Available online at http://scik.org

J. Math. Comput. Sci. 10 (2020), No. 3, 448-473

https://doi.org/10.28919/jmcs/4403

ISSN: 1927-5307

SECURE AND EFFICIENT IDENTITY-BASED PROXY SIGNATURE SCHEME WITH MESSAGE RECOVERY

SALOME JAMES[1], GOWRI THUMBUR[2] AND P. VASUDEVA REDDY[1,*]

[1]Department of Engineering Mathematics, Andhra University, Visakhapatnam, 530003, India

[2]Department of Electronics and Communication Engineering, GITAM University, Visakhapatnam, 530045, India

Abstract: Digital signature with proxy delegation, which is a secure ownership enforcement tool, allows an original signer to delegate signature rights to a third party called proxy, so that the proxy can sign messages on behalf of the original signer. In today's modern society, many applications use this mechanism. Several types of delegations are quite prevalent and the delegation of signing authority is one of them. In a traditional digital signature scheme, signer transmits signature along with message for verification, which leads to additional communication, computation cost and requires extra bandwidth. To resolve these issues, in this paper, we present an efficient ID-based proxy signature scheme with message recovery using bilinear pairings. Because of the message recovery feature, the proxy signer need not send the message to the verifier, so that the proposed scheme reduces the bandwidth requirement and communication cost. Our proposed scheme is proven secure against existential forgery under adaptively chosen message and identity attacks in the random oracle model (ROM) with the assumption that the Computational Diffie-Hellman Problem (CDHP) is intractable. We compare our scheme with related schemes. The efficiency analysis shows that the scheme is computationally efficient. Thus the proposed ID-based proxy signature scheme is secure and efficient both in terms of computation and communication costs than the related existing schemes.

Keywords: proxy signature; identity-based setting; message recovery; bilinear pairings; computational Diffie-Hellman problem.

2010 AMS Subject Classification: 94A60.

---

*Corresponding author

E-mail address: vasucrypto@andhrauniversity.edu.in

Received November 30, 2019

1. INTRODUCTION

In a traditional public key cryptography, the user generates a private/public key pair and based on the fact that this key pair has absolutely no indication to which identity it belongs, the public key needs to be certified i.e. to bind the public key to the user's identity, through a digital certificate. In this system, however, the participant should first check the certificate of the user, before using the public key of a user. Subsequently, this system requires a large amount of computing time and storage when the number of users is increasing rapidly. In view of simplifying the key management complexity which is a heavy burden in the traditional public key cryptography (PKC), the concept of identity based cryptography was introduced by Shamir [1] in 1984. The ID-based cryptography can be an excellent option for traditional public key infrastructures, particularly when it desires efficient key management and moderate security. The primary concept behind an ID-based cryptography is that, it is possible to calculate the user's public key directly from his/her identity instead of extracting it from a certificate issued by a certificate authority. The private key is obtained by a trusted authority called private key generator (PKG). The bilinear pairings [2] have discovered several cryptographic applications, as they can be used to understand certain cryptographic primitives that were earlier unidentified or impractical. More specifically, they are the fundamental tools for constructing ID-based cryptographic schemes.

Today, we live in a digital era where communications and transactions are mainly online and in various areas, including e-government and e-commerce, there might be many circumstances in which the signatory entity itself cannot apply the signature and requires delegating its rights to another entity. A method of such delegation, called the proxy signature scheme was proposed by Mambo, Usuda and Okamoto [3] in 1996. The proxy signature enables a designated person called a proxy signer to sign on behalf of the original signer. It plays a very significant role in numerous applications including e-cash system [4], mobile agents for electronic commerce [5], mobile communications [6], grid computing [7] and distributed shared object systems [8]. The proxy signature can be classified as full delegation, partial delegation and delegation by warrant on the basis of delegation of signing powers to the proxy signer. The proxy signature is also categorized as unprotected proxy signature and protected proxy signature on the basis of protection.

The computation and communication efficiency are the key requirements in the latest scenario of technological development. In any communication network, the bandwidth is one of the primary limitations. In order to design a new efficient scheme, it is necessary to ensure that the

amount of data to be communicated or transmitted is reduced, so as to improve the communication efficiency. To minimize the overall length of the message and the appended signature, the idea of digital signature scheme with message recovery was initially introduced by Nyberg and Rueppel [9] in 1993. In this signature scheme, it is not necessary to transmit the original message along with the signature, as it is appended to the signature and can be recovered during the process of verification/message recovery. ID-based message recovery signature schemes are more convincing, as they avoid using a complicated certification system which is necessary in traditional message recovery signature schemes.

ID-based proxy signature schemes have enjoyed a significant interest from the cryptographic research community. The first ID-based proxy signature scheme was proposed by Zhang and Kim [10] in 2003. However, no formal security analysis was provided for their scheme. In 2005, Xu *et al.* [11] proposed an ID-based proxy signature scheme from pairings, but the security model described in their scheme did not take the case of an adaptively chosen identity attack into consideration. Subsequently in 2006, Gu *et al*. [12] proposed an ID-based proxy signature from pairings and discussed the security of the scheme in the ROM. In the same year, Mala *et al.* [13] proposed an ID-based proxy signature scheme from bilinear pairings and their scheme was based on Hess ID-based signature scheme [14]. Since then many ID-based proxy signature schemes were proposed in the literature [15-22].

Over the past decade, the cryptographer community has focused on designing secure and efficient message recovery proxy signature schemes. These schemes are efficient in terms of bandwidth, due to the message recovery property. Moreover, various message recovery proxy signature schemes [23-30] with different settings are proposed in the literature. It is, however, interesting to construct ID-based proxy signature schemes with message recovery to provide more flexible management of public keys and shorten the ID-based proxy signatures, thereby enhancing their communication overhead. The concept of ID-based proxy signature scheme with message recovery was introduced by Singh *et al*. [24] in 2012. Unfortunately, Tian *et al.* [25] reported the insecurity in Singh *et al.*'s [24] scheme. In 2013, Yoon *et al*. [26] proved that Singh *et al*.'s [24] scheme is insecure and they presented an improved ID-based proxy signature scheme with message recovery. In 2015, Zhou [28] proposed an improved ID-based proxy signature scheme with message recovery and in their scheme; they reported a security flaw in the proof of Singh *et al.* [24] scheme and discussed an improvement towards Singh *et al*. [24] scheme. In the same year, Sarde and Banerjee [20] proposed a secure ID-based proxy signature scheme from bilinear pairings. The security of their scheme is based both on CDHP resolution

and the strength and security of the hash function. Later, in 2016, Asaar *et al*. [29] proposed a short ID-based proxy signature scheme with message recovery and they claimed that Yoon *et al*. [26] scheme did not fulfill all the criteria of a proxy signature scheme with message recovery. Subsequently, in 2018, Liu *et al*. [22] proposed Strong Identity-based Proxy Signature Schemes, Revisited. In their scheme, they identified a new attack that has been neglected by many existing proven secure proxy signature schemes. They also proposed one method that can effectively prevent this attack and can also be applied in other proxy signature schemes to ensure an improved security. To fulfill the requirements of proxy signature schemes for low bandwidth communications and to maintain the merits of the message recovery property, in this paper we propose an efficient ID-based proxy signature scheme with message recovery from bilinear pairings. With smaller key sizes, our scheme achieves an appropriate level of security and thus improves the communicational efficiency.

The rest of the paper is organized accordingly. Section 2 discusses the preliminaries and computational hard problems. Framework and security model for the proposed scheme are presented in section 3.The proposed IBPSMR scheme is presented in section 4. The Security analysis and efficiency analysis are presented in section 5. Finally, the conclusion is discussed in section 6.

2. PRELIMINARIES

In this section, we briefly review the fundamental concepts of bilinear pairings and some related mathematical computational problems.

