A Secure Anonymous Proxy Multi-signature Scheme

Vishal Saraswat and Rajeev Anand Sahu
C.R.Rao Advanced Institute of Mathematics Statistics and Computer Science
Hyderabad, India

Keywords: Identity-based Cryptography, Bilinear Map, CDHP, Provable Security, Digital Signature, Signing Rights Delegation, Proxy Multi-signature, Anonymity, Accountability.

Abstract: A proxy signature scheme enables a signer to delegate its signing rights to any other user, called the proxy signer, to produce a signature on its behalf. In a proxy multi-signature scheme, the proxy signer can produce one single signature on behalf of multiple original signers. We propose an efficient and provably secure threshold-anonymous identity-based proxy multi-signature (IBPMS) scheme which provides anonymity to the proxy signer while also providing a threshold mechanism to the original signers to expose the identity of the proxy signer in case of misuse. The proposed scheme is proved secure against adaptive chosen-message and adaptive chosen-ID attacks under the computational Diffie-Hellman assumption. We compare our scheme with the recently proposed anonymous proxy multi-signature scheme and other ID-based proxy multi-signature schemes, and show that our scheme requires significantly less operation time in the practical implementation and thus it is more efficient in computation than the existing schemes.

1 INTRODUCTION

Digital signature is a cryptographic primitive to guarantee data integrity, entity authentication and signer's non-repudiation. A proxy signature scheme enables a signer, O, also called the *designator* or *delegator*, to delegate its signing rights (without transferring the private key) to another user $\mathcal{P}$, called the *proxy signer*, to produce, on the delegator's behalf, signatures that can be verified by a verifier V under the delegator O's public key. For example, the director of a company may authorize the deputy director to sign certain messages on his behalf during a certain period of his absence.

Proxy multi-signature is a proxy signature primitive which enables a group of original signers $O_1, \dots, O_n$ to transfer their signing rights to a proxy signer $\mathcal{P}$ who can produce one single signature which convinces the verifier V of the concurrence of all the original signers. Threshold anonymous proxy multi-signature provides anonymity to the proxy signer while also providing a (t,n)-threshold mechanism to the original signers to expose the identity of the proxy signer in case of misuse. The *proxy identification algorithm* in the standard proxy signature protocol is replaced by a *proxy exposure protocol* where any t (or more) out of n original signers can come together to expose the identity of the proxy signer. Note that the *threshold anonymous* proxy multi-signature is different from *threshold proxy signatures* in the sense that while in threshold proxy signatures, any t (or more) out of n proxy signers must come together to produce a valid signature, in threshold anonymous proxy multi-signature any t (or more) out of n original signers must come together to revoke the anonymity of the proxy signer.

Consider the following example in a secure multiparty computation setting, multiple parties $O_1, \dots, O_n$ start a process $\mathcal{P}$ after authenticating themselves. Once the process $\mathcal{P}$ is started, the parties do not need to stay connected while the process $\mathcal{P}$ may remain active and need access to additional resources that require further authentication. The parties thus delegate their rights to the process $\mathcal{P}$ and the resources allow access to $\mathcal{P}$ as long as the resources can verify that $\mathcal{P}$ was indeed authorised by the original parties. The resources do not need to know the 'identity' of the process at all and $\mathcal{P}$ may remain anonymous to them. In fact, most of the times the resources do not even need to know whether it is actually the set of original parties who were authenticated or their proxy $\mathcal{P}$. But in case of a malicious process, the original parties should be able to expose the process and restrict any further activities by it on their behalf.

Saraswat V. and Sahu R..
A Secure Anonymous Proxy Multi-signature Scheme.
DOI: 10.5220/0005021200550066
In *Proceedings of the 11th International Conference on Security and Cryptography* (SECRYPT-2014), pages 55-66
ISBN: 978-989-758-045-1

1.1 Related Work

The notion of proxy signature has been around since 1989 (Gasser et al., 1989) but it took almost seven years for the first construction (Mambo et al., 1996) of a proxy signature scheme to be proposed. Since then many variants of the proxy signature have been proposed and many extensions of the basic proxy signature primitive have been studied. The formal security model of proxy signatures was first formalized in (Boldyreva et al., 2003) and further strengthened and extended to the identity-based setting in (Schuldt et al., 2008). A formal security model for anonymous proxy signatures was introduced relatively recently in (Fuchsbauer and Pointcheval, 2008) by unifying the notions of proxy signatures and group signatures.

The notion of proxy-anonymous proxy signatures was introduced in (Shum and Wei, 2002). Their scheme was based on the proxy signature scheme of (Lee et al., 2001) which was shown insecure in (Sun and Hsieh, 2003). The anonymization technique itself was shown to cause insecurity – (Lee and Lee, 2005) showed that the original signer can generate valid proxy signatures, thus violating the property of the strong unforgeability.

Many proxy signature schemes have since been proposed with the aim of providing anonymity of the proxy signers — (Yu et al., 2009; Toluee et al., 2012) provide proxy-anonymity by having a large "ring" of proxy-signers; (Wu et al., 2008; Fuchsbauer and Pointcheval, 2008) require a large "group" of proxy signers with one or more *group managers* to revoke the anonymity of a malicious proxy-signer; and (Lee et al., 2005; Du and Wang, 2013) require a *trusted third-party* or *trusted authority* or *trusted dealer* to provide the required functionality. In the ring-based and group-based settings, the proxy signature schemes require that the number of proxy signers authorized by the original signer is large enough to provide sufficient anonymity to the proxy signer. The cost (time, space, etc.) of providing anonymity is rather large and the anonymity is not even absolute but only 1-out-of-n where n is the size of group or ring. The trusted third-party setting has its fair share of well-known issues including the requirement of an absolutely trustworthy authority/ dealer who is always available.

The recently proposed scheme of (Du and Wang, 2013) is most notable in its attempt but it comes with several flaws. First and foremost, their implementation of the anonymization technique is not correct and because of that the signature verification cannot be done. In particular their proxy key generation is not consistent — they are adding a group element with scalar elements (integers). Thus their scheme is not consistent and is in fact incorrect. Second, it is not even a proxy multi-signature scheme but is just a concatenation of n proxy signature schemes, where n is the number of original signers, since each original signer in the scheme issues a warrant with a different *pseudonym*: $Q_{pseu_i} = R_O + R_{\mathcal{P}_i} + Q_{\mathcal{P}}$. Third, each original signer O_i can reveal the identity of the proxy signer $\mathcal{P}$ and even try to (partially) "demonstrate" by using the signature of the proxy. Fourth, during the proxy multi-signature verification $\mathsf{PMSVeri}$, it is required that the verifier "checks whether or not the proxy signer $\mathcal{P}$ is authorized by the n original signers $O_1, \ldots, O_n$ in the warrant w". Nevertheless, if the verifier can already check the authority of the proxy signer $\mathcal{P}$ then $\mathcal{P}$ never remains anonymous! Fifth, the dealer D of the secret sharing scheme used by the original signers also knows the identity of $\mathcal{P}$ and can also compute $R_{\mathcal{P}_i}$ from the R_O (which he computes himself) and publicly available values $Q_{\mathcal{P}}$ and Q_{pseu}. Finally, the scheme of (Du and Wang, 2013) is based on the proxy multi-signature scheme of (Cao and Cao, 2009) which can be shown to be insecure (Xiong et al., 2011) when $n = 1$.

1.2 Our Contribution

To the best of our knowledge, almost all available proxy-anonymous signature schemes are either too costly or inefficient to be practical or have not been proved secure. We propose an efficient and provably secure threshold-anonymous ID-based proxy multi-signature scheme which provides anonymity to the proxy signer while also providing a threshold mechanism to the original signers to expose the identity of the proxy signer in case of misuse. The proposed scheme is proved secure against adaptive chosen-message and adaptive chosen-ID attacks under the computational Diffie-Hellman (CDH) assumption.

