Efficient ID-based multi-proxy multi-signature scheme based on CDHP

Rajeev Anand Sahu and Sahadeo Padhye
Department of Mathematics
Motilal Nehru National Institute of Technology Allahabad-211004, India
Email: rajeevs1729@gmail.com, sahadeomathrsu@gmail.com

***Abstract*—Proxy signature is useful in situation when a user wants to authorize an agent called proxy signer to sign any document on his behalf. Multi-proxy multi-signature is one of the primitives of proxy signature. Bilinear pairing makes the system efficient and provides an ease in computation. In this paper, we propose an ID-based multi-proxy multi-signature scheme from bilinear pairings based on Computational Diffie-Hellman Problem (CDHP), replacing the certificate generation of Li and Chen's scheme by delegation generation. Our scheme is proxy protected and computationally more efficient than the ID-based multi-proxy multi-signature scheme of Li and Chen. We also analyze the security properties of our scheme and show that the proposed scheme satisfies all the security requirements of a safe proxy signature scheme.**

Key Words: Bilinear pairings, Cryptography, ID-based signature scheme, Multi-proxy multi-signature, Security analysis.

I. Introduction

In the traditional public key cryptography, to communicate a message, users first obtain the authenticated public key from certificate authority. In that system, large overhead to transfer certificates and their maintenance increases the associated cost significantly. In ID-based cryptography, the users public and private keys are generated from their identities such as email address, IP-address etc. In this way the ID-based setting simplifies the key management procedure and provides added security, hence it is an economical alternative of the traditional certificate-based settings. In 1984, Shamir [18] introduced the concept of ID-based cryptosystem and signature scheme. After the work of Shamir [18], many signature schemes [3], [15], [20], [22] have been proposed using the keys generated by the identities of users. Bilinear pairings are very much useful for the ease of computation in various cryptosystems. The pairing has property of linearity in both co-ordinate which makes it computationally easy and functionally strong. Hence the notion of bilinear pairing brought a new and efficient method of computation. In 2001, Boneh and Franklin [1], presented a practical ID-based encryption scheme which took advantage of the properties of admissible bilinear parings over supersingular elliptic curves. This work of Boneh and Franklin encouraged many authors to design efficient signature schemes. Most of the ID-based key agreement protocol and signature schemes [2], [5], [16], [19], [25] have been designed using bilinear pairings (Weil pairing or Tate pairing).

Proxy signature enables any original signer to delegate its signing rights to any other user called proxy signer. It is very much applicable in scenarios when the original signer is absent at the time of signing any document. The concept of proxy signature was introduced by Mambo, Usuda and Okamoto [14] in 1996. Later in 1997, Kim et. al. [8] extended the notion by using Schnorr signature and including warrant information in partial delegation schemes.

In a proxy signature scheme, an original signer can delegate its signing capability to a proxy signer, and having the signing rights, the proxy signer can sign a message on behalf of the original signer. According to the number of signers in the original and proxy group, the proxy signature can be categorized in multi-proxy signature, proxy multi-signature and multi-proxy multi-signature. The concept of multi-proxy signature is applicable when an original signer needs to delegate its signing right to a group of proxy signers. The idea of multi-proxy signature was introduced by Hwang and Shi [7] in 2000. In contrary situations, when a single proxy signer is required to sign any document on behalf of the group of original signers, the notion of proxy multi-signature is useful. The concept was firstly proposed by Yi *et al.* [26] in 2000. The third type of signature, called multi-proxy multi-signature, was proposed by Hwang and Chen [6] in 2004. A multi-proxy multi-signature is a signature, generated by a group of proxy signers on behalf of the group of original signers. Proxy signature schemes are useful in many applications, particularly in distributed computing where delegation of rights is quite common. Some applications discussed in the literature include grid computing, global distribution networks, mobile agent applications, distributed shared objects, mobile communications etc.

In [14] Mambo *et al.* classified the proxy signatures based on delegation types as full delegation, partial delegation, and delegation by warrant. In full delegation type, the original signer delegates its private key directly to the proxy signer. So, the proxy signer can use the same signing rights as the original one. Such systems are insecure in practice. Hence, mostly this delegation is avoided in proxy signature protocols. In the second type of delegation i.e. partial delegation, the proxy signer posses a private proxy key, differ from original signer's private key. But since in this scheme, the proxy signer is independent from the original signer to sign any document, he can sign as many documents as he wants,

and hence this system lacks control on the rights of proxy signer. In many real world scenarios the facility of proxy signer, to sign unlimited documents is not acceptable. The delegation by warrant is a solution of the weakness due to above delegation types. In delegation by warrant, the original signer makes a signature on warrant using some standard signature scheme and its private key. Then he sends the signature (signed warrant) as delegation value to the proxy signer. The proxy signer then creates a signature on given message on behalf of the original signer using the delegation value and his private key. According to the privilege of original signer, the proxy signatures can be categorized in proxy protected and proxy unprotected schemes. In a proxy protected scheme, the original signer can not generate a valid proxy signature whereas in a proxy unprotected scheme, the proxy signature can be generated by either of the original and proxy signer. In this paper, we propose an ID-based multi-proxy multi-signature scheme using delegation by warrant. Moreover, our proposed scheme is proxy protected.

