doi:10.3772/j.issn.1006-6748.2016.02.012

Identity-based proxy multi-signature applicable to secure E-transaction delegations①

Liu Jianhua (刘建华)②*, Wu Qianhong**, Liu Jianwei**, Shang Tao**
(*Aviation Engineering Institute, Civil Aviation Flight University of China, Guanghan 618307, P. R. China)
(**School of Electronics and Information Engineering, Beihang University, Beijing 100191, P. R. China)

Abstract

To enhance the robustness of a proxy multi-signature scheme and improve its efficiency, a novel proxy signature paradigm is proposed referred to as identity-based proxy multi-signature (IBPMS). In this paradigm, multiple proxy signer candidates are employed to play a role of the single proxy signer in the existing model. A provably secure IBPMS scheme is presented which requires only one round broadcast operation. Performance analysis demonstrates that the new scheme outperforms the existing multi-signature schemes in robustness and communication. These properties are rendered to our IBPMS scheme as a more practical solution to secure e-transaction delegation applications of proxy signatures.

Key words: multi-signature, E-transaction, delegation, provable security, information security

0 Introduction

Digital signature protocols allow message transmissions among a group of users with non-repudiation, user identification, and message authentication. Many variants of signatures such as blind signature[1], proxy signature[2], multi-proxy signature[3], and proxy multi-signature (PMS for short)[4] have been proposed to meet different application demands, among which, the proxy signature protocol is constructed to empower a signee to issue a message on behalf of another signee.

Proxy multi-signature plays an critical role in the following scenarios. There may be real estate owned by $m(m>1)$ entities, any legal transaction that wants to sell or rent out the assets must be permitted by all the m owners. In other words, it must be signed jointly by all the entities, or signed by their designated proxy signers. For the latter case, any transaction of the real estate needs to be executed with the permission issued by all the owners' proxy signers. One practical solution to the problem is to allow multiple proxy signers in a proxy multi-signature, each owner has his/her own proxy signer.

The PMS scheme[5] needs only one round broadcast operation for each original signer during the proxy key generation phase. The proxy multi-signature schemes in Refs[4-6] do not provide formal definition or security model. Cao et al. proposed an ID-based proxy multi-signature scheme which used bilinear pairings[7].

The existing PMS schemes (e.g., Ref.[5-9]) only allow one proxy signer candidate. This limitation may incur a bottleneck to the PMS schemes in some applications. Multi-proxy multi-signature[10,11] allow multiple original signer to delegate their signing capability to a group of proxy signers. Consider a scenario where a real estate owned by multiple owners needs to be rent out or sold. Suppose the owners delegate their signing capability to some proxy signers. If there is only one single proxy signer allowed to sign on behalf of the owners, then the single proxy signer's relationships with the owners are different from each other. He/She may issue some transaction documents which will meet some owners' interests but damage the interests of other owners. To address the drawback, a plausible solution is to allow the owners to choose their own proxy signers, any owner can designate a proxy signer. A transaction document is legal if and only if all the proxy signers sign on it. Another issue in existing PMS schemes is their communication complexity. Each orig-

① Supported by the National Basic Research Program of China (No. 2012CB315905), the National Natural Science Foundation of China (No. 61272501), the Fund of Tianjin Key Laboratory of Civil Aircraft Airworthiness and Maintenance in CAUC and a General grant from Civil Aviation Flight University of China (No. J2013-31, Q2014-48).

② To whom correspondence should be addressed. E-mail: ljh2583265@163.com

Received on Aug. 22, 2015

inal signer needs two-round broadcast operations. It's critical to reduce the number of interaction rounds.

Contribution of the study: Motivated by the above observations and the work of Ref. [12], this work revisits proxy multi-signatures. The contribution consists of two folds. First, a general framework to identity-based proxy multi-signature (IBPMS) is presented. In IBPMS, the original signers of a group are allowed to transfer their signing rights to a group of proxy signer candidates and any proxy signer candidate can sign a document on behalf of all the original signers alone. Second, an IBPMS scheme is proposed which is provably secure under standard computational assumptions. A striking feature of the IBPMS scheme is that it demands only one time of broadcasting operation for each original signer during the proxy key generation phase.

The rest of this paper is organized as follows. Some background knowledge associated with the work is given in Section 1. The outline of the proxy multi-signature scheme and security model are given in Section 2. In Section 3, a proxy multi- signature scheme from bilinear pairing is presented. Its formal security proofs will be given in Section 4. In Section 5, the efficiency of the proposed scheme is compared with some related work. conclusion is given in Section 6.

1 Syntax

In the conventional proxy signature definition, a proxy signer can sign a message on behalf of an original signer under the delegation of the original signer. In Ref. [13], Huang, et al. proposed a security model of a proxy signature scheme. This model is the most widespread used one for the security analysis. A delegation usually is produced by the original signer through an algorithm whose inputs are private key and a certain message. Therefore, a delegation can be seen as a special signature signed by the original signer(In most cases, it is a signature on a warrant).

Usually, there is only a single proxy signer in a PMS scheme[5-8]. All the original signers delegate their signing capability to one proxy signer. If the designated proxy signer is unavailable for something unexpected, then the protocol will be collapsed. Therefore, the proxy signer may be a bottleneck of a PMS scheme. One efficient solution to reduce the bottleneck effect is to increase the number of proxy signers and each proxy signer can sign a message on behalf of all the original signers separately.