In this paper, we build our scheme on the technique of *pseudonym* and secret sharing as suggested by (Du and Wang, 2013) to provide the required functionality – the identity of the proxy signer is hidden but in case of misuse of the delegated rights, t or more of the n original signers can come together to reveal the proxy signer's identity.

In our scheme we modify the structure of the warrant slightly. As in usual proxy signature schemes, the warrant in our scheme includes the nature of message to be delegated, period of delegation, identity information of original signers, etc. But unlike usual proxy signature schemes, it does not include the identity information of the proxy signer. Instead, the warrant includes the proxy signer's *pseudonym*, which is a proxy signature verification key that cannot be linked to the identity of the proxy signer easily — t or more original signers must come together to reveal the proxy signer's identity.

Compared with the scheme of (Du and Wang, 2013), our scheme allows the proxy signer to act as the dealer of the secret sharing scheme and uses a verifiable secret sharing scheme (Pedersen, 1991) to restrict the proxy from acting as a malicious dealer. Our scheme requires only $2n$ broadcasts compared to $3n+1$ of Du's scheme to construct the *pseudonym* of the proxy and thus our scheme requires 33% less broadcasts to provide anonymity. Also we use a much more efficient and provably secure proxy multi-signature scheme of (Sahu and Padhye, 2012) as our basic scheme so that the overall proxy signature has less operation time and thus more efficient (14%-23% more) than the existing best schemes in computation.

1.3 Outline of the Paper

The rest of this paper is organized as follows. In Section 2, we introduce some related mathematical definitions, problems and assumptions. In Section 3, we present the formal definition of an anonymous ID-based proxy multi-signature scheme and a security model for it. Our proposed anonymous ID-based proxy multi-signature scheme is presented in Section 4. In Section 5 we analyze the security of our scheme. Finally, Section 6 includes the efficiency comparison.

2 PRELIMINARIES

In this section, we introduce some relevant definitions, mathematical problems and assumptions and briefly discuss the verifiable secret sharing scheme.

2.1 Bilinear Map

Let G_1 be an additive cyclic group with generator P and G_2 be a multiplicative cyclic group with generator g. Let the both groups are of the same prime order q. Then a map $e: G_1 \times G_1 \rightarrow G_2$ satisfying the following properties, is called a *cryptographic* bilinear map:

1. *Bilinearity*: For all $a,b \in \mathbb{Z}_q^*$, $e(aP,bP) = e(P,P)^{ab}$, or equivalently, for all $Q,R,S \in G_1$, $e(Q+R,S) = e(Q,S)e(R,S)$ and $e(Q,R+S) = e(Q,R)e(Q,S)$.
2. *Non-Degeneracy*: There exists $Q,R \in G_1$ such that $e(Q,R) \neq 1$. Note that since G_1 and G_2 are groups of prime order, this condition is equivalent to the condition $e(P,P) \neq 1$, which again is equivalent to the condition that $e(P,P)$ is a generator of G_2.
3. *Computability*: There exists an efficient algorithm to compute $e(Q,R) \in G_2$, for any $Q,R \in G_1$.

2.2 Discrete log (DL) Assumption

Let G_1 be a cyclic group with generator P.

Definition 1. Given a random element $Q \in G_1$, the *discrete log problem* (DLP) in G_1 is to compute an integer $n \in \mathbb{Z}_q^*$ such that $Q = nP$.

Definition 2. The *DL assumption* on G_1 states that the probability of any polynomial-time algorithm to solve the DL problem in G_1 is negligible.

2.3 Computational Diffie-Hellman (CDH) Assumption

Let G_1 be a cyclic group with generator P.

Definition 3. Let $a,b \in \mathbb{Z}_q^*$ be randomly chosen and kept secret. Given $P, aP, bP \in G_1$, the *computational Diffie-Hellman problem* (CDHP) is to compute $abP \in G_1$.

Definition 4. The (t,ε)-*CDH assumption* holds in G_1 if there is no algorithm which takes at most t running time and can solve CDHP with at least a non-negligible advantage ε.

2.4 Verifiable Secret Sharing

The notion of secret sharing was introduced independently by (Shamir, 1979) and (Blakley, 1979) to enable a secret to be shared among a group of users so that the secret can be reconstructed only when a sufficient number of them come together. For integers n and t such that $1 < t \leq n$, an (t,n)-secret sharing scheme consists of two phases:

1. in the *splitting phase*, a dealer shares a secret σ among n users;
2. in the *combining phase*, only t or more users in the group can reconstruct the secret σ.

Verifiable secret sharing (VSS), introduced in (Chor et al., 1985), enables each user to verify the correctness of their shares to prevent malicious attack performed by the dealers. For the purpose of this paper, we use Pedersen's non-interactive and information theoretic secure VSS (Pedersen, 1991). This scheme protects the secret to be distributed unconditionally for any value of t, $(1 < t \leq n)$, and the correctness of the shares depends on the assumption that the dealer cannot find discrete logarithms before the distribution has been completed.

3 ANONYMOUS ID-BASED PROXY MULTI-SIGNATURE SCHEME AND ITS SECURITY MODEL

Here we give a formal definition of an anonymous ID-based proxy multi-signature scheme and a formal security model for it as presented in (Cao and Cao, 2009; Sahu and Padhye, 2012) built upon the work of (Boldyreva et al., 2003) and (Schuldt et al., 2008).

3.1 Anonymous ID-based Proxy Multi-Signature Scheme

In a (t,n)-threshold anonymous ID-based proxy multi-signature scheme, group of n original signers are authorized to transfer their signing rights to a single proxy signer to sign any document anonymously on their behalf but in case of misuse of the delegated rights by the proxy signer, t or more of the original signers can come together to reveal and demonstrate the identity of the proxy signer. Public and private keys of original and proxy signers are generated by a Private Key Generator (PKG), using their corresponding identities. Let the n original signers O_i have the identities ID_{O_i}, $i = 1,\ldots,n$, and the proxy signer $\mathcal{P}$ has the identity $ID_{\mathcal{P}}$. A (t,n)-threshold anonymous ID-based proxy multi-signature scheme can be defined consisting the following:

Setup: For a security parameter k, the PKG runs this algorithm and generates the public parameters *params* and a master secret of the system. Further, the PKG publishes *params* and keeps the master secret confidential.

Extract: This is a private key generation algorithm. For a given identity ID, public parameters *params* and master secret, PKG runs this algorithm to generate private key S_{ID} of the user with identity ID, and provides this private key through a secure channel to the user corresponding to the identity ID.

Proxy multi-generation: This is an interactive protocol among the original signers and the proxy signer. In this phase, the group of original signers interact with the proxy signer to agree on a *pseudonym* to anonymize the identity of the proxy signer and a warrant w which includes the nature of message to be delegated, period of delegation, identity information of original signers, the *pseudonym* for the proxy signer etc. Finally the original signers delegate their signing rights to the proxy signer and the proxy signer produces the (secret) proxy signing key. This algorithm takes as input, the identities $ID_{O_i}, ID_{\mathcal{P}}$ and private keys $S_{ID_{O_i}}, S_{ID_{\mathcal{P}}}$ of all the users and outputs the *pseudonym* Q_{ID_Q}, the warrant w, the shares ρ_{O_i} of the original signers, the delegation V_{O_i}, $i = 1,\ldots,n$, and the proxy signing key $S_{\mathcal{P}}$.

Proxy multi-signature: This is a randomized algorithm, the proxy signer runs this algorithm to generate a proxy multi-signature on an intended message m. This algorithm takes proxy signing key of the proxy signer, the warrant w, message m and outputs the proxy multi-signature $\sigma_{\mathcal{P}}$.