In 2005, Li and Chen [9] proposed the ID-based multi-proxy signature, proxy multi-signature and multi-proxy multi-signature schemes from bilinear pairings. Their multi-proxy signature and proxy multi-signature schemes can be regarded as special cases of corresponding variants of ID-based threshold signature schemes. In their paper [9], they have proposed the multi-proxy multi-signature scheme, combining the multi-proxy signature and proxy multi-signature generating a certificate for the group of proxy signers. The building blocks of all the signature schemes given in [9] is the ID-based signature scheme of Hess [5]. The security of Hess's signature scheme depends on the security of CDHP, hence signatures proposed in [9] are unforgeable due to the hardness of CDHP.

A. Our Contribution:

In [9], Li and Chen has proposed an ID-based multi-proxy multi-signature scheme from bilinear pairings generating a certificate for the proxy signers. To improve the efficiency of multi-proxy multi-signature scheme proposed in [9], in this paper, we have designed an ID-based multi-proxy multi-signature scheme from bilinear pairings. Instead of generating any certificate for proxy signers like [9], we have designed the scheme using delegation generation, which reduces the associated computational cost. We also analyze the security properties of our scheme and show that the proposed scheme satisfies all the security requirements of a safe and sound proxy signature described in [12].

B. Organization of this paper:

The rest of this paper is organized as follows: In Section 2, we introduce the bilinear pairing, some related mathematical problems and the security requirements of a proxy signature. The formal model of a proxy signature scheme is a described in section 3. In Section 4, we briefly review the Hess's signature scheme and the ID-based multi-proxy multi-signature scheme of Li and Chen. Our proposed scheme is described in Section 5. In Section 6, We analyze the security properties of our scheme and compare the computational efficiency of our scheme with that of [9]. Finally Section 7 concludes the paper.

II. Preliminaries

In this section, we briefly introduce the concept of bilinear pairing, some related mathematical problems and the security requirements for a proxy signature.

A. Bilinear Pairing:

Given a cyclic additive group G_1 and a cyclic multiplicative group G_2. Order of both the groups is a prime number say q, then a map $e : G_1 \times G_1 \rightarrow G_2$ satisfying the following properties, is called bilinear pairing:

(a)Bilinearity: $e(aP, bQ) = e(P, Q)^{ab}$, $\forall a, b \in Z_q^*$ and $P, Q \in G_1$.
(b)Non-Degeneracy: There exists $P, Q \in G_1$ such that $e(P, Q) \neq 1$, or in other words if P is generator of G_1, then $e(P, P)$ is generator of G_2.
(c)Computability: There exists an efficient algorithm to compute $e(P, Q) \in G_2$, $\forall P, Q \in G_1$.

Modified Weil pairing and Tate pairing are examples of cryptographic bilinear pairings.

B. Discrete logarithm problem (DLP):

For given two elements $P, Q \in G_1$, to compute an integer $n \in Z_q^*$, such that $P = nQ$.

C. Computational Diffie-Hellman Problem(CDHP):

For given $P, aP, bP \in G_1$, to compute $abP \in G_1$, where $a, b \in Z_q^*$.

D. Bilinear Pairing Inversion Problem (BPIP):

Given $P \in G_1$, and $e(P, Q) \in G_2$, to find $Q \in G_1$.

E. Security requirements for a proxy signature:

In 2001, Lee et. al. proposed some extensions on security requirements of a proxy signature scheme proposed by Mambo et. al. [14] in 1996. According to [12], a secure proxy signature scheme should satisfy the following security properties [12]:

Strong unforgeability: No one, other than the proxy signer, can generate a valid proxy signature.

Verifiability: The signature can be verified by anyone, and the signed message should confirm to the delegation warrant. That means, any verifier can be convinced of the original signer's agreement on the signed message.

Strong identifiability: Identity of corresponding proxy signer can be determined by anyone.

Strong undeniability: The proxy signer cannot deny his signature, he has made ever.

Prevention of misuse: The proxy signer cannot sign any message, which has not been authorized by the original signer. Or alternatively, It should be confident that proxy key cannot be used for other purposes. In the case of misuse, the responsibility of proxy signer should be determined explicitly.

According to the undeniability property, they have also classified the proxy signature schemes into weak and strong category [12].

Weak proxy signature: It represents only original signers signature. It does not provide non-repudiation of proxy signer. Weak proxy signature cannot be used in real world because of many deviations.
Strong proxy signature: It represents both original signers and proxy signers signatures. Once a proxy signer creates a valid proxy signature, he cannot repudiate his signature creation against anyone. Strong proxy signature can be used widely without relying on any trustedness assumption.

Further in proxy key issuing sense, they have classified the proxy signature schemes into designated and non-designated proxy signatures according to designation of proxy signer in proxy key issuing stage. They have also proposed a possibility of self-proxy signature in which the original signer issues a proxy key pair for itself. They have shown that the self-proxy signature can be used for construction of partially blind signature [12].

III. Formal model of ID-based proxy signature

Before going to the details of the proposed ID-based multi-proxy multi-signature scheme, we give here a formal model of the ID-based proxy signature scheme [23].

