An Improved ID based Proxy Signature Scheme based on Elliptic Curve Cryptography

Deepa Mukherjee
Electronics and Telecommunication
SGSITS
Indore, India
Email: dmukherjee1991@gmail.com

Prakash Vyavahare
Electronics and Telecommunication
SGSITS
Indore, India
Email: prakash.vyavahare@gmail.com

Manish Panchal
Electronics and Telecommunication
SGSITS
Indore, India
Email: hellopanchal@gmail.com

***Abstract*—Proxy signature schemes allow the original signer of a message to delegate his signing capability to a proxy signer to generate a valid proxy signature on behalf of the original signer. One such scheme is proposed by Zhang and Kim which is based on Elliptic Curve Cryptography and Identity based Signature. However, Zhang's scheme requires secure channel for transmission of private key, has no provision of private key revocation and signature verification by any user. In this paper, we propose an improved ID based proxy signature scheme based on bilinear pairing. The scheme employs Knapsack algorithm for key distribution which eliminates the need for secure channel for sending the private keys from Private key generator (PKG) to respective users. The scheme also supports private key revocation by concatenating time parameter with public key of proxy signers. The signature can be verified only by a designated verifier. It is shown that the proposed proxy signature scheme satisfies all security requirements. Finally, the proposed proxy signature scheme is compared with that of Zhang and Kim's scheme and is shown to have merits over the latter one. Therefore, the proposed scheme can be a potential candidate for implementation of future proxy signature schemes.**

Keywords–Proxy Signature Scheme; ID based Cryptography; designated verifier scheme; ECC; Knapsack Algorithm;

I. INTRODUCTION

Digital Signatures are used in a wide variety of modern cryptographic systems that support data integrity and authentication. In public key cryptography, prior to any communication, each user should obtain a certificate from the Certificate Authority validating their public-private key pair. Proxy signatures schemes are one of the variety of digital signatures. Proxy signature schemes are required when the original signer is not available for some duration due to some reasons. Proxy signature scheme can be of two types depending on the signing authority. In full delegation scheme, signing rights are given permanently to the proxy signer. In partial delegation, signing right is delegated for a fixed period of time. The period of delegation and the type of messages that can be signed is usually specified by message warrant issued by the original signer at the time of delegation of signing authority.

In public key cryptography, the users must obtain their public-private key pair from the Certificate Authority prior to message communication [1]. In case of ID based cryptography, a trusted third party called as Private Key Generator (PKG) generates public-private key pair for the signers and transmits it to them via secure channel [2] [3]. In the recent proposals of proxy signatures, the public key of the signers is based on their popular public IDs (such as email id, telephone number etc).

Mambo [4] has described a proxy signature scheme based on Discrete Logarithm Problem (DLP). Recently, an improved proxy signature scheme based on RSA algorithm was proposed by Akanksha et. al [5] in Secureware 2015. The first ID based proxy signature scheme was proposed by Zhang and Kim, which requires a secure channel for transmission of private keys to the respective signers. Zhang and Kim [6] have described an identity based proxy signature scheme based on Elliptic Curve Cryptography. SK Hafizul [7] has described a designated verifier proxy signature scheme. This scheme [6] requires a secure channel for transmission of private key from PKG to user. It has no provision for private key revocation and the signature can be verified by any unknown verifier. The proposed proxy signature scheme attempts to overcome the drawbacks of this particular scheme.

In this paper, we have proposed an ID based proxy signature scheme that eliminates the requirement of a secure channel for transmission of secret key from PKG to signer. It also allows for changing the private key from time to time or when it is compromised to avoid its misuse for a long time. The proposed signature scheme has a designated verifier for verification of the signature created by the signer.

The rest of the paper is organized as follows. Section 2 decribes the Zhang and Kim's scheme. In Section 3, an improved ID based proxy signature scheme is proposed with its security analysis presented in Section 4. Finally, the paper is concluded in Section 5.