Proxy multi-verification: This is a deterministic algorithm run by the verifier on receiving a proxy multi-signature $\sigma_{\mathcal{P}}$ on any message m. This algorithm takes as inputs the proxy multi-signature $\sigma_{\mathcal{P}}$, the message m, the warrant w, the identities ID_{O_i} of all the original signers, Q_{ID_Q} and outputs 1 if the signature $\sigma_{\mathcal{P}}$ is a valid proxy multi-signature on behalf of the group of original signers on m, and outputs 0 otherwise. We emphasize that the actual identity $ID_{\mathcal{P}}$ of the proxy signer is not required but the *pseudonym* Q_{ID_Q}, as in the warrant, is required for the verification.

Reveal & Demonstrate: To reveal and demonstrate the proxy signer's identity, t or more original signers combine their shares ρ_{O_i} to recover the shared secret ρ_O and proceed to reveal the proxy signer's identity from the *pseudonym*.

3.2 Security Model for Anonymous ID-based Proxy Multi-Signature Schemes

3.2.1 Unforgeability

In this model we consider a case where an adversary $\mathcal{A}$ tries to forge the proxy multi-signature working against a single user, once against an original signer say O_i and once against the proxy signer $\mathcal{P}$. We consider that ID_{O_i} $(i = 1,\ldots,n)$ denotes identities of the original signers and $ID_{\mathcal{P}}$ denotes identity of the proxy signer. The adversary $\mathcal{A}$ is allowed to access polynomial number of hash queries, extraction queries, proxy multi-generation queries and proxy multi-signature queries. The goal of the adversary $\mathcal{A}$ is to produce one of the following forgeries:

1. A proxy multi-signature for a message m by user 1 on behalf of the original signers, such that either the original signers never designated user 1, or the message m was not submitted in the proxy multi-signature queries.
2. A proxy multi-signature for a message m by some user i ($i \neq 1$) on behalf of the original signers, such that user i was never designated by the original signers, and user 1 is one of the original signers.

An ID-based proxy multi-signature scheme is said to be existential unforgeable against adaptive chosen-message and adaptive chosen-ID attack if there is no probabilistic polynomial time adversary $\mathcal{A}$ with a non-negligible advantage against the challenger $\mathcal{C}$ in the following game:

1. *Setup*: Challenger $\mathcal{C}$ runs the Setup algorithm and provides the public parameters *params* to the adversary $\mathcal{A}$.
2. *Extract query*: When the adversary $\mathcal{A}$ asks private key of any user with identity ID_i, the challenger runs the Extract algorithm and responds the private keys to the adversary.
3. *Proxy multi-generation query*: When the adversary $\mathcal{A}$ requests to interact with the user 1 for the proxy signing key by proxy multi-generation query on the warrant w' and identities ID_i of its choice where the user 1 may be either one of the original signers or the proxy signer, the challenger $\mathcal{C}$ runs the proxy multi-generation algorithm to respond the proxy signing key to the adversary and maintains corresponding lists.
4. *Proxy multi-signature query*: Proceeding adaptively when the adversary $\mathcal{A}$ requests for a proxy multi-signature on message m' and warrant w' of its choice, $\mathcal{C}$ responds by running the proxy multi-signature algorithm and maintains a query list say L_{pms} for it.
5. *Output*: After the series of queries, $\mathcal{A}$ outputs a new proxy multi-signature $(U_{\mathcal{P}}, \sigma_{\mathcal{P}}, U, w)$ on message m under a warrant w for identities ID_{O_i} and $ID_{\mathcal{P}}$. Where $\mathcal{A}$ has never requested private keys for ID_{O_i} and $ID_{\mathcal{P}}$ in extraction queries. $\mathcal{A}$ has never requested a Proxy multi-generation query including warrant w and identities ID_{O_i}. $\mathcal{A}$ has never requested a proxy multi-signature query on message m with warrant w and identity $ID_{\mathcal{P}}$.

The adversary $\mathcal{A}$ wins the above game if the new ID-based proxy multi-signature $(U_{\mathcal{P}}, \sigma_{\mathcal{P}}, U, w)$ on message m is valid.

Definition 5. An ID-based proxy multi-signature forger $\mathcal{A}$ $(t, q_H, q_E, q_{pmg}, q_{pms}, n+1, \varepsilon)$-breaks the $n+1$ users ID-based proxy multi-signature scheme by the adaptive chosen-message and adaptive chosen-ID attack, if $\mathcal{A}$ runs in at most t time; makes at most q_H hash queries; at most q_E extraction queries; at most q_{pmg} proxy multi-generation queries; at most q_{pms} proxy multi-signature queries; and the success probability of $\mathcal{A}$ is at least ε.

Definition 6. An ID-based proxy multi-signature scheme is $(t, q_H, q_E, q_{pmg}, q_{pms}, n+1, \varepsilon)$-secure against adaptive chosen-message and adaptive chosen-ID attacks, if no adversary $(t, q_H, q_E, q_{pmg}, q_{pms}, n+1, \varepsilon)$-breaks it.

3.2.2 Anonymity and Accountability

Definition 7 (Anonymity)**.** By *anonymity* we mean that no one except the original signers should be able to determine the identity of the proxy signer from the proxy signatures or the warrant.

Definition 8 (Threshold Anonymity)**.** By (t,n)-*threshold anonymity* we mean that even the original signers O_i who know the identity of the proxy signer $\mathcal{P}$ should not be able to prove that $\mathcal{P}$ is the signer of a certain proxy multi-signature unless at least t of the n original signers participate in the proof.

Definition 9 (Accountability)**.** *Accountability* ensures that the proxy signer $\mathcal{P}$ does not abuse its anonymity. Any t (or more) out of n original signers can come together to prove that $\mathcal{P}$ is the signer of any verifiable designated proxy multi-signature.

Remark: Note that each of the original signers always know the identity of the proxy signer $\mathcal{P}$ since they delegate their rights to $\mathcal{P}$. Our definitions require that any group of less than t original signers is not able to prove to a third party that $\mathcal{P}$ is indeed the proxy signer.

4 PROPOSED SCHEME

In this section, we present our efficient and provably secure threshold-anonymous identity-based proxy multi-signature (IBPMS) scheme which provides anonymity to the proxy signer while also providing a threshold mechanism to the original signers to expose the identity of the proxy signer in case of misuse. Our scheme consists of the following phases: *setup*, *extract*, *proxy multi-generation*, *proxy multi-signature*, *proxy multi-verification*, *reveal & demonstration*.

The scheme uses the following signature scheme which was proved to be secure in (Sahu and Padhye, 2012) (with *Setup* and *Extract* as defined below in the definition of the threshold anonymous proxy multi-signature):

Signature: To sign a message $m \in \{0,1\}^*$,

- randomly selects $r \in \mathbb{Z}_q^*$,
- computes $U = rP \in G_1$,
- $h = H_2(m\|U)$ and
- $V = hS_{ID} + rPub$.

The signature on message m is $\sigma = \langle U, V \rangle$.

Verification: To verify a signature $\sigma = \langle U, V \rangle$ on message m for an identity ID, the verifier first computes

- $Q_{ID} = H_1(ID)$, and
- $h = H_2(m\|U)$.

Then accepts the signature if

$$e(P,V) = e(Pub, hQ_{ID} + U),$$

and rejects otherwise.

