An efficient Multi-proxy system for Proxy signature scheme

Shivendu Mishra[1], Rajeev Anand Sahu[2], Sahadeo Padhye[2]
[1]Department of Computer Science and Engineering,
[2]Department of Mathematics
Motilal Nehru National Institute of Technology
Allahabad-211004, India.
{2009is17, rajeevs.crypto, sahadeomathrsu}@gmail.com

ABSTRACT: *In the last few years, the traditonal certificate-based setting is replaced by the ID-based setting. Proxy signatures allow the delegation of signing rights from an original user to its proxy agent. Normally, the original signer can authorize a group of proxy agents to sign any document on its behalf. Currently in our study, we have proposed an ID-based multi-proxy signature scheme, from bilinear pairings based on 'k-plus problem'. We document that the proposed scheme is secure under the inverse computational Die-Hellman (INV-CDH) assumption. Besides, we have proven that the new scheme is computationally more efficient and takes less running time than other existing schemes [5, 10]. Our proposed method meets all the security requirements of a proxy signature scheme proposed by Lee.*

Keywords: Multi-proxy signature, ID-based signature, Bilinear pairings, *k-plus* problem, Computational Die-Hellman problem

Received: 12 November 2010, Revised 27 December 2010, Accepted 4 January 2011

1. Introduction

In traditional public-key cryptography, the problem was to maintain certificates of users, storage space and large overhead to transfer certificates in users group which leads to increase the associated cost significantly. As an economical alternative of traditional certificate-based setting, Shamir [18] introduced the notion of ID-based cryptography in 1984, which removed the need of certificates for public keys and thus reduced the associated cost. In ID-based cryptography, the users public and private keys are generated directly from their identities such as email address, IP-address, phone number etc. The bilinear pairing has property of linearity in both co-ordinates which makes it computationally simple and functionally strong. Hence the bilinear pairings are found very useful for the ease of computation in various cryptosystems. In 2001, Boneh and Franklin [1], proposed a practical ID-based encryption scheme which took advantage of the properties of bilinear parings over supersingular elliptic curves. The work of Boneh and Franklin encouraged many authors to design efficient key agreement protocols, signcryption and signature schemes using bilinear pairings [2, 4, 6, 15, 19, 22].

The paradigm of proxy signature is a technique for a user to delegate signing rights to its proxy agent, so that the proxy agent can sign any document on behalfof the user within a given criteria (the criteria includes delegation warrant issues). Proxy signature is very much applicable in scenarios when the original signer is absent at the time to sign any document. Many applications of proxy signature are discussed in the literature, some of them are in distributed shared objects, grid computing, global distribution networks, mobile agent applications, mobile communications etc. The concept of proxy signature was introduced by Mambo, Usuda and Okamoto [13] in 1996. Later in 1997, Kim et. al. [9] extended the notion by using Schnorr signature and including warrant information in partial delegation schemes. In 2001, Lee et. al. [12] proposed some extensions on security requirements of a proxy signature scheme presented by Mambo et. al. [13]. The proxy signature primitive introduces other additions also, such as multi-proxy signature, proxy multi-signature, multi-proxy multi-signature, threshold proxy signature etc. The idea of multi-proxy signature was introduced by Hwang and Chen. [7] in 2000. In a multi-proxy signature scheme, the original signer delegates its signing rights to a group of its proxy agents and the nal signature is made by the group of proxy agets on behalf of the original signer. The classic scheme of multi-proxy signature presented in [7] leads to many multi-proxy signature schemes [5, 10, 11].

1.1 Our contribution

In 2005, Takeshi et. al. [14] suggested the '*k-plus*' and 'extended *k-plus*' problems using bilinear pairings. In [14], they proposed a short signature scheme based on *k-plus* problem and a proxy signature scheme based on extended *k-plus* problem. Security of their schemes depends on *k-plus* problem under the INV-CDHP assumption. In this paper, we have proposed an ID-based multi-proxy signature scheme, based on k-plus problem using the idea of Takeshi et. al. [14]. The building blocks for proposed multi-proxy signature scheme is ID-based signature scheme based on *k-plus* problem [21]. Our scheme is computationally more efficient than other existing schemes [5, 10] and satisfies all the security requirements of a safe proxy signature scheme [12].

1.2 Organization

The rest of this paper is organized as follows. In Section 2, we describe some related mathematical preliminaries and security requirements. The ID-based signature scheme based on *k-plus* problem is brіey reviewed in Section 3. Our proposed scheme is depict in Section 4. Section 5 investigates the security and efficiency analysis of our scheme and nally Section 6 gives some conclusions of this paper.

2. Preliminaries

In this section, we briefly describe some related mathematical problems and security requirements of a proxy signature scheme.