II. ZHANG AND KIM'S SCHEME

Let PKG be the private key generator. It generates public-private key pairs for the original and proxy signer. Let, Alice (A) be the alias name for original signer and Bob (B) be the alias name for proxy signer. Let Z_p be a field of order p. Let P be an element of Z_p having order p. Let p be a primitive element of Z_p. G_p be an additive cyclic subgroup of Z_p generated by P and G_M be a multiplicative group obtained by bilinear pairing of G_p and $e : G_p \times G_p \rightarrow G_M$ be a bilinear map that maps an element in G_p to an element in G_M. Table 1 summarizes the list of conventions and notation used in paper. The scheme advances as follows:

1) **Setup Phase**
In setup phase, Private Key Generator(PKG) generates its own public/private key pair. Let P_{pub} be

TABLE I. **LIST OF SYMBOLS**

SYMBOL	SIGNIFICANCE
A	Original Signer
B	Proxy Signer
C	Verifier
$C_{1,i}$, $C_{2,i}$	Encrypted r_i
G_p	Additive group over Z_p
G_M	Multiplicative Group obtained by Bilinear Mapping of G_p
H_1, H_2	Publicly Known Hash function
ID	Identity of the user e.g. email.
P	Generator element of Z_p
PKG	Private Key Generator
P_{pub}	Public key of PKG
Q_A, S_A	Public-private key pair of original signer
Q_B, S_B	Public-private key pair of proxy signer
Q_C, S_C	Public-private key pair of verifier
Q_W, S_W	Public-private key pair of proxy signer in proposed scheme to sign any message
Q_i'	Intermediate Public Key of the User i in the proposed scheme
S_1'	Signature of original signer on message warrant m_w in proposed scheme
S_g	Signature of proxy signer on message m in proposed scheme
U_A, c_A	Signature of original signer on message warrant mw in Zang and Kim's scheme
U_B, c_B	Signature of proxy signer on message m in Zang and Kim's scheme
Z_P	[0,p-1]
Z_p*	(1,p-1]
$Z \in_R [1, p{-}1]$	Random Number (Nounce) selected from Z_P
e	Bilinear map which maps an element in G_M to an element in G_P
k_A	Random number generated by original signer in Zang and Kim's scheme
k_B	Random number generated by proxy signer in Zang and Kim's scheme
l	Bitwise length of private key S_i of user i in proposed scheme
m_w	Message warrant
p	Number of elements in field Z_p
r_i	Point on elliptic curve randomly selected by user i for Knapsack algorithm
s	Master key or secret key of PKG
t_i	Time for which the generated public key in proposed scheme is valid

the PKG's public key that is generated using PKG's master key s as follows:

- a. Let G_P be an additive cyclic subgroup of Z_p and G_M be a multiplicative cyclic group obtained by bilinear mapping of G_p each of prime order p.
- b. Let P be the generator element of G_p
- c. Define a bilinear map $e : G_p \times G_p \rightarrow G_M$.
- d. PKG selects a random number $s \in_R Z_p^*$ and
- e. PKG calculates its own public key Ppub as follows
- f. $P_{pub} = sP$

The system public parameters are $params = (G_p, G_M, e, p, P, Ppub, H_1, H_2)$, where H_1 and H_2 are publicly known hash functions.

2) **Extract Phase**

In Extract phase, PKG calculates public and private key pairs (Q_A, S_A) and (Q_B, S_B) based on ID_A and ID_B for original and proxy signer respectively. Let ID be the public identity of the user such as telephone number or email id, etc.

- a. Let ID_i is the public ID of i where $i \in (A, B)$
- b. For the given identity ID of a signer(telephone number, email id, etc), PKG computes the public key Q_i as for ID as follows:
  $Q_i = H_2(ID)$
- c. The private key S_i is calculated by PKG as $S_i = sQ_{ID}$
  where s is the private key of PKG $s \epsilon Z_p$

Then, Q_{ID_i} is the public key of i where $i \in (A, B)$
S_{ID_i} is the private key of original signer where $i \in (A, B)$
PKG sends S_A and S_B to A and B respectively on secure channel.
Note that ID_A and ID_B i.e. IDs of original and proxy signers are publicly known
Since H_2 is public function, anyone can calculate Q_A and Q_B

3) **Proxy Key Generation**

To delegate his signing capability to a proxy signer, the original signer A makes signed warrant m_w that consists of public IDs of A and B, type of messages that can be signed by proxy signer (B) and validity period of proxy signer's signatures.
To delegate the signing capacity to the proxy signer, the original signer (Alice) makes the signed warrant m_w consisting of public IDs of original and proxy signer, type of messages that can be signed and valid time period for proxy signature. The proxy key S_{Bm} is generated by Bob as follows:

