A New Provably Secure Identity-Based Multi-proxy Signature Scheme

Qunshan Chen[1,4(✉)], Zhenjie Huang[2], Yong Ding[3], Yuping Zhou[1,4], and Hui Huang[1,5]

[1] College of Computer, Minnan Normal University, Zhangzhou 363000, People's Republic of China
xiamensam@163.com, hhui323@163.com, yp_zhou@mnnu.edu.cn

[2] Lab of Granular Computing, Minnan Normal University, Zhangzhou 363000, People's Republic of China
zjhuang@mnnu.edu.cn

[3] School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, People's Republic of China
stone_dingy@126.com

[4] Key Laboratory of Data Science and Intelligence Application, Fujian Province University, Zhangzhou 363000, People's Republic of China

[5] Key Laboratory of Financial Mathematics (Putian University), Fujian Province University, Putian 351100, People's Republic of China

Abstract. In a multi-proxy signature scheme, an original signer could delegate his signing power to a designated proxy group. Only the cooperation of all proxy signers in the proxy group could generate a legitimate proxy signature on behalf of the original signer. In this paper, we formalize the definition and security model of the identity-based multi-proxy signature, and we construct a new identity-based multi-proxy signature scheme using bilinear pairings. We show the security of our scheme in the random oracle model, and the security of our scheme is based on the hardness of the computational Diffie-Hellman problem.

Keywords: Identity-based signature · Multi-proxy signature · Computational Diffie-Hellman problem · Provable security

1 Introduction

The concept of an identity-based (ID-based) cryptosystem was first introduced by Shamir [20] in 1984. In an ID-based cryptosystem, the user does not generate his key pairs by himself. The system needs a trusted third authority named the private key generator (PKG) to compute the user's private key, and the user's public key can be derived as an arbitrary string that indicates the user's identity information, such as the user's e-mail address, IP address, and social security number. An ID-based cryptosystem simplifies the problem of key management in traditional public-key cryptography. Due to this advantage, many ID-based cryptosystems have been proposed [1, 3, 5, 12, 21].

J. Vaidya et al. (Eds.): CSS 2019, LNCS 11982, pp. 230–242, 2019.
https://doi.org/10.1007/978-3-030-37337-5_18

In 1996, Mambo et al. [14, 15] introduced the concept of the proxy signature, which solved the problem of the authorization of the signing capability. In a proxy signature scheme, an original signer is allowed to delegate his signing power to a designated person named a proxy signer. Provided with the proxy delegation, the proxy signer could sign a message on behalf of the original signer. Any verifier could be convinced of both the original signer's authorization and the proxy signer's signature. Proxy signatures could be used in many situations, especially in applications where the delegation of rights is highly common, such as distributed computing and mobile communications. To date, many proxy signature schemes have been proposed [2, 7, 9, 18, 19, 23].

The primitive of the multi-proxy signature was first introduced by Hwang and Shi in 2000 [8]. In a multi-proxy signature scheme, an original signer can delegate his signing power to a designated proxy group. Only the cooperation of all proxy signers in the proxy group could generate a legitimate proxy signature on behalf of the original signer. The multi-proxy signature scheme can be regarded as a special threshold proxy signature scheme [24]. Since that work, some multi-proxy signature schemes have been successively proposed [10, 11, 16, 17], but they lacked provable security, and their securities were only heuristically analyzed. In 2009, Cao and Cao [4] gave the first formal definition and security model of an ID-based multi-proxy signature and then proposed an ID-based multi-proxy signature scheme using bilinear pairings. However, Xiong et al. [22] showed that Cao-Cao's scheme was not secure under their security model. These researchers proposed an improved scheme, but they did not give the formal security proof of the improved scheme. Moreover, some provably secure multi-proxy signature schemes in the standard model have been proposed [6, 13].

In this paper, based on the work of Bellare et al. [1] and Cao and Cao [4], we give a formal definition and security model of an ID-based multi-proxy signature scheme. Then, we present a concrete ID-based multi-proxy signature scheme that meets our definition. Our scheme is provably secure in the random oracle model, and the security of our scheme is based on the hardness of computational Diffie-Hellman problem. To the best of our knowledge, to date, our scheme is the only ID-based multi-proxy signature scheme using bilinear pairings that is proved to be secure in the random oracle model.

The rest of this paper is organized as follows. In Sect. 2, we introduce some preliminaries. In Sect. 3, we give a formal definition and security model of the ID-based multi-proxy signature scheme. In Sect. 4, we propose a new ID-based multi-proxy signature scheme. In Sect. 5, we prove our scheme's security and compare the efficiency of our scheme with some similar schemes. The final section is the conclusion.

2 Preliminaries

In this section, we introduce some concepts for bilinear pairings and the computational Diffie-Hellman problem.