4.1 Our Anonymous IBPMS Scheme

Setup: For a given security parameter 1^k, let G_1 be an additive cyclic group of prime order q with generator P and G_2 be a multiplicative cyclic group of the same prime order q. Let $e : G_1 \times G_1 \to G_2$ be a cryptographic bilinear map as defined above. Let H_1 and H_2 are two hash functions defined for security purpose as $H_1 : \{0,1\}^* \to G_1$ and $H_2 : \{0,1\}^* \to \mathbb{Z}_q^*$. The PKG randomly selects $s \in \mathbb{Z}_q^*$ and sets $Pub = sP$ as public value. Finally, the PKG publishes system's public parameter $params = \langle k, e, q, G_1, G_2, H_1, H_2, P, Pub \rangle$ and keeps the master secret s confidential to itself only.

Extract: Given a user's identity ID, the PKG computes its

- public key as: $Q_{ID} = H_1(ID)$ and
- private key as: $S_{ID} = sQ_{ID}$ respectively.

Proxy multi-generation: To delegate the signing capability to the proxy signer $\mathcal{P}$, the n original signers do the following jobs to make a signed warrant w. The warrant includes the nature of message to be delegated, period of delegation, identity information of original signers, the *pseudonym* for the proxy signer etc. In successfully completion of the protocol, proxy signer gets a proxy signing key $S_{\mathcal{P}}$.

***Delegation generation*:** (a) *Pseudonym generation*: Each original signer with identity ID_{O_i} selects a random number $\rho_{O_i} \in \mathbb{Z}_q^*$ and sends it to the proxy signer $\mathcal{P}$ in a secure channel. $\mathcal{P}$ computes $\rho_O = \rho_{O_1} + \rho_{O_2} + \cdots + \rho_{O_n} \in \mathbb{Z}_q^*$ and uses a (t,n)-threshold verifiable secret sharing scheme (Pedersen, 1991) to split ρ_O into n shares ρ_{s_i}, $i = 1,2,\ldots,n$. $\mathcal{P}$ then sets $R_O = \rho_O P$ and sends R_O, ρ_{s_i} to the corresponding original signer O_i for $i = 1,2,\ldots,n$ through a secure channel. $\mathcal{P}$ also selects a random number $\rho_{\mathcal{P}} \in \mathbb{Z}_q^*$, computes $R_{\mathcal{P}} = \rho_{\mathcal{P}} P$ and its standard signature $s_{R_{\mathcal{P}}}$. Finally $\mathcal{P}$ sends $R_{\mathcal{P}}, s_{R_{\mathcal{P}}}$ to all the n original signers O_i for $i = 1,2,\ldots,n$ through a secure channel. Each original signer computes $Q_{ID_Q} = Q_{ID_{\mathcal{P}}} + R_{\mathcal{P}} + R_O$ as the proxy signer's *pseudonym*, which will be included in the warrant and will be used as the signature verification key.

Remark: Note that all the n original signers can come together with the ρ_{O_i} that they sent to $\mathcal{P}$ and compute $\rho_O = \rho_{O_1} + \rho_{O_2} + \cdots + \rho_{O_n}$ to expose the identity of the proxy signer in case of misuse. We are using a threshold secret sharing scheme to provide a threshold mechanism to the original signers so that only $t \leq n$ of the original signers are sufficient to participate to expose the identity of the proxy signer. Also note that we use a verifiable secret sharing scheme so that a malicious proxy signer cannot mislead the original signers with an incorrect ρ_{s_i} to avoid being held responsible for its proxy-signatures. The original signers can verify their shares as soon as they receive it and are assured that the ρ_{s_i} they receive will correctly construct to ρ_O corresponding to the R_O which they receive. Also, $\mathcal{P}$'s signature is required for non-repudiation.

(b) *Delegation generation*: For $i = 1,\ldots,n$, each O_i

- selects $r_i \in \mathbb{Z}_q^*$,
- computes $U_i = r_i P$ and
- broadcasts U_i to the other $n-1$ original signers.

For $i = 1,\ldots,n$, each O_i computes

- $U = \sum_{i=1}^n U_i$,
- $h = H_2(w\|U)$, and
- $V_{O_i} = hS_{ID_{O_i}} + r_i Pub$

and sends (w, U_i, V_{O_i}) to the proxy signer $\mathcal{P}$, with V_{O_i} as a delegation value.

***Delegation verification*:** For $i = 1,\ldots,n$, $\mathcal{P}$ verifies the delegation by $U = \sum_{i=1}^n U_i$ and $h = H_2(w\|U)$ and checking

$$e(P, V_{O_i}) = e(Pub, hQ_{ID_{O_i}} + U_i).$$

If the above equality does not hold for some $i = 1,\ldots,n$, $\mathcal{P}$ requests a valid delegation (w, U_i, V_{O_i}) or terminates the protocol.

***Proxy signing key generation*:** Having accepted delegations (w, U_i, V_{O_i}), $i = 1, \ldots, n$, $\mathcal{P}$ computes

$$S_{ID_Q} = S_{ID_\mathcal{P}} + \rho_\mathcal{P} Pub + \rho_O Pub$$

and sets the proxy signing key $S_\mathcal{P}$ as

$$S_\mathcal{P} = V_O + hS_{ID_Q},$$

where $V_O = \sum_{i=1}^{n} V_{O_i}$ and $h = H_2(w\|U)$.

Remark: Note that

$$\begin{aligned} S_{ID_Q} &= S_{ID_\mathcal{P}} + \rho_\mathcal{P} Pub + \rho_O Pub \\ &= sQ_{ID_\mathcal{P}} + \rho_\mathcal{P} sP + \rho_O sP \\ &= s(Q_{ID_\mathcal{P}} + \rho_\mathcal{P} P + \rho_O P) \\ &= s(Q_{ID_\mathcal{P}} + R_\mathcal{P} + R_O) \\ &= sQ_{ID_Q}. \end{aligned}$$

So, (Q_{ID_Q}, S_{ID_Q}) is a valid public-key / private-key pair.

Proxy multi-signature: To sign a message m anonymously on behalf of the group of n original signers, the proxy signer $\mathcal{P}$ computes the following:

- Randomly picks $r_\mathcal{P} \in \mathbb{Z}_q^*$, and
- computes
  - $U_\mathcal{P} = r_\mathcal{P} P$,
  - $h_\mathcal{P} = H_2(m\|U_\mathcal{P})$ and
  - $V_\mathcal{P} = h_\mathcal{P} S_\mathcal{P} + r_\mathcal{P} Pub$.

The anonymous proxy multi-signature on message m, by $\mathcal{P}$ on behalf of the n original signers is $\sigma_\mathcal{P} = (w, U_\mathcal{P}, V_\mathcal{P}, U)$.

Proxy multi-verification: To verify an anonymous proxy multi-signature $\sigma_\mathcal{P} = (w, U_\mathcal{P}, V_\mathcal{P}, U)$ for message m under a warrant w, the verifier proceeds as follows:

- Checks whether or not the message m conforms to the warrant w. If not, stop. Continue otherwise.
- Checks whether or not the *pseudonym* Q_{ID_Q} is authorized by the group of n original signers in the warrant w. If not, stop. Continue otherwise.
- Computes $h_\mathcal{P} = H_2(m\|U_\mathcal{P})$, $h = H_2(w\|U)$ and accepts the proxy signature if and only if the following equality holds:

$$e(P, V_\mathcal{P}) = e(Pub, h_\mathcal{P}(h(\sum_{i=1}^{n} Q_{ID_{O_i}} + Q_{ID_Q}) + U) + U_\mathcal{P}).$$

Remark: Note that the identity of the proxy signer $\mathcal{P}$ or its public key $Q_{ID_\mathcal{P}}$ is not required for the verification.