2.1. Bilinear Pairings

It is a significant cryptographic primitive and is widely adapted in several practical applications of cryptography. Let G_1 and G_2 be additive and multiplicative cyclic groups respectively of same prime order q with P as a generator of G_1. An admissible bilinear pairing is a map $\hat{e}$ defined by $\hat{e}: G_1 \times G_1 \to G_2$ satisfying the following properties:

1) Bilinearity: For all $P, Q \in G_1$ and $a, b \in Z_q^*$, $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$.

2) Non-Degeneracy: There exists $P \in G_1$, such that $\hat{e}(P, P) \neq 1$.

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

The Weil or Tate pairings on elliptic curves over finite fields can provide an efficient implementation of the admissible bilinear pairing.

2.2. Bilinear Pairings over Elliptic Curves

The modified Weil pairing and Tate pairing are admissible instantiations of bilinear pairings. The modified Weil pairing settings are briefly discussed below.

Let p be a sufficiently large prime that satisfies (1) $p \equiv 2 \bmod 3$; (2) $p = lq - 1$, where q is also a large prime. Let E be an elliptic curve defined by the equation $y^2 = x^3 + 1$ over F_p. Define $E(F_p)$ to be the group of points on E defined over F_p. Let $P \in E(F_p)$ be a point of order q and let G_1 be the subgroup of points generated by P. Set G_2 to be the subgroup of $F_{p^2}^*$ of order q. The modified Weil pairing is thus defined by $\hat{e} : G_1 \times G_1 \to G_2$ satisfying the conditions of a bilinear pairing.

2.3. Computational Problems

Now, we deal with some computational problems on which the security of the proposed scheme is based [31, 32].

1) Discrete Logarithm Problem (DLP): Given two group elements P and Q, find an integer n such that $Q = nP$ whenever such an integer exists.
2) Decisional Diffie-Hellman Problem (DDHP): For $a,b,c \in_R Z_q^*$, given P, aP, bP, cP decide whether $c \equiv ab \bmod q$.
3) Computational Diffie-Hellman Problem (CDHP)**:** For $a,b \in_R Z_q^*$, given P, aP, bP compute abP.

Throughout this paper, we assume that CDHP and DLP are intractable. When the DDHP is easy but the CDHP is hard on the group G, we call G, a *Gap Diffie-Hellman (GDH)* group. Such groups can be found on super singular elliptic curves or hyper elliptic curves over finite field and the bilinear pairings can be derived from the Weil or Tate pairing.

2.4. Notations

The notations and their meanings which we use in this paper are tabulated in TABLE 1 below.

TABLE 1. Notation and Description

Notation	Meaning
k	Security parameter
G_1, G_2	Additive and multiplicative cyclic groups respectively of same prime order q
H_1, H_2, H_3, F_1, F_2	Cryptographic hash functions
$a \Vert b$	Concatenation of two strings a and b
$\oplus$	X-OR computation in the binary system
$[x]_{10}$	Decimal representation of $x \in \{0,1\}^*$
$[y]_2$	Binary representation of $y \in Z$
${}_{l_2}\lvert\beta\rvert$	The first l_2 bits of β from the left side
$\lvert\beta\rvert_{l_1}$	The first l_1 bits of β from the right side
Ω	Signature on the message m

2.5. Acronyms

The acronyms which we use in this paper are tabulated in TABLE 2 below.

TABLE 2. Acronyms and explanation

Acronyms	Explanation
PKC	Public Key Cryptography
ID-based	Identity-based
CDHP	Computational Diffie-Hellman Problem
IBPSMR	Identity-based Proxy Signature Scheme with Message Recovery
PPT	Probabilistic Polynomial Time
PKG/KGC	Private Key Generator/ Key Generation Centre
ROM	Random Oracle Model
EF-ACMA	Existential Forgery under the Adaptive Chosen Message Attack

3. FRAMEWORK AND SECURITY MODEL OF THE PROPOSED IBPSMR

In this section we present the framework and security model of the proposed scheme.

3.1. Framework of IBPSMR

Let A be the original signer with identity ID_A and private key d_{ID_A}. The original signer A delegates his signing rights to a proxy signer B with identity ID_B and private key d_{ID_B}. A warrant is used to delegate signing rights. Now, we provide a formal security model for our ID-based proxy signature scheme with message recovery (IBPSMR). Precisely, our IBPSMR

scheme comprises of the following polynomial-time algorithms: System Setup, Key Extract, Delegation Generation, Delegation Verification, Proxy Key Generation, Proxy Signature Generation, Message Recovery and Proxy Signature Verification. The detailed performance of these algorithms is described below.

System Setup: The KGC takes the security parameter $k \in Z^+$ and executes this algorithm to generate the system parameters *Params* and the master private key s. *Params* will be made public and s will be kept secret. *Params* are implicit input to all the following algorithms.

Key Extract: The KGC executes this algorithm with inputs system's private key s, the *Params* and a pair of identities ID_A, ID_B, and generates the private keys of the identities through a secure channel to the corresponding user.

Delegation Generation: This algorithm takes as input, the private key d_{ID_A} of the original signer, a pair of identities ID_A, ID_B and a warrant m_w and outputs the delegation W.

Delegation Verification: This algorithm takes as input, the identity ID_A of the original signer and the delegation W and checks whether it is a valid delegation from the original signer A.

Proxy Key Generation: This is an interactive algorithm between the original signer A and the proxy signer B, during which they agree on the warrant m_w, that contains some specific information regarding the message. After successful interaction, the proxy signer outputs the proxy signing key d_{psk}, which is used to sign the messages on behalf of the original signer.

Proxy Signature Generation: This algorithm takes system parameters, proxy signing key d_{psk} and a message $m \in \{0,1\}^{l_1}$ as input and outputs a proxy signature Ω of the message m.

Message Recovery and Proxy Signature Verification: In this algorithm, the verifier receives the signature Ω and takes the original signer's identity ID_A and the proxy signer's identity ID_B as input and then recovers the message and displays acceptance or rejection.

3.2. Security Model of IBPSMR

The formal security model of the proposed IBPSMR scheme is discussed in this section. The following model/game is played between the forger/adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. The

forger $\mathcal{A}$ is permitted to adaptively choose its messages, warrant and identities. Furthermore, the forger is given access to the signing oracle for any messages for desired identities. A forger's advantage $Adv_{IBPSMR,\mathcal{A}}$ is defined as its probability of success in the following game. We classify the potential adversary $\mathcal{A}$ into the following three types.

1) **Type 1 Adversary:** The adversary $\mathcal{A}_1$ contains the public keys of the original signer and the proxy signer, and attempts to forge the delegation for a chosen warrant or to forge the proxy signature for some chosen message.
2) **Type 2 Adversary:** The adversary $\mathcal{A}_2$ contains the public keys of the original signer and the proxy signer. Furthermore, adversary $\mathcal{A}_2$ contains the private key of the proxy signer and attempts to forge the delegation by directly forging a valid signature for a chosen warrant.
3) **Type 3 Adversary:** The adversary $\mathcal{A}_3$ contains the public keys of the original signer and the proxy signer. Furthermore, the adversary $\mathcal{A}_3$ contains the private key of the original signer and attempts to forge the proxy signature for some chosen message.

Obviously, from the construction of these adversaries, if the proxy signature scheme with message recovery is capable of resisting the attacks plotted from type 2 and type 3 adversaries, then it will be secure against the type 1 adversary straight forwardly.

Definition 1. A proxy signature scheme with message recovery is said to be secure against type 2 adversary if there exists no probabilistic polynomial-time adversary $\mathcal{A}_2$ which can forge a valid signature Ω on a chosen warrant W, by playing the game with a challenger $\mathcal{C}$.

