy key generation algorithm to produce a SK_P and returns it to A.*

4) *GPSC queries: A produces a message m, a warrant m_w, an original identity ID_A, a proxy identity ID_P and a receiver's identity ID_R. Here if ID_R is null, it is equal to a proxy signature query or else it is equal to a proxy signcryption query. C runs the GPSC algorithm to produce a signature or a ciphertext σ_P to A.*

5) *UN-GPSC queries: A produces a σ_P, a warrant m_w, an original identity ID_A, a proxy identity ID_P and a receiver's identity ID_R. Here if ID_R is null, it is equal to a proxy signature verification query or else it is equal to a proxy un-signcryption query. If it is a proxy signature verification query, C runs the UN-GPSC algorithm to return true or false to A. If it is a proxy un-signcryption query, C runs the UN-GPSC algorithm to return the plaintext m or an invalid symbol $\perp$ to A.*

Challenge: *The attacker A selects two different messages m_0, m_1 with equal length, a warrant m_w^*, and three challenge identities ID_A^*, ID_P^* and ID_R^* (ID_R^* must not be null). Here A has not made the extraction query to identity ID_R^*. C randomly selects a bit $b \in \{0,1\}$, computes the m_b's proxy signcryption ciphertext σ_P^* on $(m_w^*, ID_A^*, ID_P^*, ID_R^*)$, and gives it to A.*

Phase 2: *The attacker A can adaptively make a series of queries as in the Phase 1, but he/she cannot make extraction query of identity ID_R^* nor can he/she make proxy un-signcryption query to σ_P^* under $(m_w^*, ID_A^*, ID_P^*, ID_R^*)$.*

Guess: *When the attacker A wants to end the game, he/she must give his/her guess $b' \in \{0,1\}$. If $b' = b$, he/she wins the game.*

The advantage of the adversary A is defined as: $Adv^{IND-IB-GPSC-CCA}(A) := 2Pr[b' = b] - 1$.

Note 1. *The attacker A is allowed to make a query about the private key of the original person or the proxy person in the Challenge stage, but these inside attackers cannot breach the security of the scheme.*

When the scheme run in proxy signature mode or proxy signcryption mode, it has unforgeability security. In the following security model [31], it considers proxy key exposure attacks, insider attacks and self-delegation.

Definition 2. *(unforgeability, both modes) An identity-based GPSC scheme is existentially unforgeable against the adaptive chosen message, chosen identity and chosen warrant attacks (EUF-IB-GPSC-CMA for short) in proxy signature mode or proxy signcryption mode if no PPT adversary A has a non-negligible advantage in the following game:*

Setup: *Same as in the confidentiality game.*

Attack: *Same as in the confidentiality game.*

Forgery: *If any one of the following events occurs, A wins the game.*

1) *The attacker A outputs a forged delegation σ^* on (ID_A^*, ID_P^*, m_w^*). He/she has not made the delegation query of (ID_A^*, ID_P^*, m_w^*) or extraction query of ID_A^*, and σ^* can pass the delegation verification. Here the identity ID_P^* may be equal to the identity ID_A^* (it means self-delegation).*

2) *The attacker A pretends to be a proxy person ID_P^* to output a forged ciphertext σ_P^* (which may be a proxy signature or proxy signcryption) on (ID_A^*, ID_R^*, m_w^*). The ciphertext σ_P^* is not the output of GPSC query, A has not made extraction query or proxy key query of ID_P^*, and σ_P^* can pass the validation of UN-GPSC.*

   Note 2. *The attacker A is allowed to make a query about the private key of the original person or the receiver (if ID_R^* is not null) in the Forgery stage, but these inside attackers cannot breach the security of the scheme.*

3) *The attacker A pretends to be an original person ID_A^* to output a forged ciphertext σ_P^* (which may be a proxy signature or proxy signcryption)*

on (ID_P^*, ID_R^*, m_w^*). The ciphertext σ_P^* is not the output of GPSC query. A does not make the delegation query with (ID_A^*, ID_P^*, m_w^*), the extraction query with ID_A^* or the proxy key query with ID_P^*, and σ_P^* can pass the validation of UN-GPSC.

Note 3. *The attacker A is allowed to make a query about the private key of the proxy person or the receiver (if ID_R^* is not null) in the Forgery stage, but these inside attackers cannot breach the security of the scheme.*

A's advantage is its probability of victory.

Note 4. *In the above Forgery stage, ID_R^* may be null. If ID_R^* is null, it runs in proxy-signature mode or else it runs in proxy-signcryption mode. So the two modes share the same game.*

4 An Identity-based Generalized Proxy Signcryption Scheme

Based on Yoon *et al.*'s [28] identity-based signature scheme, we give out our identity-based GPSC scheme.