Reveal & Demonstrate: To reveal the identity of the proxy signer, any original signer O_i can reveal R_O and $R_\mathcal{P}$ and show that

$$Q_{ID_Q} = Q_{ID_\mathcal{P}} + R_\mathcal{P} + R_O.$$

That $R_\mathcal{P}$ was indeed sent by $\mathcal{P}$ is proved using the signature $s_{R_\mathcal{P}}$. To prove that R_O is not just a solution to the equation $Q_{ID_Q} = Q_{ID_\mathcal{P}} + R_\mathcal{P} + R_O$, t or more original signers combine their shares to recover the secret ρ_O and show that $R_O = \rho_O P$.

5 SECURITY ANALYSIS

In this section, we analyze the correctness, security, threshold-anonymity and accountability of our scheme. First we prove the correctness of the scheme, then we prove that the underlying IBPMS scheme is existential unforgeable against adaptive chosen-message and adaptive chosen-ID attacks and finally we analyze the threshold-anonymity and accountability of the proposed anonymous proxy multi-signature scheme.

5.1 Correctness

Theorem 10. *The presented threshold anonymous proxy multi-signature scheme is correct.*

Proof. This follows since

$$\begin{aligned} e(P, V_\mathcal{P}) &= e(P, h_\mathcal{P} S_\mathcal{P} + r_\mathcal{P} Pub) \\ &= e(P, h_\mathcal{P}(V_O + hS_{ID_Q}) + r_\mathcal{P} Pub) \\ &= e(P, h_\mathcal{P}(\sum_{i=1}^{n}(hS_{ID_{O_i}} + r_i Pub) + hS_{ID_Q}) \\ &\quad + r_\mathcal{P} Pub) \\ &= e(Pub, h_\mathcal{P}(\sum_{i=1}^{n}(hQ_{ID_{O_i}} + r_i P) + hQ_{ID_Q}) \\ &\quad + r_\mathcal{P} P) \\ &= e(Pub, h_\mathcal{P}(\sum_{i=1}^{n} hQ_{ID_{O_i}} + \sum_{i=1}^{n} r_i P + hQ_{ID_Q}) \\ &\quad + U_\mathcal{P}) \\ &= e(Pub, h_\mathcal{P}(h(\sum_{i=1}^{n} Q_{ID_{O_i}} + Q_{ID_Q}) + U) \\ &\quad + U_\mathcal{P}). \end{aligned}$$

5.2 Security Proof of the IBPMS Scheme

We now prove that the underlying IBPMS scheme is existential unforgeable against adaptive chosen-message and adaptive chosen-ID attacks.

We facilitate the adversary to adaptively select the identity on which it wants to forge the signature. Further the adversary can obtain the private keys associated to the identities. The adversary also can access the proxy multi-generation oracles on warrants w' of its choice, and proxy multi-signature oracles on the warrant, messages pair (w', m') of its choice, as many times it wants.

Theorem 11. *We consider the random oracle for reply to hash queries. If there exists an adversary*

$$\mathcal{A}(t, q_{H_1}, q_{H_2}, q_E, q_{pmg}, q_{pms}, \varepsilon)$$

which breaks the proposed ID-based proxy multi-signature scheme, then there exists an adversary

$$\mathcal{B}(t', q'_{H_1}, q'_{H_2}, q'_E, q'_{pmg}, q'_{pms}, \varepsilon')$$

which solves CDHP in time at most

$$t' \geq t + (q_{H_1} + q_E + 2q_{pmg} + 4q_{pms} + 1)C_{G_1}$$

with success probability at least

$$\varepsilon' \geq \frac{\varepsilon(1 - 1/q)}{M(q_E + q_{pmg} + q_{pms} + n + 1)}$$

where C_{G_1} denotes the number of scalar multiplications in group G_1.

Proof. First of all the challenger runs the setup algorithm and provides the

$$params = \langle q, G_1, G_2, e, P, sP, bP \rangle$$

to $\mathcal{B}$. Here, $\mathcal{A}$ is a forger algorithm whose goal is to break the underlying ID-based proxy multi-signature scheme. The adversary $\mathcal{B}$ simulates the challenger and interacts with $\mathcal{A}$. The goal of $\mathcal{B}$ is to solve CDHP by computing $sbP \in G_1$.

***Key Generation*:** For security parameter 1^k, $\mathcal{B}$ generates the system's public parameter

$$params = \langle q, G_1, G_2, e, P, Pub, H_1, H_2 \rangle$$

and provides $Pub = sP$ to $\mathcal{A}$.

***H_1-queries*:** To respond to the H_1 hash function queries, $\mathcal{B}$ maintains a list $L_{H_1} = \{\langle ID, h, a, c \rangle\}$. When $\mathcal{A}$ queries the H_1 hash function on some identity $ID_i \in \{0,1\}^*$, $\mathcal{B}$ responds as follows:

1. If the query ID_i already appears in the list L_{H_1} in some tuple $\langle ID_i, h_i, a_i, c_i \rangle$ then algorithm $\mathcal{B}$ responds to $\mathcal{A}$ with $H_1(ID_i) = h_i$.
2. Otherwise $\mathcal{B}$ picks a random integer $a_i \in \mathbb{Z}_q^*$ and generates a random coin $c_i \in \{0,1\}$ with probability $Pr[c_i = 0] = \lambda$, for some $\lambda \in [0,1]$.
3. If $c_i = 0$, $\mathcal{B}$ computes $h_i = a_i(bP)$ and if $c_i = 1$, $\mathcal{B}$ computes $h_i = a_i P$.
4. Algorithm $\mathcal{B}$ adds the tuple $\langle ID_i, h_i, a_i, c_i \rangle$ to the list L_{H_1} and responds to $\mathcal{A}$ with h_i.

***H_2-queries*:** To respond to the H_2 hash function queries, $\mathcal{B}$ maintains a list $L_{H_2} = \{\langle w, U, f \rangle\}$. When $\mathcal{A}$ requests the H_2 query on (w', U') for some warrant w', $\mathcal{B}$ responds as follows:

1. If the query (w', U') already appears on the list L_{H_2} in some tuple $\langle w', U', f \rangle$ then algorithm $\mathcal{B}$ responds to $\mathcal{A}$ with $H_2(w' \| U') = f$.
2. Otherwise $\mathcal{B}$ picks a random integer $f \in \mathbb{Z}_q^*$ and adds the tuple $\langle w', U', f \rangle$ to the list L_{H_2} and responds to $\mathcal{A}$ with $H_2(w' \| U')$ as $H_2(w' \| U') = f$.

***Extraction queries*:** If $\mathcal{A}$ requests a private key on identity ID_i, $\mathcal{B}$ responds as follows:

1. $\mathcal{B}$ runs the above algorithm for responding to H_1 queries on ID_i and obtains the corresponding tuple $\langle ID_i, h_i, a_i, c_i \rangle$ on the list L_{H_1}.
2. If $c_i = 0$, then $\mathcal{B}$ outputs 'failure' and terminates.
3. If $c_i = 1$, then $\mathcal{B}$ responds to $\mathcal{A}$ with $S_{ID_i} = a_i Pub \in G_1$.

Remark: Note that $H_1(ID_i) = h_i = a_i P$ so that

$$\begin{aligned} e(S_{ID_i}, P) &= e(a_i Pub, P) \\ &= e(a_i P, Pub) \\ &= e(H_1(ID_i), Pub) \\ &= e(Q_{ID_i}, Pub). \end{aligned}$$

Thus, S is a valid private key corresponding to the identity ID_i and the probability of success is $(1 - \lambda)$, because we have considered the case for $c_i = 1$.

