An Improved Threshold Proxy Signature Scheme

Akanksha Gupta, Prakash D.Vyavahare, Manish Panchal

Department of Electronics and Telecommunication Engineering
S. G. S. Institute of Technology and Science, Indore, India

Email: akkuregister.90@gmail.com, prakash.vyavahare@gmail.com, hellopanchal@gmail.com

***Abstract*—Threshold proxy signature schemes allow original signer to delegate his signing capability to a group of *n* proxy signers in which *t*, $(1 < t \leq n)$ or more proxy signers out of *n* can generate a valid proxy signature on behalf of original signer. The first Rivest, Shamir and Adleman (RSA) based threshold proxy signature scheme was proposed by Hwang et. al. . Later on Wang in his paper, commented on the security weakness in Hwang's scheme. In this paper, we propose a new RSA based scheme that removes weakness of Hwang's scheme. The proposed scheme employs efficient generation of shared RSA key using algorithm proposed by Boneh and Fanklin and RSA threshold cryptosystem of Nguyen. The proposed scheme provides secrecy, unforgeabilty, nonrepudiation, proxy protection and removes the need of trusted combiner. Therefore, it can be a potential candidate for implementation of electronic proxy signature system.**

Keywords– Proxy Signature; Threshold Signature; Secret Sharing; RSA.

I. INTRODUCTION

Digital signature is a cryptographic scheme used to authenticate the identity of the sender of a message and sender can not repudiate the message once it is signed by him. It also assures the recipient for the integrity of message. Many practical applications implement it either directly or in some other form. Proxy signature is one such example, where digital signature on a message is performed by *proxy signer* on behalf of original signer in his absence, to whom original signer has delegated his signing power.

Mambo *et. al.* [1] in 1996 first proposed the concept of proxy signature and classified it on the basis of delegations namely, *full delegation*, *partial delegation* and *delegation by warrant*. Full delegation was not secure and hence was not used in practical systems. In such a case, original signer gives his private key to proxy signer. Therefore, proxy signature is indistinguishable from original signature. Partial delegation and delegation by warrant are more secure than full delegation scheme.

In 1997, Kim [2] combined the idea of secret sharing and threshold crypto system to proxy signature scheme with less trust on single proxy signer. Initially proposed threshold proxy signature schemes were based on principle of discrete logarithm cryptosystem [3][4]. In (t, n) threshold scheme, t number of proxy signers $(1 < t \leq n)$ can cooperatively generate valid proxy signature on a message, but $(t - 1)$ or less proxy signers can not generate valid sign on the document. Threshold proxy signature scheme is more secure and practical than conventional proxy signature scheme. Number of publications of threshold proxy signature based on discrete logarithm have been reported [5][6][7][8]. Hwang et. al. [9] proposed the first RSA based threshold proxy signature scheme in which author described six requirements that should be satisfied by a secure (t, n) threshold proxy signature scheme.

1) **Secrecy:** Original signer's private key must be kept secret with original signer and it should not be possible to derive it by any proxy signer, not even by cooperation among them.
2) **Proxy protected:** Partial proxy signature of a designated proxy signer can only be generated by him. Even original signer can not be masquerade partial proxy signature. Partial proxy signature key must not known to original signer.
3) **Unforgeability:** Only t or more designated proxy signers can cooperatively generate valid proxy signature.
4) **Non repudiation:** Once t or more proxy signers cooperatively generate valid proxy signature, they can not deny their signatures and original signer also can not deny delegating the signing power to the proxy signers.
5) **Time constraint:** The proxy signature key can be used during the delegation period. After this period, proxy signature generated by proxy signer will be considered to be invalid.
6) **Known signers:** Scheme must be able to identify the actual group of signers from proxy group in threshold scheme.

Wang [10] analyzed the security aspects of Hwang's scheme and commented that Hwang's scheme was unable to fulfill the security requirements of threshold proxy signature system. Number of publications of the threshold proxy signature scheme based on RSA has been reported [11][12][13][14], that modified the Hwang's scheme and removed various security weakness. The most recent publication [15] adds a new feature by allowing n proxy signers, renew their own proxy shares periodically without changing the secret. In the proposed scheme the combiner need not be trusted. The verifier can independently verify if the combiner has done untrustworthy operation on proxy signature. All the above proposed schemes require trusted combiner and trusted original signer for generation and verification secret shares.