4.1 The Concrete Scheme

Setup: Given a security parameter k, the PKG chooses an additive cyclic group G_1, a multiplicative cyclic group G_2, a generator P of G_1, a bilinear pairing $e : G_1 \times G_1 \rightarrow G_2$ and four secure hash functions $H_1 : \{0,1\}^* \rightarrow G_1^*$, $H_2, H_4 : \{0,1\}^* \rightarrow Z_q^*$, $H_3 : \{0,1\}^* \rightarrow \{0,1\}^m$. m represents the bit length of a message. G_1 and G_2 have the same prime order q. The PKG randomly chooses $s \in Z_q^*$ as the master private key, and computes $P_{pub} = sP$ as the master public key. Let $ID \in \{0,1\}^*$ be an identity of a user. The PKG defines a function $f(ID)$: If $ID \in null$ then $f(ID) = 0$; else $f(ID) = 1$. The PKG publishes the system public parameters as $\{e, G_1, G_2, P, P_{pub}, m, H_1, H_2, H_3, H_4, f\}$, and keeps the master private key s secret.

Extraction: Given an identity ID_i of a user i, the PKG computes the user's private key as $D_i = sH_1(ID_i) = sQ_i$.

Delegation: The original person ID_A first generates a warrant m_w, which records the identities and public keys of original person and proxy person, the type and scope of messages, the time period and so on. ID_A randomly selects $k_A \in Z_q^*$, and computes $R_A = k_A \cdot P$, $h_A = H_2(m_w, R_A)$ and $V_A = h_A D_A + k_A Q_A$. The delegation is $\sigma = (m_w, R_A, V_A)$. ID_A transmits $\sigma = (m_w, R_A, V_A)$ to ID_P publicly. ID_P can verify it through the equation $e(V_A, P) = e(Q_A, h_A P_{pub} + R_A)$. If σ is valid, ID_P accepts it; else σ is re-produced.

Proxy-Key Generation: ID_P randomly selects $k_P \in Z_q^*$, and computes $R_P = k_P \cdot P$, $h_P = H_2(m_w, R_P)$ and $V_P = h_P D_P + k_P Q_P$. At last, the proxy key is $SK_P = V_P + V_A$.

GPSC: Let $M \in \{0,1\}^m$ and $tag \in \{0,1\}$. The proxy person ID_P first computes $f(ID_R)$. If $f(ID_R) = 0$ then $tag = 0$; else $tag = 1$. ID_P randomly selects $t \in Z_q^*$, and computes $R = tP$, $T = e(P_{pub}, Q_R)^{t \cdot tag}$, $h_3 = tag \cdot H_3(R, T, ID_P, Q_P, ID_A, Q_A)$, $C = M \oplus h_3$, $h_4 = H_4(m_w, C, R, ID_R, Q_R)$ and $X = h_4 \cdot SK_P + t(Q_A + Q_P)$. At last, the ciphertext is $\sigma_P = (m_w, R, C, X, R_A, R_P, tag)$.

UN-GPSC:

1) $tag = 0$. $\sigma_P = (m_w, R, C = M, X, R_A, R_P, tag)$ is a proxy signature. Anyone can compute $h_A = H_2(m_w, R_A)$, $h_P = H_2(m_w, R_P)$ and $h_4 = H_4(m_w, C, R, null, null)$, and then verifies the equation $e(X, P) = e(Q_P, h_4 h_P P_{pub} + h_4 R_P + R)e(Q_A, h_4 h_A P_{pub} + h_4 R_A + R)$. If it holds true, ID_P accepts it; else he/she rejects it.

2) $tag = 1$. $\sigma_P = (m_w, R, C, X, R_A, R_P, tag)$ is a proxy signcryption. ID_R can compute $h_A = H_2(m_w, R_A)$, $h_P = H_2(m_w, R_P)$ and $h_4 = H_4(m_w, C, R, ID_R, Q_R)$, and then verifies the equation $e(X, P) = e(Q_P, h_4 h_P P_{pub} + h_4 R_P + R)e(Q_A, h_4 h_A P_{pub} + h_4 R_A + R)$. If it does not hold true, ID_P rejects it; else he/she accepts it and recovers the message $M = C \oplus H_3(R, e(R, D_R), ID_P, Q_P, ID_A, Q_A)$.

4.2 Adaptation

The scheme is an adaptive one and able to switch to two different modes according to the receivers identity ID_R. If the input to the receiver's identity ID_R is null, it will work in proxy signature mode or else it will work in proxy signcryption mode. So the two modes share the same algorithm, so we can use the same key pair to proxy-sign or proxy-signcrypt documents.

5 Analysis of the Proposed Scheme

5.1 Correctness

$$\begin{aligned}
e(X,P) &= e(h_4 \cdot SK_P + t(Q_A + Q_P), P) \\
&= e(h_4 \cdot (h_P D_P + k_P Q_P + h_A D_A + k_A Q_A) \\
&\quad + t(Q_A + Q_P), P) \\
&= e(Q_P, h_4 h_P P_{pub} + h_4 R_P + R) e(Q_A, h_4 h_A P_{pub} \\
&\quad + h_4 R_A + R) \\
T &= e(P_{pub}, Q_R)^t \\
&= e(tP, sQ_R) \\
&= e(R, D_R).
\end{aligned}$$