2.1 Bilinear pairing

Let G_1 and G_2 be two groups of prime order q. Then a map e : $G_1 x\, G_1 \rightarrow G_2$ satisfying the following properties, is called bilinear pairing:

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

(b) Non-Degeneracy: *There exists* $P,Q \in G_1$ *such that* $e(P,Q) = 1$.

(c) Computability: *There must exist an efficient algorithm to compute* $e(P,Q) \in G_2$ *for any* $P,Q \in G_1$.

2.2 Discrete logarithm problem (DLP)

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

2.3 Computational Diffie-Hellman problem(CDHP)

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

2.4 Inverse computational Diffie-Hellman problem (INV-CDHP)

Given $P, aP \in G_1$, to compute $a^{-1}P \in G_1$, where $a \in Z_q^*$.

2.5 Bilinear pairing inversion problem (BPIP)

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

2.6 The *k-plus* problem

For given P, $Pub = sP \in G_1$, $V = g^e \in G_2$, $e_1, e_2 \ldots e_k \in Z_q^*$ $\{\frac{e+e_1}{s}P, \frac{e+e_2}{s}P, \ldots. \frac{e+e_k}{s}P\} \in G_1$, To nd a pair $\{e', \frac{e+e'}{s}P\}$, where $e', e, s \in Z_q^*$, $e' \notin \{e_1, e_2 \ldots e_k\}$ and k is a constant number.

2.7 Security requirements of a proxy signature

A safe and sound proxy signature should satisfy the following security requirements [12]:

Strong unforgeability: Only the legal proxy signer can generate a valid proxy signature on behalf of original signer. Even the original signer cannot make proxy signature.

Verifiability: Signature can be verified by anyone, and delegation warrant should be confirmed by the signed message.

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

Strong undenifiability: The proxy signer cannot deny his signature, which he generates ever.

Prevention of misuse The proxy signer should be unable to sign any unauthorized message. 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.

3. ID-based signature scheme based on *k-plus* problem

To construct an ID-based multi-proxy signature scheme, we firstly review an ID-based signature scheme from bilinear pairings based on k-plus problem [21] which uses the short signature scheme proposed by Takeshi *et. al*. [14]. The ID- based signature scheme based on *k-plus* problem [21] can be regarded as building blocks for our ID-based multi-proxy signature scheme. The scheme [21] is as follows:

Setup phase: For a given security parameter K, the PKG generates system's public parameter param = $(K, G_1, G_2, q, e, H, P, g, Pub)$ and system's master secrets. Where G_1 is an additive cyclic group of prime order q, and G_2 is a multiplicative cyclic group of the same prime order q. Generators of the groups G_1 and G_2 are P and $g = e(P, P)$ respectively. Bilinear pairing $e: G_1 \times G_1 \rightarrow G_2$ and hash function $H: \{0, 1\}^* \rightarrow Z_q^*$ are defined. $Pub = sP \in G_1$ is system's public key and $s \in Z_q^*$ is system's master secret. PKG publishes param and keeps the system's master secret s unrevealed.

Extract phase: Given an identity ID of a user, the PKG computes public key and private key of the user as follows:

public key: $Q_{ID} = H(ID)$ and private key: $S_{ID} = \frac{Q_{ID}}{s}P$.

PKG sends this S_{ID} to the user having identity ID, as his private key by a secure channel.

Sign phase: Signer first selects a random integer $r \in Z_q^*$ and computes $V_s = g_r$, broadcasts Vs as public parameter, keeping r secret.

Then signer computes $h = H(m)$ and $\mathbf{S} = (\mathbf{r} + \mathbf{h})\mathbf{S}_{\mathbf{ID}}$.

Signature on the message m is $(\mathbf{S}, \mathbf{V}_s)$.

Verification phase: Having the system's public parameter Param and signature (S, Vs) on message m, the verifier first computes $h = H(m)$ and accepts the signature on message m iff the following holds:

$e(\mathbf{Pub}, \mathbf{S}) = (\mathbf{V}_s . g^h)^{Q_{ID}}$

3.1 Security analysis

In this section, we analyze the security of above scheme. It is proved as follows that the proposed scheme is secure against existential forgery on adaptive chosen message and *ID* attack [6].

Theorem: The proposed signature scheme is secure against existential forgery on adaptive chosen message and *ID* attack if INV-CDHP in G_1 is hard.