- a. A randomly selects $k \in_R Z_p^*$ and computes
  $r_A = e\left(P, P\right)^k$
  $c_A = H_1\left(m_w \parallel r_A\right)$
  $U_A = c_A S_A + kP$
- b. A then sends (m_w, c_A, U_A) to B on secure channel.
  Note that S_A and P lie on elliptic curve on Z_p and c_A and k_A are scalar quantities.
  and r_A is not sent explicitly from A to B
- c. On receiving the above information from A, proxy signer B computes the following:
  $r_A = e\left(U_A, P\right) e\left(Q_A P_{pub}\right)^{-c_A}$
  and accepts the signature to be valid if and only if
  $c_A = H_1\left(m_w \parallel r_A\right)$
  This validates that B has received information from A only(authentication).
- d. If the signature on message warrant is valid, B computes his private proxy key as follows:
  $S_{Bm} = c_A S_B + U_A$ where S_{Bm} is a modified proxy key created by proxy signer using the original proxy key sent by PKG to user.

4) **Proxy Signature Generation**

The message m is signed by proxy signer B using his proxy key S_{Bm} as follows:

- a. Proxy signer B selects a random number $k_B \epsilon Z_p*$
- b. B computes $r_B = e\left(P, P\right)^{k_B}$
- c. B computes the proxy signature on message m using his proxy signature key S_B as follows:
  $c_B = H_1\left(m \parallel r_B\right)$
  $U_B = c_B S_{Bm} + k_B P$
- d. B broadcasts (m, c_B, U_B).
  where m is the message, r_B is an intermediate value and (c_B,U_B) is the signature of B

on message m.
The signature generated by this scheme is proxy protected as it can be created by the proxy signer only.

5) **Verification Phase**
Any verifier can verify signature on message m to be valid as follows:
  a) Verifier computes
  $r_B = e\left(U_B, P\right)\left(e\left(Q_A + Q_B, P_{pub}\right)^{H_1(m_w \| r_A)} r_A\right)^{-c_A}$
  b) Verifier accepts signature to be valid on message m if and only if
  $c_B = H_1\left(m \parallel r_B\right)$

A. Security Analysis of Zhang and Kim's scheme

The security analysis of Zhang and Kim's scheme is as follows:

1) Secure channel is needed for transmission of secret key from PKG to original signer A and proxy signer B.
2) If the private keys of original signer A and proxy signer B has been compromised, even then since people use their popular public IDs as public key, the system is no longer secure.
3) Validity of generated signature can be verified by anyone which may not be desirable in some situations.

III. Proposed Scheme

In previous section, the Zang and Kim's ID based proxy signature which did not fulfill all the security requirements. An ID based proxy signature scheme has been proposed that overcomes some of the shortcomings pointed out in the previous section. The given scheme consists of seven phases namely, 1. Setup phase, 2. Public Key Generation phase, 3. Private Key Generation phase, 4. Secret Key Sharing Phase 5. Proxy Key Generation Phase, 6. Proxy signature generation, and 7. Proxy signature verification.

Let PKG be the private key generator. It generates public-private key pairs for the original and proxy signer and verifier. Let Z_p be a field of order p. Let P be an element of Z_p having order p. Let G_p be an additive cyclic subgroup of Z_p generated by P and G_M be a multiplicative group obtained by bilinear pairing of G_p and $e : G_p \times G_p \rightarrow G_M$ be a bilinear map that maps an element in G_p to an element in G_M.

The various steps involved in the proposed proxy signature scheme are as follows:

1) **Setup phase**
In this phase, the PKG generates its own public private key pair(P_{pub}, s) as follows:
  a. PKG selects an elliptic curve E over Z_p and broadcasts it.
  PKG randomly selects $s \in Z_p$ where s is the private key of PKG.
  b. Let P be a point on elliptic curve. PKG generates its public key P_{pub} as follows:
  $$P_{pub} = sP \quad (1)$$
  where s is the private key of PKG
  PKG then broadcasts P_{pub} and P.

2) **Public Key Generation**
In this phase, PKG generates public keys of original signer A, proxy signer B and verifier C as follows:
  a. PKG calculates intermediate public key $Q_i{}'$ using public ID of signer(such as email ID, telephone number etc) and a publicly known hash function H_1.
  $Q_i{}' = H_1\left(ID_i\right)$
  b. The intermediate public key $Q_i{}'$ is concatenated with time parameter t_i which indicates the validity period of proxy signature key.
  $Q_i = Q_i' \parallel t_i$
  Q_i is the public key for entity where $i \in (A, B, C)$
  Note that public key is changed by PKG from time to time so that even if the private key is compromised, it cannot be misused for a longer time.