5.2 Semantic Security

Theorem 1. *In the random oracle model, if there is a PPT attacker A with a non-negligible advantage ε against the IND-IB-GPSC-CCA security of our scheme running in proxy signcryption mode in time T and performing at most q_E extraction queries, q_{SK_P} proxy key queries, q_{DE} delegation queries, q_{GPSC} GPSC queries, $q_{UN-GPSC}$ UN-GPSC queries, q_{H_1} H_1 queries, q_{H_2} H_2 queries, q_{H_3} H_3 queries and q_{H_4} H_4 queries, then the GBDH problem can be solved with probability $\varepsilon' \geq \varepsilon \cdot 1/q_{H_1} \cdot (1 - 1/q_{H_1}) \cdot (1 - q_{UN-GPSC}/2^k)$ in time $T' \leq T + O((q_{GPSC} + q_{UN-GPSC} + q_E + q_{SK_P} + q_{DE}) \cdot t_m + q_{GPSC} \cdot t_e + (q_{GPSC} + q_{UN-GPSC}) \cdot t_p)$, where t_m, t_e and t_p represent the time for a scalar multiplication on G_1, an exponentiation on G_2 and a pairing operation, respectively.*

Proof. Our proof is partially similar to scheme [2]. Challenger C is given $(P, aP, bP, cP) \in G_1^4$ for random $a, b, c \in Z_q^*$. C does not know the values of a, b and c, and is asked to compute $e(P,P)^{abc}$ with the help of a DBDH oracle. To utilize adversary A, challenger C will simulate all the oracles defined in Definition 1. C maintains four lists L_1, L_2, L_3 and L_4, which are initially empty. We assume all queries in the following are distinct and A will ask for $H_1(ID)$ before ID is used in any other queries. In the beginning, C gives the system parameters *params* to A with $P_{pub} = aP$ and he/she randomly selects a number $\theta \in \{1, 2, ..., q_{H_1}\}$.

H_1 **queries:** On the i-th query ID, if $i \neq \theta$, C randomly selects $x \in Z_q^*$ and repeats the process until x is not in list L_1 and sets $Q_{ID} = xP$. Then C stores (i, ID, x) in list L_1 and returns Q_{ID} to A. Otherwise, C stores $(\theta, ID, -)$ in list L_1 and returns $Q_\theta = bP$ to A.

H_2 **queries:** A supplies an item (m_w, R_{ID}). C randomly selects $h_2 \in Z_q^*$ and repeats the process until h_2 is not in list L_2. C stores the item (m_w, R_{ID}, h_2) in list L_2, and returns h_2 to A.

H_4 **queries:** A supplies an item (m_w, C, R, ID_R, Q_R). C randomly selects h_4 and repeats the process until h_4 is not in list L_4. C stores the item $(m_w, C, R, ID_R, Q_R, h_4)$ in list L_4, and returns h_4 to A.

H_3 **queries:** A supplies an item $(R$, T, ID_P, Q_P, ID_A, $Q_A)$. C does the following.

1) C checks if the DBDH oracle returns 1 when queried with the tuple (aP, bP, cP, T). If it does, C returns T and stops.
2) Otherwise, C goes through list L_3 with entries $(R, *, ID_P, Q_P, ID_A, Q_A, h_3)$, so that for different values of h_3, the DBDH oracle returns 1 when queried on the tuple (aP, bP, cP, T). If such a tuple exists, C returns h_3 and replaces the symbol $*$ with T.
3) Otherwise, C randomly selects $h_3 \in \{0,1\}^m$ and repeats the process until h_3 is not in list L_3. C stores the item $(R, T, ID_P, Q_P, ID_A, Q_A, h_3)$ in list L_3, and returns h_3 to A.

Extraction queries: A supplies an identity ID. C searches in list L_1 on ID and obtains (i, ID, x). If $i = \theta$, then C outputs failure and aborts. Otherwise, C returns $x(aP)$.

Delegation queries: A produces a warrant m_w, a proxy identity ID_P and an original identity ID_A.

1) $ID_A \neq ID_\theta$. C runs the delegation algorithm as normal because C can get the private key of ID_A.
2) $ID_A = ID_\theta$. C produces the delegation as follows.
   a. Randomly selects $k, h_A \in Z_q^*$ and computes $V_A = kQ_A$ and $R_A = kP - h_A P_{pub}$.
   b. Saves the item (m_w, R_A, h_A) to list L_2. If a collision occurs in list L_2, C repeats the Step (a).
   c. Outputs the delegation $\sigma = (m_w, R_A, V_A)$ to ID_P.

Here the delegation may be a self-delegation, *i.e.*, the identity ID_P may be equal to the identity ID_A.

Proxy-Key queries: A produces a warrant m_w, a proxy identity ID_P and an original identity ID_A. C first runs delegation query to get V_A.

1) $ID_P \neq ID_\theta$. C runs the proxy-key algorithm as normal because C can get the private key of ID_P.
2) $ID_P = ID_\theta$. C produces the proxy-key as follows.
   a. Randomly selects $k, h_P \in Z_q^*$ and computes $V_P = kQ_P$ and $R_P = kP - h_P P_{pub}$.
   b. Saves the item (m_w, R_P, h_P) to list L_2. If a collision occurs in list L_2, C repeats the Step (a).

c. Outputs the proxy-key $SK_P = V_P + V_A$.