1. *ParamGen*: This algorithm outputs the system's public parameter *param* and system's master key s, taking the security parameter k as an input.
2. *KeyExtract*: This algorithm gives secret keys S_{ID_A}, S_{ID_B} of original signers A and proxy signers B, taking their identities ID_A and ID_B as inputs.
3. *StandardSign*: This algorithm outputs the standard signature σ_s, taking message m, system's parameter *param* and secret key S_{ID} as input.
4.*StandardVerify*: Taking system's parameter *param*, standard signature σ_s, message m, the signer's identity ID, this algorithm outputs **True** if σ_s is a valid signature, **False** otherwise.
5. *DelegationGen*: Inputs in this algorithm are system's parameter *param*, the original signer's secret key S_{ID_A}, and the warrant to be signed. And output is delegation σ_w, which is generated using the standard signing algorithm.
6. *ProxySign*: This algorithm takes system's parameter *param*, the warrant w, delegation σ_w, the secret key S_{ID_B} of proxy signer, the message m to be signed and outputs proxy signature σ.
7. *ProxyVerify*: This algorithm takes inputs the system's parameter *param*, original signer's identity ID_A, proxy signer's identity ID_B, the warrant w, the message m and the signature σ on message m and outputs **True** if the proxy signature σ is a valid signature on message m, **False** otherwise.

IV. Review of some ID-based signature schemes

In this section, we briefly review the ID-based signature scheme of Hess [5] and the ID-based multi-proxy multi-signature scheme of Li and Chen [9] with the same notations as in [5] and [9]. The ID-based multi-proxy multi-signature scheme in [9] is designed combining the ID-based multi-proxy signature and the ID-based proxy multi-signature schemes proposed in [9]. The security of our scheme and all the schemes proposed in [9] depends on the security of Hess's signature scheme [5].

A. The Hess's signature scheme

Here, we briefly review the ID-based signature scheme of Hess [5]. This scheme is itself one of the classic ID-based signature schemes which uses bilinear pairings. Many proxy signature and signcryption [4], [10], [21], [27] have been proposed using the scheme [5] as a building block. The signature scheme proposed by Hess in [5] is as follow:

Setup: Let G be a cyclic additive group and V be a cyclic multiplicative group. Order of both the groups is a prime number l. Let P be the generator of G and $e : G \times G \rightarrow V$ be a bilinear pairing. Define hash functions $h : \{0,1\}^* \times V \rightarrow (Z/lZ)^\times$ and $H : \{0,1\}^* \rightarrow G^*$ where $G^* = G\{0\}$. The Private Key Generator (PKG) selects randomly $t \in (Z/lZ)^\times$, computes the system's public key $Q_{TA} = tP$ and keeps t secret.

Extraction: Computing $H(ID)$ as public key for the user with identity ID, the PKG generates its corresponding private key $S_{ID} = tH(ID)$.

Sign: To sign a message $m \in \{0,1\}^*$, the user with identity ID firstly chooses $P_1 \in G^*$, picks a random $k \in (Z/lZ)^\times$ then computes:

$r = e(P_1, P)^k$
$v = h(m, r)$
$u = vS_{ID} + kP_1.$

The signature on message m is the pair $(u, v) \in (G, (Z/lZ)^\times)$.

Verify: To verify the received signature $(u, v) \in (G, (Z/lZ)^\times)$, the verifier

computes $r = e(u, P)e(H(ID), -Q_{TA})^v$

and accepts the signature if and only if the equality $v = h(m, r)$ holds.

Security: Anybody who wants to recover the private key S_{ID} will have to compute $tH(ID)$. But since the cryptographic hash function H is defined as $H : \{0,1\}^* \rightarrow G^*$ so, $H(ID) \in G^*$ can be written as xP, as P is generator of G. So, to compute $tH(ID)$, one has to compute txP, while $tP = Q_{TA}$ and $xP = H(ID)$ are given. But this is an instance to solve $CDHP$. Hence the security of Hess's scheme depends on the hardness of CDHP.

B. ID-based multi-proxy multi-signature scheme of Li and Chen

In this section, we briefly review the ID-based multi-proxy multi-signature scheme of Li and Chen [9]. For security analysis and other details one can refer [9].

System setup: For a given security parameter k, let G_1 and G_2 be two groups of prime order q, and P be the generator of G_1. Define a bilinear map $e : G_1 \times G_1 \rightarrow G_2$. The PKG selects master key $s \in_R Z_q^*$, computes public key $P_{pub} = sP$ and keeps the master key s secret. Define hash functions $H_1 : \{0,1\}^* \rightarrow Z_q^*$, $H_2\{0,1\}^* \rightarrow G_1$. System's public parameter is $param = \{G_1, G_2, k, e, q, P, P_{pub}, H_1, H_2\}$.

Extraction: For $1 \leq i \leq n$, $Q_{ID_{A_i}} = H_2(ID_{A_i})$ and $d_{ID_{A_i}} = sH_2(ID_{A_i})$ are public and private keys respectively, for the n original signers A_i, with identity $\{ID_{A_i}\}$. Similarly, for $1 \leq j \leq l$, $Q_{ID_{B_j}} = H_2(ID_{B_j})$ and $d_{ID_{B_j}} = sH_2(ID_{B_j})$ are public and private keys respectively, for the l proxy signers B_j, with identity $\{ID_{B_j}\}$.

Proxy certificate generation: In this phase, all proxy signers cooperate with all of the original signers to generate the certificate. Here m_w is the message warrant. In successfully completion of this phase, each proxy signer B_j, for $1 \leq j \leq l$ gets a proxy certificate (U, V).

• For $1 \leq i \leq n$, each A_i
chooses $x_{ai} \in_R Z_q^*$
computes $U_{ai} = x_{ai}P$,
broadcasts U_{ai} to the other $(n-1)$ original signers and l proxy signers.

• For $1 \leq j \leq l$, each B_j
chooses $x_{bj} \in_R Z_q^*$
computes $U_{bj} = x_{bj}P$,
broadcasts U_{bj} to the other $(l-1)$ proxy signers and n original signers.