2.1 Bilinear Pairings

Let $(G_1, +)$ and $(G_2, \cdot)$ be two groups of prime order q. We call a map $e : G_1 \times G_1 \rightarrow G_2$ a bilinear pairing if it satisfies the following properties:

- **Bilinear:** For any P, $Q \in G_1$, and any α, $\beta \in Z_q$, we have $e(\alpha P, \beta Q) = e(P, Q)^{\alpha\beta}$;
- **Nondegenerate:** Their exists P, $Q \in G_1$, such that $e(P, Q) \neq 1$;
- **Computable:** For any P, $Q \in G_1$, there is an efficient algorithm to compute $e(P, Q)$.

2.2 Computational Assumption

The security of our scheme is based on the hardness of the computational Diffie-Hellman problem.

Definition 1. Computational Diffie-Hellman (CDH) problem. Let G_1 be a group of prime order q with generator P. Given aP, $bP \in G_1$, where a, $b \in Z_q^*$, compute abP.

Definition 2. CDH Assumption. We say that the (t, ε)-CDH assumption holds in group G_1 if no probabilistic algorithm can solve the CDH problem in G_1 with a non-negligible probability of at least ε within polynomial time t.

3 Definition and Security Model of Identity-Based Multi-proxy Signature

In this section, we give a formal definition and security model for our ID-based multi-proxy signature scheme.

3.1 Definition of Identity-Based Multi-proxy Signature

We give the definition of the ID-based multi-proxy signature scheme as follows. More details can be found in [4].

Definition 3. An identity-based multi-proxy signature scheme consists of the following algorithms: Setup, User-Key-Gen, Delegation-Gen, Multi-Proxy-Sign, and Multi-Proxy-Verify. It is composed of the following entities: the key generation center KGC, the original signer U_0, the proxy signers $U_i (i = 1, 2, \cdots, n)$, and the verifier.

- **Setup**: This algorithm is run by the KGC on the input security parameter 1^k, and it generates the system's master key s and public parameters *params*.
- **User-Key-Gen**: This algorithm is run by the KGC. It takes as inputs the *params*, the identity ID_0 of the original signer U_0, or the identity ID_i of the proxy signer $U_i (i = 1, 2, \cdots, n)$, and then returns the corresponding private key $S_{ID_i} (i = 0, 1, 2, \cdots, n)$.
- **Delegation-Gen**: This algorithm is run by the original signer U_0. It takes as inputs the *params*, his private key S_{ID_0}, the identity ID_i of the proxy signer $U_i (i = 1, 2, \cdots, n)$, and a warrant message w, and it outputs a proxy delegation σ_0.
- **Multi-Proxy-Sign**: This algorithm is run by every proxy signer U_i. It takes as inputs the *params*, the private key S_{ID_i}, the delegation σ_0 and the message m, and it outputs a partial proxy signature S_i $(i = 1, 2, \cdots, n)$. Then, U_i sends S_i to a clerk who is a designated proxy signer in the proxy group. The clerk verifies the validity of S_i, and it returns the multi-proxy signature S if all of S_i are accepted; otherwise, the algorithm stops.

- **Multi-Proxy-Verify**: It takes as inputs the *params*, the identities ID_0 and $ID_i (i = 1, 2, \cdots, n)$, the message m and the multi-proxy signature S. The algorithm outputs 1 if S is a valid multi-proxy signature, and it outputs 0 otherwise.

3.2 Security Model of Identity-Based Multi-proxy Signature

According to the security model of the proxy signature that is proposed in [14, 15], the security of the proxy signature is mainly considered based on the unforgeability of the delegation and the proxy signature. The unforgeability of the delegation means that an adversary could not forge an efficient delegation on behalf of the original signer, and the proxy signer could not generate a valid proxy signature without the delegation. The unforgeability of the proxy signature means that nobody (including the original signer) could generate a legitimate proxy signature without the proxy signer's private key.

In our model, we consider the adversary who can adaptively choose an identity ID_0 for an original signer or an identity ID_i for a proxy signer, and then acts as a user with the identity of ID_0 or ID_i when executing the multi-proxy signature scheme with other users. Therefore, we can divide the potential adversaries into the following two kinds:

Type I: Adversary A_{I} attempts to forge a multi-proxy signature without the delegation. For the adversary A_{I}, we mainly model the malicious proxy signer. Moreover, adversary A_{I} can be considered as a collusion attack from multiple proxy signers.

Type II: Adversary A_{II} attempts to forge a multi-proxy signature without the proxy signer's private key. For the adversary A_{II}, we mainly model the malicious original signer.

According to the work of [4], we define the security model of an ID-based multi-proxy signature as follows.

Definition 4. Let A_{I} and A_{II} be adversaries that act as the malicious proxy signer and the original signer, respectively. The security of an ID-based multi-proxy signature scheme is modeled by the following games between a challenger C and A_{I} and A_{II}, respectively.