GPSC queries: A produces a message m, a warrant m_w, an original identity ID_A, a proxy identity ID_P and a receiver's identity ID_R. Here if ID_R is null, it is equal to a proxy signature query or else it is equal to a proxy signcryption query. C first makes a proxy-key query to get ID_P's proxy key, then runs the GPSC algorithm as normal to produce a signature or a ciphertext σ_P to A.

UN-GPSC queries: A produces a σ_P, a warrant m_w, an original identity ID_A, a proxy identity ID_P and a receiver's identity ID_R. If ID_R is null, it is equal to a proxy signature verification query or else it is equal to a proxy un-signcryption query. If it is a proxy signature verification query, it just needs public parameters. If it is a proxy un-signcryption query, we consider two cases:

1) $ID_R \neq ID_\theta$. C runs the UN-GPSC algorithm as normal because C can get the private key of ID_R.

2) $ID_R = ID_\theta$. C first runs the verification part of the UN-GPSC algorithm, which just needs public parameters. If the verification does not succeed, C returns $\perp$. Otherwise, it means the verification of the UN-GPSC algorithm holds true. In this situation, C checks if a tuple $(R, T, ID_P, Q_P, ID_A, Q_A, h_3)$ exists in list L_3, so that for some T, the DBDH oracle returns 1 when queried on (aP, bP, R, T). If such a tuple exists, C recovers the message m using the hash value h_3. Otherwise, C randomly selects $h_3 \in \{0,1\}^m$ and repeats the process until h_3 is not in list L_3. C stores the item $(R, *, ID_P, Q_P, ID_A, Q_A)$ in list L_3 and recovers the message m using the hash value h_3. The symbol * denotes an unknown value of pairing.

At last, attacker A selects two different messages M_0, M_1 with equal length, a warrant m_w^*, and three challenge identities ID_A^*, ID_P^* and ID_R^* (ID_R^* must not be null). If $ID_R^* \neq ID_\theta$, C outputs failure and aborts; otherwise C proceeds to construct a challenge as follows. C selects a random $b \in \{0,1\}$ and a random hash $h_3^* \in \{0,1\}^m$, and sets $R^* = cP$ and $C^* = h_3^* \oplus M_b$. Then C makes a delegation query to get (R_A^*, V_A^*), makes a proxy-key query to get (R_P^*, SK_P^*), makes H_4 query on the tuple $(m_w^*, C^*, R^*, ID_R^*, Q_R^* = bP)$ to get h_4^* and computes $X^* = h_4^* \cdot SK_P^* + (x_A^* + x_P^*)(cP)$ ($Q_A^* = x_A^* P, Q_P^* = x_P^* P, x_A^*, x_P^*$ can be obtained from list L_1). At last, the challenge ciphertext is $\sigma_P^* = (m_w^*, R^*, C^*, X^*, R_A^*, R_P^*, tag = 1)$.

In the second stage of the confidentiality game, A can adaptively make a series of queries like before with the restrictions as in Definition 1. At last, A must give his/her guess. A cannot find out that σ_P^* is not a valid ciphertext unless he/she asks for the hash value of $H_3(R^* = cP, T = e(P,P)^{abc}, ID_P^*, Q_P^*, ID_A^*, Q_A^*)$. If this happens, C will solve the GDBH problem due to the first step of H_3 oracle.

Now, we assess the probability of success. In the Challenge stage, the probability that $ID_R^* = ID_\theta$ is $1/q_{H_1}$. The probability of A querying the private key of ID_θ is $1/q_{H_1}$. In the UN-GPSC queries, the probability of C rejecting a valid ciphertext does not exceed $q_{UN-GPSC}/2^k$.

The time complexity of C depends on the scalar multiplication on G_1, the exponentiation on G_2 and pairing operations needed in all the above queries. The extraction queries, delegation queries and proxy-key queries need $O(1)$ scalar multiplications on G_1. The GPSC queries need $O(1)$ scalar multiplications on G_1, $O(1)$ exponentiation on G_2 and $O(1)$ pairings. The UN-GRSC queries need $O(1)$ scalar multiplications on G_1 and $O(1)$ pairings. □

5.3 Unforgeability

Theorem 2. *In the random oracle model, if there is a PPT attacker A with a non-negligible advantage against the EUF-IB-GPSC-CMA security of our scheme running in proxy signature mode or proxy signcryption mode in polynomial time and performing at most q_E extraction queries, q_{SK_P} proxy key queries, q_{DE} delegation queries, q_{GPSC} GPSC queries, $q_{UN-GPSC}$ UN-GPSC queries, q_{H_1} H_1 queries, q_{H_2} H_2 queries, q_{H_3} H_3 queries and q_{H_4} H_4 queries, then the GDH$'$ problem can be solved with a non-negligible advantage in a polynomial time.*

This theorem follows from Lemmas 1, 2 and 3.