• All of the signers A_i and B_j
compute $U = \sum_{i=1}^{n} U_{ai} + \sum_{j=1}^{l} U_{bj}$.

• For $1 \leq i \leq n$, each A_i
computes $V_{ai} = H_1(m_w \parallel U)d_{ID_{A_i}} + x_{ai}P_{pub}$
broadcasts V_{ai} to the chairman of original group.

• For $1 \leq j \leq l$, each B_j
computes $V_{bj} = H_1(m_w \parallel U)d_{ID_{B_j}} + x_{bj}P_{pub}$
broadcasts V_{bj} to the chairman of original group.

The chairman confirms V_{ai} and V_{bj} by checking:

$e(P, V_{ai}) = e(U_{ai}, P_{pub})e(P_{pub}, Q_{ID_{A_i}})^{H_1(m_w \| U)}$, (for $1 \leq i \leq n$) and

$e(P, V_{bj}) = e(U_{bj}, P_{pub})e(P_{pub}, Q_{ID_{B_j}})^{H_1(m_w \| U)}$, (for $1 \leq j \leq l$), respectively.

If all of the above equalities hold, the chairman computes $V = \sum_{i=1}^{n} V_{ai} + \sum_{j=1}^{l} V_{bj}$, and broadcasts V to the all original and proxy signers. Finally, members of the proxy group are authorized to act as proxy agents for the group of n original signers with certificate (U, V).

Multi-proxy multi-signature generation: If the l proxy signers want to sign a message m on behalf of the n original signers, they perform the following steps. One proxy signer in the proxy group, plays the role of clerk to combine all partial proxy signatures to generate the final multi-proxy multi-signature on message m with warrant m_w.

• For $1 \leq j \leq l$, each proxy signer B_j
chooses $t_j \in_R Z_q^*$
computes $R_j = t_jP$
broadcasts R_j to the other $(l-1)$ proxy signers.

• For $1 \leq j \leq l$, each proxy signer B_j
computes $R = \sum_{j=1}^{l} R_j$ and
$S_j = H_1(m \parallel R)d_{ID_{B_j}} + t_jV$
sends (R, S_j) to the clerk as his partial proxy signature on message m.

• For $1 \leq j \leq l$, the clerk verifies the partial proxy signatures by checking the equation:

$e(P, S_j) = e(R_j, V)e(P_{pub}, Q_{ID_{B_j}})^{H_1(m \| R)}$.

If the above equality holds for $1 \leq j \leq l$, the final multi-proxy multi-signature on message m is generated as $(m_w, (R, S), (U, V))$ by the clerk, where $S = \sum_{j=1}^{l} S_j$.

Verification: Receiving the multi-proxy multi-signature $(m_w, (R, S), (U, V))$, and the message m, the verifier proceeds as follows:

(i) Checks whether or not the message m conforms to the

warrant m_w. If not, stop. Continue, otherwise.
(ii) Checks whether or not the l proxy signers are authorized by the n original signers in the warrant m_w. If not, stop. Continue, otherwise.
(iii) Verifies the warrant and the certificate (U, V) by the equation:

$$e(P,V) = e(U,P_{pub}) \times e(P_{pub}, \sum_{i=1}^{n} Q_{ID_{A_i}} + \sum_{j=1}^{l} Q_{ID_{B_j}})^{H_1(m_w \| U)}$$

(iv) Accepts the multi-proxy multi-signature if and only if the following equality holds:

$$e(P,S) = e(R,V)e(P_{pub}, \sum_{j=1}^{l} Q_{ID_{B_j}})^{H_1(m \| R)}$$

Discussion on the scheme [9]: For the security reason of proposed scheme, the proxy certificate must be generated by the cooperation of the original group and the proxy group. So, the proxy certificate is verified by the equation

$$e(P,V) = e(U,P_{pub}) \times e(P_{pub}, \sum_{i=1}^{n} Q_{ID_{A_i}} + \sum_{j=1}^{l} Q_{ID_{B_j}})^{H_1(m_w \| U)}$$

using the public keys of all original signers and all proxy signers. Here, it is clear that, the original group is not able to arbitrarily announce any group as its proxy group without the agreement of the proxy group. Therefore, in this scheme proxy group is seems to be protected. On the other hand, since the proxy group had agreed on the proxy authorization, hence no proxy signer can deny that are the proxy agent, this leads to protection of proxy group. Moreover, since the proxy signer's private keys are required in the multi-proxy multi-signature generation phase, the multi-proxy multi-signature has to be generated by the cooperation of all members in the proxy group.

On the other hand, unforgeability of the scheme can be observed as they use a modified Hess's scheme [5], which is proven to be secure. As to generate the multi-proxy multi-signature, any third party who can even get signatures of the original signers on the warrant m_w cannot forge the multi-proxy multi-signature. Also alternatively, the group of original signers cannot generate a valid multi-proxy multi-signature since those private keys $d_{ID_{B_j}}$ of proxy signers are used in the multi-proxy multi-signature generation algorithm. Secondly, even the clerk, who has more privilege than other proxy signers in the proxy group, cannot forge a multi-proxy multi-signature. To see this, suppose that the clerk wants the proxy group to sign a false message m_0, He can change his own R_j therefor R. Then the clerk tries to compute S such that the equation $e(P,S) = e(R,V)e(P_{pub}, \sum_{j=1}^{l} Q_{ID_{B_j}})^{H_1(m_0 \| R)}$ holds. But it is equivalent to solve the bilinear pairing inversion problem (BPIP), Now since the BPIP is reducible to CDHP in G_2 and can be reduced to DLP in G_2, and CDHP and DLP are intractable in G_2, hence the clerk cannot forge a valid multi-proxy multi-signature by this way. Lastly, all other proxy signers cannot obtain more information than the clerk, hence they cannot generate a valid multi-proxy multi-signature.