***Proxy multi-generation queries*:** When the adversary $\mathcal{A}$ requests to interact with either the proxy signer or anyone from the original signers, then challenger $\mathcal{B}$ responds as follows:

1. Suppose, $\mathcal{A}$ requests to interact with the user ID_{O_i}, who is playing the role of one of the original signers. For this, $\mathcal{A}$ creates a warrant w' and requests ID_{O_i} to sign the warrant. $\mathcal{B}$ queries w' to its signing oracle and upon receiving a response $\langle U'_{O_i}, V'_{O_i} \rangle$, sends $\langle w', U'_{O_i}, V'_{O_i} \rangle$ to $\mathcal{A}$ and adds the warrant w to the delegation generation list say L_{del}.
2. Suppose, $\mathcal{A}$ requests to interact with user $ID_{\mathcal{P}}$, where $ID_{\mathcal{P}}$ is playing the role of the proxy signer. For this, $\mathcal{A}$ creates a warrant w' and computes the

signatures $V'_{O_i} = H_2(w' \| U')S_{ID_{O_i}} + x'_i Pub$. Where $U' = \sum_{i=1}^{n} x'_i P$ for randomly selected $x'_i \in \mathbb{Z}_q^*$ and $S_{ID_{O_i}}$ is private key of the original signer O_i which can be collected by $\mathcal{A}$ in the extraction query. Then $\mathcal{A}$ sends (w', V'_{O_i}) (for $i = 1, \ldots, n$) to $\mathcal{B}$. $\mathcal{B}$ provides the corresponding proxy signing key $S'_{\mathcal{P}}$ to $\mathcal{A}$ and adds the tuple $\langle w', S_{\mathcal{P}} \rangle$ to the proxy multi-generation list say L_{pmg}.

In either of the above cases,

1. $\mathcal{B}$ runs the above algorithm for responding to H_2 queries on w' obtaining the corresponding tuple $\langle w', U', f \rangle$, on L_{H_2} list.

2. For H_1 query, if $c = 0$, then $\mathcal{B}$ reports 'failure' and terminates. If $c = 1$, then, $H_1(ID_{O_i}) = a_{O_i} P$.

Then for $V'_{O_i} = f a_{O_i} Pub + x'_i Pub$, one can check that:

$$\begin{aligned} & e(Pub, fQ_{ID_{O_i}} + U'_i) \\ & = e(Pub, fH_1(ID_{O_i}) + U'_i) \\ & = e(Pub, fa_{O_i}P + x'_i P) \\ & = e(P, fa_{O_i}Pub + x'_i Pub) \\ & = e(P, V'_{O_i}). \end{aligned}$$

Hence the above provided proxy signing key is valid. The success probability is $(1 - \lambda)$, because we have considered the case for $c = 1$.

***Proxy multi-signature queries*:** Proceeding adaptively when adversary $\mathcal{A}$ requests for a proxy multi-signature on message m' of its choice (with satisfying the warrant w'), by the proxy signer $\mathcal{P}$ on behalf of the n original signers O_i, $(i = 1, 2, .., n)$. $\mathcal{B}$ does the following:

1. runs the above algorithm to respond H_2-queries on w', obtaining the tuple $\langle w', U', f \rangle$ on L_{H_2} list.

2. If $c = 0$ then reports 'failure' and terminates. If $c = 1$, then by the corresponding H_1-query $h = aP$.

Now $\mathcal{B}$ randomly selects $r'_{\mathcal{P}}, r' \in \mathbb{Z}_q^*$ and computes $U'_{\mathcal{P}} = r'_{\mathcal{P}} P$ and $U' = r'P$ then having $H_2(w' \| U') = f$ from H_2 query, for the tuple $\langle w', U', f \rangle$ and $H_2(m' \| U'_{\mathcal{P}}) = f_{\mathcal{P}}$ from H_2 query, for the tuple $\langle m', U'_{\mathcal{P}}, f_{\mathcal{P}} \rangle$, $\mathcal{B}$ again computes $Q_{\mathcal{P}} = f(\sum_{i=1}^{n} Q_{ID_{O_i}} + Q_{ID_{\mathcal{P}}}) + U'$. Finally $\mathcal{B}$ computes $V'_{\mathcal{P}} = [f_{\mathcal{P}}\{f(a_{O_1} + \cdots + a_{O_n} + a_{\mathcal{P}}) + r'\} + r'_{\mathcal{P}}]Pub$ for the signature on message m'. One can check that:

$$\begin{aligned} & e(Pub, f_{\mathcal{P}}\{f(\sum_{i=1}^{n} Q_{ID_{O_i}} + Q_{ID_{\mathcal{P}}}) + U'\} + U'_{\mathcal{P}}) \\ & = e(Pub, f_{\mathcal{P}}\{f(H_1(ID_{O_1}) + \cdots + H_1(ID_{O_n}) \\ & \quad + H_1(ID_{\mathcal{P}})) + U'\} + U'_{\mathcal{P}}) \\ & = e(Pub, f_{\mathcal{P}}\{f(a_{O_1}P + \cdots + a_{O_n}P + a_{\mathcal{P}}P) + r'P\} \\ & \quad + r'_{\mathcal{P}}P) \qquad \text{(for the case when } c = 1) \\ & = e(Pub, f_{\mathcal{P}}\{f(a_{O_1} + \cdots + a_{O_n} + a_{\mathcal{P}}) + r'\}P \\ & \quad + r'_{\mathcal{P}}P) \\ & = e(P, f_{\mathcal{P}}\{f(a_{O_1} + \cdots + a_{O_n} + a_{\mathcal{P}}) + r'\}Pub \\ & \quad + r'_{\mathcal{P}}Pub) \\ & = e(P, [f_{\mathcal{P}}\{f(a_{O_1} + \cdots + a_{O_n} + a_{\mathcal{P}}) + r'\} \\ & \quad + r'_{\mathcal{P}}]Pub) \\ & = e(P, V'_{\mathcal{P}}). \end{aligned}$$

Hence, the produced proxy multi-signature $(w', U'_{\mathcal{P}}, V'_{\mathcal{P}}, U')$ on message m' is valid, which satisfies

$$e(P, V'_{\mathcal{P}}) = e(Pub, h_{\mathcal{P}}(h(\sum_{i=1}^{n} Q_{ID_{O_i}} + Q_{ID_{\mathcal{P}}}) + U') + U'_{\mathcal{P}}).$$

The success probability is $(1 - \lambda)$, because we have considered the case for $c = 1$.

Hence, the probability that $\mathcal{B}$ does not abort during the simulation is

$$(1 - \lambda)^{q_E + q_{pmg} + q_{pms}}.$$

***Output*:** If $\mathcal{B}$ never reports 'failure' in the above game, $\mathcal{A}$ outputs a valid ID-based proxy multi-signature $(w, U_{\mathcal{P}}, V_{\mathcal{P}}, U)$ on message m which satisfies

$$e(P, V_{\mathcal{P}}) = e(Pub, h_{\mathcal{P}}(h(\sum_{i=1}^{n} Q_{ID_{O_i}} + Q_{ID_{\mathcal{P}}}) + U) + U_{\mathcal{P}}).$$

If $\mathcal{A}$ does not query any hash function, that is, if responses to all the hash function queries are picked randomly then the probability that verification equality holds is less than $1/q$. Hence, $\mathcal{A}$ outputs a new valid ID-based proxy multi-signature $(w, U_{\mathcal{P}}, V_{\mathcal{P}}, U)$ on message m with the probability

$$(1 - \lambda)^{q_E + q_{pmg} + q_{pms}}(1 - 1/q).$$

Now we compute the success probability of $\mathcal{B}$ for the solution of CDHP using the above forgeries (by $\mathcal{A}$). We consider both the possible cases, viz. , success probability in case when $\mathcal{A}$ plays against an original signer and when $\mathcal{A}$ plays against the proxy signer.