Lemma 1. *If A can forge a delegation of our scheme with a non-negligible advantage $\varepsilon \geq 10(q_{H_1}+q_{DE})(q_{DE}+1)/2^k$ in polynomial time t, then the GDH$'$ problem can be solved with probability $\varepsilon' \geq 1/9 \cdot 1/q_{H_1} \cdot (1 - 1/q_{H_1}) \cdot (1 - q_{UN-GPSC}/2^k)$ in time $t' \leq 23 q_{H_2}(t + (q_{GPSC} + 4q_{UN-GPSC})t_e)/\varepsilon$, where t_e represents the time for a pairing operation.*

Lemma 2. *If A can pretend to be a proxy person to forge our scheme (which may be a proxy signature or proxy signcryption) with a non-negligible advantage ε in polynomial time t, then the GDH$'$ problem can be solved with probability $\varepsilon' \geq 1/q_{H_1} \cdot (1 - 1/q_{H_1}) \cdot (1 - q_{UN-GPSC}/2^k) \cdot (\varepsilon^3/(q_{H_2} + q_{H_4})^6 - 3/2^k)\varepsilon$ in time $t' \leq 2t + (q_{GPSC} + 4q_{UN-GPSC})t_e$, where t_e represents the time for a pairing operation.*

Lemma 3. *If A can pretend to be an original person to forge our scheme (which may be a proxy signature or proxy signcryption) with a non-negligible advantage ε in polynomial time t, then the GDH$'$ problem can be solved with probability $\varepsilon' \geq 1/q_{H_1} \cdot (1 - 1/q_{H_1}) \cdot (1 - q_{UN-GPSC}/2^k) \cdot (\varepsilon^3/(q_{H_2} + q_{H_4})^6 - 3/2^k)\varepsilon$ in time $t' \leq 2t + (q_{GPSC} + 4q_{UN-GPSC})t_e$, where t_e represents the time for a pairing operation.*

Proof. (Lemma 1) Suppose challenger C is given $(P, aP, bP) \in G_1^3$ for random $a, b \in Z_q^*$. C does not know the values of a and b, and is asked to compute abP with the help of a DBDH oracle. To utilize adversary A, challenger C will simulate all the oracles defined in Definition 2. C maintains four lists L_1, L_2, L_3 and L_4, which are initially empty. We assume all queries in the following are distinct and A will ask for $H_1(ID)$ before ID is used in any other queries. In the beginning, C gives the system parameters $params$ to A with $P_{pub} = aP$ and randomly selects a number $\theta \in \{1, 2, ..., q_{H_1}\}$.

The H_1, H_2, H_4, extraction, delegation, proxy-key, GPSC and UN-GPSC queries are the same as in Theorem 1.

H_3 queries: A supplies an item $(R, T, ID_P, Q_P, ID_A, Q_A)$. C does the following.

1) C goes through list L_3 with entries $(R, *, ID_P, Q_P, ID_A, Q_A, h_3)$, so that for different values of h_3, the DBDH oracle returns 1 when queried on the tuple (aP, bP, R, T). If such a tuple exists, C returns h_3 and replaces the symbol * with T.

2) Otherwise, C randomly selects $h_3 \in \{0, 1\}^m$ and repeats the process until h_3 is not in list L_3. C stores the item $(R, T, ID_P, Q_P, ID_A, Q_A, h_3)$ in list L_3, and returns h_3 to A.

At last, attacker A outputs a forged delegation σ^* on (ID_A^*, ID_P^*, m_w^*). He/she does not make delegation query with $(ID_A^*, ID_P^*, m_w^*, \sigma^*)$ or extraction query with ID_A^*, and σ^* can pass the delegation verification. Here the identity ID_P^* may be equal to the identity ID_A^* (it means self-delegation). If $ID_A^* \neq ID_\theta$, C outputs failure and stops. Otherwise, we suppose the forged delegation is $\sigma^* = (m_w^*, R_A^*, V_A^*)$. According to the forking lemma [19], we can get two valid delegations $\sigma^* = (m_w^*, R_A^*, V_A^*)$ and $\sigma' = (m_w^*, R_A', V_A')$, so that $V_A^* = h_A^* D_A^* + k_A^* Q_A^*$, $V_A' = h_A' D_A^* + k_A^* Q_A^*$ and $h_A^* = H_2^*(m_w^*, R_A^*) \neq h_A' = H_2'(m_w^*, R_A^*)$. Therefore we can get $V_A^* - V_A' = (h_A^* - h_A')D_A^* = (h_A^* - h_A') \cdot abP$ and $abP = (V_A^* - V_A')(h_A^* - h_A')^{-1}$.

Now, we assess the probability of success. In the forgery stage, the probability that $ID_A^* = ID_\theta$ is $1/q_{H_1}$. The probability of A querying the private key of ID_θ is $1/q_{H_1}$. In the UN-GPSC queries, the probability of C rejecting a valid ciphertext does not exceed $q_{UN-GPSC}/2^k$. Combined with the forking lemma, the probability of C success is $\varepsilon' \geq 1/9 \cdot 1/q_{H_1} \cdot (1 - 1/q_{H_1}) \cdot (1 - q_{UN-GPSC}/2^k)$.