Proof: According to [6], if there exists a polynomial time algorithm A_1 for adaptive chosen message and *ID* attack to the proposed scheme then there exists an algorithm A_2 with the same advantage. For the given identity *ID* and public key Q_{ID}, the Forking lemma [16] says, if there exists an efficient algorithm B_1 for adaptive chosen message and *ID* attack for the proposed scheme then there is an algorithm B_2 by which one can derive two valid signatures (m, h, S_1, V_s) and (m, h', S_2, V_s) provided that $h \neq h'$. Now according to [6], an algorithm B_3, based on B_2 can be produced for given public values Pub and Q_{ID} which gives two forgeries (m, h, S_1, V_s) and (m, h', S_2, V_s) provided that $h \neq h'$ as $e(\mathbf{Pub}, \mathbf{S_1})^{Q_{ID}} = (\mathbf{V}_s . g^h)$ and $e(\mathbf{Pub}, \mathbf{S_2})^{Q_{ID}} = (\mathbf{V}_s . g^{h'})$.
Taking the first equality:

$$e(Pub, S_1) = \left(V_s . g^h\right)^{Q_{ID}}$$
$$= \left(g^r . g^h\right)^{Q_{ID}}$$

$$= \left(g^{r+h}\right)^{Q_{ID}}$$
$$= g^{(r+h)Q_{ID}}$$
$$= e(P,P)^{(r+h)Q_{ID}}$$

i.e. $e(Pub, S_1) = e(P, (r+h)Q_{ID}P)$

$$e(sP, S_1) = e(P, (r+h)Q_{ID}P)$$
$$e(P, sS_1) = e(P, (r+h)Q_{ID}P) \text{ or}$$
$$e(P, sS_1 - (r+h)Q_{ID}P) = 1 \qquad (1)$$

Similarly one can get,

$$e(P, sS_2 - (r+h')Q_{ID}P) = 1 \qquad (2)$$

From (1) and (2) the following can be derived:

$e(P, s(S_1 - S_2) - (h - h')Q_{ID}P) = 1$ or $s(S_1 - S_2) - (h - h')Q_{ID}P = O$. Where O is point at infinity i.e. identity element of defined elliptic-curve. From above, one can have $s(S_1 - S_2) = (h - h')Q_{ID}P$ (by the property of bilinear pairing). The above gives $\frac{Q_{ID}}{s}P = \frac{S_1 - S_2}{h-h'}$ or $S_{ID} = \frac{S_1 - S_2}{h-h'}$ That means, algorithm B_3 solves SID, an instance of INV-CDHP in G_1. But INV-CDHP in G_1 is assumed to be hard hence the proposed scheme is secure against existential forgery on adaptive chosen message and ID attack.

4. Proposed Scheme

In this section, we describe our proposed ID-based multi-proxy signature scheme. In our scheme, the delegation security depends on the '*k-plus* problem' and security of the partial signature generation depends on the combination of '*k- plus* problem' and INV-CDHP. Our scheme is designed into five phases: System setup, Extraction, Proxy key generation, Multi-proxy signature and Verification.

4.1 System Setup

PKG generates the system's pram = $(K, G_1, G_2, q, e, H, H_1, P, g, Pub)$, where K is given security parameter, G_1 is an additive cyclic group of prime order q, and G_2 is a multiplicative cyclic group of the same prime order q. Bilinear pairing $e: G_1 \times G_1 \rightarrow G_2$ is defined as above. $H: \{0,1\}^* \rightarrow Z_q^*$ and $H_1: \{0,1\}^* \times G_2 \rightarrow Z_q^*$ are two cryptographic hash functions for the security purpose. Let P is a generator of G_1 and $g = e(P, P)$ is generator of G_2. System's public key is $Pub = sP \in G_1$, and $sP \in Z_q^*$ is system's master key. PKG publishes the *param* and keeps the master-key s secret.

4.2 Extraction

For given identity *ID*, the *PKG* computes public key and private key as follows

Public key: $Q_{ID} = H(ID)$

Private Key: $S_{ID} = \frac{Q_{ID}}{s}P$, where $P \in G_1$ is generator of G_1.

Thus the original signer (say A), has his public key Q_{ID_A}, and consequent private key S_{ID_A}. Similarily, for the l proxy signers, the public key is $Q_{ID_{P_i}}$ and consequent private key is $S_{ID_{P_i}}$ (for $1 \leq i \leq l$).

4.3 Proxy key generation

Through the signing warrant w, the original signer A delegates the signing capability to the l proxy signers in proxy group.The warrant w includes the delegation time, identity of original and proxy signers etc. Following is the process of delegation of warrant and proxy key generation.

Warrant Delegation: The original signer A randomly choses $r_A \in Z_q^*$ and computes

$V_A = g^{r_A}$,
$h = H(w)$ and
$S_A = (r_A + h)\, S_{ID_A}$,

then sends (S_A, V_A, w) to each proxy signer as a delegation value.
Each proxy signer P_i for $1 \le i \le l$, accepts the delegation value S_A on warrant w, if the equality $e(Pub, S_A) = (V_A . g^h)^{Q_{ID_A}}$ holds. Finally, each proxy signer generates their proxy key as $d_{P_i} = S_A + S_{ID\, P_i}$, (for $1 \le i \le l$).