1.1 Protocol variables and member relationship

It is assumed for simplicity that a polynomial-size set P is $= \{U_0, U_1, \cdots, U_l\}$ of potential players. Let P_0 denote the initial group. $\prod_{P_i}^{\iota_i}$ denotes an instance ι_i of a group member P_i. An original group $P_i^{\sum_O}$ denotes the group containing all original members through an instance $\sum$. $P_i^{\sum_P}$ denotes the group containing all proxy members and corresponds to $P_i^{\sum_O}$. Obviously, it always holds that $P_i^{\sum_O} \cup P_i^{\sum_P} = P_i$. Let $|A|$ denote the cardinality of a group A. Proxy signature schemes (PS) can be classified into six types.

Type T_1: If $|P_i^{\sum_P}| = P_i^{\sum_O}| = 1$, $P|_i^{\sum_P} \cap P_i^{\sum_O} = \varnothing$, then the PS scheme is called a type T_1 scheme.

Type T_2: If $|P_i^{\sum_O}| > 1$, $|P_i^{\sum_P}| = 1$, and $P|_i^{\sum_P} \cap P_i^{\sum_O} = \varnothing$, then the PS scheme is called a type T_2 scheme.

Type T_3: If $|P_i^{\sum_O}| > 1$, $|P_i^{\sum_P}| = 1$, $P|_i^{\sum_P} \cap P_i^{\sum_O} = P|_i^{\sum_P}$, then the PS scheme is called a type T_3 scheme.

Type T_4: If $|P_i^{\sum_O}| = 1$, $|P_i^{\sum_P}| > 1$, $P|_i^{\sum_P} \cap P_i^{\sum_O} = \varnothing$, then the PS scheme is called a type T_4 scheme.

Type T_5: If $|P_i^{\sum_O}| > 1$, $|P_i^{\sum_P}| > 1$, $P|_i^{\sum_P} \cap P_i^{\sum_O} = \varnothing$, then the PS scheme is called a type T_5 scheme.

Type T_6: If $|P_i^{\sum_O}| > 1$, $|P_i^{\sum_P}| > 1$, $P|_i^{\sum_P} \cap P_i^{\sum_O} = \varnothing$, then the PS scheme is called a type T_6 scheme.

Traditional proxy multi-signature[7] is a type T_2 scheme, a traditional multi-proxy signature scheme[3] is a type T_4 scheme, a (t, n) threshold proxy signature schemes[14] is a type T_4 scheme.

1.2 Bilinear parings

Let G_1 be an additive cyclic group with prime order q, and G_2 be a multiplicative cyclic group of the same order q. Admissible bilinear pairing map $\hat{e}: G_1 \times G_1 \rightarrow G_2$ is called an admissible bilinear paring if it satisfies bilinearity, non-degeneracy and computability.

The bilinear map can be constructed by suitable modification in Weil[15] or Tate pairings[16]. The group equipped with such a map is called a bilinear group, on which the Decisional Diffie-Hellman problem is able to be solved within a polynomial-time while the computational Diffie-Hellman problem is believed hard[17].

2 Modelling identity-based proxy multi-signature

Inspired by the works of Cao, et al. [7], Wang, et al. [18], and Rajeev, et al. [8], and Pointcheval[19], a formal definition and security model for identity-based proxy multi-signature schemes are given.

2.1 Definition of identity-based proxy multi-signature schemes

In an identity-based proxy multi-signature scheme, the original signers of a group are allowed to transfer their signing rights to a group of proxy signer candidates, in such a way that any proxy signer candidate can sign a document on behalf of all the original signers alone. Let $A_1, A_2, \cdots, A_m$ be the original signers and $B_1, B_2, \cdots, B_n$ be the proxy signer candidates designated by $A_1, A_2, \cdots, A_m$. For $1 \leqslant i \leqslant m$, A_i has an identity ID_{A_i}, for $1 \leqslant j \leqslant n$, B_j has an identity ID_{B_j}.

Definition 1 An identity-based proxy multi-signature scheme is a tuple IBPMS = (Setup; Extract; Sign; Veri; PMGen; PMSign; PMVeriT).

Setup: On the input security parameter l, PKG generates public parameters *Para* of the system and a master secret key s. PKG publishes *Para* and keeps confidential master key s.

Extract: Input master secret key s, public parameters *Para* and an identity ID, and output the private key S_{ID} of ID. PKG will use this algorithm to generate private keys for all entities participating in the scheme and send the private keys to their respective owners through a secure channel.

Sign: Input public parameters *Para*, signer's identity ID, his private key S_{ID}, and a message $\hat{m}$, and output a signature σ on $\hat{m}$.

Veri: On input public parameters *Para*, signer's identity ID, message $\hat{m}$ and a signature σ on $\hat{m}$, the algorithm outputs 1 if σ is a valid signature on $\hat{m}$ for identity ID, otherwise output 0.

PMGen: This is a protocol jointly executed by all the candidate proxy signers and all original signers. Input $ID_{A_1}, \cdots, ID_{A_m}, ID_{B_1}, \cdots, ID_{B_n}$, the original signers' private keys $S_{ID_{A_1}}, \cdots, S_{ID_{A_m}}$ and the delegation warrant ω which includes the type of the information delegated, the period of delegation, all the candidate proxy signers, etc. Any candidate proxy signer $B_j (j = 1, \cdots, n)$ can output a proxy signing key sk_{B_j} by inputting his secret key $S_{ID_{B_j}}$. The proxy signing key sk_{B_j} can be used by B_j to produce proxy multi-signature on behalf of the original signers.

PMSign: Input the proxy signing key sk_{B_j}, the warrant ω and the message $\hat{m} \in \{0,1\}^*$. Output a proxy multi-signature σ_{B_j} on $\hat{m}$.

PMVeri: Input $\hat{m}$, ω, σ_{B_j}, the identities $ID_{A_1}, \cdots, ID_{A_m}, ID_{B_j}$. If outputs 1 then σ_{B_j} is a valid proxy multi-signature for $\hat{m}$ by the proxy signer B_j, or outputs 0 otherwise.