In terms of the time complexity, GPSC needs one pairing operation and UN-GPSC needs four pairing operations. Combined with the forking lemma, the running time of C is $t' \leq 23q_{H_2}(t + (q_{GPSC} + 4q_{UN-GPSC})t_e)/\varepsilon$. □

Proof. (Lemma 2) Suppose challenger C is given $(P, aP, bP) \in G_1^3$ for random $a, b \in Z_q^*$. C does not know the values of a and b, and is asked to compute abP with the help of a DBDH oracle. To utilize adversary A, challenger C will simulate all the oracles defined in Definition 2. C maintains four lists L_1, L_2, L_3 and L_4, which are initially empty. We assume all queries in the following are distinct and A will ask for $H_1(ID)$ before ID is used in any other queries. In the beginning, C gives the system parameters $params$ to A with $P_{pub} = aP$ and randomly selects a number $\theta \in \{1, 2, ..., q_{H_1}\}$.

The H_1, H_2, H_3, H_4, extraction, delegation, proxy-key, GPSC and UN-GPSC queries are the same as in Lemma 1.

At last, attacker A pretends to be a proxy person ID_P^* to output a forged ciphertext σ_P^* (which may be a proxy signature or proxy signcryption) on $(ID_A^*, ID_R^*, m^*, m_w^*)$. The ciphertext σ_P^* is not the output of a GPSC query. A does not make extraction query or proxy key query on ID_P^*, and σ_P^* can pass the validation of UN-GPSC. If $ID_P^* \neq ID_\theta$, C outputs failure and stops. Otherwise, we suppose the forged ciphertext is $\sigma_P^* = (m_w^*, R^*, C^*, X^*, R_A^*, R_P^*, tag)$. According to the multiple-forking lemma [3], with the same inputs and different oracle instance to H_2 and H_4, we can get 4 forged signatures

$$\begin{aligned}
\sigma_P^{(1)} &= (m^*, R^*, X^{(1)}, R_A^*, R_P^*), \\
\sigma_P^{(2)} &= (m^*, R^*, X^{(2)}, R_A^*, R_P^*), \\
\sigma_P^{(3)} &= (m^*, R^*, X^{(3)}, R_A^*, R_P^*), \\
\sigma_P^{(4)} &= (m^*, R^*, X^{(4)}, R_A^*, R_P^*).
\end{aligned}$$

Let $h_P^{(1)}$ and $h_P^{(2)}$ be two different hash values of H_2, and $h_4^{(1)}$, $h_4^{(2)}$, $h_4^{(3)}$ and $h_4^{(4)}$ be four different hash values of H_4. We have

$$\begin{aligned}
X^{(1)} &= h_4^{(1)} \cdot (h_P^{(1)} D_P + k_P Q_P + h_A D_A + k_A Q_A) + t(Q_A + Q_P), \\
X^{(2)} &= h_4^{(2)} \cdot (h_P^{(1)} D_P + k_P Q_P + h_A D_A + k_A Q_A) + t(Q_A + Q_P), \\
X^{(3)} &= h_4^{(3)} \cdot (h_P^{(2)} D_P + k_P Q_P + h_A D_A + k_A Q_A) + t(Q_A + Q_P), \\
X^{(4)} &= h_4^{(4)} \cdot (h_P^{(2)} D_P + k_P Q_P + h_A D_A + k_A Q_A) + t(Q_A + Q_P).
\end{aligned}$$

Thus, we have

$$\begin{aligned}
&((X^{(1)} - X^{(2)})(h_4^{(1)} - h_4^{(2)})^{-1} - (X^{(3)} - X^{(4)})(h_4^{(3)} \\
&\quad - h_4^{(4)})^{-1})(h_P^{(1)} - h_P^{(2)})^{-1} = abP.
\end{aligned}$$

Now, we assess the probability of success. In the forgery stage, the probability that $ID_P^* = ID_\theta$ is $1/q_{H_1}$. The probability of A querying the private key of ID_θ is $1/q_{H_1}$. In the UN-GPSC queries, the probability of C rejecting a valid ciphertext does not exceed $q_{UN-GPSC}/2^k$. Combined with the multiple-forking lemma, the probability of C success is $\varepsilon' \geq 1/q_{H_1} \cdot (1 - 1/q_{H_1}) \cdot (1 - q_{UN-GPSC}/2^k) \cdot (\varepsilon^3/(q_{H_2} + q_{H_4})^6 - 3/2^k)\varepsilon$.