4.4 Multi-proxy signature

Each proxy signer in proxy group, generates his partial proxy signature on message *m* that verifies the warrant *w*. One proxy signer in the proxy group is assigned as a clerk, whose task is to combine all the partial proxy signatures to generate the final multi-proxy signature. For that, each proxy signer P_i for $1 \le i \le l$
chooses randomly $r_i \in Z_q^*$ and
computes $V_i = g^{r_i Q_{ID_{P\,i}}}$
then broadcasts their Vi to the other (l-1) proxy signers.
Each P_i then computes
$V_P = \Pi_{i=1}^{l} V_i$
$h' = H_1(m, V_P)$, and
$S_{P_i} = h' d_{P_i} + r_i S_{ID_{P_i}}$
where *m* is the intended message. The partial proxy signature on message *m* is (S_{P_i}, V_P). Each proxy signer P_i sends their partial proxy signatures to the clerk in proxy group.

Receiving the partial proxy signatures (S_{P_i}, V_P), for $1 \le i \le l$, the clerk verifies them checking whether the equality $e\left(Pub, S_{P_i}\right) = V_A^{h' Q_{ID_A}} . g^{h'[h Q_{ID_A} + Q_{ID_{P_i}}]} . V_i$ holds or not.
Once if all the partial proxy signatures are verified correct by the clerk, he finally generates the multi-proxy signature on message m as (**S_P, V_P, V_A, w**). Where $S_P = \sum_{i=1}^{l} S_{P_i}$

4.5 Verification

Getting a multi-proxy signature (S_P, V_P, V_A, w) and message m, the verifier proceeds as follows

(1) Checks whether or not the message *m* validates to the warrant *w*. If not, stop, continue otherwise

(2) Checks the authorization of *l* proxy signers by original signer in the warrant *w*. Stop the verification, if all or any one is not authorized by the warrant. Continue otherwise.

(3) Agree to the multi-proxy signature on message *m*, if and only if the following equality holds

$$e\left(\mathrm{Pub}, S_P\right) = V_A^{l h' Q_{ID_A}} . g^{h'[l h Q_{ID_A} + \sum_{i=1}^{l} Q_{ID_{P_i}}]} . V_i$$

Where, $Q_{ID_A} = H(ID_A)$, $Q_{ID_{P_i}} = H(ID_{P_i})$, $h' = H_1(m, V_P)$ and $h = H(w)$.

5. Analysis of proposed scheme

In this section, we prove the correctness of verification and compare the efficiency of our scheme with those of [5, 10]. We show that our scheme is computationally more efficient than [5, 10]. We also prove that the proposed scheme satifises all the security requirements of a proxy signature scheme given in [12].

5.1 Correctness

The property of correctness is satised as follows-

$$
\begin{aligned}
\mathrm{e}(\mathrm{Pub},\mathrm{S_P}) &= \mathrm{e}\left(\mathrm{Pub},\textstyle\sum_{i=1}^{l}(\mathrm{S_{Pi}})\right)\\
&= e\left(Pub,\textstyle\sum_{i=1}^{l}\left[h'd_{P_i}+r_iS_{ID_{P_i}}\right]\right)\\
&= e\left(Pub,\textstyle\sum_{i=1}^{l}\left[h'\left(S_A+S_{ID_{P_i}}\right)+r_iS_{ID_{P_i}}\right]\right)\\
&= e\left(Pub,\textstyle\sum_{i=1}^{l}\left[h'S_A+h'S_{ID_{P_i}}+r_iS_{ID_{P_i}}\right]\right)\\
&= e\left(Pub,\textstyle\sum_{i=1}^{l}\left[h'S_A+(h'+r_i)S_{ID_{P_i}}\right]\right)\\
&= e\left(Pub,\textstyle\sum_{i=1}^{l}h'S_A\right)e\left(Pub,\textstyle\sum_{i=1}^{l}(h'+r_i)S_{ID_{P_i}}\right)\\
&= e(Pub,S_A)^{\sum_{i=1}^{l}h'}\,e\left(Pub,\textstyle\sum_{i=1}^{l}(h'+r_i)\frac{Q_{ID_{P_i}}}{s}P\right)\\
&= e(Pub,S_A)^{lh'}\,e\left(Pub,\frac{P}{s}\textstyle\sum_{i=1}^{l}(h'+r_i)Q_{ID_{P_i}}\right)\\
&= e(sP,S_A)^{lh'}\,e\left(sP,\frac{P}{s}\textstyle\sum_{i=1}^{l}(h'+r_i)Q_{ID_{P_i}}\right)\\
&= e\left(sP,(r_A+h)S_{ID_A}\right)^{lh'}\,e\left(P,P\textstyle\sum_{i=1}^{l}(h'+r_i)Q_{ID_{P_i}}\right)\\
&= e\left(sP,(r_A+h)\frac{Q_{ID_A}}{s}P\right)^{lh'}\,e\left(P,P\textstyle\sum_{i=1}^{l}(h'+r_i)Q_{ID_{P_i}}\right)\\
&= e\left(sP,\frac{P}{s}(r_A+h)Q_{ID_A}\right)^{lh'}\,e\left(P,P\textstyle\sum_{i=1}^{l}(h'+r_i)Q_{ID_{P_i}}\right)\\
&= e(P,P)^{lh'(r_A+h)Q_{ID_A}}\;e(P,P)^{\sum_{i=1}^{l}(h'+r_i)Q_{ID_{P_i}}}\\
&= \left\{g^{r_A}g^{h}\right\}^{lh'Q_{ID_A}}\;g^{h'\sum_{i=1}^{l}Q_{ID_{P_i}}}\,.g^{\sum_{i=1}^{l}riQ_{ID_{P_i}}}\\
&= \left\{V_Ag^{h}\right\}^{lh'Q_{ID_A}}\;g^{h'\sum_{i=1}^{l}Q_{ID_{P_i}}}\,.\Pi_{i=1}^{l}g^{riQ_{ID_{P_i}}}\\
&= V_A^{\,lh'Q_{ID_A}}\,.g^{h'lh'Q_{ID_A}}\,g^{h'\sum_{i=1}^{l}Q_{ID_{P_i}}}\,.V_P\\
&= V_A^{\,lh'Q_{ID_A}}\,.g^{h'\left[lh'Q_{ID_A}+\sum_{i=1}^{l}Q_{ID_{P_i}}\right]}.V_P
\end{aligned}
$$