2.2 Security model

A formal security model for an identity-based proxy multi-signature scheme based on the work of Refs [7,8,20] is given. It is considered that adversary A tries to forge a proxy multi-signature working against a single honest user 1. User 1 can be an original signer or a proxy signer adaptively. A is allowed to access standard signing oracle, delegation oracle, and proxy multi-signature oracle.

The goal of adversary A is to produce one of the following forgeries:

(1) A standard signature by user 1 for message $\hat{m}$ that was not submitted to the standard signature signing oracle.

(2) A proxy multi-signature for message $\hat{m}$ by user 1 on behalf of the original signers such that either the original signers never designated user 1, or $\hat{m}$ was not in a query made to the proxy multi-signing oracle.

(3) A proxy multi-signature for message $\hat{m}$ by some user $ID_i (ID_i \neq ID_1)$ on behalf of the original signers, such that user ID_i was never designed by the original signers, and user 1 is one of the original signers.

Consider the following game:

(1) Setup: The challenger runs the algorithm Setup of the proxy multi-signature scheme and provides the public parameters *Para* to A.

(2) Hash query: A can access the hash oracle, challenger X responds through the hash oracle and maintains L_{H_1}, L_{H_2} and L_{H_3} for each hash query.

(3) Extract query: Adversary A can ask for the private key of any user $ID_i (ID_i \neq ID_1)$. The challenger responds by running the Extract algorithm and returns the private key S_{ID_i} to A.

(4) Signing query: A can query oracle $O_S(S_{ID_i}, \cdot)$ on $\hat{m}$ of his choice, and obtains a standard signature for $\hat{m}$ by user 1.

(5) Delegation query: A is allowed to request for the proxy signing key on the warrant ω and the identity $ID_i (ID_i \neq ID_1)$. The user 1 may be either one of the original signers or one of the proxy signers.

(6) Proxy multi-signature query: Proceeding adaptively, A can request for a proxy multi-signature

on message $\hat{m}'$ with warrant ω' of its choice.

Definition 2 ID-based proxy multi-signature forger A $(t, q_H, q_E, q_s, q_{ps}, q_{pms}, m+n, \varepsilon)$-breaks the $m+n$ users ID-based proxy multi-signature scheme by the adaptive chosen message and given *ID* attack if: A runs in time at most t, and A makes at most q_H queries to the hash queries, at most q_E queries to the extraction queries, at most q_s queries to the signing queries, at most q_{ps} queries to the delegation queries and at most q_{pms} queries to the proxy multi-signature queries, and the success probability of A is ε at least.

Definition 3 An ID-based proxy multi-signature scheme is $(t, q_H, q_E, q_s, q_{ps}, q_{pms}, m+n, \varepsilon)$-secure against adaptive chosen message and given *ID* attack, if there is no adversary who can $(t, q_H, q_E, q_s, q_{ps}, q_{pms}, m+n, \varepsilon)$-break it.

3 Proposed identity-based proxy multi- signature scheme

In this section, an identity-based proxy multi-signature (IBPMS for short) scheme is presented based on the ID-based aggregate signature scheme[12] and IBPMS scheme[8]. The scheme has following phases: Setup, Extract, Sign, Veri, PMGen, PMSign, PMVeri.

Setup: It takes as input the system's parameter l, PKG generates two cyclic groups $(G_1, +)$ and $(G_2, \cdot)$ of order $q(q > 2^l)$, where P is a generator of the additive group G_1, and $\hat{e}$ is an admissible bilinear map $\hat{e}: G_1 \times G_1 \to G_2$. PKG randomly chooses an integer $s \in Z_q^*$, and sets $P_{pub} = sP \in G_1$. PKG selects four collision resistant hash functions $H_1: \{0,1\}^* \to G_1$, $H_2, H_3, H_4: \{0,1\}^* \to Z_q^*$. The public system parameters are $Para = (\hat{e}, l, q, P, G_1, G_2, H_1, H_2, H_3, H_4, P_{pub})$. The master key is s.

Extract: For a user with *ID*, PKG computes its public key as $Q_{ID} = H_1(ID) \in G_1$ and private key as $S_{ID} = sQ_{ID}$. Thus original signer A_i, has its public key Q_{A_i} (for $i = 1, \cdots, m$) and corresponding private key $S_{ID_{A_i}}$. Similarly, for the n proxy signers, the public keys are $Q_{ID_{B_j}}$ and corresponding private keys are $S_{ID_{B_j}}$ (for $j = 1, \cdots, n$).

Sign: To sign a message $\hat{m} \in \{0,1\}^*$, with a private key S_{ID}, randomly select $x \in Z_q^*$, and compute $V_s = xP$, $H = H_2(\hat{m})$, and $Ws = S_{ID} + xHP_{pub}$. The signature on message $\hat{m}$ is $\sigma = (V_s, W_s)$.

Veri: To verify a signature $\sigma = (V_s, W_s)$ on message $\hat{m}$ for an *ID*, the verifier computes $Q_{ID} = H_1(ID)$ and $H = H_2(\hat{m})$. The signature is accepted if $\hat{e}(W_s, P) = \hat{e}(Q_{ID} + HV_s, P_{pub})$, otherwise it is rejected.

The Sign and Veri algorithm above is the same algorithm as the Shim's IBS scheme[12], and for short the Shim's IBS scheme is denoted as SIBS.