Case 1. Suppose, $\mathcal{A}$ simulates $\mathcal{B}$ and requests to interact with any user say ID_{O_1}, where the user ID_{O_1} is playing the role of one original signer. For ID_{O_1}, $\mathcal{A}$ did not request the private key in Extraction queries, $\mathcal{A}$ did not request a Proxy multi-generation query including $\langle w, ID_{O_1}\rangle$ and $\mathcal{A}$ did not request a Proxy multi-signature query including $\langle ID_{O_1}, w, m\rangle$. If $c = 1$, then $H_1(ID_{O_i}) = a_{O_i}P$ for $i = 2,\ldots,n$, and $H_1(ID_{\mathcal{P}}) = a_{\mathcal{P}}P$ from the H_1-query. Further $\mathcal{B}$ computes $V^*_{\mathcal{P}} = V'_{\mathcal{P}} - ([f_{\mathcal{P}}\{f(a_{O_2} + \cdots + a_{O_n} + a_{\mathcal{P}}) + r'\} + r'_{\mathcal{P}}]Pub)$, then proceeds to solve CDHP using the equality:

$$\begin{aligned}
e(P,V'_{\mathcal{P}}) &= e(Pub, h_{\mathcal{P}}(h(\sum_{i=1}^{n} Q_{ID_{O_i}} + Q_{ID_{\mathcal{P}}}) + U') \\
&\quad + U'_{\mathcal{P}}) \\
&= e(Pub, f_{\mathcal{P}}\{f(H_1(ID_{O_1}) + \cdots + H_1(ID_{O_n}) \\
&\quad + H_1(ID_{\mathcal{P}})) + U'\} + U'_{\mathcal{P}}) \\
&= e(Pub, f_{\mathcal{P}}\{f(a_{O_2} + \cdots + a_{O_n} + a_{\mathcal{P}}) + r'\}P \\
&\quad + r'_{\mathcal{P}}P)e(Pub, f_{\mathcal{P}}\{fH_1(ID_{O_1})\}) \\
&= e(P, [f_{\mathcal{P}}\{f(a_{O_2} + \cdots + a_{O_n} + a_{\mathcal{P}}) + r'\} \\
&\quad + r'_{\mathcal{P}}]Pub)e(Pub, f_{\mathcal{P}}\{fH_1(ID_{O_1})\})
\end{aligned}$$

or, by above we can write

$$\begin{aligned}
e(P,V^*_{\mathcal{P}}) &= e(Pub, f_{\mathcal{P}}\{fH_1(ID_{O_1})\}) \\
&= e(Pub, f_{\mathcal{P}} f a_{O_1}(bP)) \\
&= e(P, f_{\mathcal{P}} f a_{O_1}(bsP)) \\
&= e(P, k(bsP))
\end{aligned}$$

where $k = f_{\mathcal{P}} f a_{O_1} \in Z^*_q$.

Comparing the components on both sides, $\mathcal{B}$ gets

$$V^*_{\mathcal{P}} = k(bsP)$$

which implies that $k^{-1}V^*_{\mathcal{P}} = bsP$. Thus $\mathcal{B}$ can solve an instance of CDHP.

The probability of success is $\lambda(1-\lambda)^n$.

Case 2. When $\mathcal{A}$ simulates $\mathcal{B}$ and requests to interact with a user $ID_{\mathcal{P}}$, where user $ID_{\mathcal{P}}$ is the proxy signer. For $ID_{\mathcal{P}}$, $\mathcal{A}$ did not request the private key, $\mathcal{A}$ did not request a proxy multi-generation query including $\langle w, ID_{\mathcal{P}}\rangle$ and $\mathcal{A}$ did not request a proxy multi-signature query including $\langle ID_{\mathcal{P}}, w, m\rangle$. As the above case, we can show that $\mathcal{B}$ can derive sbP with the same success probability $\lambda(1-\lambda)^n$.

Hence the overall success probability that $\mathcal{B}$ solves the CDHP in the above attack game is:

$$\varepsilon' = \lambda(1-\lambda)^{q_E + q_{pmg} + q_{pms} + n}(1 - 1/q)\varepsilon.$$

Now the maximum possible value of the above probability occurs for

$$\lambda = \frac{1}{q_E + q_{pmg} + q_{pms} + n + 1}.$$

Hence the optimal success probability is

$$\frac{\varepsilon(1-1/q)}{M(q_E + q_{pmg} + q_{pms} + n + 1)}$$

where $\frac{1}{M}$ is the maximum value of

$$(1-\lambda)^{q_E + q_{pmg} + q_{pms} + n}$$

for

$$\lambda = \frac{1}{q_E + q_{pmg} + q_{pms} + n + 1}.$$

Therefore

$$\varepsilon \le \frac{\varepsilon' M(q_E + q_{pmg} + q_{pms} + n + 1)}{1 - 1/q}.$$

Now taking care of running time, one can observe that the running time of algorithm $\mathcal{B}$ is same as $\mathcal{A}$'s running time plus the time taken to respond to the hash, extraction, proxy multi-generation and proxy multi-signature queries, that is,

$$q_{H_1} + q_{H_2} + q_E + q_{pmg} + q_{pms}.$$

Hence, the maximum running time is given by

$$t + (q_{H_1} + q_E + 2q_{pmg} + 4q_{pms} + 1)C_{G_1},$$

as each H_1 Hash query requires one scalar multiplication in G_1, Extraction query also requires one scalar multiplication in G_1, proxy multi-generation query requires two scalar multiplications in G_1, proxy multi-signature query requires four scalar multiplications in G_1 and to output CDH solution from $\mathcal{A}$'s forgery, $\mathcal{B}$ requires at most one scalar multiplication in G_1. Hence

$$t' \ge t + (q_{H_1} + q_E + 2q_{pmg} + 4q_{pms} + 1)C_{G_1}.$$

5.3 Anonymity

Theorem 12. *The presented threshold anonymous proxy multi-signature scheme is anonymous.*

Proof. Since $\rho_{O_i} \in \mathbb{Z}^*_q$ are random, so is ρ_O and hence so is $R_O = \rho_O P$. Also, since $\rho_{\mathcal{P}} \in \mathbb{Z}^*_q$ is random, so is $R_{\mathcal{P}} = \rho_{\mathcal{P}}P$. Since ρ_{O_i}, ρ_{s_i}, R_O and $R_{\mathcal{P}}$ were communicated through a secure channel, these are hidden from any adversary. So, no adversary would be able to ascertain the identity of the proxy signer from the computation $Q_{ID_Q} = Q_{ID_{\mathcal{P}}} + R_{\mathcal{P}} + R_O$.

In fact, note that even a collusion of t' original signers ($t' < t$) cannot recover ρ_O and cannot really prove that $\mathcal{P}$ is indeed the proxy signer. The last statement follows from the security of the threshold verifiable secret sharing (Pedersen, 1991) and to get the value ρ_O from R_O, the adversary has to solve discrete log problem, which is assumed to be hard.

5.4 Accountability

Theorem 13. *The presented threshold anonymous proxy multi-signature scheme is accountable.*

Proof. To reveal the identity of the proxy signer, any original signer O_i can reveal R_O and $R_{\mathcal{P}}$ and show that

$$Q_{ID_{\mathcal{Q}}} = Q_{ID_{\mathcal{P}}} + R_{\mathcal{P}} + R_O. \quad (1)$$

That $R_{\mathcal{P}}$ was indeed sent by $\mathcal{P}$ is proved using the signature $s_{R_{\mathcal{P}}}$.

To prove that R_O is not just a solution to the equation (1), t or more original signers combine their shares to recover the secret ρ_O and show that $R_O = \rho_O P$. Since $R_{\mathcal{P}}$ was randomly chosen by $\mathcal{P}$, given $Q_{ID_{\mathcal{P}}}$ and $Q_{ID_{\mathcal{Q}}}$, R_O is also random, and hence dishonest original signers can produce correct ρ_O only if they can solve the discrete log problem in G_1.