Furthermore, $\mathcal{A}_2\left(t, q_{H_1}, q_{H_2}, q_E, q_D, \varepsilon\right)$ is said to break an IBPSMR scheme if $\mathcal{A}_2$ can run in time at most t; makes at most $q_{H_1+H_2}$ queries to the hash oracles H_1, H_2; at most q_E queries to the key extract queries; at most q_D queries to delegation query; with $Adv_{IBPSMR,\mathcal{A}_2}$ is at least ε, in the following game.

Setup: The challenger $\mathcal{C}$ takes a security parameter k and executes the setup algorithm of the IBPSMR scheme. It gives *Params* to the adversary $\mathcal{A}_2$ and keeps the master private key

secret with itself.

Queries: The forger $\mathcal{A}_2$ adaptively makes different queries to the challenger $\mathcal{C}$.

Hash Queries: When the involved hash functions are modeled by random oracles, $\mathcal{A}_2$ also performs adaptive queries of the hash functions. The challenger $\mathcal{C}$ responds to these queries of the forger of this oracle, providing it with consistent and totally random values.

Extract Queries: The challenger $\mathcal{C}$ executes the key extract phase on a chosen identity ID_i and returns a private key corresponding to ID_i.

Delegation Queries: The challenger $\mathcal{C}$ executes the delegation phase on a chosen warrant m_w and returns the delegation W on the warrant m_w.

Forgery/Output: The adversary $\mathcal{A}_2$ outputs $\left\{ID_A, m_{w_A}, W\right\}$ and wins the game if:

1) m_{w_A} is not m_w; and

2) W is a valid signature of m_{w_A}.

Definition 2: A proxy signature scheme with message recovery is said to be secure against any type 3 adversary, if there exists no probabilistic polynomial-time adversary $\mathcal{A}_3$ which can forge a valid proxy signature Ω on the chosen message m, by playing the game with a challenger $\mathcal{C}$. Furthermore, $\mathcal{A}_3\left(t, q_{H_1}, q_{H_2}, q_{H_3}, q_E, q_S, \varepsilon\right)$ is said to break a IBPSMR scheme if $\mathcal{A}_3$ runs in time at most t; makes at most $q_{H_1+H_2}$ queries to the hash oracles H_1, H_2; at most q_E queries to the key extract queries; at most q_S queries to sign query; with $Adv_{IBPSMR,\mathcal{A}_3}$ is at least ε, in the following game.

The setup, $H_1, H_2, F_1, F_2,$ key extract queries are same as defined above made by the adversary $\mathcal{A}_2$.

Proxy Signature Query: When the adversary $\mathcal{A}_3$ requests adaptively a proxy signature on a given message m with an identity ID_B, the challenger C executes the proxy signature

generation algorithm, outputs Ω and returns to $\mathcal{A}_3$.

All these queries can be made adaptively, i.e. each query may depend on the answer obtained to the previous queries.

Output/Forgery: The adversary $\mathcal{A}_3$ outputs $\left\{m^*, ID_B^*, \Omega^*\right\}$ and wins the game if:

1) $m^* \neq m$ and
2) Ω is a valid signature.

4. PROPOSED ID-BASED PROXY SIGNATURE SCHEME WITH MESSAGE RECOVERY

In this section, we present the concrete description of our ID-based proxy signature scheme with message recovery (IBPSMR) using bilinear pairings. This scheme deals with messages of fixed length.

System Setup: For a given security parameter $k \in Z^+$, the KGC executes this algorithm as follows.

1) Chooses two groups G_1, G_2 of same prime order $q \geq 2^k$ with a bilinear pairing $\hat{e}: G_1 \times G_1 \to G_2$; G_1 is an additive cyclic group with $P \in G_1$ as a generator and G_2 is a multiplicative cyclic group.
2) Selects $s \in Z_q^*$ randomly and computes the system public key $P_{pub} = sP$.
3) Chooses the cryptographic hash functions

   $H_1: \{0, 1\}^* \to G_1$, $H_2: \{0, 1\}^* \times G_2 \to Z_q^*$, $H_3: \mathrm{G}_2 \to \{0,1\}^{|q|}$, $F_1: \{0, 1\}^{l_1} \to \{0, 1\}^{l_2}$,

   $F_2: \{0, 1\}^{l_2} \to \{0, 1\}^{l_1}$, where $l_1 + l_2 = |q|$.
4) KGC picks $s \in Z_q^*$ randomly and keeps $P_{pub} = sP$ as the master public key.
5) KGC publishes the system parameters as

   $Params = \{G_1, G_2, \hat{e}, q, P, P_{pub}, H_1, H_2, H_3, F_1, F_2\}$ as public and keeps the master key $< s >$ as secret.

Key Extract: Given $Params$, $< s >$, the user's identity ID_i, the KGC computes the corresponding private key $d_{ID_i} = sQ_{ID_i}$, where $Q_{ID_i} = H_1(ID_i)$ is the public key of the user. The private key is sent to the user with identity ID_i, over a secure and authenticated channel.

Delegation Generation: The original signer generates a warrant m_w, that maintains the record of proxy information such as the identities of the original signer, proxy signer, proxy validation period etc. The delegation W is generated as follows.The original signer A does the following.

1) Selects $r_A \in Z_q^*$ and computes $X_A = \hat{e}\left(P_{pub}, r_A Q_{ID_A}\right)$.
2) Computes $h_A = H_2\left(ID_A, X_A, m_w\right)$.

$$Y = \left(r_A + h_A\right) d_{ID_A}.$$

The original signer A outputs the delegation $W = \left(m_w, X_A, Y\right)$ on the warrant m_w, and forwards it to the proxy signer B.

Delegation Verification: Given an original signer's identity ID_A and the delegation $W = \left(m_w, X_A, Y\right)$, the proxy signer B checks whether $\hat{e}\left(Y, P\right) = \hat{e}\left(P_{pub}, h_A Q_{ID_A}\right) X_A$ holds or not. If it holds, the proxy signer B accepts the delegation W on the warrant m_w, otherwise rejects.

Proxy Key Generation: After the delegation W is validated, the proxy signer B computes the proxy signing key d_{psk} by using the original signer $A's$ delegation key and proxy signer $B's$ private key d_{ID_B}.

Compute $d_{psk} = h_A d_{ID_B}$ where $h_A = H_2\left(ID_A, X_A, m_w\right)$.

Proxy Signature Generation: In order to generate a valid proxy signature, this algorithm takes system parameters, proxy signing key d_{psk} and a message $m \in \{0,1\}^{l_1}$ as input and does the following.

1) Chooses $r_B \in Z_q^*$ and computes $X_B = \hat{e}\left(P_{pub}, r_B Q_{ID_B}\right)$.

2) Computes $\alpha = H_3(ID_B, X_B)$.

3) $\beta = F_1(m) \parallel (F_2(F_1(m)) \oplus m)$.

4) $v = [\alpha \oplus \beta]_{10}$.

5) $V = r_B d_{ID_B} + d_{psk}$.

The proxy signature on the message m is $\Omega = (m_w, V, v, X_A)$.

Message Recovery and Proxy Signature Verification: Given identities ID_A and ID_B and the signature Ω, the verifier performs the following.

1) Compute $\tilde{\alpha} = H_3\left(ID_B, \hat{e}(V,P)\hat{e}\left(P_{pub}, -h_A Q_{ID_B}\right)\right)$, where $h_A = H_2(ID_A, X_A, m_w)$.

2) Compute $\tilde{\beta} = [v]_2 \oplus \tilde{\alpha}$.

3) Recover the message $m' = \left|\tilde{\beta}\right|_{l_1} \oplus F_2(_{l_2}\left|\tilde{\beta}\right|)$.