Game 1

The challenger C inputs a security parameter 1^k, performs the Setup algorithm, generates the system parameters *params*, and then C sends *params* to A_{I}. A_{I} can carry out the following queries in polynomial bounded times.

- **Hash query**: A_{I} can query the value of all Hash functions in the scheme.
- **User-Key query**: A_{I} can input an arbitrary user's identity ID_i to query the private key S_{ID_i}. C performs the User-Key-Gen algorithm to generate S_{ID_i} and returns the result to A_{I}.
- **Delegation query**: A_{I} can query the proxy delegation certificate σ_0 of a chosen warrant message w. C performs the Delegation-Gen algorithm and then returns the result to A_{I}.
- **Multi-Proxy-Sign query**: A_{I} can input the original signer's identity ID_0, the proxy signer's identity $ID_i (i = 1, 2, \cdots, n)$, the warrant message w and the message m, and queries the multi-proxy signature on m. C performs the Multi-Proxy-Sign algorithm to generate a multi-proxy signature S and then returns the result to A_{I}.

Finally, A_{I} outputs a multi-proxy signature S^* on message m using the proxy signers $U_i(i = 1, 2, \cdots, n)$ with the warrant message w. We say that A_{I} wins the game if and only if the multi-proxy signature S^* is accepted by the verifier, the warrant message w has not been queried in the Delegation query, and (w, m) has not been queried in the Multi-Proxy-Sign query.

Game 2

The challenger C inputs a security parameter 1^k, performs the Setup algorithm, generates the system parameters *params*, and then C sends *params* to A_{II}. A_{II} can perform the same queries as *Game* 1 in polynomial bounded times.

Finally, A_{II} outputs a multi-proxy signature S^* on message m using the proxy signers $U_i(i = 1, 2, \cdots, n)$ with the warrant message w. We say that A_{II} wins the game if and only if the multi-proxy signature S^* is accepted by the verifier, there is at least one $ID_I \in \{ID_1, ID_2, \cdots, ID_n\}$ that has not been queried in the User-Key query, and (w, m) has not been queried in the Multi-Proxy-Sign query.

Definition 5. An identity-based multi-proxy signature scheme is existentially unforgeable against the chosen massage attack and chosen warrant attack if and only if there is no probabilistic polynomial-time (PPT) adversary that could win the above games with a non-negligible probability.

4 Identity-Based Multi-proxy Signature Scheme

In this section, we propose a new ID-based multi-proxy signature scheme based on the ID-based signature scheme that was constructed by Sakai-Ogishi-Kasahara [1, 12].

Setup: Given a security parameter 1^k, let G_1 be an additive group of prime order q with a generator P, G_2 is a multiplicative group with the same prime order, and $e : G_1 \times G_1 \to G_2$ is a bilinear map. The KGC chooses the hash functions $H_1 : \{0,1\}^* \to G_1$, $H_2 : \{0,1\}^* \times G_1 \to G_1$ and $H_3 : \{0,1\}^* \times G_1 \times G_1 \to Z_q^*$. The KGC randomly chooses $s \in Z_q^*$ as the master key, computes the system public key $P_{pub} = sP$, and then publishes the system parameters $params = (G_1, G_2, q, e, P, P_{pub}, H_1, H_2, H_3)$.

User-Key-Gen: Given the identity ID_0 of the original signer U_0 or the identity ID_i of the proxy signer U_i, where $1 \le i \le n$, the KGC computes $Q_{ID_i} = H_1(ID_i)$ and $S_{ID_i} = sQ_{ID_i}$, and then it sends S_{ID_i} to $U_i(i = 0, 1, 2, \cdots, n)$ via a secure channel.

Delegation-Gen: The original signer U_0 generates the proxy warrant message w, which includes the identity information of the original signer and proxy signers, the scope of the proxy authority, and the delegation period.

1. U_0 randomly chooses $r_0 \in Z_q^*$, and computes $R_0 = r_0 P$.
2. U_0 computes $h_0 = H_2(w, R_0)$, $V_0 = r_0 h_0 + S_{ID_0}$.
3. U_0 sends a proxy warrant message w and its signature $\sigma_0 = (R_0, V_0)$ to the proxy signers $U_i(i = 1, 2, \cdots, n)$ via a secure channel.

The proxy signer U_i computes $h_0 = H_2(w, R_0)$, and then checks the following equation:

$$e(P, V_0) = e(P_{pub}, Q_{ID_0})e(R_0, h_0).$$

If the equality holds, U_i accepts σ_0 as a valid delegation.