In this paper, we propose an improved threshold scheme based on RSA algorithm [16]. In the proposed scheme, n proxy signers execute the Boneh and Franklin protocol [17] to generate RSA based signature scheme modulo N, whose prime factors are not known to proxy signers and they do not

ISBN: 978-1-61208-427-5

TABLE I. CONVENTION AND NOTATION

C	Combiner
D	Proxy signature key to generate proxy signatures
D_i	Additive share of D for S_i
E	Proxy verification key to verify proxy signatures
N	RSA modulo
S	Proxy signature on message m
R_i	Random number
S_0	Original signer
S_i	i^{th} Proxy signer $(1 \leq i \leq n)$
T	Subset of n Proxy Signers who cooperatively generate signature on message m
V	Verifier
d_0	Original signer's private key
d_i	i^{th} Proxy signer's private key
e_0	Original signer's public key
e_i	i^{th} Proxy signer's public key
k_i	Partial proxy signature key of S_i to generate partial proxy signature
m	message to be signed
n	Number of proxy signer
r_i	Random number
s_i	Partial signature of S_i on message m
t	Threshold number, $(1 < t \leq n)$ of proxy signers require to generate proxy signature on message m
w	warrant generated by S_0
$\phi(N)$	Euler totient function
$\|\|$	concatenation of bit strings

know $\phi(N)$. Each proxy signer will derive its additive share D_i for Proxy signature key D without revealing it to others, such that $D = D_1 + D_2 + ... + D_n$. These shares are used to compute partial proxy signature, which are used to generate proxy signature S later on the need of trusted combiner has been removed in the proposed scheme.

The rest of the paper is organized as follows. Hwang's scheme is reviewed in Section II. Section III describes the weakness of Hwang's scheme. In Section IV proposed improved threshold proxy signature scheme is presented which is analyzed in Section V. Finally, the paper is concluded in section VI.

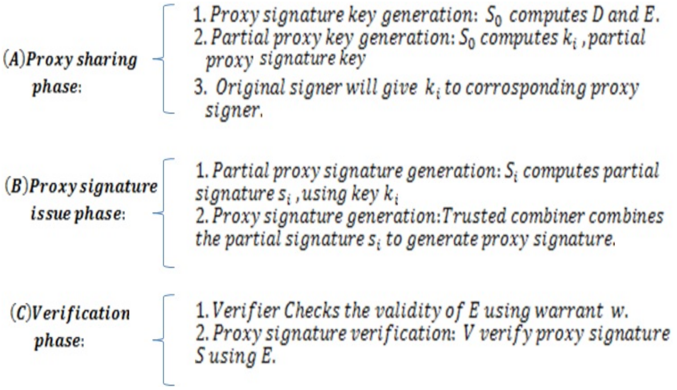
II. REVIEW OF HWANG'S SCHEME

Let S_0 denote the original signer and $S_1, S_2.....S_n$ be n proxy signers. C is a signature combiner and V is a signature verifier. Let $N_i = p_i * q_i$ be a public RSA modulus for S_i, where p_i and q_i are two secret large prime numbers for i= 1, 2, 3,..., n. Each of the proxy signers S_i has its own public key e_i and private key d_i such that $d_i * e_i = 1mod\phi(N_i)$. Let d_0 and e_0 be the private and public key of original signer and w is be a warrant generated by S_0. The warrant includes delegation period for validity of proxy signature, the proxy signers' identities and identity of the original signer. Other notations and conventions are shown in Table I.
Hwang's scheme consists of three phases which are shown in Figure 1 :

A. Proxy sharing phase: The original signer generates share of proxy signature key D.
B. Proxy signature issue phase: Proxies generate partial proxy signatures which, after combining, will create proxy signature on message and
C. Verification phase: Proxy signature is verified by proxy verification key E.

A. The Proxy Sharing Phase

The original signer S_0 delegates his signing capability to n proxy signers as follows:



(A) Proxy sharing phase:
1. Proxy signature key generation: S_0 computes D and E.
2. Partial proxy key generation: S_0 computes k_i , partial proxy signature key
3. Original signer will give k_i to corresponding proxy signer.

(B) Proxy signature issue phase:
1. Partial proxy signature generation: S_i computes partial signature s_i , using key k_i
2. Proxy signature generation: Trusted combiner combines the partial signature s_i to generate proxy signature.

(C) Verification phase:
1. Verifier Checks the validity of E using warrant w.
2. Proxy signature verification: V verify proxy signature S using E.