In terms of the time complexity, GPSC needs one pairing operation and UN-GPSC needs four pairing operations. Combined with the multiple-forking lemma, the running time of C is $t' \leq 2t + (q_{GPSC} + 4q_{UN-GPSC})t_e$. □

Proof. (Lemma 3) The proof is similar to that of Lemma 2. □

5.4 Comparison of Performance

We compare our scheme running in proxy signcryption mode with other identity-based proxy signcryption schemes that use bilinear pairings, which include Li *et al.*'s scheme [15], Duan *et al.*'s scheme [5], Wang *et al.*'s scheme [25] and Tian *et al.*'s scheme [22]. The comparison results are listed in Table 1 and Table 2. The pairing computations that can be pre-computed are not included in Table 1. P, M and E represent a pairing computation, a scalar multiplication on G_1 and an exponentiation on G_2, respectively. From Table 1, we can see that scheme [15] and [5] need 8 and 7 pairing computations in the proxy un-signcryption stage, respectively. Therefore, the computational costs of these two schemes are higher than those of the other schemes, indicating they are inefficient. Scheme [25] is the most efficient one in all stages, but it is vulnerable to the proxy key exposure attacks. Once the proxy key is exposed in scheme [25], the original person can compute the private key of the proxy person. Scheme [22] needs 4 pairing computations in the delegation stage, and thus the cost greatly exceeds those of the others, indicating it is inefficient. About the ciphertext size, our scheme is slightly longer. In general, our scheme is one of the efficient schemes.

6 Conclusions

Generalized proxy signcryption can realize both proxy signature and proxy signcryption with only one key pair and one algorithm, which significantly improves the efficiency of a system with a large number of users, or with limited storage space or whose functions may be changed. In this paper, we propose an identity-based GPSC scheme in the random oracle model by using bilinear pairings. Our scheme can perform public verification in proxy signcryption mode, resist proxy key exposure attacks, resist insider attacks and supports self-delegation. What is more, it needs no secure channel between the original person and the proxy person. Under the adaptive chosen ciphertext, chosen identity and chosen warrant attacks, the confidentiality of our scheme can be reduced to the GBDH hard problem. Under the adaptive chosen message, chosen identity and chosen warrant attacks, the unforgeability of our scheme can be reduced to the GDH' hard problem. Through performance evaluation, our scheme is found to be practical.

Acknowledgments

This work was supported by the National Natural Science Foundation of China [grant numbers 61462048, 61562047, 61662039]. We express our thanks to Ms. Yan Di, who checked our manuscript.

References

[1] M. H. Au, J. Chen, J. K. Liu, Y. Mu, D. S. Wang, and G. M. Yang, "Malicious kgc attacks in certificateless cryptography," in *Proceedings of the 2007 ACM Symposium on Information, Computer and Communications Security (ASIACCS'07)*, pp. 302–311, Mar. 2007.

[2] M. Barbosa and P. Farshim, "Certificateless signcryption," in *Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security (ASIACCS'08)*, pp. 369–372, Mar. 2008.

[3] A. Boldyreva, A. Palacio, and B. Warinschi, "Secure proxy signature schemes for delegation of signing rights," *Cryptology ePrint Archive*, vol. 25, no. 1, pp. 57V115, Jan. 2012.

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

[5] S. S. Duan, Z. F. Cao, and Y. Zhou, "Secure delegation-by-warrant id-based proxy signcryption scheme," in *Computational Intelligence and Security, International Conference (CIS'05)*, pp. 445–450, Dec. 2005.

[6] Y. L. Han, Y. C. Bai, D. Y. Fang, and X. Y. Yang, "The new attribute-based generalized signcryption scheme," in *Intelligent Computation in Big Data Era - International Conference of Young Computer Scientists, Engineers and Educators (ICYCSEE'15)*, pp. 353–360, Jan. 2015.

[7] Y. L. Han and X. L. Gui, "Adaptive secure multicast in wireless networks," *International Journal of Communication Systems*, vol. 22, no. 9, pp. 1213–1239, 2009.

[8] Y. L. Han, X. Y. Yang, P. Wei, Y. M. Wang, and Y. P. Hu, "Ecgsc: Elliptic curve based generalized signcryption," in *Ubiquitous Intelligence and Computing, Third International Conference (UIC'06)*, pp. 956–965, Sep. 2006.

[9] M. S. Hwang, J. W. Lo, S. C. Lin, "An efficient user identification scheme based on ID-based cryptosystem", *Computer Standards & Interfaces*, vol. 26, no. 6, pp. 565–569, 2004.

[10] 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.

[11] H. F. Ji, W. B. Han, and L. Zhao, "Certificateless generalized signcryption," *Cryptology ePrint Archive*, no. 33, pp. 962-967, 2012.

[12] P. Kushwah and S. Lal, "Efficient generalized signcryption schemes," *Cryptology ePrint Archive*, 2010. (http://eprint.iacr.org/2010/346)

[13] S. Lal and P. Kushwah, "Id based generalized signcryption," *Cryptology ePrint Archive*, 2008. (http://eprint.iacr.org/2008/084)

Table 1: Comparison of computational costs

Schemes	**Dele**	**Dele-Veri**	**P-Signcrypt**	**P-UnSigncrypt**
[15]	3M	3P+E	P+2M+2E	8P+4E
[5]	2M	3P	P+2M+2E	7P+2E
[25]	2M	2P+M	P+2M+E	3P+M
[22]	4P+5M+2E	2P	P+2M+2E	4P+2E
Ours	3M	2P+M	P+3M+E	4P+4M

Table 2: Comparison of communication overhead

Schemes	**CipherText-Size**
[15]	$2\|G_1\| + \|q\| + \|m\| + \|m_w\|$
[5]	$3\|G_1\| + \|q\| + \|m_w\|$
[25]	$3\|G_1\| + 2\|ID\| + \|m\| + \|m_w\|$
[22]	$\|G_1\| + \|q\| + \|m\| + \|m_w\|$
Ours	$4\|G_1\| + \|m\| + \|m_w\|$

[14] 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.