V. Our ID-based multi-proxy multi-signature scheme:

In this section, we propose an efficient ID-based multi-proxy multi-signature scheme with delegation by warrant. Our scheme is divided into six phases: System setup phase, Extraction phase, Delegation generation phase, Proxy secret key generation phase, Multi-proxy multi-signature generation phase and Verification phase.

Setup: For a given security parameter k, let G_1 and G_2 be two cyclic groups of prime order q, and P be the generator of G_1. Define a bilinear map $e : G_1 \times G_1 \rightarrow G_2$. The PKG randomly selects a master key $s \in_R Z_q^*$ and computes public key $P_{pub} = sP$. Define cryptographic hash functions $H_1, H_2, H_3 : \{0,1\}^* \rightarrow G_1$ and $H_4 : \{0,1\}^* \rightarrow Z_q Z_q^*$. The PKG publishes system's public parameters $param = \{G_1, G_2, k, e, q, P, P_{pub}, H_1, H_2, H_3, H_4\}$ and keeps the master key s secret.

Extract: Let for $1 \leq i \leq n$, A_i be the n original signers with identity ID_{A_i}, and for $1 \leq j \leq l$, B_j be the l proxy signers with identity ID_{B_j}. The PKG computes public and private keys of A_i as $Q_{ID_{A_i}} = H_1(ID_{A_i})$ and $S_{ID_{A_i}} = sQ_{ID_{A_i}}$ respectively. Similarly the public and private keys of B_j as $Q_{ID_{B_j}} = H_1(ID_{B_j})$ and $S_{ID_{B_j}} = sQ_{ID_{B_j}}$ respectively.

Delegation generation: To delegate the signing capability to the l proxy signers $B_1, B_2, .., B_l$, the n original signers $A_1, A_2, .., A_n$ do the following job to make a signed warrant. The warrant w includes some information like the period of delegation, nature of message, identity information of original and proxy signers etc.

• For $1 \leq i \leq n$, Each original signer A_i
selects $x_{ai} \in_R Z_q^*$ and
computes $U_{ai} = x_{ai}P$.
broadcasts U_{ai} to the other $(n-1)$ original signers.

• For $1 \leq i \leq n$, Each original signer A_i
computes $U = \sum_{i=1}^{n} U_{ai}$ and
$V_{a_i} = S_{ID_{A_i}} + x_{ai} H_2(ID_{A_i} \| w \| U)$.

Each A_i sends their V_{a_i} to a chairman in the group of original signers.
The chairman receiving V_{a_i} from each A_i, confirms the validity of V_{ai} by checking:

$$e(P, V_{a_i}) = e(P_{pub}, Q_{ID_{A_i}}) \times e(U_{ai}, H_2(ID_{A_i} \| w \| U))$$

If all V_{a_i} are valid, the chairman combines them as $V = \sum_{i=1}^{n} V_{a_i}$ and sends (U, w, V) to the group of l proxy signers.

Receiving (U, w, V), each proxy signer confirms its validity by checking:

$$e(P,V) = e(P_{pub}, \sum_{i=1}^{n} Q_{ID_{A_i}}) \times e(U, H_2(ID_{A_i} \| w \| U))$$

Each proxy signer accepts the delegation, if it is a valid delegation, otherwise requests for a new one or terminates the protocol.

Proxy secret key generation: If each proxy signer B_j, for $1 \leq j \leq l$ accepts the delegation, they generate their proxy private key

$$S_{pj} = V + H_4(ID_{Bj} \| w \| U) S_{ID_{Bj}} \text{ (for } 1 \leq j \leq l).$$

Multi-proxy multi-signature generation: To sign a message m with warrant w, on behalf of the group of n original signers $A_1, A_2, .., A_n$, the l proxy signers $B_1, B_2, .., B_l$ perform the following steps:

• For $1 \leq j \leq l$ each proxy signer B_j
selects $x_{bj} \in_R Z_q^*$
computes $U_{bj} = x_{bj}P$
broadcasts U_{bj} to the other $(l-1)$ proxy signers.

• For $1 \leq j \leq l$, each proxy signer B_j
computes $U_p = \sum_{j=1}^{l} U_{bj}$ and
$\sigma_j = S_{Pj} + x_{bj} H_3(w \parallel m \parallel U_p)$
sends their partial proxy signature σ_j to a clerk in the proxy group.

The clerk verifies all the partial proxy signature by checking:

$$\begin{aligned} e(P, \sigma_j) = e(P_{pub}, Q_{ID_{Bj}})^{H_4(ID_{Bj} \parallel w \parallel U)} \\ \times e(U_{bj}, H_3(w \parallel m \parallel U_p)) \\ \times e(P_{pub}, \sum_{i=1}^{n} Q_{ID_{Ai}}) e(U, H_2(ID_{Ai} \parallel w \parallel U)) \end{aligned}$$

If all the partial proxy signatures are correct, clerk combines them as:
$\sigma = \sum_{j=1}^{l} \sigma_j$.