Figure 1. Three phase of Hwang's Scheme

1) S_0 computes $D = d_0^w mod\phi(N_0)$ and $E = e_0^w mod\phi(N_0)$ as proxy signature key and proxy verification key respectively. S_0 publishes w, E, and (w || E)$^{d_0} modN_0$.

2) S_0 generates partial proxy signature key k_i, shares of secret D for corresponding proxy signer S_i, using Shamir's secret sharing scheme [18]. For this, S_0 selects a polynomial

$$f(x) = D + R_1x + ... + R_{t-1}x^{t-1}mod\phi(N_0) \tag{1}$$

of degree $(t-1)$, where $R_1, R_2, ...R_{t-1}$ are random numbers chosen by S_0. Then, S_0 computes

$$k_i = f(i) \tag{2}$$

for S_i, where $i = 1, 2, ...n$ and sends $({k_i}^{d_0} modN_0 \| k_i)^{e_i} modN_i$ to S_i on public channel.

3) After receiving the data from S_0, S_i computes k_i by first decrypting it with its private key d_i and then by public key e_0 of S_0 as $k_i = ((({k_i}^{d_0} modN_0 \| k_i)^{e_i} modN_i)^{d_i} modN_i)^{e_0} modN_0$, where $i = 1, 2, ...n$.

B. The Proxy Signature Issue Phase

To sign a the message m on behalf of original signer S_0, any t or more proxy signers form a subset T of proxy signers. They generate proxy signature as follows:

1) Each proxy signer of subset T computes its partial proxy signature s_i on message m as follows:

$$s_i = m^{L_i k_i} modN_0 \tag{3}$$

where, i indicates the i^{th} proxy signer from subset T. L_i is Lagrange interpolation coefficient given by

$$L_i = \prod_{i,j \epsilon T, j \neq i} \frac{-j}{i-j} \tag{4}$$

and sends $s_i \| {s_i}^{d_i} modN_i$ to combiner C.

2) Combiner verifies s_i using public key e_i of S_i and then computes proxy signature S as follows:

$$S = \prod_{i=1}^{n} s_i modN_0 = m^D modN_0 \qquad (5)$$

C. The Verification Phase

After receiving proxy signature S, the verifier V verifies it as follows:

1) V receives w, E, (w ∥ E)$^{d_0} modN_0$ from S_0 and verifies it using $(w \| E) = ((w \| E)^{d_0} modN_0)^{e_0} modN_0$. V first checks the validity of E by checking the valid period mentioned in the warrant. If the period has expired then E is invalid and it can not be used for verification.

2) If E is valid, then V computes $S^E modN_0$ and checks whether it is equal to m.

$$S^E modN_0 = m \qquad (6)$$

Since
$S^E modN_0 = \left(m^D\right)^E modN_0$
$= m^{{d_0}^w {e_0}^w} modN_0$
$= m^{(d_0 e_0)^w} modN_0 = m$, and

3) The actual proxy signers identity can be identified using his public key e_i on ${s_i}^{d_i} modN_i$.

III. SECURITY ANALYSIS BY WANG

Wang [10] claimed that Hwang's scheme was not able to satisfy the security requirements of threshold proxy signature scheme and indicated various security weaknesses in it, such as:

- Secrecy: Proxy signers from the set T can cooperatively compute proxy signature key D and $(DE - 1)$, factors of $\phi(N_0)$. Knowing these factors is equivalent to factoring N_0 [21]. Finally, with the factors of N_0, the proxy signers can compute the value of $\phi(N_0)$. Once factors of $\phi(N_0)$ are calculated, it is easy to calculate d_0 by using $d_0 e_0 = 1 mod\phi(N_0)$. Hence, private key of original signer will not remain secret.

- Proxy Protected: Original signer knows the partial signing key k_i of corresponding proxy signer S_i. Therefore, he can create the partial proxy signature s_i on message m on behalf of S_i as k_i is just the share of D created by S_0 himself. Another security weakness is that the proxy signers have to trust on original signer that given partial proxy signature key k_i is correct for generating valid proxy signature S.