6 EFFICIENCY COMPARISON

Here, we compare the efficiency of our scheme with the IBPMS schemes of (Cao and Cao, 2009), (Du and Wang, 2013) and (Shao, 2009), and show that our scheme is more efficient in the sense of computation and operation time than these schemes. For the computation of operation time, we refer to (Debiao et al., 2011) where the operation time for various cryptographic operations have been obtained using MIRACL (MIRACL), a standard cryptographic library, and the hardware platform is a PIV 3 GHZ processor with 512 M bytes memory and the Windows XP operating system. For the pairing-based scheme, to achieve the 1024-bit RSA level security, Tate pairing defined over the supersingular elliptic curve $E = F_p : y^2 = x^3 + x$ with embedding degree 2 was used, where q is a 160-bit Solinas prime $q = 2^{159} + 2^{17} + 1$ and p a 512-bit prime satisfying $p + 1 = 12qr$. We note that the OT for one pairing computation is $20.04ms$, for one scalar multiplication it is $6.38ms$, for one map-to-point hash function it is $3.04ms$ and for one general hash function it is $< 0.001ms$. To evaluate the total operation time in the efficiency comparison tables, we use the simple method from (Cao et al., 2010; Debiao et al., 2011). In each of the three phases: proxy multi-generation, proxy multi-signature and proxy multi-verification, we compare the total number of bilinear pairings (P), scalar multiplications (SM), map-to-point hash functions (H) and the consequent operation time (OT) while omitting the operation time due to a general hash function which is negligible compared to the other three operations. Further, across all the compared schemes, in the computation table for proxy multi-generation, we take into consideration the computations of only one of the n original signers following the methodology of (Cao et al., 2010; Debiao et al., 2011).

For example, the proxy multi-generation phase of our scheme takes 2 pairing operations, 7 scalar multiplications and 1 map-to-point hash function. Hence the total operation time for this phase can be calculated as: $2 \times 20.04 + 7 \times 6.38 + 1 \times 3.04 = 87.78ms$. Similarly, we have computed the total OT in other phases for all the schemes.

Table 1: Efficiency Comparison

Proxy multi-generation:

Scheme	P	H	SM	OT (ms)
(Cao and Cao, 2009)	3	3	3	88.38
(Du and Wang, 2013)	3	4	4*	97.80
(Shao, 2009)	3	3	2	82.00
Our scheme	**2**	**1**	**7***	**87.78**

Proxy multi-signature:

Scheme	P	H	SM	OT (ms)
(Cao and Cao, 2009)	0	1	2	15.80
(Du and Wang, 2013)	0	1	2	15.80
(Shao, 2009)	0	1	2	15.80
Our scheme	**0**	**0**	**3**	**19.14**

Proxy multi-verification:

Scheme	P	H	SM	OT (ms)
(Cao and Cao, 2009)	4	3	1	95.66
(Du and Wang, 2013)	4	3	1	95.66
(Shao, 2009)	4	3	0	89.28
Our scheme	**2**	**1**	**2**	**55.88**

Overall Time:

Scheme	P	H	SM	OT (ms)
(Cao and Cao, 2009)	7	7	6	199.84
(Du and Wang, 2013)	7	8	7	209.26
(Shao, 2009)	7	7	4	187.08
Our scheme	**4**	**2**	**11**	**162.80**

* The scalar multiplications due to *pseudonym* generation are not considered.

From the efficiency comparison table (1), it is clear that our scheme is computationally more efficient and having less operation time than the schemes (Cao and Cao, 2009; Du and Wang, 2013; Shao, 2009).

REFERENCES

Blakley, G. R. (1979). Safeguarding cryptographic keys. In *National Computer Conference*, pages 313–317.

Boldyreva, A., Palacio, A., and Warinschi, B. (2003). Secure proxy signature schemes for delegation of signing rights. *IACR Cryptology ePrint Archive*, 2003:096.

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

Cao, X., Kou, W., and Du, X. (2010). A pairing-free identity-based authenticated key agreement protocol with minimal message exchanges. *Information Sciences*, 180(15):2895–2903.

Chor, B., Goldwasser, S., Micali, S., and Awerbuch, B. (1985). Verifiable secret sharing and achieving simultaneity in the presence of faults (extended abstract). In *FOCS*, pages 383–395.

Debiao, H., Jianhua, C., and Jin, H. (2011). An id-based proxy signature schemes without bilinear pairings. *Ann. Telecommun.*, 66(11-12):657–662.

Du, H. and Wang, J. (2013). An anonymous but accountable proxy multi-signature scheme. *Journal of Software*, 8(8):1867–1874.

Fuchsbauer, G. and Pointcheval, D. (2008). Anonymous proxy signatures. In *Security and Cryptography for Networks*, pages 201–217.

Gasser, M., Goldstein, A., Kaufman, C., and Lampson, B. (1989). The digital distributed system security architecture. In *NCSC'89*, pages 305–319.

Lee, B., Kim, H., and Kim, K. (2001). Strong proxy signature and its applications. In *Proc of SCIS*, volume 1, pages 603–608.

Lee, N.-Y. and Lee, M.-F. (2005). The security of a strong proxy signature scheme with proxy signer privacy protection. *Applied mathematics and computation*, 161(3):807–812.

Lee, Y.-H., Hong, S.-M., and Yoon, H. (2005). A secure strong proxy signature scheme with proxy signer privacy protection. In *CCCT'05*. (International Conference on Computing, Communications and Control Technologies).

Mambo, M., Usuda, K., and Okamoto, E. (1996). Proxy signatures: Delegation of the power to sign messages. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 79(9):1338–1354.

MIRACL. Multiprecision integer and rational arithmetic cryptographic library. http://certivox.org/display/EXT/MIRACL.

Pedersen, T. P. (1991). Non-interactive and information-theoretic secure verifiable secret sharing. In *Advances in Cryptology – CRYPTO'91*, pages 129–140.

Sahu, R. A. and Padhye, S. (2012). Efficient id-based proxy multi-signature scheme secure in random oracle. *Frontiers of Computer Science*, 6(4):421–428.

Schuldt, J. C. N., Matsuura, K., and Paterson, K. G. (2008). Proxy signatures secure against proxy key exposure. In *Public Key Cryptography*, volume 4939 of *LNCS*, pages 141–161.

Shamir, A. (1979). How to share a secret. *Communications of the ACM*, 22(11):612–613.

Shao, Z. (2009). Improvement of identity-based proxy multi-signature scheme. *Journal of Systems and Software*, 82(5):794–800.

Shum, K. and Wei, V. K. (2002). A strong proxy signature scheme with proxy signer privacy protection. In *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2002. WET ICE 2002. Proceedings. Eleventh IEEE International Workshops on*, pages 55–56. IEEE.

Sun, H.-M. and Hsieh, B.-T. (2003). On the security of some proxy signature schemes. *IACR Cryptology ePrint Archive*, 2003:068.

Toluee, R., Asaar, M. R., and Salmasizadeh, M. (2012). An anonymous proxy signature scheme without random oracles. *IACR Cryptology ePrint Archive*, 2012:313.

Wu, K.-L., Zou, J., Wei, X.-H., and Liu, F.-Y. (2008). Proxy group signature: A new anonymous proxy signature scheme. In *Machine Learning and Cybernetics, 2008 International Conference on*, volume 3, pages 1369–1373.

Xiong, H., Hu, J., Chen, Z., and Li, F. (2011). On the security of an identity based multi-proxy signature scheme. *Computers & Electrical Engineering*, 37(2):129–135.

Yu, Y., Xu, C., Huang, X., and Mu, Y. (2009). An efficient anonymous proxy signature scheme with provable security. *Computer Standards & Interfaces*, 31(2):348–353.