Multi-Proxy-Sign: Given a message m to be signed, for $1 \leq i \leq n$, the proxy signer U_i can sign the message m as follows:

1. U_i randomly chooses $r_i \in Z_q^*$, computes $R_i = r_i P$, and broadcasts R_i to the other proxy signers.
2. U_i computes $R = \sum_{i=1}^{n} R_i$, $h_0 = H_2(w, R_0)$, $h = H_3(m, R_0, R)$, and $V_i = r_i h_0 + hS_{ID_i} + V_0$.
3. U_i sends the partial proxy signature $\sigma_i = (w, R_0, R_i, V_i)$ to the clerk who is a designated proxy signer in the proxy group.
4. The clerk computes $R = \sum_{i=1}^{n} R_i$, $h_0 = H_2(w, R_0)$, $h = H_3(m, R_0, R)$, then verifies the validity of σ_i using the following equation:

$$e(P, V_i) = e(P_{pub}, hQ_{ID_i} + Q_{ID_0})e(R_i + R_0, h_0),\ i = 1, 2, \cdots, n$$

If all of the equalities hold, the clerk computes $V = \sum_{i=1}^{n} V_i$, and then the multi-proxy signature on message m is $\sigma = (w, R_0, R, V)$.

Multi-Proxy-Verify: To verify a multi-proxy signature $\sigma = (w, R_0, R, V)$, the verifier computes $h_0 = H_2(w, R_0)$, $h = H_3(m, R_0, R)$, and accepts the signature if and only if the following equation holds:

$$e(P, V) = e(P_{pub}, nQ_{ID_0} + h\sum_{i=1}^{n} Q_{ID_i})e(nR_0 + R, h_0).$$

5 Analysis of Our Scheme

5.1 Correctness

The correctness of a partial proxy signature can be proved by the following:

$$\begin{aligned} e(P, V_i) &= e(P, r_i h_0 + hS_{ID_i} + V_0) \\ &= e(P, r_i h_0)e(P, hS_{ID_i})e(P, r_0 h_0)e(P, S_{ID_0}) \\ &= e(r_i P, h_0)e(P_{Pub}, hQ_{ID_i})e(r_0 P, h_0)e(P_{Pub}, Q_{ID_0}) \\ &= e(P_{pub}, hQ_{ID_i} + Q_{ID_0})e(R_i + R_0, h_0). \end{aligned}$$

The proposed ID-based multi-proxy signature scheme is correct according to the following:

$$\begin{aligned} e(P,V) &= e(P, \sum_{i=1}^{n}(r_i h_0 + hS_{ID_i} + V_0)) \\ &= e(P, \sum_{i=1}^{n} r_i h_0) e(P, h\sum_{i=1}^{n} S_{ID_i}) e(P, nr_0 h_0) e(P, nS_{ID_0}) \\ &= e(\sum_{i=1}^{n} r_i P, h_0) e(P_{Pub}, h\sum_{i=1}^{n} Q_{ID_i}) e(nr_0 P, h_0) e(P_{Pub}, nQ_{ID_0}) \\ &= e(P_{pub}, nQ_{ID_0} + h\sum_{i=1}^{n} Q_{ID_i}) e(nR_0 + R, h_0). \end{aligned}$$

5.2 Security Proof of Our Scheme

In this section, we will prove that our scheme is secure in the random oracle model. The Delegation-Gen algorithm in our scheme is the ID-based signature scheme that was constructed by Sakai-Ogishi-Kasahara [1, 12], which was proved to be secure, such that an adversary could not forge a valid delegation certificate without the original signer's private key.

Theorem 1. In the random oracle model, let A_{I} be a PPT adversary with the non-negligible probability ε to win *Game* 1 in time t. Assume that A_{I} makes at most q_{H_i} queries to the hash functions $H_i (i = 1, 2, 3)$, at most q_K queries to the User-Key-Gen oracle, at most q_D queries to the Delegation-Gen oracle, and at most q_P queries to the Multi-Proxy-Sign oracle. Then, there exists an algorithm C with the probability $\varepsilon' \geq \varepsilon \frac{1}{(q_K + q_D + n + 1)e}$ to solve the CDH problem in time $t' < t + (q_{H_1} + q_K + q_{H_2} + 3q_D + 3nq_P + n + 4)t_s + t_i$, where t_i is the time of an inversion computation in Z_q^*, and t_s is the time of a scalar multiplication in G_1.

Proof: Let (P, aP, bP) be a random instance of the CDH problem in G_1 acting as a challenger of A_{I}. C could compute abP via *Game* 1 as follows.

C runs the setup algorithm, sets the system public key $P_{Pub} = aP$, and generates the system parameters *params*. Then, it gives *params* to A_{I}. C maintains an H_1-list $(ID_i, Q_{ID_i}, t_i, c_i)$ to hold the value of hash function H_1, a UK-list (ID_i, S_{ID_i}) to hold the user's private key, an H_2-list (w, R_0, h_0, d_0) to hold the value of hash function H_2, a Del-list (ID_0, w, r_0, R_0, V_0) to hold the delegation certificate, and a H_3-list (m, R_0, R, h) to hold the value of hash function H_3.