3) **Private key generation phase**
PKG computes each user i's private key as follows:
$S_i = sQ_i$
where
s is the secret key of PKG
Q_i is the public key of user i and
S_i is the secret key of user i

4) **Secret Sharing Phase**
  a. To obtain its private key, each user i selects a random point r_i on elliptic curve where $i \in (A, B, C)$
  Let $r_i = (r_{ix}, r_{iy})$ where r_{ix} and r_{iy} are the x and y coordinates of r_i respectively.
  b. User then computes $n_i = |r_{ix} + r_{iy}|$
  c. User i then selects another random number $k_i \in Z_p$.
  d. Each user i then encrypts the point r_i using PKG's public key according to the following equations [8]:
  $C_{1,i} = k_i P$
  $C_{2,i} = r_i + k_i P_{pub}$
  Where P_{Pub} is the public key of PKG.
  Note that P, r_i, $C_{1,i}$, $C_{2,i}$ and P_{pub} are points on an elliptic curve over Z_p and k_i is a scalar quantity
  e. User i then sends $C_{1.i}$ and $C_{2,i}$ to PKG on public channel.
  f. The PKG then decrypts $C_{1,i}$ and $C_{2,i}$ and obtain r_i as follows
  $r_i = C_{2,i} - sC_{1,i} = C_{2,i} - sk_iP = r_i + k_iP_{pub} - k_iP_{pub} = r_i$
  g. PKG then computes $n_i = |r_{ix} + r_{iy}|$
  h. PKG calculates a series N_i using number n_i as
  $N_i = (1, n_i, n_i^2, ...n_i^j, n_i^{l-1})$
  where $i \in (A, B, C)$ and $j \in (0, 1, 2....., l - 1)$
  where l is the bitwise length of the private key.
  i. PKG converts S_i into binary form as $S_i = (b_{l-1}, b_{l-2}, ..., b_1, b_0)$
  Where b_{1-1} is the Most Significant

Bit(MSB) and b_0 is the Least Significant Bit(LSB)

j. PKG computes R_i for each user i using KNAPSACK algorithm [9]
$R_i = \sum n_i^j b_j$, $0 \leq j \leq l-1$

k. Then PKG sends R_i to the signer on public channel.

l. Signer i recovers $S_i = (b_{l-1}, ..., b_0)$ as follows:
Let R_I be an intermediate value derived from R_i

I $k = 1$.
II $R_i' = R_i$
III $R_I = R_i' - n_i^{l-k}$.
IV If $R_I < 0$
$b_{l-k} = 0$
IV If $R_I \geq 0$
$b_{l-k} = 1$, $R_i' = R_I$
V $k = k + 1$
VI If $k \leq l$, go to step III
If $k > l$, then end the process

In this way user i recovers his secret key S_i.

5) **Proxy Key Generation**
Original signer creates a message warrant m_w specifying public identities of original and proxy signer, validity period of signing of the proxy signature and type of messages that can be signed.

a. The original signer computes

$$S_1 = H_2 (m_w \parallel S_A) \tag{2}$$

Where H_2 is publicly known hash function. and sends (m_w, S_1) to B on public channel.

b. B computes

$$S_1{}' = H_2 (S_1 \parallel S_B) \tag{3}$$

and sends $\left(m_w, S_1{}'\right)$ to PKG on public channel.

c. PKG accepts $\left(m_w, S_1{}'\right)$ if the following equation holds true:

$$S_1{}' = H_2 (H_2 (m_w \parallel S_A) \parallel S_B) \tag{4}$$

This verification can be done by PKG since S_A and S_B are known to PKG only

d. Then, PKG finally computes public key (Q_w) and private key (S_w) of proxy signer for signing a message.

$$Q_w = H_1 (m_w) \tag{5}$$

$$S_w = sQ_w \tag{6}$$

e. PKG then sends S_w to B on public channel using Knapsack algorithm. PKG also broadcasts the public key Q_w.

f. B accepts (S_w, Q_w) only if the following equation holds true:

$$e (S_w, P) = e (H_1 (m_w), P_{pub}) \tag{7}$$

This step ensures data integrity of S_w and Q_w.