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

[16] 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.

[17] Y. Ming and Y. M. Wang, "Cryptanalysis of an identity based signcryption scheme in the standard model," *International Journal of Network Security*, vol. 18, no. 1, pp. 165–171, 2016.

[18] C. H. Pan, S. P. Li, Q. H. Zhu, C. Z. Wang, and M. W. Zhang, "Notes on proxy signcryption and multi-proxy signature schemes," *International Journal of Network Security*, vol. 17, no. 1, pp. 29–33, 2015.

[19] D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures," *Journal of Cryptology*, vol. 13, no. 3, p. 361-396, 2000.

[20] J. C. N. Schuldt, K. Matsuura, and K. G. Paterson, "Proxy signatures secure against proxy key exposure," in *11th International Workshop on Practice and Theory in Public-Key Cryptography (PKC'08)*, pp. 141–161, Mar. 2008.

[21] X. Q. Shen, M. Yang, and J. Feng, "Identity based generalized signcryption scheme in the standard model," *Entropy*, vol. 19, no. 3, Article Number 121, 2017.

[22] X. X. Tian, J. P. Xu, H. J. Li, Y. Peng, and A. Q. Zhang, "Secure id-based proxy signcryption scheme with designated proxy signcrypter," in *2009 International Conference on Multimedia Information Networking and Security (MINES'09)*, pp. 351–355, Nov. 2009.

[23] S. Vollala, B. S. Begum, A. D. Joshi, and N. Ramasubramanian, "High-radix modular exponentiation for hardware implementation of public-key cryptography," in *Proceedings of the 2016 International Conference on Computing, Analytics and Security Trends (CAST'16)*, pp. 346–350, Dec. 2016.

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

[25] Q. Wang and Z. F. Cao, "Efficient id-based proxy signature and proxy signcryption form bilinear pairings," in *Computational Intelligence and Security, International Conference, Part II (CIS'05)*, pp. 167–172, Dec. 2005.

[26] X. A. Wang, X. Y. Yang, and Y. L. Han, "Provable secure generalized signcryption," *Cryptology ePrint Archive*, 2007. (`http://eprint.iacr.org/2007/173`)

[27] G. Y. Wei, J. Shao, Y. Xiang, P. P. Zhu, and R. X. Lu, "Obtain confidentiality or/and authenticity in big data by id-based generalized signcryption," *Information Sciences*, vol. 318, pp. 111–122, 2015.

[28] H. J. Yoon, J. H. Cheon, and Y. D. Kim, "Batch verifications with id-based signatures," in *7th International Conference of Information Security and Cryptology (ICISC'04)*, pp. 233–248, Dec. 2004.

[29] G. Yu, X. X. Ma, Y. Shen, and W. B. Han, "Provable secure identity based generalized signcryption scheme," *Theoretical Compute Science*, vol. 411, no. 40-42, pp. 3614–3624, 2010.

[30] C. X. Zhou, "An improved multi-receiver generalized signcryption scheme," *International Journal of Network Security*, vol. 17, no. 3, pp. 340–350, 2015.

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

[32] C. X. Zhou, W. Zhou, and X. W. Dong, "Provable certificateless generalized signcryption scheme," *Designs, Codes and Cryptography*, vol. 71, no. 2, pp. 331–346, 2014.

Biography

Caixue Zhou received BS degree in Computer Science Department from Fudan University in 1988, Shanghai, China and MS degree in Space College of Beijing University of Aeronautics and Astronautics in 1991, Beijing, China. He is an Associate Professor in the School of Information Science and Technology, Jiujiang University, Jiujiang, China since 2007. He is a member of the CCF (China Computer Federation) and a member of CACR(Chinese Association for Cryptologic Research). His research interests include applied cryptography, security of computer networks.

Yue Zhang received BS degree in College of Computer Science from Chongqing University in 2004, Chongqing, China and MS degree in School of Computer Science and Technology from Huazhong University of Science and Technology in 2008, Wuhan, China. She is a lecturer in the School of Information Science and Technology, Jiujiang University, JiuJiang, China since 2007. Her research interests include applied cryptography, network and information security.

Lingmin Wang received BS degree in computer science and technology from Shenyang Normal University in 2003, Shenyang, China and MS degree in School of Computer Science and Technology from Huazhong University of Science and Technology in 2009, Wuhan,China. She is a lecturer in the School of Information Science and Technology, Jiujiang University, Jiujiang, China since 2007. She is a member of the CCF(China Computer Federation). Her research interest includes security of computer networks.