PMGen: In this phase, the original signers perform the following job to make a message warrant ω, jointly with the proxy signers. ω includes some specific information about the message, restrictions on the message, time of delegation, identity of original and proxy signers, period of validity, and so on. Unlike the traditional proxy multi-signature schemes, the warrant ω in our scheme will declare a proxy signer group $\{B_j \mid 1 \leqslant j \leqslant n\}$, any B_j can sign a message on the behalf of the original signers $\{A_i \mid 1 \leqslant i \leqslant m\}$.

Delegation: To delegate the signing rights to the proxy signers, each original signer A_i (for $1 \leqslant i \leqslant m$) randomly chooses $t_i \in Z_q^*$ and computes $V_i = t_iP$, $h_2 = H_2(\omega) \in Z_q^*$, $W_i = S_{ID_{A_i}} + t_ih_2P_{pub}$, and broadcasts (W_i, V_i, ω) to the group of proxy signers.

Delegation verification: Each proxy signer B_j (for $1 \leqslant j \leqslant n$) computes $h_2 = H_2(\omega)$ and accepts the delegation value (W_i, V_i, ω) on warrant ω, if the equality $\hat{e}(W_i, P) = \hat{e}(Q_{ID_{A_i}} + h_2V_i, P_{pub})$ holds. Otherwise, B_j terminates the protocol.

Proxy key generation: After receiving the valid delegation value (W_i, V_i, ω) (for $1 \leqslant i \leqslant m$), each proxy signer B_j (for $1 \leqslant j \leqslant n$) sets $V = \sum_{i=1}^{m} V_i$ and $h_3 = H_3(\omega \| V)$, generates its proxy signing key as $sk_{B_j} = \sum_{i=1}^{m} W_i + h_3S_{ID_{B_j}}$.

PMSign: Using sk_{B_j}, $B_j (1 \leqslant j \leqslant n)$ can sign the message $\hat{m}$ under ω on behalf of the originals $A_i (1 \leqslant i \leqslant m)$. In this phase, B_j chooses $x_j \in_R Z_q^*$, and computes

$$V_{B_j} = x_jP; h_4 = H_4(\hat{m} \| \omega \| V) \in Z_q^*;$$
$$W_{B_j} = sk_{B_j} + x_jh_4P_{pub}$$

The proxy signature for message $\hat{m}$ issued by B_j on behalf of the original signers $A_1, \cdots, A_m$ is $\sigma_{B_j} = (W_{B_j}, V_{B_j}, V, \omega)$.

PMVeri: To verify a proxy multi-signature $\sigma_{B_j} = (W_{B_j}, V_{B_j}, V, \omega)$ on message $\hat{m}$ under a warrant ω, the verifier operates as follows:

1. Checks whether or not the message $\hat{m}$ conforms to the warrant ω. If not, stop. Otherwise, continue.

2. Checks whether or not the proxy signer B_j is on the authorized list in the warrant ω. If not, stop. Otherwise, continue.

3. Computes $\bar{h}_2 = H_2(\omega)$, $\bar{h}_3 = H_3(\omega \| V)$, $\bar{h}_4 = H_4(\hat{m} \| \omega \| V)$. Then checks if it holds that

$$\hat{e}(W_{B_j},P) = \hat{e}(\sum_{i=1}^{m} Q_{ID_{A_i}} + \bar{h}_2 V + \bar{h}_3 Q_{B_j} + \bar{h}_4 V_{B_j}, P_{\text{pub}})$$

Our PMS scheme allows n candidates of proxy signing $B_1, \cdots, B_n$. If $n = 1$, then our PMS scheme is a traditional PMS scheme, namely a T_2 proxy signature scheme; If $n > 1$, then the PMS scheme is a T_5 proxy signature scheme.

If set the number of proxy signers $n > 1$ in the PMS scheme, and use the PMS scheme to be a traditional PMS scheme, that is to say, a message can be checked as valid when one of the proxy signers signs on it. In this situation, the proposed PMS scheme enjoys better robustness than a traditional PMS scheme, because, a traditional PMS scheme will not work if the unique proxy signer is not available. Thus a T_5 scheme enjoys better robustness than a T_2 scheme if a message needs only one proxy signer to sign on it.

An e-transaction application instance: Suppose there is a real estate owned by multiple entities needed to be sold out. The owners authorize proxy signers to sign e-transaction documents. If there is only one single proxy signer allowed to sign messages on behalf of all the owners, then how to choose the single proxy signer admitted by all owners is another open problem. The relationship between the single proxy signer and owner A_i is different from that between the single proxy signer and owner $A_j (i \neq j)$. By using the proposed PMS scheme, each owner is allowed to choose her/his own proxy signer, then m original signers have n proxy signers, there $m = n$. Any e-transaction document can be verified as valid only if all the n proxy signers sign on it. The whole process contains m times PMSign signing and m times PMVeri verifying.

4 Analysis of the scheme

4.1 Correctness

1. The correctness of delegation process:

$$\begin{aligned}\hat{e}(W_i,P) &= \hat{e}(S_{ID_{A_i}} + t_i h_2 P_{\text{pub}}, P)\\ &= \hat{e}(sQ_{ID_{A_i}} + t_i h_2 sP, P)\\ &= \hat{e}(Q_{ID_{A_i}} + h_2 V_i, P_{\text{pub}})\end{aligned}$$

2. The correctness of PMSign and PMVeri algorithms from the following equalities:

$$\begin{aligned}\hat{e}(W_{B_j},P) &= \hat{e}(sk_{B_j} + x_j h_4 P_{\text{pub}}, P)\\ &= \hat{e}(\sum_{i=1}^{m} W_i + h_3 S_{ID_{B_j}} + x_j h_4 P_{\text{pub}}, P)\\ &= \hat{e}(\sum_{i=1}^{m} Q_{ID_{A_i}} + \bar{h}_2 V + \bar{h}_3 Q_{B_j} + \bar{h}_4 V_{B_j}, P_{\text{pub}})\end{aligned}$$