4) Accept the signature Ω as a valid signature and message $\tilde{m}(=m)$ if and only if $F_1(\tilde{m}) = _{l_2}\left|\tilde{\beta}\right|$.

5. ANALYSIS OF THE PROPOSED IBPSMR SCHEME

This section provides the security analysis and efficiency analysis of the proposed IBPSMR scheme.

5.1. Security Analysis of the Proposed Scheme

In the following we will analyse the security of our IBPSMR scheme.

Proof of Correctness

The correctness of the above scheme may be easily validated according to the following equation.

Consider $\hat{e}(V,P)\hat{e}\left(P_{pub}, -h_A Q_{ID_B}\right)$

$= \hat{e}\left(r_B d_{ID_B} + d_{psk}, P\right)\hat{e}\left(P_{pub}, -h_A Q_{ID_B}\right)$

$$= \hat{e}\left(r_B d_{ID_B}, P\right)\hat{e}\left(d_{psk}, P\right)\hat{e}\left(P_{pub}, -h_A Q_{ID_B}\right)$$

$$= \hat{e}\left(r_B s Q_{ID_B}, s^{-1}P_{pub}\right)\hat{e}\left(h_A d_{ID_B}, P\right)\hat{e}\left(P_{pub}, -h_A Q_{ID_B}\right)$$

$$= \hat{e}\left(r_B Q_{ID_B}, P_{pub}\right)\hat{e}\left(h_A s Q_{ID_B}, s^{-1}P_{pub}\right)\hat{e}\left(P_{pub}, -h_A Q_{ID_B}\right)$$

$$= \hat{e}\left(r_B Q_{ID_B}, P_{pub}\right)\hat{e}\left(h_A s Q_{ID_B}, s^{-1}P_{pub}\right)\hat{e}\left(P_{pub}, -h_A Q_{ID_B}\right)$$

$$= \hat{e}\left(r_B Q_{ID_B}, P_{pub}\right)\hat{e}\left(h_A Q_{ID_B}, P_{pub}\right)\hat{e}\left(P_{pub}, -h_A Q_{ID_B}\right)$$

$$= X_B.$$

If $\Omega = \left(m_w, V, v, X_A\right)$ is a valid signature, then $H_3(ID_B, X_B) = \alpha$ and

$$F_1(m) \parallel (F_2(F_1(m)) \oplus m) = \beta = [v]_2 \oplus \alpha.$$

Hence, we obtain $\left|\tilde{\beta}\right|_{l_1} \oplus F_2(_{l_2}\left|\tilde{\beta}\right|) = (F_2(F_1(m)) \oplus m) \oplus F_2(F_1(m)) = m.$

Finally, the integrity of m is justified if $F_1(m) = {}_{l_2}|\beta|$.

Unforgeability

We discuss the security analysis of the proposed IBPSMR scheme against Type 2 and Type 3 adversaries and also demonstrate that our proposed scheme is secure against existential forgery in the random oracle model with the assumption that the CDH problem is intractable. We prove the security of the proposed scheme by the following theorems.

Theorem 1. The proposed IBPSMR scheme is existentially unforgeable under the adaptive chosen message and identity attacks against the type 2 adversary $\mathcal{A}_2$ in the random oracle model provided the CDH problem is intractable by any polynomial time-bounded algorithm.

Proof. Suppose $\mathcal{A}_2$ is a probabilistic polynomial time forger who can break the proposed IBPSMR scheme with non negligible advantage. We will now construct an algorithm $\mathcal{B}$ which outputs the CDH solution abP for a given CDH instance (P, aP, bP) in G_1. Algorithm $\mathcal{B}$ executes the following simulation by interacting with the forger $\mathcal{A}_2$. Algorithm $\mathcal{B}$ simulates an original signer to attain a valid signature from the forger $\mathcal{A}_2$, and by doing so can solve the CDH problem.

Setup: $\mathcal{B}$ sets the system's overall public key as $P_{pub} = aP$ and starts by giving $\mathcal{A}_2$ the system parameters *Params* including P and P_{pub}.

The random oracles H_1, H_2, Key Extract and Delegation Queries can be queried by $\mathcal{A}_2$ at any time.

H_1- **Queries:** Algorithm $\mathcal{B}$ maintains an initial-empty H_1-list, which contains tuples of the form (*ID, x, y, z*). When $\mathcal{A}_2$ makes queries on the oracle H_1 at a point $ID \in \{0,1\}^*$, $\mathcal{B}$ responds as follows:

1) If the query *ID* already exists on the H_1-list in a tuple (*ID, x, y, z*) then $\mathcal{B}$ responds with $H_1(ID) = z \in G_1$.
2) If not, $\mathcal{B}$ selects a coin $y \in \{0, 1\}$ randomly, such that $pr[y = 0] = 1/(q_E + 1)$.
3) Algorithm $\mathcal{B}$ selects $x \in Z_q^*$ randomly,

   If $y = 0$, $\mathcal{B}$ computes $z = x(bP) \in G_1$.

   If $y = 1$, $\mathcal{B}$ computes $z = xP \in G_1$.
4) $\mathcal{B}$ adds the tuple (*ID, x, y, z*) to the H_1-list and responds to $\mathcal{A}_2$ with $H_1(ID) = z \in G_1$.

H_2- **Queries:** Algorithm $\mathcal{B}$ maintains an initial-empty H_2-list, which contains tuples of the form (ID, m_w, X, h). When $\mathcal{A}_2$ makes queries on the oracle H_2 at a point $ID \in \{0,1\}^*$, $\mathcal{B}$ responds as follows:

1) If H_2-list is with the queried tuple (ID, m_w, X) then $\mathcal{B}$ responds with $H_2(ID, m_w, X) = h \in Z_q^*$.
2) If not, $\mathcal{B}$ selects $h \in Z_q^*$ randomly and adds the tuple (ID, m_w, X, h) in the H_2-list and responds to $\mathcal{A}_2$ with $H_2(ID, m_w, X) = h \in Z_q^*$.

Key Extract Queries: After obtaining the private key query on an identity ID by $\mathcal{A}_2$, $\mathcal{B}$ initially recovers the corresponding tuple (ID, x, y, z) from the $H_1 - list$ and performs the following.

1) If y=0, then $\mathcal{B}$ outputs failure and halts.
2) If not, $\mathcal{B}$ computes $d_{ID} = xP_{pub} = x(aP) \in G_1$ by using the tuple (ID, x, y, z) in the $H_1 - list$ and returns d_{ID} to $\mathcal{A}_2$.

Delegation: After obtaining $\mathcal{A}_2$'s query on a given warrant m_{w_A} for an original signer with the identity ID_A, $\mathcal{B}$ initially confirms that $\{ID_A, m_{w_A}\}$ was not requested previously. If $\{ID_A, m_{w_A}\}$ was requested previously, then $\mathcal{B}$ returns failure and halts, or else performs the following.

1) Executes H_1 query on ID_A and get the corresponding instance of tuple (ID_A, x_A, y_A, z_A) from $H_1 - list$.
2) Computes $X_A = \hat{e}\left(P_{pub}, r_A Q_{ID_A}\right)$, where $r_A \in Z_q^*$ is randomly chosen.
3) Executes H_2 query on (ID_A, m_{w_A}, X_A) and get the corresponding instance of tuple $(ID_A, m_{w_A}, X_A, h_A)$ from $H_2 - list$.
4) If $y_A = 0$ holds, then returns failure and halts, or else computes $Y = (r_A x_A + h_A x_A) P_{pub}$ and returns $W = (X_A, Y)$ as a signature/delegation on m_{w_A}.