A_{I} can conduct queries as follows.

H_1 Query: When A_{I} makes H_1 query $ID_i (i = 0, 1, 2, \cdots, n)$, ID_0 is the identity of the original signer, and $ID_i (i = 1, 2, \cdots, n)$ are the identities of the proxy signers. C responds as follows.

(1) For $0 \le i \le n$, if ID_i has been queried, C returns Q_{ID_i} from the H_1-list.
(2) Otherwise, C randomly chooses $t_i \in Z_q^*$, and generates a random coin $c_i \in \{0,1\}$ such that $\Pr[c_i = 0] = \delta$ and $\Pr[c_i = 1] = 1 - \delta$, where $0 < \delta < 1$.

Then, C sets $Q_{ID_i} = H_1(ID_i) = t_i P$ if $c_i = 0$, and sets $Q_{ID_i} = t_i(bP)$ if $c_i = 1$. Finally, C adds $(ID_i, Q_{ID_i}, t_i, c_i)$ into the H_1-list, and returns Q_{ID_i} to A_I.

User-Key Query: When A_I queries the private key of ID_i, C responds as follows.

(1) C searches the UK-list. If ID_i has been queried, then C returns the corresponding private key S_{ID_i} to A_I.
(2) Otherwise, C searches the H_1-list to get $(ID_i, Q_{ID_i}, t_i, c_i)$. When there is no record of ID_i in the H_1-list, C will create $(ID_i, Q_{ID_i}, t_i, c_i)$ according to the H_1 query.

If $c_i = 0$, C computes $S_{ID_i} = t_i(aP)$, returns S_{ID_i} to A_I and adds (ID_i, S_{ID_i}) into the UK-list. If $c_i = 1$, C outputs "failure" and terminates the simulation.

H_2 Query: When A_I makes an H_2 query on (w, R_0), C responds as follows.

(1) If (w, R_0) has been queried, C returns h_0 from the H_2-list;
(2) Otherwise, C randomly chooses $d_0 \in Z_q^*$, and d_0 has not been in the H_2-list. C computes $h_0 = d_0 P$ and returns h_0 to A_I. Then, C adds (w, R_0, h_0, d_0) into the H_2-list.

H_3 Query: When A_I makes an H_3 query on (m, R_0, R), C responds as follows.

(1) If (m, R_0, R) has been queried, C returns h from the H_3-list;
(2) Otherwise, C randomly chooses $h \in Z_q^*$, and h has not been in H_3-list. Then, C returns h to A_I and adds (m, R_0, R, h) into the H_3-list.

Delegation Query: A_I can query the proxy delegation certificate σ_0 of a chosen warrant message w from ID_0, and C responds as follows.

(1) If (ID_0, w) has been queried, C returns $\sigma_0 = (R_0, V_0)$ from the Del-list.
(2) Otherwise, C searches the UK-list to get $(ID_0, Q_{ID_0}, t_0, c_0)$.

If $c_0 = 0$, C randomly chooses $r_0 \in Z_q^*$ and computes $R_0 = r_0 P$. Then, C searches the H_2-list to get (w, R_0, h_0, d_0), computes $V_0 = r_0 h_0 + t_0(aP)$, returns $\sigma_0 = (R_0, V_0)$ to A_I, and adds (ID_0, w, r_0, R_0, V_0) to the Del-list.

If $c_i = 1$, C outputs "failure" and terminates the simulation.

When there are no records of ID_i or (w, R_0) in the UK-list and the H_2-list, C will create the corresponding values according to the User-Key query and H_2 query.

Multi-Proxy-Sign Query: A_I can input a proxy signer's identity $ID_i (i = 1, 2, \cdots, n)$, an original signer's identity ID_0, a warrant message w and a message m, and it then queries the multi-proxy signature. C responds as follows.

(1) C searches the H_1-list to get $(ID_i, Q_{ID_i}, t_i, c_i)$, where $i = 0, 1, 2, \cdots, n$. If $c_0 = 1$ or $c_i = 1$, C outputs "failure" and terminates the simulation.

(2) Otherwise, for $1 \le i \le n$, C randomly chooses $r_i \in Z_q^*$, and computes $R_i = r_i P$, $R = \sum_{i=1}^{n} R_i$.

C searches the UK-list to get $(ID_i, S_{ID_i})(i = 0, 1, 2, \cdots, n)$, and then searches the Del-list, H_2-list and H_3-list to get the records (ID_0, w, r_0, R_0, V_0), (w, R_0, h_0, d_0) and (m, R_0, R, h), respectively. If there are no corresponding records in the lists, C generates the corresponding values according to the above queries.