5.2 Security analysis

In this section, we examine the security properties of our scheme. We will show that all the security requirements of a safe proxy signature scheme, mentioned in section 2 [12] are satisfied by our scheme.

(i) Strong unforgeability: **Theorem:** The proposed ID-based multi- proxy signature is unforgeable under the DLP and INV-CDHP assumptions, if the '*k-plus* problem' is hard in G_1.

Proof: The attempt to forge the multi-proxy signature, can be made by either of the three parties, (1) The original signer (2) Proxy signers, and (3) Any third party who never take part in the entire protocol.

1.The original signer: The original signer can not generate a valid multi-proxy signature, because to do this, he will need to get the private keys $S_{ID_{P_i}}$ of each proxy signer. But as $S_{ID_{P_i}} = \frac{Q_{ID_{P_i}}}{s}$, the attacker will have to solve the INV-CDHP in G_1, which is assumed to be hard.

In other way if the original signer wants to generate a valid partial proxy signture S_{P_i}, he will have to compute $\frac{r_i+h'}{s} Q_{ID_{P_i}} P$ as

$$S_{P_i} = h'd_{P_i} + r_i S_{ID_{P_i}}$$

$$S_{P_i} = h'\left(S_A + S_{ID_{P_i}}\right) + r_i S_{ID_{P_i}}$$

$$S_{P_i} = h'S_A + \left(r_i + h'\right) S_{ID_{P_i}}$$

$S_{P_i} = h'S_A + \left[\frac{r_i+h'}{s}\right] Q_{ID_{P_i}} P$. But computing $\left[\frac{r_i+h'}{s}\right]$ is equivalent to soloving k-plus problem, which is assumed to be hard.

Hence the original signer is unable to get any valid multi-proxy signature.

2.Proxy signers: Suppose, the clerk in proxy group wants to sign any unauthorized message, he can maximum change his V_i, that leads to change in V_P and finaly change in h'. Then he will try to compute $S_P \in G_1$, such that the equality $e\left(Pub, S_P\right) = V_A^{lh'Q_{ID_A}} . g^{h'\left[lh'Q_{ID_A} + \sum_{i=1}^{l} Q_{ID_{P_i}}\right]} . V_P$ holds. But this is equivalent to solving the BPIP, which is is reducible to CDHP in G2 and can be condensed to DLP in G_2. Now since DLP is intractable in G_2 according to assumptions, hence the clerk cannot generate a valid multi-proxy signature on any unauthorized message. In other way, if the clerk tries to get the partial proxy signatures on the false message, he will need to break the combination of '*k-plus* problem' and INV-CDHP to find $d_{P_i} = S_A + S_{ID_{P_i}}$, because S_A is based on *k-plus* problem and $S_{ID_{P_i}}$ is based on INV-CDHP, which are hard to solve. So, the clerk in proxy group can not forge the proposed multi-proxy signature. Moreover, since all other proxy signers are less priviledged than the clerk in our scheme, hence no proxy signer can forge the signature.

3.Third party: Any third party can not forge the proposed multi- proxy signature, even having signature of the original signer. Because to forge the signature, he will be required the private key of original signer, which is impossible to get due to the hardness of '*k-plus* problem'.

Hence, it is proved that the proposed scheme is strongly unforgeable.