Output/Forgery: Finally, $\mathcal{A}_2$ terminates by admitting failure, as does $\mathcal{B}$ or returns a forgery $W = (X_A, Y)$ for the given warrant m_{w_A} under ID_A. Algorithm $\mathcal{B}$ attains (ID_A, x_A, y_A, z_A) from $H_1 - list$, declares failure if $z_A = 1$ and halts. Otherwise, computes $Q_{ID_A} = x_A(bP)$, for $z_A = 0$. This forged signature/delegation W should satisfy $\hat{e}(Y, P) = \hat{e}\left(P_{pub}, h_A Q_{ID_A}\right) X_A$.

Consider $\hat{e}(Y, P)$

$$= \hat{e}\left((r_A + h_A) d_{ID_A}, P\right)$$

$$= \hat{e}\left(r_A d_{ID_A}, P\right)\hat{e}\left(h_A d_{ID_A}, P\right)$$

$$= \hat{e}\left(r_A s Q_{ID_A}, s^{-1} P_{pub}\right)\hat{e}\left(h_A s Q_{ID_A}, s^{-1} P_{pub}\right)$$

$$= \hat{e}\left(r_A Q_{ID_A}, P_{pub}\right)\hat{e}\left(h_A Q_{ID_A}, P_{pub}\right)$$

$$= \hat{e}\left(h_A Q_{ID_A}, P_{pub}\right) X_A.$$

Now $\mathcal{B}$ recovers the respective tuple $(ID_A, m_{w_A}, X_A, h_A)$ from $H_2 - list$ and computes $Y = (r_A x_A + h_A x_A) P_{pub}$.

Consider $\hat{e}(Y, P)$

$$= \hat{e}\left(h_A Q_{ID_A}, P_{pub}\right) X_A$$

$$= \hat{e}\left(h_A Q_{ID_A}, P_{pub}\right)\hat{e}\left(P_{pub}, r_A Q_{ID_A}\right)$$

$$= \hat{e}\left(h_A x_A (bP), aP\right)\hat{e}\left(aP, r_A x_A (bP)\right)$$

$$= \hat{e}\left(aP, h_A x_A (bP) + r_A x_A (bP)\right)$$

$$= \hat{e}\left(P, h_A x_A (abP) + r_A x_A (abP)\right)$$

$$= \hat{e}\left(P, (h_A + r_A) x_A (abP)\right)$$

$$\Rightarrow Y = (h_A + r_A) x_A (abP)$$

$$\Rightarrow abP = Y (h_A + r_A)^{-1} x_A^{-1}.$$

This concludes the proof of theorem 1 and hence the description of algorithm $\mathcal{B}$.

Theorem 2. If the CDH problem is (T', ε') - hard, the scheme IBPSMR is $(T, q_{H_1}, q_{H_2}, q_{H_3}, q_{F_1}, q_{F_2}, q_E, q_S, \varepsilon)$ - secure against existential forgery under adaptive chosen-message and ID attacks for any T and ε satisfying $\varepsilon \geq e(q_E + 1)\, \varepsilon'$, $T \leq T' - T_{EM}(1q_{H_1} + 1q_E + 2q_S + 2)$, where e is the base of the natural logarithm,

T_{EM} is the time for computing a scalar multiplication in G_1. Also q_{H_1}, q_{H_2}, q_{H_3}, q_{F_1}, q_{F_2} denote the number of queries made to the hash oracle, q_E denotes the number of queries made to the key extract oracle and is q_S denotes the number of queries made to the sign oracle.

Proof. Suppose $\mathcal{A}_3$ is a probabilistic polynomial time forger who can break the proposed IBPSMR scheme with non negligible advantage. We will now construct an algorithm $\mathcal{B}$ which outputs the CDH solution abP for a given CDH instance (P, aP, bP) in G_1. Algorithm $\mathcal{B}$ executes the following simulation by interacting with the forger $\mathcal{A}_3$. Algorithm $\mathcal{B}$ simulates an original signer to attain a valid signature from the forger $\mathcal{A}_3$, and by doing so can solve the CDH problem.

The setup phase, queries to the oracles H_1, H_2, key extraction, made by the forger $\mathcal{A}_3$ are similar to that of the forger $\mathcal{A}_2$, described in the proof under Theorem 1.

The oracles H_1, H_2, H_3, F_1, F_2, key extraction and proxy sign with $\mathcal{B}$ can be queried by $\mathcal{A}_3$, at any time.

H_3- **Queries:** $\mathcal{A}_3$ makes queries to the oracle H_3 at a point $X_B \in G_2$, at any time. To respond to the queries made by $\mathcal{A}_3$ to X_B, the algorithm $\mathcal{B}$ maintains an initial-empty H_3-list, which contains tuples of the form (ID_B, X_B, γ) and proceeds as follows.

1) If the queried X_B already exists on the H_3-list in a tuple (ID_B, X_B, γ) then $\mathcal{B}$ responds with $H_3(ID_B, X_B) = \gamma$.
2) If not, $\mathcal{B}$ selects $\gamma \in \{0,1\}^{|q|}$, and adds the tuple (ID_B, X_B, γ) in the H_3-list and responds to $\mathcal{A}_3$ with $H_3(ID_B, X_B) = \gamma$.

$F_1(\cdot)$ **and** $F_2(\cdot)$ **Queries:** $\mathcal{A}_3$ makes queries to the random oracles $F_1(\cdot)$ and $F_2(\cdot)$ at any time. To respond to the queries made by $\mathcal{A}_3$ to the oracles $F_1(\cdot)$ and $F_2(\cdot)$, $\mathcal{B}$ simulates the

oracles $F_1(\cdot)$ and $F_2(\cdot)$ in a similar way as that of the $H_1(\cdot)$ oracle, by maintaining F_1-list and F_2-list of tuples respectively.

Proxy Sign Queries: When $\mathcal{A}_3$ queries a signature on a message m for an identity ID_B, $\mathcal{B}$ proceeds as follows.

1) Chooses a random integer $r_B \in Z_q^*$ and computes $X_B = \hat{e}\left(P_{pub}, r_B Q_{ID_B}\right)$.
2) Retrieves the H_3-list and sets $\gamma = H_3(ID_B, X_B)$.
3) Computes $\beta = F_1(m) \parallel (F_2(F_1(m)) \oplus m)$ and $v = [\alpha \oplus \beta]_{10}$.
4) Also computes $V = (r_B + h_A) x_B P_{pub}$.

Outputs the proxy signature $\Omega = (m_w, V, v, X_B)$ on a message m.

All responses to the sign queries are valid; in fact the output of sign queries is a valid proxy signature on m under ID_B. In order to see this,

Consider $\hat{e}(V,P)\hat{e}\left(P_{pub}, -h_A Q_{ID_B}\right)$

$$= \hat{e}\left(r_B d_{ID_B} + d_{psk}, P\right)\hat{e}\left(P_{pub}, -h_A Q_{ID_B}\right)$$

$$= \hat{e}\left(r_B d_{ID_B}, P\right)\hat{e}\left(d_{psk}, P\right)\hat{e}\left(P_{pub}, -h_A Q_{ID_B}\right)$$

$$= \hat{e}\left(r_B s Q_{ID_B}, s^{-1} P_{pub}\right)\hat{e}\left(h_A d_{ID_B}, P\right)\hat{e}\left(P_{pub}, -h_A Q_{ID_B}\right)$$

$$= \hat{e}\left(r_B Q_{ID_B}, P_{pub}\right)\hat{e}\left(h_A s Q_{ID_B}, s^{-1} P_{pub}\right)\hat{e}\left(P_{pub}, -h_A Q_{ID_B}\right)$$

$$= \hat{e}\left(r_B Q_{ID_B}, P_{pub}\right)\hat{e}\left(h_A s Q_{ID_B}, s^{-1} P_{pub}\right)\hat{e}\left(P_{pub}, -h_A Q_{ID_B}\right)$$

$$= \hat{e}\left(r_B Q_{ID_B}, P_{pub}\right)\hat{e}\left(h_A Q_{ID_B}, P_{pub}\right)\hat{e}\left(P_{pub}, -h_A Q_{ID_B}\right)$$

$$= X_B.$$

Output/Forgery: Finally, $\mathcal{A}_3$ terminates. It either admits failure, in which case so does $\mathcal{B}$ or it returns a forged proxy signature Ω on m under ID_B. Algorithm $\mathcal{B}$ attains