Finally, (U_p, w, U, σ) is the multi-proxy multi-signature on message m with warrant w made by the group of l proxy signers $\{B_1, B_2, .., B_l\}$ on behalf of the group of n original signers $\{A_1, A_2, .., A_n\}$.

Verification: Receiving a multi-proxy multi-signature (U_p, w, U, σ) and message m, the verifier checks the following:

(1) Checks whether or not the message m confirms to the warrant w. If not, stop. Continue otherwise.
(2) Checks whether or not the l proxy signers are authorized by the group of n original signers in the warrant w. If not, stop. Continue otherwise.
(3)Recovers the public keys $Q_{ID_{Ai}} = H_1(ID_{Ai})$ for $1 \leq i \leq n$ and $Q_{ID_{Bj}} = H_1(ID_{Bj})$ for $1 \leq j \leq l$.
(4)Accepts the multi-proxy multi-signature if and only if the following equality holds:

$$\begin{aligned} e(P, \sigma) = e(P_{pub}, l \sum_{i=1}^{n} Q_{ID_{Ai}}) \\ \times e(U, H_2(ID_{Ai} \parallel w \parallel U)) \\ \times e(P_{pub}, \sum_{j=1}^{l} Q_{ID_{Bj}})^{H_4(ID_{Bj} \parallel w \parallel U)} \\ \times e(U_p, H_3(w \parallel m \parallel U_p)). \end{aligned}$$

VI. Analysis of proposed scheme

In this section, we prove the correctness of verification and compare the efficiency of our scheme with that of [9], we show that our scheme is more efficient than [9]. We also show that the proposed scheme satisfies all the security requirements of a proxy signature given in [12].

A. Correctness

The correctness of verification is satisfied as follows:

$$\begin{aligned} e(P, \sigma) &= e(P, \sum_{j=1}^{l} \sigma_j) \\ &= e(P, \sum_{j=1}^{l} (S_{pj} + x_{bj} H_3(w \parallel m \parallel U_p))) \\ &= e(P, \sum_{j=1}^{l} [(V + H_4(ID_{Bj} \parallel w \parallel U) S_{ID_{Bj}}) + \\ &\quad x_{bj} H_3(w \parallel m \parallel U_p)]) \\ &= e(P, \sum_{j=1}^{l} (V + H_4(ID_{Bj} \parallel w \parallel U) S_{ID_{Bj}})) \\ &\quad \times e(P, \sum_{j=1}^{l} x_{bj} H_3(w \parallel m \parallel U_p)) \\ &= e(P, lV) e(P, \sum_{j=1}^{l} S_{ID_{Bj}})^{H_4(ID_{Bj} \parallel w \parallel U)} \\ &\quad \times e(P, \sum_{j=1}^{l} x_{bj} H_3(w \parallel m \parallel U_p)) \\ &= e(P, l \sum_{i=1}^{n} (S_{ID_{Ai}} + x_{ai} H_2(ID_{Ai} \parallel w \parallel U))) \\ &\quad \times e(P, \sum_{j=1}^{l} S_{ID_{Bj}})^{H_4(ID_{Bj} \parallel w \parallel U)} \\ &\quad \times e(P, \sum_{j=1}^{l} x_{bj} H_3(w \parallel m \parallel U_p)) \\ &= e(P, l \sum_{i=1}^{n} S_{ID_{Ai}}) e(P, \sum_{i=1}^{n} x_{ai} H_2(ID_{Ai} \parallel w \parallel U)) \\ &\quad \times e(P, \sum_{j=1}^{l} S_{ID_{Bj}})^{H_4(ID_{Bj} \parallel w \parallel U)} \\ &\quad \times e(P, \sum_{j=1}^{l} x_{bj} H_3(w \parallel m \parallel U_p)) \\ &= e(P_{pub}, l \sum_{i=1}^{n} Q_{ID_{Ai}}) e(U, H_2(ID_{Ai} \parallel w \parallel U)) \\ &\quad \times e(P_{pub}, \sum_{j=1}^{l} Q_{ID_{Bj}})^{H_4(ID_{Bj} \parallel w \parallel U)} \\ &\quad \times e(U_p, H_3(w \parallel m \parallel U_p). \end{aligned}$$

B. Security Analysis

In this section, we analyze the security of our scheme. We will show that our scheme satisfies all the security requirements of a proxy signature mentioned in section 2.

(i)*Strong unforgeability*: The attempt to forge the signature can be made by either of the original signers, proxy signers or by any third party who do not participate in the protocol.

Claim 1: The proposed ID-based signature scheme is strongly unforgeable if the CDHP in G_1 is hard.
For this, we see that if any proxy/original signer or third party wants to forge the private key of original/proxy signers respectively, he will have to solve an instance of CDHP. As for the ith original signer, the private key is $S_{ID_{A_i}} = sQ_{ID_{A_i}}$. Also $Q_{ID_{A_i}} \in G_1$ can be written as $k_i P \in G_1$, for any $k_i \in Z_q^*$ and being P a generator of G_1. So, to compute the private key $S_{ID_{A_i}}$, one has to compute $sk_i P \in G_1$ for given $sP = P_{pub} \in G_1$ and $k_i P = Q_{ID_{A_i}} \in G_1$. But this is an instance to solve the CDHP, which is assumed to be hard in our system. The same instance happens in case of proxy signer. Hence, none of the above three parties can forge the private key of any original or proxy signer.

Claim 2: The proposed ID-based multi-proxy multi-signature scheme is strongly unforgeable if the CDHP and DLP in G_2 are intractable.