For $1 \le i \le n$, C computes $V_i = r_i h_0 + h S_{ID_i} + V_0 = r_i h_0 + h t_i (aP) + r_0 h_0 + t_0 (aP)$, $V = \sum_{i=1}^{n} V_i$, and then returns $\sigma = (w, R_0, R, V)$ to A_1.

Finally, A_1 stops the simulation, and outputs a multi-proxy signature tuple $\{ID_0, ID_i, m, \sigma^* = (w, R_0, R, V)\}$. C searches the H_1-list to get $(ID_i, Q_{ID_i}, t_i, c_i)$ $(0 \le i \le n)$. If $c_0 = 0$ or $c_i = 1$ $(1 \le i \le n)$, C outputs "failure" and terminates the simulation. Otherwise, $c_0 = 1$ and $c_i = 0$ $(1 \le i \le n)$, C gets (w, R_0, h_0, d_0) and (m, R_0, R, h) in the H_2-list and H_3-list, respectively. The forged multi-proxy signature satisfies the following equation.

$$\begin{aligned} e(P, V) &= e(P_{pub}, h \sum_{i=1}^{n} Q_{ID_i} + n Q_{ID_0}) e(R + n R_0, h_0) \\ &= e(aP, h \sum_{i=1}^{n} t_i P + n t_0 b P) e(R + n R_0, d_0 P) \\ &= e(P, h \sum_{i=1}^{n} t_i a P + n t_0 a b P + d_0 (R + n R_0)) \end{aligned}$$

Then, C computes $abP = (nt_0)^{-1}(V - h \sum_{i=1}^{n} t_i P_{Pub} - d_0(R + nR_0))$. Therefore, C can solve the CDH problem.

To analyze the probability of C succeeding in the above game, we define the following five events, which are needed for C to succeed.

E_1: C does not abort in the User-Key query.

E_2: C does not abort in the Delegation-Gen query.

E_3: C does not abort in the Multi-Proxy-Sign query.

E_4: A_1 succeeds to forge a valid multi-proxy signature.

E_5: Event E_4 occurs, $c_0 = 1$, and $c_i = 0 (1 \le i \le n)$. Here, c_0 and c_i are the c-components of the tuple on the H_1-list.

Therefore, the probability that C can solve the instance of CDH problem is

$$\begin{aligned} &\Pr[E_1 \wedge E_2 \wedge E_3 \wedge E_4 \wedge E_5] \\ &= \Pr[E_1] \Pr[E_2 | E_1] \Pr[E_3 | E_1 \wedge E_2] \Pr[E_4 | E_1 \wedge E_2 \wedge E_3] \ \Pr[E_5 | E_1 \wedge E_2 \wedge E_3 \wedge E_4] \ . \end{aligned}$$

From the simulation, we have the following results.

$\Pr[E_1] \ge \delta^{q_K}$,

$\Pr[E_2 | E_1] \ge \delta^{q_D}$,

$\Pr[E_3|E_1 \wedge E_2] = 1$,
$\Pr[E_4|E_1 \wedge E_2 \wedge E_3] \geq \varepsilon$, and
$\Pr[E_5|E_1 \wedge E_2 \wedge E_3 \wedge E_4] \geq (1-\delta)\delta^n$.
Thus, we have $\Pr[E_1 \wedge E_2 \wedge E_3 \wedge E_4 \wedge E_5] \geq \varepsilon(1-\delta)\delta^{q_K+q_D+n}$.
When $\delta = \frac{q_K+q_D+n}{q_K+q_D+n+1}$, $(1-\delta)\delta^{q_K+q_D+n}$ obtains the minimum value $\frac{1}{(q_K+q_D+n+1)e}$. Then, the probability that C succeeds is $\varepsilon' \geq \varepsilon\frac{1}{(q_K+q_D+n+1)e}$.
The total running time of C is $t' < t + (q_{H_1} + q_K + q_{H_2} + 3q_D + 3nq_P + n + 4)t_s + t_i$.

Theorem 2. In the random oracle model, let A_{II} be a PPT adversary with a non-negligible probability ε to win *Game* 2 in time t. Assume that A_{II} makes at most q_{H_i} queries to hash functions $H_i(i = 1, 2, 3)$, at most q_K queries to the User-Key-Gen oracle, at most q_D queries to the Delegation-Gen oracle, and at most q_P queries to the Multi-Proxy-Sign oracle. Then, there exists an algorithm C with the probability $\varepsilon' \geq \varepsilon\frac{n}{(q_K+q_D+n+1)e}$ to solve the CDH problem in time $t' < t + (q_{H_1} + q_K + q_{H_2} + 3q_D + 3nq_P + n + 4)t_s + t_i$, where t_i is the time of an inversion computation in Z_q^*, and t_s is the time of a scalar multiplication in G_1.