(ii) Verifiability: The correctness of the verification is discussed above so any verifier can validate the signature and can check whether the signed message authenticate to the delegation warrant or not.

(iii) Identifiability: Through the attached warrant, any one can determine the identity of proxy signers and original signer. (iv) Strong undeniability: No proxy signer in proxy group can refuse their signature, they made in earlier session because the clerk validates all the partial proxy signatures by checking $e\left(Pub, S_{P_i}\right) = V_A^{h'Q_{ID_A}} . g^{h'\left[h'Q_{ID_A} + Q_{ID_{P_i}}\right]} . V_i$

(v) Prevention of misuse: Due to the warrant, the proxy signers cannot sign any message which does not validates to the warrant and has not been authorized by the original signer.

5.3 Efficiency comparison

Here, we compare the efficiency of our scheme with those of other ID-based multi-proxy signature scheme given in [5, 10].

Proxy key generation:

Scheme	Pairing	Hashing	Exponentiation
Li and Chen's scheme (2005) [10]	3	2	1
Cao and Cao's scheme (2009) [5]	3	3	0
Our scheme	**1**	**2**	**3**

Multi-proxy signature generation:

Scheme	Pairing	Hashing	Exponentiation
Li and Chen's scheme (2005) [10]	3	1	1
Cao and Cao's scheme (2009) [5]	5	1	1
Our scheme	**1**	**1**	**3**

Verification:

Scheme	Pairing	Hashing	Exponentiation
Li and Chen's scheme (2005) [10]	3	2	1
Cao and Cao's scheme (2009) [5]	3	3	0
Our scheme	**1**	**2**	**3**

From the above comparisons, it is clear that our scheme is computationally more ecient than other existing schemes [5,10].

5.4 Advantage and application

Previously some ID-based multi-proxy signature schemes have been proposed [5, 10] whose security depends on CDHP. Here, our scheme generates a multi-proxy signature employing the *k-plus* problem which is supposed to be more strong than CDHP, as the hardness of *k-plus* problem depends on computation of two unknown integers whereas hardness of CDHP depends on computation of a single unknown integer. Hence, our scheme is supposed to be more strong than others, whose security is based on CDHP. The proposed signature scheme is also applicable in many real word scenarios as in grid computing, mobile agent environments, distributed system etc. In distributed system, where the delegation of right is common in practice, this scheme can be used to delegate the right of execution to the person sitting in a connected computer in a network. Also in commercial transitions, this scheme can be employed in grid computing by any agent who wish to transfer his rights to some other person. This scheme also enjoys application in global distributed networks and distributed shared object system. To implement the proposed scheme, one can employ the proposed signature algorithm in various open source tools like Sage [17], PBC library (http://crypto.stanford.edu/pbc/) etc.

6. Implementation

The concept of identity-based cryptography (IBC) eliminates much of the over-heads associated with key management in conventional public-key infrastructure.Therefore, it became a very fashionable area of research for the last couple of decades. The implementations of IBC is currently a big task. There are currently only a few software libraries and toolkits which support implementations of IBC schemes. Some available libraries and toolkits for implementations are: MIRACL, Sage, PBC (Pairing-Based Cryptography) library etc.

We use PBC library for implementations of various ID-based multi-proxy signature schemes [5, 10, 20]. PBC library [16] is a free C library. It is based on GMP library which performs mathematical operations underlying pairing-based cryptosystems. To implement our ID- based multi- proxy signature scheme,we use the following congured PC :

Operating System: Linux
RAM: 2 GB
Processor: Intel Core 2 Duo CPU T5670@1.80 GHZ.
In our implementation, we test our scheme in the following curves:

-*Type A*: Type A [16] pairings are constructed on the elliptic curve $y^2 = x^3 + x$ over the eld Fq for some prime q = 3 mod 4. Group G_1 is the group of points on $E(F_q)$ and G_2 is a subgroup of F_{q^2} The value r is taken as some prime factor of $q+1$. For efficiency, r is picked to be a Solinas prime, that is, r has the form $2^a \pm 2^b \pm 1$ for some integers $0 < b < a$, such that $q+1 = r * h$, for an integer h. Type A curve parameter leads to generate *a.param*, which are command line inputs to our algorithm. Precisely, *a.param* elds are:

exp2, exp1, sign1, sign0, r:
r = $2^{exp2} + sign1\ \ 2^{exp1} + sign0*1$ (Solinas prime)
q, h : $r * h$ = q + 1
q is a prime, h is a multiple of 12 .
E : $y^2 = x^3 + x$

We take the following values of *a.param*.
q = 8780710799663312522437781984754049815806883199414208211028653399266475630880222957078625179422662221423155858769582317459277713367317481324925129998224791.