Firstly, anyone from the group of original signers can not generate a valid multi-proxy multi-signature, because it involves the private keys of all the proxy signers.

Secondly, the group of proxy signers can not forge the multi-proxy multi-signature. To see this, suppose firstly the clerk in proxy group wants to sign a false message m'. He can maximum change his U_{bj} therefore U_p and hence σ_j. Then finally he will try to compute σ such that, the equality

$$\begin{aligned} e(P, \sigma) = e(P_{pub}, l \sum_{i=1}^{n} Q_{ID_{Ai}}) \\ \times e(U, H_2(ID_{Ai} \parallel w \parallel U)) \\ \times e(P_{pub}, \sum_{j=1}^{l} Q_{ID_{Bj}})^{H_4(ID_{Bj} \parallel w \parallel U)} \\ \times e(U_p, H_3(w \parallel m' \parallel U_p)). \end{aligned}$$

holds. But it is equivalent to solve the bilinear pairing inversion problem (BPIP). And since the bilinear pairing inversion problem is reducible to computational Diffie-Hellman problem (CDHP) in G_2 and can be condensed in discrete logarithm problem (DLP) in G_2, and CDHP and DLP are intractable in G_2, the clerk cannot forge a valid multi-proxy signature. But since in our scheme, the clerk in the proxy group is supposed to be most privileged than other proxy signers, so no proxy signer in that group can forge a valid multi-proxy multi-signature by this way. Also, if the group of proxy signers wants to sign a false message m'. For this, they will require the value V to generate their proxy secret key, but construction of V involves the private keys of all the original signers. Selection of an invalid V, will make the verification failure.

Finally, any third party who do not participate in the protocol can not forge the signature even having signatures of all the original signers, since for that, he will be needed the private keys of all the original signers.

Getting the private keys of proxy and original signers in above phases is equivalent to forging the Hess's signature scheme [5], which is proved to be secure. Hence the proposed multi-proxy multi-signature is strongly unforgeable.

(ii)*Verifiability:* Any verifier can verify the multi-proxy multi-signature and can check whether the signed message confirms to the delegation warrant or not. Correctness of the verification is described above.

(iii)*Strong identifiability:* By warrant anyone can determine the identity of proxy signers.

(iv)*Strong undeniability:* For $1 \leq j \leq l$, partial proxy signature σ_j of each B_j is constructed using his proxy secret key S_{pj}, which involves the value V provided by the group of original signers and the private key $S_{ID_{B_j}}$ provided by the PKG. Also the attached warrant includes the identity of proxy signers, and the signature will be verifiable only for legally used private keys of proxy signers provided by the PKG against the public keys $Q_{ID_{B_j}}$. Also, The clerk validates the proxy signer's partial signatures on message m, by checking whether or not the equation

$$\begin{aligned} e(P,\sigma_j) = e(P_{pub}, Q_{ID_{Bj}})^{H_4(ID_{Bj} \| w \| U)} \\ \times e(U_{bj}, H_3(w \parallel m \parallel U_p)) \\ \times e(P_{pub}, \sum_{i=1}^{n} Q_{ID_{Ai}}) \\ \times e(U, H_2(ID_{Ai} \parallel w \parallel U)) \end{aligned}$$

holds, so no proxy signer can deny his signature of earlier session.

(v)*Prevention of misuse:* Since the message warrant is attached specifying the delegation period, nature of message, identities of original signers etc., the group of proxy signers can not sign any message which does not confirms to the warrant and has not been authorized by the group of original signers.

C. Efficiency comparison

Here we compare the efficiency of our scheme with that of multi-proxy multi-signature scheme given by Li and Chen [9]. Firstly in table below, we compare the computational costs due to the certificate generation of Li and Chen scheme and due to delegation generation of our scheme

Certificate generation/Delegation generation

Scheme	Pairing	Exponentiation	Hashing
Li and Chen scheme	6	2	2
Our scheme	6	0	2

Also in verification phase, **1 pairing** and **1 exponent** reduces in our scheme with compare to Li and Chen's scheme. Hence due to delegation by warrant, our scheme is economical and computationally more efficient than Li and chen's ID-based multi-proxy multi-signature scheme [9].

D. Application and implementation

There are many real world scenarios where delegation of right is quit common like distributed database, grid computing, organizational knowledge, global distributed networks etc [11], [13], [17], [24]. The proposed ID-based multi-proxy multi-signature scheme can be apply to transfer the execution right in above distributed systems. Moreover, to examine the size of keys, total running time, signature size etc, signature algorithm of proposed scheme can be implemented through programming on many open source tools like SAGE, PBC Library (http://crypto.stanford.edu/pbc/) etc.

VII. Conclusion

In this paper, we have proposed an ID-based multi-proxy multi-signature scheme, replacing the certificate generation phase of the ID-based multi-proxy multi-signature scheme of Li and Chen [9], by delegation generation. Due to this replacement, computational cost reduces in our scheme with comparison to [9]. We have also analyzed the security properties of our scheme and showed that the proposed scheme is strongly unforgeable under the CDHP and DLP assumptions. Moreover, proposed scheme fulfils all the security requirements of a proxy signature scheme [12].

ACKNOWLEDGEMENT The work carried out in this paper is supported by Cryptology Research Society of India (CRSI) and Department of Science and Technology (DST), India under the Young Scientist ITS grant (DST/ITS/03366).