4.2 Security proof

Theorem 1. Given a security parameter l, let G_1 be a (t', ε')-*CDH* group of prime order $q > 2^l$, P be a generator of G_1, and $\hat{e}: G_1 \times G_1 \to G_2$ be a bilinear map. Then the identity-based proxy multi-signature scheme on G_1 is $(t, q_H, q_E, q_s, q_{ps}, q_{pms}, m+n, \varepsilon)$-secure against forgery for any t and ε satisfying

$$\begin{aligned}\varepsilon &\geqslant e(q_E+1)\cdot(1+(1-1/(q_E+1))^{q_{pms}})\\ &\quad + (1-1/(qE+1))^{q_s+q_{ps}+q_{pms}})^{-1}\varepsilon'\\ t &\leqslant t' - C_{G_1}(q_{H_1}+q_{H_2}+q_{H_3}+q_{H_4}+2q_E\\ &\quad + 3q_s + 3q_{ps} + 3q_{pms} + 9)\end{aligned}$$

where e is the base of natural logarithms, and C_{G_1} is the time of computing a scalar multiplication and inversion on G_1.

Proof. Suppose adversary $(t, q_H, q_E, q_s, q_{ps}, q_{pms}, m+n, \varepsilon)$-breaks the proxy multi-signature scheme. X is given $X = xP$ and $Y = yP$. Its goal is to output $xY = xyP$. X interacts with A as follows:

Setup: Algorithm X initializes A with $P_{pub} = X$ as a system's master public key. A selects an identity ID_1.

H_1-queries: At any time A can query the random oracle O_{H_1}. To respond to these queries, X maintains a list L_{H_1} of tuples (ID_i, Q_i, b_i, c_i). When an identity ID_i is submitted to the O_{H_1}, X responds as follows:

(1) If the query ID_i already appears on the list L_{H_1} in some tuple (ID_i, Q_i, b_i, c_i), then algorithm X responds with $H_1(ID_i) = Q_i$.

(2) Otherwise, X generates a random coin $c \in \{0,1\}$ such that $\Pr[c = 0] = \lambda$.

(3) Algorithm X picks a random $b_i \in Z_q^*$. If $c = 0$, algorithm X sets $Q_i = b_i Y$, If $c = 1$, X sets $Q_i = b_i P$.

(4) Algorithm X adds the tuple (ID_i, Q_i, b_i, c_i) to the list L_{H_1} and responds to A with $H_1(ID_i) = Q_i$.

H_2-queries: A can query the random oracle O_{H_2}. X maintains a list L_{H_2} of tuples (ω_i, v_i). When a warrant ω_i is submitted to the O_{H_2}, X responds as follows:

(1) If the query ω_i already appears on the list L_{H_2} in some tuple (ω_i, v_i) then algorithm X responds with $H_2(\omega_i) = v_i$.

(2) Otherwise, X picks a random $v_i \in Z_q^*$, adds the tuple (ω_i, v_i) to the list L_{H_2} and responds to A with $H_2(\omega_i) = v_i$.

H_3-queries: At any time A can query the random oracle O_{H_3} with (ω, V). X maintains a list L_{H_3} of tuples (ω, V, η). X responds as follows:

(1) If the query $(\hat{m}, \omega)$ already appears on the L_{H_3} then X responds with $H_3(\omega \| V) = \eta$.

(2) Otherwise, X picks a random $\eta \in Z_q^*$, adds (ω, V, η) to L_{H_3} and returns $H_3(\omega \| V) = \eta$.

H_4-queries: At any time A can query the random oracle O_{H_4}. To respond to these queries, X maintains a list L_{H_4} of tuples $(\hat{m}, \omega, V, \gamma)$. When a tuple $(\hat{m}, \omega, V)$ is submitted to O_{H_4}, algorithm X responds as follows:

(1) If query $(\hat{m}, \omega, V)$ already appears on the list L_{H_4} in some tuple $(\hat{m}, \omega, V, \gamma)$, then algorithm X responds with $H_4(\hat{m} \| \omega \| V) = \gamma$.

(2) Otherwise, X picks $\gamma \in_R Z_q^*$, and adds $(\hat{m}, \omega, V, \gamma)$ to L_{H_4} and returns γ.

Extraction queries: Let $ID_i (i \neq 1)$ be a private key extraction query issued by algorithm A.

(1) X runs the above algorithm for responding to H_1-queries to obtain a $Q_i \in G_1$ such that $H_1(ID_i) = Q_i$. Let (ID_i, Q_i, b_i, c_i) be the corresponding tuple on the list L_{H_1}. If $c_i = 0$, then X outputs "failure" and terminates.

(2) Otherwise $c_i = 1$ and $Q_i = b_i P$. Define $S_{ID_i} = b_i P_{pub}$. It is seen that $S_{ID_i} = b_i x P = x Q_i$ and therefore S_{ID_i} is the private key associated with the public key ID_i. Returns S_{ID_i}. The probability of success is $1 - \lambda$.

Signing queries: A is allowed to requests O_s for standard signature with $(ID_i, \hat{m}_i)$. X maintains a list L_s of tuples $(ID_i, m_i, V_{s_i}, W_{s_i})$. When $(ID_i, \hat{m}_i)$ is submitted to the O_s, X responds as follows:

(1) If the query $(ID_i, \hat{m}_i)$ already appears on the list L_s in some tuple $(ID_i, m_i, V_{s_i}, W_{s_i})$, then algorithm X responds with (W_{s_i}, V_{s_i}).