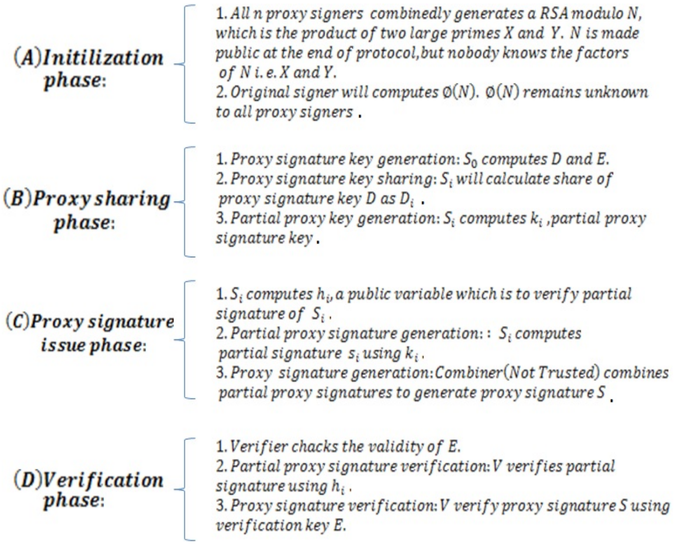
- Unforgeability: Hwang's scheme is weak in yet another aspect that an unauthorized third person can compute proxy signature S as he can calculate factors of N_0 because $({e_0}^w - E)$ is also a factor of $\phi(N_0)$. Using similar strategy as mentioned above, he can compute d_0 as well as $D = d_0^w mod\phi(N_0)$. However, it is a time consuming process.

- Non Repudiation: D is generated by original signer. Therefore, he can create proxy signature S on message m as $m^D modN_0$ by surpassing the combiner. Moreover, t proxy signers can also cooperatively compute proxy signature without combiner. In such cases Verifier will accept a signature S by checking $(S^E modN_0 = m)$. In future, if such a valid proxy signature S causes dispute on the identity of signer of message, it can not be found out as who was responsible for generating signature, whether the original signer or proxy signers.

- Known Signer: In Hwang's scheme, a trusted combiner is required. Otherwise, partial proxy signature s_i can be altered, replaced or deleted by the combiner.

IV. PROPOSED SCHEME

The proposed scheme consists of four phases namely (A). Initialization phase, (B). Proxy sharing phase, (C). Proxy signature issue phase, (D). Verification phase. These phases are described as follows in Figure 2:



(A)Initilization phase:
1. All n proxy signers combinedly generates a RSA modulo N, which is the product of two large primes X and Y. N is made public at the end of protocol,but nobody knows the factors of N i.e. X and Y.
2. Original signer will computes Ø(N). Ø(N) remains unknown to all proxy signers .

(B)Proxy sharing phase:
1. Proxy signature key generation: S_0 computes D and E.
2. Proxy signature key sharing: S_i will calculate share of proxy signature key D as D_i .
3. Partial proxy key generation: S_i computes k_i ,partial proxy signature key .

(C)Proxy signature issue phase:
1. S_i computes h_i, a public variable which is to verify partial signature of S_i .
2. Partial proxy signature generation: : S_i computes partial signature s_i using k_i .
3. Proxy signature generation:Combiner(Not Trusted) combines partial proxy signatures to generate proxy signature S .

(D)Verification phase:
1. Verifier chacks the validity of E.
2. Partial proxy signature verification:V verifies partial signature using h_i .
3. Proxy signature verification:V verify proxy signature S using verification key E.

Figure 2. Four phases of the proposed scheme

A. Initialization Phase

1) All n proxy signers combinedly generate a number N, which is the product of two large primes by employing Boneh and Franklin protocol [17]. N is made public but nobody knows the factors of N at the end of protocol. Following steps are followed to generate N :
Participants secret and Distributed Sieving:
   a) Each proxy signer S_i randomly picks up two secret numbers x_i, y_i.
   b) All proxy signers determine whether or not the sums $X = \sum_{i=1}^{n} x_i$, $Y = \sum_{i=1}^{n} y_i$ are not divisible by any prime number between 0 and some bound B1.

Computation of N: All proxy signers will compute N without reveling any information about their secrets x_i and y_i . They agree on a big prime number $M > N$ and an element g of high order in Z_N^* and use BGW protocol [19] as follows to generate N.

$$N = \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i = X * YmodM \tag{7}$$

Trial Division: This is required to ensure that N is not divisible by any number between B1 and B2, numbers which are agreed by all the parties.

Primality Test: Extended Fermat's Primality test is performed to check whether N is a product of two prime numbers or not. If it is not, then protocol is repeated from the first stage with new values of x_i, y_i until it passes the primality test.