6) **Proxy Signature Generation**
In this phase, proxy signer (B) generates proxy signature on message m in following manner:

a. B computes

$$T = e (S_w, Q_C) \tag{8}$$

where Q_C is the public key of verifier.

b. B then computes

$$S_g = H_2 (m \parallel m_w \parallel T) \tag{9}$$

c. B sends (m_w, m, S_g) to the verifier for verification

7) **Proxy Signature Verification**
To accept the signature is accepted by the verifier by calculating the following:

a. PKG calculates an intermediate value $\bar{T}$ as follows:

$$\bar{T} = e (H_1 (m_w,), S_C) \tag{10}$$

Where S_C is the private key of verifier given by following equation:

$$S_C = sQ_C \tag{11}$$

Where s is the private key of PKG and Q_C is the public key of verifier.

b. PKG calculates an intermediate variable s' as follows:

$$s' = H_2 \left(m \parallel m_w \parallel \bar{T}\right) \tag{12}$$

c. The signature is accepted by PKG if the following equation holds true:
$s' = S_g$

As the proxy signer B uses his own private key S_W, neither the original signer nor PKG can create a valid proxy signature.

Only a designated verifier can verify the proxy signature as the designated verifier's public key (Q_C) is also involved in creating the signature for message m and it can be verified by the designated verifier only by using his own private key.

A. An Implementation Example of the Proposed Scheme

The scheme can be implemented using an example given below. The elliptic curve considered is $E : y^2 = x^3 + 4x + 20$ and the calculations have been done using elliptic curve calculator [10]. The various steps of the proposed scheme can be exemplified as follows:

a **Setup Phase**
Let $E : y^2 = x^3 + 4x + 20$ be an Elliptic Curve defined over $Z_{29} = (0, 28)$.
Let P=(1,5) be a point on E over Z_p.
We assume that the order of P in 29.
Let the private key of PKG, s=3.
The public key of PKG P_{pub} is calculated as follows:
P_{pub} = sP = $3(1, 5)$ = (20,3)

b **Public Key Generation Phase**
Let ID_i be the publicly known ID of user i.
Let H_1 be a hash function that maps ID_i to a point on E.
$Q_i{}' = H_1 (ID_i)$

The intermediate public key ${Q_i}'$ is concatenated with time parameter t_i which indicates the validity period of proxy signature key.
$Q_i = Q_i' \parallel t_i$
Q_i is the public key for entity where $i \in (A, B, C)$ where Q_i is the public ID of user i where $i \in (A, B, C)$
Where A is the original signer, B is the proxy signer and C is the verifier.
Let $Q_A = (20, 3)$, $Q_B = (4, 19)$ and $Q_C = (15, 27)$.

c **Private Key Generation Phase**
Private key of A i.e. S_A is calculated as follows:
$S_A = sQ_A$=$S_A = 3(20, 3)$=(14,23)
Similarly, $S_B = (17, 19)$ and $S_C = (19, 13)$

d **Secret Key Sharing Phase**
Let user A selects $r_A = (3, 1)$.
Therefore n_A = $|3 + 1|$=4
The generated series N_A = 1, 4, 16,........
A selects a random number k_A=2
A encrypts r_A as follows:
$C_{1,A} = k_A P$ = 2(1, 5) = (4, 19)
$C_{2,A} = r_A + k_A P_{pub}$ = (3,1) + 2(20,3) = (0, 7)
(4, 19) and (0, 17) is sent by A to PKG instead of (3, 1) on public channel.
PKG recovers r_A as follows:
$r_A = C_{2,A} - sC_{1,A}$ = (0, 17) - 3(4,19)= (3,1)
PKG calculates n_A = $|3 + 1|$ = 4
PKG generates N_A = 1, 4, 16,........
PKG converts S_A = (14, 23), the private key of A into binary form (01110, 10111).
14 is encrypted as follows:
$(14)_{10}$ = $(01110)_2$ = (0x256) + (1x64) + (1x16) + (1x4) + (0x1) = 84.
Similarly 23 is encrypted as 277.
PKG sends (84, 277) instead of (14, 23) to A on public channel.
84 is decrypted as follows:
Let R_I be an intermediate variable.
R_I = 84 - 4^4 = -172 which is negative, hence $b_4 = 0$.
R_I = 84 - 4^3 = 20 which is positive, hence $b_3 = 1$.
R_I = 20 - 16 = 4 which is positive, hence $b_2 = 1$.
R_I = 4 - 4^1 = 0 which is 0, hence $b_1 = 1$.
R_I = 0 - 1 = -1 which is negative, hence $b_0 = 0$.
Hence, 84 is decrypted into $(01110)_2$ = $(14)_{10}$.
Similarly, 277 is decrypted into $(10111)_2$ = $(23)_{10}$.
In this way, A recovers its private key S_A = (14, 23).
Similarly, B and C receive their private key S_B = (17, 19) and S_C = (19, 13).