(2) Otherwise, algorithm X recovers (ID_i, Q_i, b_i, c_i) and (ω_i, v_i), chooses $t_i \in_R Z_q^*$ sets $V_{s_i} = t_i P \in G_1$ and $W_{s_i} = (t_i v_i + b_i) P_{pub} \in G_1$. The pair (W_{s_i}, V_{s_i}) is a valid signature on message $\hat{m}_i$ under ID_i. Then algorithm X returns (W_{s_i}, V_{s_i}) and adds it to L_s.

Delegation queries:

Case 1: A requests to interact with ID_1 where ID_1 plays the role of one of the proxy signers. For this, A generates a warrant ω, obtaining $H_2(\omega) = v_i$ by accessing the O_{H_2} oracle. Then algorithm X chooses $t_i \in_R Z_q^*$ and computes $W_{A_i} = S_{ID_{A_i}} + t_i H_2(\omega) P_{pub}$, where $S_{ID_{A_i}}$ is the private key of the original signer A_i and algorithm X sets $V_{A_i} = t_i P$. Then A submits $(W_{A_i}, V_{A_i}, \omega)$ to X. X returns a corresponding partial proxy signing key sk_{B_j} which involves all (W_{A_i}, V_{A_i}), $(1 \leqslant i \leqslant m)$. The tuple $((W_{A_1}, V_{A_1}), \cdots, (W_{A_m}, V_{A_m}), \omega, sk_{B_j})$ is added to the proxy key generation list L_{ps_p}.

Case 2: A requests to interact with ID_1, where ID_1 plays the role of one of the original signers. To responds to this query, A generates a warrant ω, and requests ID_1 to sign ω and receives a response $(W_{A_1}, V_{A_1}, \omega)$. X returns a partial proxy signing key sk_{B_j} which involves $(W_{A_1}, V_{A_1}, \omega)$ and adds $((W_{A_1}, V_{A_1}), \cdots, (W_{A_m}, V_{A_m}), \omega, sk_{B_j})$ to L_{ps_o}.

In either of the above cases, X recovers (ω_i, v_i) on L_{H_2} and gets $H_2(\omega_i) = v_i$. If $c = 0$, X returns "failure" and terminates. If $c = 1$, it is known that $H_1(ID_{A_i}) = b_{A_i} P$ and $H_1(ID_{B_j}) = b_{B_j} P$. Considering these scenarios let $W_{A_i} = (b_{A_i} + t_i v_i) P_{pub}$ then one can check the following equation for $1 \leqslant i \leqslant m$: $\hat{e}(W_{A_i}, P) = \hat{e}(Q_{ID_{A_i}} + H_2(\omega) V_{A_i}, P_{pub})$.

Hence the above provided proxy signing key which involves (W_{A_i}, V_{A_i}) is valid. The probability of success is $1 - \lambda$.

Proxy multi-signature queries: Proceeding adaptively, when the adversary A requests for a proxy multi-signature on message $\hat{m}$ with the proxy signer B_j, satisfying the warrant ω, X responds as follows:

(1) If the query $(B_j, \hat{m}, \omega)$ already appears on the list L_{pms} in some tuple $(B_j, \hat{m}, W_{B_j}, V_{B_j}, V, \omega)$, then algorithm X responds with (W_{B_j}, V_{B_j}, V).

(2) X runs the above algorithm for responding to H_2-queries on ω, recovers (ω, v) on L_{H_2} list.

(3) If $c = 0$, then X returns "failure" and terminates. If $c = 1$, $H_1(ID_{A_i}) = b_{A_i} P$ or $H_1(ID_{B_j}) = b_{B_j} P$.

Now X chooses $x_j \in_R Z_q^*$ and computes $V = \sum_{i=1}^{m} t_i P$ and $V_{B_j} = x_j P$, gets (ω, v), (ω, V, η) and $(\hat{m}, \omega, V, \gamma)$ from requesting H_2, H_3, H_4 queries respectively. Then X sets $W_{B_j} = (\sum_{i=1}^{m} b_{A_i} + v \sum_{i=1}^{m} t_i + \eta b_{B_j} + \gamma x_j) P_{pub}$. So $\sigma_{B_j} = (W_{B_j}, V_{B_j}, V, \omega)$ is the proxy multi-signature on message $\hat{m}$ with a warrant ω'_i issued by B_j on behalf of $A_1, \cdots, A_m$. The tuple $(B_j, \hat{m}, W_{B_j}, V_{B_j}, V, \omega)$ is added to the list L_{pms}. The probability of success is at least λ.

Hence, the probability that C does not abort during the simulation is $(1 - \lambda)^{q_E + q_{ps} + q_{pms}}$.

Output: If X does not terminate as a result of A's extraction query and proxy multi-signature query, then A's view is identical to its view in the real attack.

Case 1. According to the Forking Lemma[20], after replaying A using the same random tape, X obtains two valid signatures by running the SIBS scheme[14]. X obtains $(V^*, W^*, \hat{m}^*, H_2(\hat{m}^*))$ and $(V^*, W^{*\prime}, \hat{m}^*, H'_2(\hat{m}^*))$ within a polynomial time, where

$\hat{e}(W^*, P) = \hat{e}(H(\hat{m}^*) Q_{ID_{A_1}} + V^*, P_{pub})$;

$\hat{e}(W^{*\prime}, P) = \hat{e}(H'_2(\hat{m}^*) Q_{ID_{A_1}} + V^*, P_{pub})$

X recovers $(ID_{A_1}, Q_{A_1}, b_{A_1}, c_{A_1})$ from L_{H_1}. If C_{A_1}

$= 1$, X outputs "failure" and terminates. Otherwise, it continues, sets $H_2(\hat{m}^*) = v^*$ and $H'_2(\hat{m}^*) = v^{*'}$, Then X can deduce