$h = 120160122648911460793888213667405342048029544012513118229$
$196151310472072893597045311028448021839065377867 76.$

$r = 730750818665451621361119245571504901405976559617.$

$exp2 = 159$
$exp1 = 1074$
$sign1 = 1$
$sign0 = 1$

- *Type E*: This curve [16] leads to generation of *e.param*, it results slower pairing and large storage space to represent group elements. In particular, *e.param* elds are:

$exp2$, $exp1$, $sign1$, $sign0$, r:
$r = 2^{exp2} + sign1\ 2^{exp1} + sign0*1$ (Solinas prime)
q, h: $q = h * r^2 + 1$ where r is prime, and h is 28 times a perfect square, a, b
$\mathrm{E} : y^2 = x^3 + ax + b$

We consider the following values of *e.param*.

$q = 7245986106510086080714203333362098431608853 3358$
$674258779609169284966291829916296649036541 0021$
$490094645005387278662999586944569372400129904165$
$7434948257845644905153122838458864000479326695430719 25$
$8600053239930483226650953770354174712511646273 51$
$6974069245462534034085895319225452125649979474047163305307830001$

$r = 730750862221594424981965739670091261094297337857$

$h = 1356934311091878183983524902148297025260321658798803$
$004483610694882551693017327097861748903233400 1$
$006615524543925753725725046733884363846960470 44$
$440474724128774377374668218852173872879715376027511$
6924829183670000

$a = 71309704540257990000679461375944460755515699495838 1594$
$339010872328239697373779427339724689227498188 38$
$0798952559954063085564496842679492921 5$
$5993804252696258727638014859680071360004717183351857872068 76$
$24287104269777860887513907871162183685823742940305$
$2273312335081163896980825048123655535355411494046493419999$

$b = 7169309004853894693616698536183663527570664411678352588 2$
$470447916871410434890727372327159615882882380220$
$10974661903752526911876859197052490 95$
$206526626569913014425203159149104533380758778860076455 7$
$450846327338626261289568016170532652061787582791926724 5$
$973624013988045630936251827909870167282900504660982233 33$

$exp2 = 159$
$exp1 = 135$
$sign1 = 1$
$sign0 = 1$

Remarks: Due to excess of length, we are omitting the coding of our scheme done on PBC. In the above implementation through PBC library [16], we consider one original signer and three proxy signers.We take 2009is17@gmail.com as ID of the original signer and *ID's* of proxy signers are rajeevs1729@gmail.com, sahadeomathrsu@gmail.com and rsy@mnnit.ac.in respectively. The above mentioned curve A and E are used as inputs to our scheme. We have also implemented schemes [5, 10] in PBC individually on the curves A and E with the same *ID's* and other inputs.

6.1 Comparison of running time of various schemes

In this section, we briey compare the total running time of various ID-based multi-proxy signature schemes, on the basis of outputs of corresponding algorithms done in PBC library on the above environment. Based on analysis of various outputs of above inputs, we compare the average running time of our ID-based multi-proxy signature scheme with other schemes [5, 10] in the following table.

Curve	Scheme of Li and Chen [10]	Scheme of Cao and Cao [5]	Our Scheme
A	0.387116 s	0.463559 s	0.184587 s
E	1.336742 s	1.961743 s	0.509513 s

Table 1. Comparison of running time of various ID-based multi-proxy signature schemes. Time is counted in seconds

We also sketch graphs with above values of running time of various ID-based multi-proxy signature schemes with respect to curves A and E as above and observe the efficiency in running time represented graphically. In the following graphs, the X-axis represents curve type and Y-axis represent the running time in miliseconds. The graph with curve type A is as follows:

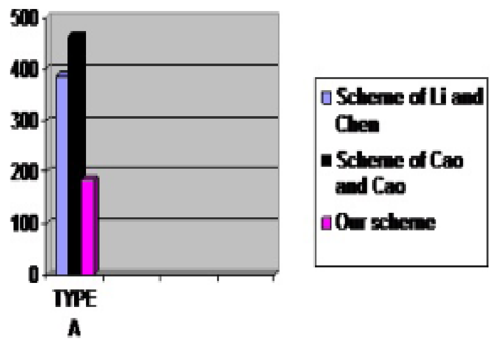

Figure 1. Graph with curve type A

The graph with cuve type E is below:

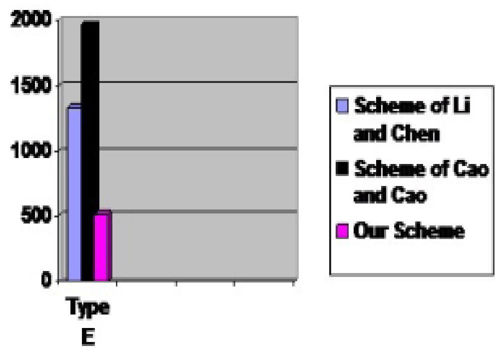

Figure 2. Graph with curve type E

From the above table and graphs it is clear that our ID-based multi-proxy signature scheme is more efficient in terms of total running time than schemes [5,10].