References

[1] D. Boneh and M. Frankline, Identity-based encryption from Weil pairings, *Crypto* 2001 Springer-Verlag LNCS 2011, 2139 pp. 213-229

[2] J. C. Cha and J. H.Cheon, An identity based signature from gap Diffie-Hellman groups, *PKC 2003,* Springer-Verlag, LNCS 2567, pp. 18-30.

[3] Y. Desmdet and J. Quisquater, Public-key Systems based on the diffiulty of tampering, *Crypto 86*, LNCS, Vol 263, Springer-Verlag 1987, pp. 111-117.

[4] C. X. Gu, H. pan and Y. F. Zhu, A new ID-based proxy multi-signature scheme from bilinear pairings, In *Wuhan University Journal of natural Sciences* Vol 11, No. 1, 2006, pp 193-197.

[5] F. Hesss, Efficient identity based signature scheme based on pairings, *SAC2002,* Springer-Verlag, LNCS 2595, pp. 310-324.

[6] S. Hwang and C. Chen, New multi-proxy multi-signature schemes, *Appl. Math. Comput.* 147, 2004, pp. 5767.

[7] S. Hwang and C. Shi, A simple multi-proxy signature scheme, *Proceedings of the 10th national conference on information security, Hualien, Taiwan,* ROC; 2000, pp. 134138.

[8] S. Kim, S. Park, and D. Won, Proxy signatures, revisited, In Proc. of *ICICS97, International Conference on Information and Communications Security*, Springer, LNCS 1334, 1997, pp. 223-232.

[9] X. Li and K. Chen, ID-based multi-proxy signature, proxy multi-signature and multi-proxy multi-signature schemes from bilinear pairings, *Applied Mathematics and Computation*, 169, 2005, pp. 437-450.

[10] X. Li, K. Chen and S. Li, Multi-proxy signature and proxy multi-signature schemes from bilinear pairings, *PDCAT 2004*, LNCS 3320, Springer 2004, pp. 591-595.

[11] M. Liu and K. Gao, High efficient scheduler for distributed data mining application, *3rd WSEAS Intl. Conf. on Comp. Engg. Appl. CEA'09*, Ningbo, China, 2009, pp. 87-92.

[12] B. Lee, H. Kim and K. Kim, Strong proxy signature and its applications, *Proceedings of SCIS,* 2001, pp. 603-608.

[13] M. I. Muntean and D. A. Tarnaveanu, Information technology and organizational knowledge management, *Proc. 13th WSEAS Intl. Conf. on Computers*, Greece, 2009, pp 335-339.

[14] M. Mambo, K. Usuda and E. Okmamoto, Proxy signatures: delegation of the power to sign message, *IEICE Transaction Functional* E79-A (9), 1996, pp. 1338 - 1354.

[15] U. Maurer and Y. Yacobi, Non-interective public-key cryptography, *Proc. of Eurocrypto 91*, Lecture Nores in Computer Sciences, Vol. 547, Springer-Verlag, 1992 pp. 498-507.

[16] K. G. Paterson, ID-based signatures from pairings on elliptic curves, *IEEE Electronic Letters*, 2002, Vol. 38, No. 18, pp. 1025-1026.

[17] S. Pukdesree, V. Lacharoj and P. Sirisang, An empirical study of distributed database on PC cluster computers, *Proc. 10th WSEAS Intl. Conf. on Appl. Comp. Sc. ACS'10*, Japan, 2010, pp. 111-115.

[18] A. Shamir, Identity based cryptosystem and signature scheme , *Proc. Crypto'84,* Springer-Verlag, LNCS Vol. 196, , 1984, pp. 47-53.

[19] N. P. Smart, An identity-based authenticated key agreement protocol based on the Weil pairings, *Electronics Letters* 38(13), 2002, pp. 630-632.

[20] H. Tanaka, A realization scheme for the identity-based cryptosystem, *Proc. of Crypto 87*, Lecture Nores in Computer Sciences, Vol. 293, Springer-Verlag, 1987, pp. 341-349.

[21] Q. Wang and Z. cao, Two proxy signcryption schemes from bilinear pairings: In *CANS 2005* LNCS 3810, Springer 2005, pp. 161-171.
[22] S. Tsuji and T. Itoh, An ID-based cryptosystem based on the discrete logarithm problem, *IEEE Journal of Selected Areas in Communications*, Vol. 7, No. 4, 1989 pp. 467-473, .
[23] W. Wu, Y. Mu, W. Susilo, J. Seberry and X. Huang, Identity-based proxy signature from pairing: In *ATC 2007*, LNCS 4610, Springer 2007, pp. 22-31.
[24] T. Wang, A. Yang and Y. ren, Application of priority in grid calling, *3rd WSEAS Intl. Conf. on Comp. Engg. Appl. CEA'09*, Ningbo, China, 2009, pp. 158-163.
[25] X. Yi, An Identity-based signature scheme from the Weil pairing, In: *IEEE Communication Letters*, Vol. 7, No. 2, 2003, pp. 76-78.
[26] L. Yi, G. Bai and G. Xiao, Proxy multi-signature scheme: a new type of proxy signature scheme, *Electronics Letters* 36 (6),2000, pp. 527528.
[27] F. Zhang and K. Kim: Efficient ID-based blind signature and proxy signature from bilinear pairings. In: *R. Safavi-Naini and J. Seberry (eds.): Australasian Conference on Information Security and Privacy*. LNCS, Vol. 2727. Springer 2003, pp. 312323.