(ID_B, x_B, y_B, z_B) from $H_1 - list$, declares failure if $y_B = 1$ and halts. Otherwise, computes $z_B = x_B(bP)$, for $y_B = 0$. This forged proxy signature should satisfy

$$\hat{e}(V,P)\hat{e}\left(P_{pub}, -h_A Q_{ID_B}\right) = X_B.$$

$$\Rightarrow \hat{e}(V,P) = X_B\ \hat{e}\left(P_{pub}, Q_{ID_B}\right)^{h_A}$$

$$= \hat{e}\left(r_B Q_{ID_B}, P_{pub}\right)\hat{e}\left(P_{pub}, Q_{ID_B}\right)^{h_A}$$

$$= \hat{e}\left(Q_{ID_B}, P_{pub}\right)^{r_B}\ \hat{e}\left(P_{pub}, Q_{ID_B}\right)^{h_A}$$

$$= \hat{e}\left(Q_{ID_B}, P_{pub}\right)^{r_B + h_A}$$

$$= \hat{e}\left(x_B(bP), aP\right)^{r_B + h_A}$$

$$= \hat{e}\left(x_B(abP)(r_B + h_A), P\right)$$

$$\Rightarrow V = x_B(abP)(r_B + h_A)$$

$$\Rightarrow abP = V x_B^{-1}(r_B + h_A)^{-1}.$$

This completes the description of algorithm $\mathcal{B}$.

Algorithm $\mathcal{B}'s$ running time is the same as $\mathcal{A}_3's$ running time plus the time it takes to respond to $\left(q_{H_1} + q_{H_2} + q_{H_3}\right)$ hash queries, q_E key extract queries and q_S sign queries and the time to transform $\mathcal{A}_3's$ final forgery in the CDH solution. From the above simulation, we observe that there exists: $1T_{EM}$ operations in each H_1 query, $1T_{EM}$ operations in each key extraction query, $2T_{EM}$ operations in each sign query and $2T_{EM}$ operations in the output phase. Hence the total running time is at most $T \leq T' - T_{EM}(1q_{H_1} + 1q_E + 2q_S + 2)$, as required.

This concludes the proof of Theorem 2.

Therefore, the security reduction of our proposed scheme does not use Forking lemma [33], and as discussed in [32, 33], the obtained security is tightly related to the CDH problem.

5.2. Efficiency of the Proposed IBPSMR

We present the performance analysis of our proposed IBPSMR scheme by comparing it with the relevant schemes [24,26,28,20,29,22] in terms of computational and communicational (signature length) cost point of view. We consider various cryptographic operations and their conversions, which are provided in TABLE 3. The conversions of these cryptographic operations have been taken from the experimental results [34-37]. Besides, we have mentioned all the cryptographic operations in terms of modular multiplications.

TABLE 3. Notations and descriptions of different cryptographic operations and their conversions

Notations	Time required to perform these operations
T_{ML}	Time required to perform the modular multiplication operation
T_{EM}	Elliptic curve point multiplication (Scalar multiplication in G_1) : $T_{EM} \approx 29T_{ML}$
T_{BP}	Bilinear pairing operation in G_2 : $T_{BP} \approx 87T_{ML}$
T_{PX}	Pairing-based exponentiation operation in G_2 : $T_{PX} \approx 43.5T_{ML}$
T_{EX}	Modular exponentiation operation in Z_q^* : $T_{EX} \approx 240T_{ML}$
T_{IN}	Modular inversion operation in Z_q^* : $T_{IN} \approx 11.6T_{ML}$
T_{MTP}	Map-to-point (hash function): $T_{MTP} \approx T_{EM} \approx 29T_{ML}$
T_{PA}	Addition of 2 elliptic curve points (point addition in G_1): $T_{PA} \approx 0.12T_{ML}$

5.2.1. Computational Efficiency

TABLE 4 gives an overview of the comparison between our proposed IBPSMR scheme and the existing proxy signature schemes in terms of computation. It clearly shows that the total computational cost of our proposed scheme is $725.12\ T_{ML}$ which is 13.81% less than Singh and Verma [24] scheme, 15.27% less than Yoon *et al*. [26] scheme, 20.67% less than Zhou [28] scheme, 19.10% less than Sarde and Banerjee [20] scheme, 18.04% less than Asaar *et al*. [29] scheme and 35.92% less than Liu *et al*. [22] scheme. Hence, compared to the existing schemes [24,26,28,20,29,22], our scheme is more efficient in terms of computational complexity.

TABLE 4. Computational efficiency comparison of our proposed scheme with the related schemes

Scheme	Delegation Generation Cost	Delegation Verification Cost	Proxy Signature Cost	Proxy Verification Cost	Total Cost
Singh and Verma (2012) [24]	$1T_{BP}+1T_{PX}+2T_{EM}+1T_{PA}$	$1T_{MTP}+2T_{BP}+1T_{PX}$	$1T_{BP}+1T_{PX}+2T_{EM}+2T_{PA}$	$2T_{BP}+1T_{PX}$	841.36 T_{ML}
Yoon *et al.* (2013) [26]	$1T_{MTP}+2T_{EM}+1T_{PA}$	$3T_{BP}+1T_{EM}$	$1T_{MTP}+2T_{EM}+1T_{PA}$	$4T_{BP}+1T_{PX}+1T_{PA}$	855.86 T_{ML}
Zhou (2015) [28]	$1T_{BP}+1T_{PX}+2T_{EM}+1T_{PA}$	$1T_{MTP}+2T_{BP}+1T_{PX}$	$1T_{BP}+1T_{PX}+3T_{EM}+2T_{PA}$	$2T_{BP}+2T_{PX}+2T_{PA}$	914.10 T_{ML}
Sarde and Banerjee (2015) [20]	$1T_{MTP}+4T_{EM}+1T_{PA}$	$4T_{BP}+2T_{PX}$	$2T_{EM}+1T_{IN}$	$2T_{BP}+1T_{PX}+1T_{EM}+1T_{PA}$	896.34 T_{ML}
Asaar *et al.* (2016) [29]	$1T_{BP}+1T_{PX}+1T_{EM}$	$2T_{BP}+1T_{PX}$	$1T_{BP}+1T_{PX}+1T_{EM}+1T_{PA}$	$3T_{BP}+2T_{PX}$	884.62 T_{ML}
Liu *et al.* (2018) [22]	$1T_{MTP}+2T_{EM}+1T_{PA}$	$2T_{MTP}+3T_{BP}$	$2T_{MTP}+4T_{EM}+4T_{PA}$	$4T_{MTP}+5T_{BP}$	1131.60 T_{ML}
Our Proposed IBPSMR Scheme	$1T_{BP}+2T_{EM}$	$2T_{BP}+1T_{EM}$	$1T_{BP}+3T_{EM}+1T_{PA}$	$2T_{BP}+1T_{EM}$	725.12 T_{ML}

5.2.2. Communicational Efficiency

TABLE 5 gives an overview of the comparison between our proposed IBPSMR scheme and the existing proxy signature schemes in terms of communication. Our proposed scheme and the related proxy signature schemes are constructed on bilinear pairings. In order to attain a security level of 80 bits, in bilinear pairing, we consider $\hat{e}: G_1 \times G_1 \to G_T$ where G_1 is an additive group generated by $\hat{P}$ with the order $\hat{q}$ on the super singular elliptic curve $\hat{E}: y^2 = x^3 + x \bmod \hat{p}$ with embedding degree 2. Here $\hat{p}$ comprises of 512 bit prime number and $\hat{q}$ is of 160 bit solinas prime number. In order to attain the same 80 bit security level, in ECC, we consider G as an additive cyclic group generated by a point P on a non-singular elliptic curve $E: y^2 = x^3 + ax + b \bmod p$ and its order is q where p, q are prime numbers of 160 bit each and $a, b \in Z_q^*$. Accordingly, the size of $\hat{p}$ is 512 bits (i.e. 64 bytes) and the size of p is 160 bits (i.e. 20 bytes). Hence the size of elements in G_1 is $512 \times 2 = 1024$ bits and size of elements in G is $160 \times 2 = 320$ bits. Moreover, the size of the elements in Z_q^* is 160 bits.