$\hat{e}(W^*, P) = \hat{e}(v^* b_{A_1} Y + V^*, P_{pub})$;

$\hat{e}(W^{*'}, P) = \hat{e}(v^{*'} b_{A_1} Y + V^*, P_{pub})$;

And it holds that

$\hat{e}(W^* - W^{*'}, P) = \hat{e}((v^* - v^{*'}) b_{A_1} Y, P_{pub})$

X outputs the required xY as $(v^* - v^{*'})^{-1} b_{A_1}^{-1}(W^* - W^{*'})$. There are three events needed for X to succeed: $\boldsymbol{E}_1$: X does not abort as a result of any of A'sσ Extraction queries. $\boldsymbol{E}_2$: A generates a valid and nontrivial signature forgery $\sigma = (V_s, W_s)$ on m. $\boldsymbol{E}_3$:$\boldsymbol{E}_2$ occurs and $c = 0$ for the related tuple on the L_{H_1}.

X succeeds if all of these events happen.

$\Pr[E_1 \wedge E_2 \wedge E_3]$

$= \Pr[E_1] \Pr[E_2 \mid E_1] \Pr[E_3 \mid E_1 \wedge E_2]$.

Since A makes at most q_E queries to the Extraction oracle and $\Pr[c = 1] = 1 - \lambda$, then $\Pr[E_1] = \lambda(1 - \lambda)^{q_E}$. If X does not abort as a result of A's Extraction query then A's view is identical to its view in the real attack. Hence, $\Pr[E_2 \mid E_1] \geqslant \varepsilon$. X will abort only if A generates a forgery such that $c = 1$. Hence, $\Pr[E_3 \mid E_1 \wedge E_2] \geqslant 1/(1 - \lambda)$. Thus, the probability of success is at least $\lambda(1 - \lambda)^{q_E}\varepsilon$.

Case 2: A simulates as a malicious proxy signer. User ID_1 is playing the role of one original signer ID_{A_1}. For ID_{A_1}, A does not access the Extraction query, and does not request a delegation query, and does not request proxy multi-signature query related with ID_{A_1}. Eventually, A outputs a valid proxy multi-signature forgery $\sigma_{B_j} = (W_{B_j}, V_{B_j}, V, \omega)$ on $\hat{m}$ issued by B_j on behalf of $A_1, \cdots, A_m$. The $\sigma_{B_j} = (W_{B_j}, V_{B_j}, V, \omega)$ should satisfy PMVeri equation

$\hat{e}(W_{B_j}, P) = \hat{e}(\sum_{i=1}^{m} Q_{ID_{A_i}} + h_2 V + h_3 Q_{B_j} + h_4 V_{B_j}, P_{pub})$

Sets $W'_{B_j} = \sum_{i=2}^{m} S_{ID_{A_i}} + h_2 \sum_{i=2}^{m} V_{s_i} + h_3 S_{ID_{B_j}} + h_4 t_{B_j} P_{pub}$ and $W_{A_1} = W_{B_j} - W'_{B_j}$, then we have

$\hat{e}(W'_{A_1}, P) = \hat{e}(Q_{ID_{A_1}} + h_2 V_{s_1}, P_{pub})$;

So (W'_{A_1}, V_{s_1}) is a valid forgery of the SIBS scheme. The probability of success is $\lambda \cdot (1 - \lambda)^{q_E + q_{pms}} \cdot \varepsilon$.

Case 3: When A requests to interact with a user ID_1, where ID_1 is playing the role of proxy signer ID_{B_1}. For ID_{B_1}, A does not request the private key in extraction query, does not request proxy multi-signature query using $(B_j, \hat{m}^*)$. Similarly to the above Case (2), it can show that X outputs a valid forgery of the SIBS scheme with the success probability $\lambda \cdot (1 - \lambda)^{q_E + q_s + q_{ps} + q_{pms}} \cdot \varepsilon$.

Hence the success probability that X solves the CDHP in the above game is at least:

$((1 - \lambda)^{q_E} + (1 - \lambda)^{q_E + q_{pms}} + (1 - \lambda)^{q_E + q_s + q_{ps} + q_{pms}})\lambda\varepsilon$

Set $\lambda = 1/(q_E + 1)$, we can deduce that

$((1 - 1/(q_E + 1))^{q_E} + (1 - 1/(q_E + 1))^{q_E + q_{pms}} + (1 - 1/(q_E + 1))^{q_E + q_s + q_{ps} + q_{pms}}) 1/(q_E + 1)\varepsilon \geqslant (1/e) \cdot 1/(q_E + 1) \cdot (1 + (1\text{-}1/(q_E + 1))^{q_{pms}} + (1 - 1/(qE + 1))^{q_E + q_s + q_{ps} + q_{pms}})\varepsilon \geqslant \varepsilon'$.

Therefore

$\varepsilon \geqslant e(q_E + 1) \cdot (1 + (1 - 1/(q_E + 1))^{q_{pms}} + (1 - 1/(qE + 1))^{q_E + q_s + q_{ps} + q_{pms}})^{-1} \varepsilon'$

For the running time, one can observe that the running time of X is the same as A running time plus the time taken to respond to $q_{H_1}, q_{H_2}, q_{H_3}, q_{H_4}$ hash queries, q_E extraction queries, q_s signing queries, q_{ps} delegation queries and q_{pms} proxy multi-signature queries, and the time to transform A's final forgery into the CDH solution. Hence, the total running time is at most $t + C_{G_1}(q_{H_1} + q_{H_2} + q_{H_3} + q_{H_4} + 2q_E + 3q_s + 3q_{ps} + 3q_{pms} + 9) \leqslant t'$ as required.