Proof: Let (P, aP, bP) be a random instance of the CDH problem in G_1. C could conduct the same computation as in Theorem 1, and A_{II} could also conduct the same queries as in Theorem 1.

Finally, A_{II} stops the simulation, and outputs a multi-proxy signature tuple $\{ID_0, ID_i, m, \sigma^* = (w, R_0, R, V)\}$. C searches the H_1-list to get $(ID_i, Q_{ID_i}, t_i, c_i)$ $(0 \leq i \leq n)$. If $c_0 = 1$ or $c_i = 0(1 \leq i \leq n)$, C outputs "failure" and terminates the simulation. Otherwise, $c_0 = 0$ and at least one $c_i = 1$. Without the loss of generality, we assume that $c_1 = 1$, and C gets (w, R_0, h_0, d_0) and (m, R_0, R, h) in the H_2-list and H_3-list, respectively. The forged multi-proxy signature satisfies the following equation.

$$\begin{aligned} e(P,V) &= e(P_{pub}, h\sum_{i=1}^{n} Q_{ID_i} + nQ_{ID_0})e(R + nR_0, h_0) \\ &= e(aP, h\sum_{i=2}^{n} t_iP + ht_1bP + nt_0P)e(R + nR_0, d_0P) \\ &= e(P, h\sum_{i=2}^{n} t_iaP + ht_1abP + nt_0aP + d_0(R + nR_0)). \end{aligned}$$

Then, C computes $abP = (ht_1)^{-1}(V - h\sum_{i=2}^{n} t_iP_{Pub} - nt_0P_{Pub} - d_0(R + nR_0))$. Therefore, C can solve the CDH problem. As with the proof in Theorem 1, the probability that C succeeds in the game is $\varepsilon' \geq \varepsilon\frac{n}{(q_K+q_D+n+1)e}$.
The total running time of C is $t' < t + (q_{H_1} + q_K + q_{H_2} + 3q_D + 3nq_P + n + 4)t_s + t_i$.

5.3 Efficiency Comparison

We compare the efficiency of our scheme with some ID-based multi-proxy signature schemes based on bilinear pairings in Table 1. We only consider the computational

costs for a single user and compare the algorithmic efficiency of delegation-gen, multi-proxy-sign and multi-proxy-verify, respectively. In Table 1, M denotes the point scalar multiplication operation in G_1, E denotes the exponentiation operation in G_2, and P denotes the pairing operation. We ignore other operations, such as hashing, in all the schemes.

Table 1. Comparison of efficiency for similar schemes

Schemes	Delegation-gen	Multi-proxy-sign	Multi-proxy-verify	Provable security
Scheme [4]	$2M + 3P$	$3M + 5P + 1E$	$3M + 4P$	Yes
Scheme [11]	$3M + 3P + 1E$	$5M + 3P + 1E$	$nM + 3P + 1E$	No
Scheme [16]	$1M + 1P + 3E$	$2M + 1P + 3E$	$1P + 2E$	No
Scheme [17]	$2M + 3P + 2E$	$4M + 4P + 3E$	$3P + 3E$	No
Scheme [22]	$2M + 3P$	$3M + 5P + 1E$	$3M + 4P$	No
Our scheme	$2M + 3P$	$4M + 3P$	$3M + 3P$	Yes

As shown in Table 1, we can see that our scheme is more efficient than the scheme in [4] which is provable secure. Moreover, we proved that our scheme was secure under the computational Diffie-Hellman problem. Although the scheme in [16] is more efficient than ours, there was no formal security proof in the scheme, and the schemes in [11, 17, 22] lacked provable security, as well. Meanwhile, it was proved that scheme [4] was not secure in [22]. Therefore, to the best of our knowledge, our scheme is the only ID-based multi-proxy signature scheme using bilinear pairings that is proved to be secure in the random oracle model.

6 Conclusion

In this paper, we describe the definition and security model of an identity-based multi-proxy signature scheme and propose a new identity-based multi-proxy signature scheme. Based on the hardness of the computational Diffie-Hellman problem, our scheme was proven to be secure in the random oracle model. Moreover, compared with previous ID-based multi-proxy signature schemes based on bilinear pairings, our scheme is provably secure and more efficient.

Acknowledgements. The authors would like to thank the anonymous reviewers for their helpful comments. This work is supported in part by the National Natural Science Foundation of China under Grant No. 61772150 and No. 61862012; the Natural Science Foundation of Fujian Province of China under Grant No. 2019J01750 and No. 2015J01662; the Research Project of Fujian Provincial Education Department of China under Grant No. JAT170345, No. JAT170346, and No. JK2017031; the Guangxi Key R&D Program under Grant No. AB17195025, the Guangxi Natural Science Foundation under Grant No. 2018GXNSFDA281054 and No. 2018GXNSFAA 281232, the National Cryptography Development Fund of China under Grant No. MMJJ2017 0217, and the Project of Key Laboratory of Financial Mathematics of Fujian Province University (Putian University) under Grant No. JR201806.