The signature length of our proposed scheme is $|q|+|G_1|+|G_2|$ and the communication cost is $20+128+128=276$ bytes. Being a message recovery scheme, the original message in our proposed scheme need not be transmitted along with the signature. TABLE 5, clearly demonstrates that the communication complexity of our scheme is greatly reduced and hence our scheme is more efficient compared to the existing proxy signature schemes [24,26,28,20,29,22].

TABLE 5. Communication efficiency comparison of our proposed scheme with related schemes

Scheme	Message Recovery	Signature length	In Bytes
Singh and Verma (2012) [24]	✓	$\|q\|+\|G_1\|+\|G_2\|$	276 *bytes*
Yoon *et al.* (2013) [26]	×	$\|q\|+3\|G_1\|+\|m\|$	504 *bytes*
Zhou (2015) [28]	✓	$\|q\|+\|G_1\|+\|G_2\|$	276 *bytes*
Sarde and Banerjee (2015) [20]	×	$\|q\|+4\|G_1\|+\|m\|$	632 *bytes*
Asaar *et al.* (2016) [29]	✓	$\|q\|+\|G_1\|+\|G_2\|$	276 *bytes*
Liu *et al.* (2018) [22]	×	$3\|G_1\|+\|m\|$	484 *bytes*
Our Proposed Scheme	✓	$\|q\|+\|G_1\|+\|G_2\|$	276 *bytes*

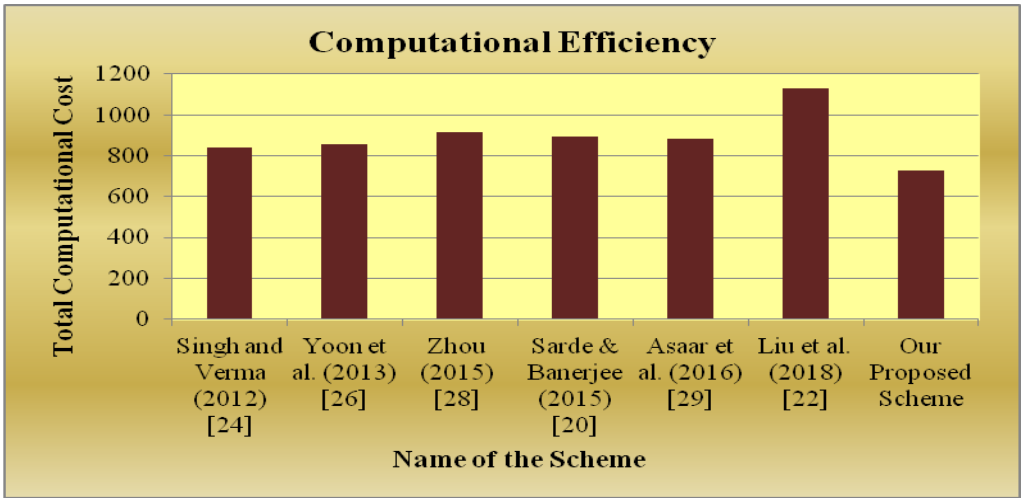

FIGURE 1. Graphical Representation of Total Computation Cost

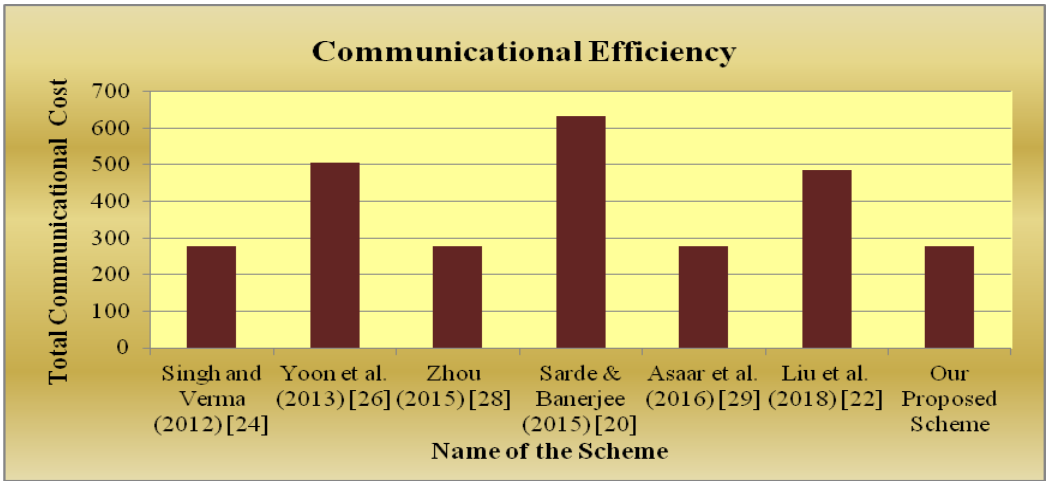

FIGURE 2. Graphical Representation of Total Communication Cost

The above graphical representations (FIGURE 1 and FIGURE 2) clearly indicate that our scheme is considerably more efficient than the existing proxy signature schemes [24,26,28,20,29,22] in terms of computation and communication overhead.

6. CONCLUSION

This paper proposes an efficient ID-based proxy signature scheme with message recovery based on bilinear pairings. In view of the desirable advantages, such as proxy with message recovery feature in the ID-based setting, our proposed scheme can be implemented across a broad range of practical applications and provides an innovative approach to low bandwidth proxy signatures with more flexible management of public keys. The proposed scheme is secure against existential forgery under adaptively chosen message and identity attacks in the random oracle model with the assumption that the CDH problem is intractable. Additionally, in order to attain tight security, the proposed scheme does not use the forking lemma. Hence our proposed scheme has the best performance and is efficient in terms of computation and communication overhead.

CONFLICT OF INTERESTS

The author(s) declare that there is no conflict of interests.

REFERENCES

[1] A. Shamir, Identity-Based Cryptosystems and Signature Schemes. In: Blakley G.R., Chaum D. (eds) Advances in Cryptology. CRYPTO 1984. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, 196 (1985), 47-53.

[2] S. G. Galbraith, K. Harrison and D. Soldera, Implementing the Tate pairing, Algorithmic Number Theory, 5th International Ymposium, ANTS-V, Springer-Verlag, Lecture Notes in Computer Science, 2369 (2002), 324-337.

[3] M. Mambo, K. Usuda and E. Okamoto, Proxy signatures: delegation of the power to sign messages, IEICE Trans. Fundam. Electron. Commun. Computer Sci. 79 (9) (1996), 1338-1354.

[4] T. Okamoto, M. Tada and E. Okamoto, Extended Proxy Signatures for Smart Cards, Extended Proxy Signatures for Smart Cards. In: Information Security. ISW 1999. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, 1729 (1999), 247-258.