7. Conclusion

In this paper, we have proposed an ID-based multi-proxy signature scheme based on *k-plus* problem. Security of our scheme is based on *k-plus* problem under the INV-CDHP and DLP assumptions. Describing various applications of proposed scheme, we have given an implementation of the scheme using PBC library. Our scheme is computationally more efficient than other existing schemes [5, 10]. Moreover, total running time of our scheme is significantly less than other similar schemes [5, 10]. Additionally we have also shown that our scheme satisfies all the security requirements of a proxy signature scheme mentioned in [12].

References

[1] Boneh, D., Franklin, M. (2001). Identity Based Encryption from theWeil Pairing, Proc.of Crypto 01, Lecture Notes in Computer Science, Vol. 2139, Springer-Verlag, p. 213-229.

[2] Barreto, P, S, L, M.,Libert, B., McCullagh, N. (2005). Efficient and provably-secure identity-based signature and signcryption from bilinear maps, B. Roy (Ed.):Asiacrypt2005, LNCS 3788, 2005, p. 515-532.

[3] Boneh, D., Lynn, B., Shacham, H. (2001). Short signatures from the Weil pairing, *In:* C.Boyd, editor, Advanced in Cryptology-Asiacrypt 2001, LNCS 2248, gold Coast,Springer-Verlag, p. 514-532.

[4] Cha, J, C., Cheon, J, H. (2003). An identity-based signature from gap Diffie-Hellman groups, PKC 2003, LNCS 2567, Springer-Verlag, p. 18-30.

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

[6] Hesss, F. (2002).Efficient identity based signature scheme based on pairings, SAC2002, LNCS 2595, Springer-Verlag, p. 310-324.

[7] Hwang, S., Chen ,C., (2004). New multi-proxy multi-signature schemes, *Appl. Math.Comput*. 147, p. 57-67.

[8] Hwang, S., Shi, C. (2000). A simple multi-proxy signature scheme, *In:* Proceedings of the 10th national conference on information security, Hualien, Taiwan, ROC; 2000. p. 134138.

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

[10] Li, X., K.Chen. (2005). ID-based multi-proxy signature, proxy multi-signature and multi-proxy multi-signature schemes from bilinear pairings. Appl Math Comput.

[11] Li, X., Chen, K., Li, Li., (2004). Multi-proxy signature and proxy multi-signature schemes from bilinear pairings, K.M.Liew et. al. (Eds): PDCAT 2004, LNCS 3320, Springer-Verlag, p. 591-595.

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

[13] Mambo, M., Usuda, K., Okmamoto, E. (1996). Proxy signatures: delegation of the power to sign message, IEICE Transaction Functional E79-A (9) 1p. 1338-1354.

[14] Okamoto, T., Inomata, A., Okamoto, E. (2005). A Proposal of Short Proxy Signature using Pairing, International Conference on Information Technology: Coding and Computing (ITCC'05) - Volume I, p. 631-635.

[15] Paterson, K,G.. (2002). Id-based signatures from pairings on elliptic curves, Cryptology ePrint Archive, Report 2002/004.

[16] Ben lynn. The Pairing-Based Cryptography Library Version 0.5.12. Available: http://crypto.stanford.edu/pbc.

[17] William Stein.(2009). Sage: Open Source Mathematical Software (Version 3.2.3)". The Sage Group, 2009. http://www.sagemath.org.

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

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

[20] Mishra, S., Sahu, R, A., Padhye, S., Yadav, R,S. (2011). Efficient ID-based multi- proxy signature scheme from bilinear pairing based on *k-plus* problem, *In*: Proc. of INTECH 2011 , Communications in Computer and Information Science (CCIS) series of LNCS, Springer-Verlag, p. 113-121.

[21] Mishra, S., Sahu, R, A., Padhye, S., Yadav, R, S.(2011). An ID-based signature scheme from bilinear Pairing Based on *k-plus* Problem, *In*: Proc. of IEEE Intl. Conf.on Electronics Computer Tech. (ICECT 2011) 3 /11.

[22] Yi, X. (2003). An identity-based signature scheme from the Weil pairing , *IEEE Communication Letters*, 7 (2) February. p. 76-78.

[23] Yi, L.,Bai, G., Xiao, G. (2000). Proxy multi-signature scheme: a new type of proxy signature scheme, *Electronics Letters* 36 (6) p. 527-528.

[24] Zhang, F., Kim, K., (2003). Efficient ID-Based Blind Signature and Proxy Signature from Bilinear Pairing, ACISP 2003, LNCS 2727,p.312-323.

[25] Zhang, F., Safavi-Naini, R., Susilo, W., (2004).An Efficient Signature Scheme from Bilinear Parings and Its Applications, PKC 2004, LNCS 2947, 2004, p.277-290.