e **Proxy Key Generation**
Original signer selects a message warrant m_w = 6.
The original signer computes
$S_1 = H_2 (m_w \parallel S_A) = H_2 (3 \parallel (14, 23))$
Where H_2 is publicly known hash function that gives a point S_1 on elliptic curve E.
Let S_1 = (10, 4)
and sends $(6, (10, 4))$ to B on public channel.
B computes
${S_1}' = {S_1}' = H_2 (S_1 \parallel S_B) = H_2 ((10, 4) \parallel (17, 19))$
Let ${S_1}'$ = (1, 24)
and sends $(6, (1, 24))$ to PKG on public channel.
PKG accepts $(6, (1, 24))$ if the following equation holds true:
$H_2 (H_2 (6 \parallel (14, 23)) \parallel (17, 19))$ = (1, 24).
PKG computes public-private key pair (S_w, Q_w) of proxy signer B as follows:
$Q_w = H_1 (6)$
Where H_1 is a publicly known hash function that maps m_w to a point Q_w on elliptic curve E
Let Q_w be (8, 10).
The private key S_w is calculated as follows:
$S_w = sQ_w$ = 3(8, 10) = (16, 2).
PKG sends (16, 2) to B on public channel using Knapsack algorithm.
PKG also broadcasts the public key Q_w.
B accepts $(16, 2)$ only if the following equation holds true:
$e ((16, 2), (1, 5)) = e (H_1 (6), (20, 3))$, where e is a bilinear pairing that maps a pair of elements in additive cyclic group G_p to an element in multiplicative group G_M.
The above condition holds true if S_w is valid.
This step ensures data integrity of S_w and Q_w.

f **Proxy Signature Generation**
Let the message to be signed by proxy signer be m = 8.
(B) generates proxy signature on message m = 8 in following manner:
B computes
$T = e (S_w, Q_C) = e ((16, 2), (15, 27))$
where Q_C is the public key of verifier. e is a bilinear pairing that maps a pair of elements in additive cyclic group G_p to an element in multiplicative group G_M.
Let us assume that bilinear pairing e maps (16, 2) and (15, 27) to an element (20, 26).
$e ((16, 2), (15, 27))$ = (20, 26)
B then computes $S_g = H_2 (8 \parallel 6 \parallel (20, 26))$
Assuming that the hash function gives (13, 6) as output, we obtain the following equation:
$S_g = H_2 (8 \parallel 6 \parallel (20, 26))$ = (13, 6)
B sends $(6, 8, (13, 6))$ to the verifier for verification.

g **Proxy Signature Verification**
To accept the signature is accepted by the verifier by calculating the following:
PKG calculates an intermediate value $\bar{T}$ as follows:
$\bar{T} = e (H_1 (6,), (19, 13))$
Let us assume that $H_1(6)$ = (2, 6).
We also assume the following:
$\bar{T} = e (H_1 (6,), (19, 13))$ = (2, 6)
Where (19, 13) is the private key of verifier.
PKG calculates an intermediate variable s' as follows:
$s' = H_2 (6 \parallel 8 \parallel (2, 6))$
The signature is accepted by PKG if the following equation holds true:
$s' = S_g$
The above equation holds true if the authorized proxy signer B signs the message m = 8 and designated verifier C verifies the signature.