[5] B. Lee, H. Kim and K. Kim, Secure Mobile Agent using Strong Non-Designated Proxy Signature, Inf. Security and Privacy (ACISP'01), Springer-Verlag, Lecture Notes in Computer Science, 2119 (2001), 474-486.

[6] H. U. Park and L.Y. Lee, A Digital Nominative Proxy Signature Scheme for Mobile Communication. In: Qing S., Okamoto T., Zhou J. (eds) Information and Communications Security. ICICS 2001. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, 2229 (2001), 451-455.

[7] I. Foster, C. Kesselman, G. Tsudik and S. Tuecke, A Security Architecture for Computational Grids, 5th ACM Conference on Computer and Communications Security Conference, San Francisco, USA, (1998), 83-92.

[8] J. Leiwo, C. Hanle, P. Homburg and A. S. Tanenbaum, Disallowing Unauthorized State Changes of Distributed Shared Objects. In: Qing S., Eloff J.H.P. (eds) Information Security for Global Information Infrastructures. SEC 2000. IFIP - The International Federation for Information Processing, Springer, Boston, MA, 47 (2000), 381-390.

[9] K. Nyberg and R. A. Rueppel, A New Signature Scheme based on the DSA giving Message Recovery, Proceedings of the 1st ACM conference on Computer and communications security, Virginia, USA, (1993), 58-61.

[10] F. Zhang and K. Kim, Efficient ID-Based Blind Signature and Proxy Signature from Bilinear Pairings. In: Safavi-Naini R., Seberry J. (eds) Information Security and Privacy. ACISP 2003. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, 2727 (2003), 312–323.

[11] J. Xu, Z. Zhang and D. Feng, ID-Based Proxy Signature Using Bilinear Pairings. In: Chen G., Pan Y., Guo M., Lu J. (eds) Parallel and Distributed Processing and Applications - ISPA 2005 Workshops. ISPA 2005. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, 3759 (2005), 359-367.

[12] C. Gu and Y. Zhu, An Efficient ID-Based Proxy Signature Scheme from Pairings. In: Pei D., Yung M., Lin D., Wu C. (eds) Information Security and Cryptology. Inscrypt 2007. Lecture Notes in Computer Science, vol 4990. Springer, Berlin, Heidelberg (2008), 40-50.

[13] H. Mala, M. Dakhil-Alian and M. Brenjkoub, A New Identity-based Proxy Signature Scheme from Bilinear Pairings, 2nd International Conference on Information & Communication Technologies, IEEE, (2006),

3304-3308.

[14] F. Hess, Efficient Identity Based Signature Schemes Based on Pairings. In: Nyberg K., Heys H. (eds) Selected Areas in Cryptography. SAC 2002. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, 2595 (2003), 310-324.

[15] K. Shim, An Identity-Based Proxy Signature Scheme from Pairings. In: Ning P., Qing S., Li N. (eds) Information and Communications Security. ICICS 2006. Lecture Notes in Computer Science, 4307 (2006), 60-71.

[16] W. Wu, Y. Mu, W. Susilo, J. Seberry and X. Huang, Identity-Based Proxy Signature from Pairings. In: Xiao B., Yang L.T., Ma J., Muller-Schloer C., Hua Y. (eds) Autonomic and Trusted Computing. ATC 2007. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, 4610 (2007), 22-31.

[17] B.Wang, A New Identity Based Proxy Signature Scheme, IACR Cryptology ePrint Archive, 323 (2008).

[18] M. A. Behesthi, M. Gardeshi and M. Bayat, Formal Security of an Identity Based Proxy Signature Scheme in the Random Oracle Model, Int. J. Mach. Learn. Comput. 2(3) (2012), 298-300.

[19] H. Elkamchouchi, E. A. El-kheir and Y. Abouelseoud, A pairing-free Identity based proxy signature scheme, Int. J. Sci. Eng. Res. 5(4) (2014), 1104-1108.

[20] P. Sarde and A. Banerjee, A secure ID-based proxy signature scheme from bilinear pairings, Int. J. Computer Appl. 124(9) (2015), 1-4.

[21] D. Mukherjee, P. Vyavahare and M. Panchal, An improved ID based proxy signature scheme based on elliptic curve cryptography, (SECURWARE 2016), The Tenth International Conference on Emerging Security Information, Systems and Technologies, (2016), 235-240.

[22] W. Liu, Y. Mu, G. Yang and Y. Tian, Strong Identity-Based Proxy Signature Schemes: Revisited, Wireless Commun. Mobile Comput. 2018 (2018), Article ID 6925019.

[23] T. Wu, C. Hsu and H. Lin, Self certified multiproxy signature scheme with message recovery, J. Zhejiang Univ. Sci. A, 10(2) (2009), 290-300.

[24] H. Singh and G.K. Verma, ID-based proxy signature scheme with message recovery, J. Syst. Softw. 85(1) (2012), 209–214.

[25] M. Tian, L. Huang and W. Yang, Cryptanalysis of an ID-based proxy signature scheme with message recovery, Appl. Math. Inf. Sci. 6 (2012), 419 - 422.

[26] E. Yoon, Y. S. Choi and C. Kim, New ID-Based Proxy Signature Scheme with Message Recovery. In: Park J.J..H., Arabnia H.R., Kim C., Shi W., Gil JM. (eds) Grid and Pervasive Computing. GPC 2013. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, 7861 (2013), 945-951.

[27] S. Padhye and N. Tiwari, ECDLP based Certificateless Proxy Signature Scheme with message recovery, Trans. Emerg. Telecomm. Tech. 26(3) (2015), 346-354.

[28] C. Zhou, An Improved ID-based Proxy Signature Scheme with Message Recovery, Int. J. Security Appl. 9 (9) (2015), 151-164.

[29] M. Asaar, M. Salmasizadeh and W. Susilo, A short ID-based proxy signature scheme, Int. J. Commun. Syst. 29 (5) (2016), 859-873.

[30] A. Mahmoodi, J. Mohajery and M. Salmasizadeh, A certificate-based proxy signature with message recovery without bilinear pairings, Security Commun. Networks, 9 (18) (2016), 4983-4991.

[31] E. J. Goh and S. Jarecki, A Signature Scheme as Secure as the Diffie-Hellman Problem. In: Biham E. (eds) Advances in Cryptology - EUROCRYPT 2003. EUROCRYPT 2003. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, 2656 (2003), 401-415.

[32] J. Katz and N. Wang, Efficiency Improvements for Signature Schemes with Tight Security Reductions, Proc. 10th ACM Conf. Computer Commun. Security, (2003), 155-164.

[33] D. Pointcheval and J. Stern, Security arguments for digital signatures and blind signatures, J. Cryptol. 13 (3) (2000), 361-396.

[34] P. S. L. M. Barreto, B. Libert, N. McCullagh and J. Quisquater, Efficient and Provably-Secure Identity-Based Signatures and Signcryption from Bilinear Maps. In: Roy B. (eds) Advances in Cryptology - ASIACRYPT 2005. ASIACRYPT 2005. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, 3778 (2005), 515-532.

[35] X. Cao, W. Kou and X. Du, A Pairing-free Identity Based Authenticated Key Agreement Protocol with Minimal Message Exchanges, Inf. Sci. 180 (15) (2010), 2895–2903.

[36] S. H. Tan, S. H. Heng and B. M. Goi, Java Implementation for Pairing-Based Cryptosystems. In: Taniar D., Gervasi O., Murgante B., Pardede E., Apduhan B.O. (eds) Computational Science and Its Applications – ICCSA 2010. ICCSA 2010. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, 6019 (2010), 188-198.

[37] Shamus Software Ltd. Miracl Library, Available: http://certivox.org/display/EXT/MIRACL.