2) Computation of $\phi(N)$

a) Each proxy signer S_i will calculate ϕ_i, the share of $\phi(N)$ as follows:

$$\phi_i = \{^{N-x_i-y_i+1 \quad if \;\; i=1}_{-x_i-y_i \quad if \;\; i>1} \tag{8}$$

and sends it to S_0 as $(\phi_i \| \phi_i^{d_i} modN_i)^{e_0} modN_0$.

b) After receiving data from all S_i, the original signer extracts ϕ_i by decrypting it with the private key d_0 of original signer and public key e_i of corresponding i^{th} proxy signer. S_0 then computes $\phi(N)$ as:

$\phi(N) = \sum_{i=1}^{n} \phi_i$

$= N - \sum_{i=1}^{n} x_i - \sum_{i=1}^{n} y_i + 1$

$= (X-1)(Y-1)$

Note that proxy signers can not determine $\phi(N)$ unless all proxy signers combine their shares ϕ_i for the generation of $\phi(N)$. Therefore $\phi(N)$ is (n-1) private.

B. Proxy Sharing Phase

This phase involves generation of proxy signature key D and its corresponding verification key E, by S_0 and generation of partial proxy signature key k_i by S_i. Note that k_i is generated by S_0 in Hwang's scheme.

Proxy Signature Key Generation:

1) S_0 chooses a random number a, where $a\epsilon Z_M$ and $gcd(a, \phi(N)) = 1$, and calculates $b = a^{-1} mod\phi(N)$ using Euclids extended algorithm.

2) Original signer then computes proxy signature key D and proxy verification key E, as follows:

$$D = b^w mod\phi(N), E = a^w mod\phi(N) \tag{9}$$

$DE = 1 mod\phi(N)$

3) S_0 will compute $\psi = \phi(N) modE$ and $\psi^{-1} modE$.

4) Proxy signature key D remains secret with original signer, Corresponding verification key E is made public for the verification of proxy signature.

5) S_0 publishes $w, E, a, \psi, (w\|E\|a\|\psi)^{d_0} modN_0$,

Proxy Signature Key Sharing:

1) Each Proxy signer S_i receives the published $w, E, a, \psi, (w\|E\|a\|\psi)^{d_0} modN_0$, from original signer, and verifies that $(w\|E\|a\|\psi) = ((w\|E\|a\|\psi)^{d_0} modN_0)^{e_0} modN_0$ using public key e_0 of original signer.

2) S_i gets proxy verification key E and warrant w, Therefore, each proxy signer can check the validity of E by checking the valid period mentioned in warrant w. If E finds it to be valid, then proxy signer will accepts it to derive shares of proxy signature D. Otherwise, he will reject it and request S_0 for a valid warrant and signature. Else, he stops this protocol.

3) Each S_i will calculate the additive share D_i of proxy signature key D using Boneh and Franklin protocol [17].

Each S_i will compute D_i as follows:

$$D_i = \{^{(1-\phi_1\psi^{-1})/E \quad if \;\; i=1}_{-(\phi_i\psi^{-1})/E \quad if \;\; i>1} \tag{10}$$

Finally, D is calculated by S_0 as follows:

$$D = \sum_{i=1}^{n} D_i =$$
$$(1-\phi_1\psi^{-1})/E + \sum_{i=2}^{n} (-\phi_i\psi^{-1})/E.$$

Calculated value of D is kept secret with S_0.

Partial Proxy Signature Key Generation: Proxy signers compute their own partial proxy signature keys k_i, as the share of actual proxy signature key D. However, D is not known to the proxy signers. k_i is calculated by S_i using following steps:

1) Each S_i selects a random polynomial $f_i(x)\epsilon Z_M$ of degree $(t-1)$, with $f_i(0) = D_i$. Let $f_i(x)$ be

$$f_i(x) = D_i + r_{i,1}x + ... + r_{i,t-1}x^{t-1} \tag{11}$$

S_i proxy signer computes $f_{i,j} = f_i(j)$ which is the share of D_i for S_j and sends $(f_{i,j} \parallel {f_{i,j}}^{d_i} modN_i)^{e_j} modN_j$ to S_j for $1 \leq j \leq n$ on the public channel. S_i broadcasts $c_{i,j} = g^{r_{i,j}} modN$ for j = 0, 1...., (t-1) to others.