5 Comparison

The efficiency of the scheme is compared with the schemes in Refs[6-8]. In Table 1, E denotes the exponentiation operation in G_2, M denotes the point scalar multiplication operation in G_1, P denotes the pairing operation and NoPS denotes the number of proxy signer candidates. The broadcasting round in which each original signer needs to execute in the PMGen is denoted by BREO.

Table 1 Performance analysis

Schemes	PMGen	PMSign	PMVeri	Provable security	BREO	NoPS
Scheme in Ref. [6]	$(3m + 1)M + 3mP + mE$	$1P + 1E + 2M +$	$3P + 2E$	No	1	1
Scheme in Ref. [7]	$(2m + 1)M + 3mP$	$2M$	$4P + M$	Yes	2	1
Scheme in Ref. [8]	$(3m + 1)M + 3mP + mE$	$2M$	$M + 3P + E$	Yes	2	1
Our scheme	$(3m + 1)M + mP$	$2M$	$3M + 2P$	Yes	1	multiple

From Table 1, it can be seen that the proposed scheme is more efficient than the schemes in Refs[6-8]. Especially, the PMS scheme demands only one round broadcasting operation for each original signer, and the proxy signer candidates is not unique. These two characteristics make our PMS scheme more practical and efficient than the other schemes. It is supposed a valid message only needs the signature of one of the proxy signers.

6 Conclusion

In this work, a novel proxy multi-signature scheme is presented. The proposed proxy multi-signature scheme demands only one round broadcasting operation for each original signer during the proxy key generation phase. The scheme allows multiple proxy signers, improves the reliability of the PMS scheme. A formal security proof for the proposed scheme is also proposed.

References

[1] Boldyreva A. Threshold signatures, multisignatures and blind signatures based on the Gap-Diffie-Hellman-group signature scheme. In: Proceeding of the Public Key Cryptography, Miami, USA, 2003. 31-46

[2] Mambo M, Usuda K, Okamoto E. Proxy signature: delegation of the power to sign messages. *IEICE Transactions on Fundamentals of Electronics Communications & Computer Sciences*, 1996, E79-A(9):1338-1353

[3] Cao F, Cao Z F. A secure identity-based multi-proxy signature scheme. *Computers and Electrical Engineering*, 2009, 35:86-95

[4] Yi L, Bai G, Xiao G. Proxy multi-signature scheme: a new type of proxy signature scheme. *Electronics Letters*, 2000, (36):527-528

[5] Li X X, Chen K F, Li S Q. Multi-proxy signature and proxy multi-signature schemes from bilinear pairings. In: Proceedings of the 5th International Conference on Parallel and Distributed Computing, Application and Technologies, Singapore, 2004, 2004. 591-595

[6] Li X X, Chen K F. ID-based multi-proxy signature, proxy multi-signature and multi-proxy multi-signature schemes from bilinear pairings. *Applied Mathematics and Computation*, 2005, 169(1):437-450

[7] Cao F, Cao Z. A secure identity-based proxy multi-signature scheme. *Information Sciences*, 2009, 179(3):292-302

[8] Rajeev A S, Sahadeo P. Efficient ID-based proxy multi-signature scheme secure in random oracle. *Frontiers of Computer Science*, 2012, 6(4):421-428

[9] Du H, Wen Q. Certificateless proxy multi-signature. *Information Sciences*, 2014, 276:21-30

[10] Asaar M R, Salmasizadeh M, Susilo W. An identity-based multi-proxy multi-signature scheme without bilinear pairings and its variants. *Computer Journal*, 2013, 58(4):1021-1039

[11] Sahu R A, Padhye S. Identity-based multi-proxy multisignature scheme provably secure in random oracle model. *Transactions on Emerging Telecommunications Technologies*, 2015, 26(4):547-558

[12] Shim K. An ID-based aggregate signature scheme with constant pairing computations. *The Journal of Systems and Soft-ware*, 2010, 83:1873-1880

[13] Huang X Y, Mu Y, Willey S, et al. Proxy signature without random oracles. In: Proceedings of the Mobile ad-hoc and Sensor Networks, Hong Kong, China, 2006. 473-484

[14] Zhang K. Threshold proxy signature schemes. *Lecture Notes in Computer Science*, 1997, 1396: 191-197

[15] Boneh D, Franklin M. Identity-based encryption from the weil pairing. In: Proceedings of the Crypto, Santa Barbara, USA, 2001. 213-229

[16] Miyaji A, Nakabayashi M, Takano S. New explicit conditions of elliptic curve traces for fr-reduction. *IEICE Transactions on Fundamentals*, 2001, 5:1234-1243

[17] Okamoto T, Pointcheval D. The gap-problems: a new class of problems for the security of cryptographic schemes. In: Proceedings of the Public key Cryptopraphy, Cheju Island, Korea, 2001. 104-118

[18] Wang Q, Cao Z, Wang S. Formalized security model of multiproxy signature schemes. In: Proceedings of the 5th International Conference on Computer and Information Technology, Shanghai, China, 2005. 668-672

[19] Pointcheval D, Stern J. Security arguments for digital signatures and blind signatures. *Journal of Cryptology*, 2000, 13:361-396

[20] Shao Z. Improvement of identity-based proxy multi-signature scheme. *Journal of Systems and Software*, 2009, 85:794-800

Liu Jianhua, born in 1983. He received his Ph. D degrees in School of Electronic and Information Engineering of Beihang University in 2013. He also received his B. S. and M. S. degrees from China West Normal University in 2006 and 2009 respectively. His research interests include information security.