B. Security Analysis of the Proposed Scheme

In this section we discuss about the security aspects of the proposed scheme such as trusted PKG, proxy key revocation, designated verifier, proxy protected, unforgability, non repudiation and secure channel. They are as follows:

1) **Trusted PKG**
The security of ID based signatures is based on the fact that PKGs should be trusted. If the PKG is not trusted then the scheme is not secure. However, given a trusted PKG, the scheme is secure.
2) **Private key revocation**
Even if private key of user is compromised, it cannot be misused for a long time as public key is valid only for particular time for which the time parameter t_i remains unchanged.
3) **Designated verifier**
Only designated verifier C can verify the proxy signature which is desirable in some situations.This is done by using the public key of verifier Q_C in creating the signature S_g which can be verified only if the verifier has the corresponding private key S_C. This happens because the designated verifiers public key is also involved in signing the message m and it can be verified by the designated verifier using his own public key.
4) **Proxy protected**
Only the proxy signer should be able to create a valid proxy signature, not the original signer. In this scheme, the secret key of the proxy signer S_w is calculated by PKG using his own secret key s which cannot be calculated by the original signer due to Elliptic Curve Discrete Logarithm Problem(ECDLP). Hence the proposed scheme is proxy protected.
5) **Unforgability**
Only the proxy signer should be able to create a valid proxy signature. In the proposed scheme, as the proxy signer creates the signature S_g using his own private key S_w, no one else can sign on behalf of proxy signer, neither the original signer himself nor a third party.
6) **Non-repudiation**
The proxy signer should not be able to deny his signature later on. In this scheme the proxy signer creates signature S_g by using his private key S_w and is verified by verifier using proxy signer's public key Q_w using his public key. Hence, the proxy signer cannot deny his signature.
7) **Secure channel**
In Zang and Kim's scheme, a secure channel is required for transmission of secret key from PKG to signers. In our proposed scheme, the PKG uses KNAPSACK algorithm to encrypt the secret keys and signers use reverse knapsack to extract back the keys. Therefore communication can take place on insecure channel.

TABLE II. **COMPARISON BETWEEN ZHANG AND KIM's SCHEME AND PROPOSED SCHEME**

Parameters	Zhang and Kim's Scheme	Proposed Scheme
Number of Steps	5	7
Secure channel requirement	Yes	No
Proxy key revocation	No	Yes
Designated verifier	No	Yes

The proposed scheme eliminates the need for a secure channel for transmission of private key from PKG to signers. It also provides the feature of private key revocation as per the need. The scheme has a provision for designated verifier only. Table 2 summarizes the comparison between Zhang and Kim's scheme and proposed scheme.

IV. CONCLUSION

In this paper, we have proposed a new ID based proxy signature scheme. The scheme has eliminated the use of secure channel for transmission of private key from PKG to original signer, proxy signer and verifier using KNAPSACK algorithm. This scheme also exhibits Private key revocation feature such that if a private key is exposed, it cannot be used for a long time. This scheme allows the proxy signature to be verified by a designated verifier only. As it satisfies all security requirements, it can be used in future proxy applications. This scheme is designed for a single proxy signer only, which can be extended to multiple proxy signers. However, the proposed needs a trusted PKG. This condition can be removed as part of future work.

REFERENCES

[1] D. Hankerson , D. Menezes and D. Vanstone, Guide to Elliptic Curve Cryptography, Springer, 2004
[2] https//courses.cs.washington.edu/courses/csep590/06wi/finalprojects
[3] A. K. Kommera , K. Kommera and P. K. Gunda, A Closer Look at ECC and RSA, "International Journal of Computer Science and Information Technologies", 2011, pp. 2220–2224
[4] M. Mambo , K. Usuda and E. Okamoto, Proxy signatures:Delegation of the power to sign messages, IEICE Transactions Fundamentals E79-A9, 1996, pp. 1338–1353
[5] A. Gupta , P.D. Vyavahare and M. Panchal, An Improved Threshold Proxy Signature Scheme, "SECURWARE: The Ninth International Conference on Emerging Security Information, System and Technologies", 2015, pp. 49–54
[6] F. Zhang and Kim, Efficient ID-based blind signature and proxy signature from bilinear pairings, Proceedings of the "8th Australasian Conference on Information Security and Privacy", 2003, pp. 312–307
[7] S.K. Hafizul and G.P. Biswas, Design of an Efficient ID-based Short Designated Verifier Proxy Signature Scheme, "Conference on Recent Advances in Information Technology", 2012, pp. 64–72
[8] W. Stallings, Cryptography and Network Security: Pearson, 2012
[9] R.R. Rajaram , M.A. Prabakar , M.I. Devi and M. Suguna, Knapsack based ECC encryption and decryption, "International Journal of Network Security", 2009, pp. 218–226
[10] http://www.christelbach.com/ECCalculator.aspx