References

1. Bellare, M., Namprempre, C., Neven, G.: Security proofs for identity-based identification and signature schemes. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 268–286. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_17
2. Boldyreva, A., Palacio, A., Warinschi, B.: Secure proxy signature schemes for delegation of signing rights. J. Cryptol. **25**(1), 57–115 (2012)
3. Boneh, D., Franklin, M.: Identity-based encryption from the Weil pairing. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44647-8_13
4. Cao, F., Cao, Z.: A secure identity-based multi-proxy signature scheme. Comput. Electr. Eng. **35**(1), 86–95 (2009)
5. Choon, J.C., Hee Cheon, J.: An identity-based signature from gap Diffie-Hellman groups. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 18–30. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-36288-6_2
6. Gu, K., Jia, W., Deng, Y., Nie, X.: Secure and efficient multi-proxy signature scheme in the standard model. Chin. J. Electron. **25**(1), 93–99 (2016)
7. Huang, X., Susilo, W., Mu, Y., Wu, W.: Proxy signature without random oracles. In: Cao, J., Stojmenovic, I., Jia, X., Das, S.K. (eds.) MSN 2006. LNCS, vol. 4325, pp. 473–484. Springer, Heidelberg (2006). https://doi.org/10.1007/11943952_40
8. Hwang, S., Shi, C.: A simple multi-proxy signature scheme. In: Proceedings of the 10th National Conference on Information Security, Hualien, Taiwan, ROC, pp. 134–138 (2000)
9. Li, J., Xu, L., Zhang, Y.: Provably secure certificate-based proxy signature schemes. J. Comput. **4**(6), 444–452 (2009)
10. Li, S., Zhang, F.: A new multi-proxy signature from bilinear pairing. Chin. J. Electron. **24**(1), 90–94 (2007)
11. Li, X., Chen, K.: ID-based multi-proxy signature, proxy multi-signature and multi-proxy multi-signature schemes from bilinear pairings. Appl. Math. Comput. **169**(1), 437–450 (2005)
12. Libert, B., Quisquater, J.J.: The exact security of an identity based signature and its applications. Cryptology ePrint Archive, Report 2004/102 (2004). http://eprint.iacr.org/2004/102
13. Liu, Z., Hu, Y., Zhang, X., Ma, H.: Provably secure multi-proxy signature scheme with revocation in the standard model. Comput. Commun. **34**(3), 494–501 (2011)
14. Mambo, M., Usuda, K., Okamoto, E.: Proxy signature: delegation of the power to sign messages. IEICE Trans. Fundamentals **E79-A**(9), 1338–1353 (1996)
15. Mambo, M., Usuda, K., Okamoto, E.: Proxy signature for delegating signing operation. In: Proceedings of the 3rd ACM Conference on Computer and Communications Security, pp. 48–57. ACM, New York (1996)
16. Mishra, S., Sahu, R.A., Padhye, S., Yadav, R.S.: Efficient ID-based multi-proxy signature scheme from bilinear pairing based on *k-plus* problem. In: Hruschka, E.R., Watada, J., do Carmo Nicoletti, M. (eds.) INTECH 2011. CCIS, vol. 165, pp. 113–122. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22247-4_10
17. Rao, B.U., Reddy, P.V.: ID-based directed multi-proxy signature scheme from bilinear pairings. Int. J. Comput. Sci. Secur. **5**(1), 717–727 (2011)
18. Seo, S., Choi, K., Hwang, J., Kim, S.: Efficient certificateless proxy signature scheme with provable security. Inf. Sci. **188**, 322–337 (2012)

19. Singh, H., Verma, G.: ID-based proxy signature scheme with message recovery. J. Syst. Softw. **85**(1), 209–214 (2012)
20. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Blakley, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 47–53. Springer, Heidelberg (1985). https://doi.org/10.1007/3-540-39568-7_5
21. Waters, B.: Efficient identity-based encryption without random oracles. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 114–127. Springer, Heidelberg (2005). https://doi.org/10.1007/11426639_7
22. Xiong, H., Hu, J., Chen, Z., Li, F.: On the security of an identity based multi-proxy signature scheme. Comput. Electr. Eng. **37**(1), 129–135 (2011)
23. Zhang, F., Kim, K.: Efficient ID-based blind signature and proxy signature from bilinear pairings. In: Safavi-Naini, R., Seberry, J. (eds.) ACISP 2003. LNCS, vol. 2727, pp. 312–323. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-45067-X_27
24. Zhang, K.: Threshold proxy signature schemes. In: Okamoto, E., Davida, G., Mambo, M. (eds.) ISW 1997. LNCS, vol. 1396, pp. 282–290. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0030429