2) S_j verifies validity of the share $f_{i,j}$ received from S_i, using following formula :

$$g^{f_{i,j}} = g^{f_i(j)} = g^{D_i + r_{i,1}j + ... + r_{i,t-1}j^{t-1}} modN$$
$$= g^{D_i}.g^{r_{i,1}j}.....g^{r_{i,t-1}j^{t-1}} modN$$
$$= \prod_{k=0}^{t-1} c_{i,k}^{j^k} modN$$

If verification fails, then S_j sends error message to the original signer.

3) Partial proxy signature key k_i for S_i is calculated as

$$k_i = \sum_{j=1}^{n} f_j(i) \tag{12}$$

k_i is kept as secret by S_i.

4) The above computation of k_i, as done by corresponding proxy signer S_i, is same as the one obtained by original signer S_0 in Hwang's scheme from polynomial $f(x) = D + R_1x + ... + R_{t-1}x^{t-1}$ of degree (t-1) having $f(0) = D$ and $k_i = f(i)$. The group of n proxy signers can obtain a polynomial f(x) of at most $(t-1)$ degree in the following form :

$$f(x)$$
$$= \sum_{j=1}^{n} D_i + (\sum_{j=1}^{n} r_{j,1})x + ...$$
$$.. + (\sum_{j=1}^{n} r_{j,t-1})x^{t-1}$$
$$= D + R_1 x + ... + R_{t-1}x^{t-1}$$
$$k_i = f(i)$$

C. Proxy Signature Issue Phase

1) Each proxy signer S_i computes

$$\sigma_i = L_i * k_i mod M \tag{13}$$

for $i = 1, 2, ..., n$. L_i is Lagrange interpolation coefficient which is calculated as follows:

$$L_i = \prod_{i,j\epsilon T, j\neq i} \frac{-j}{i-j} mod M \tag{14}$$

σ_i is kept secret by corresponding proxy signer S_i.

2) To sign the message m, proxy signers S_i, from the set T, will sign on behalf of original signer by generating his partial proxy signatures s_i on m as follows:

$$s_i = m^{\sigma_i} mod N \tag{15}$$

where $i\epsilon T$

3) Each proxy signer computes $h_i = g^{\sigma_i} mod N$, which will be used to verify the signature shares generated by proxy signers at verification stage.

4) S_i publishes its partial signature s_i, h_i and $(s_i \| h_i)^{d_i} mod N_i$.

5) Now a combiner (not necessary to be a trusted one) or any one proxy signer from group of n proxy signers, will combine all partial signatures to generate proxy signature S by collecting s_i's, and calculates as follows:

$$S = \prod_{i\epsilon T} s_i mod N \tag{16}$$

$= \prod_{i\epsilon T} m^{L_i k_i} mod N = m^{\sum_{i\epsilon T} D_i} mod N = m^D mod N$ The combiner publishes message m with its signature S and $(s_i \| h_i)^{d_i} mod N_i (i\epsilon T)$ to the verifier.

D. Verification Phase

To verify the signature S on message m, the verifier V will carry out the following steps:

1) The verifier V receives $w, E, a, \psi, (w\|E\|a\|\psi)^{d_0} mod N_0$, from S_0 and verifies that $(w\|E\|a\|\psi) = ((w\|E\|a\|\psi)^{d_0} mod N_0)^{e_0} mod N_0$. Verifier will also check the validity of E by checking the valid period mentioned in the warrant. If the period has expired, E is invalid and can not be used for verification.

2) The verifier V also receives $(s_i \| h_i)^{d_i} mod N_i (i\epsilon T)$. He retrieves s_i by decrypting with the public key e_i of S_i and verifies partial signatures as

$$Dlog_m s_i = \sigma_i \tag{17}$$

$$= Dlog_g(g^{s_i})$$
$$= Dlog_g(h_i)$$

3) After verifying partial signature, V will verify proxy Signature S using E. Finally, the message m is retrieved as

$$S^E mod N = m^{DE} mod N = m \tag{18}$$

V. SECURITY ANALYSIS

In this section, it is shown that the proposed scheme satisfies all security requirements of threshold proxy signature scheme.

1) **Secrecy:** In Hwang's scheme, proxy signature key D is computed with the original signer's private key d_0, as

$$D = d_0^w mod \phi(N_0)$$

which can be cooperatively computed by proxy signers. Wang explained that knowing D and $(DE-1)$, it is easy to calculate d_0 with $d_0 e_0 = 1 mod \phi(N_0)$. In our scheme, proxy signature key D is computed with a new randomly selected number b by S_0 as follows.

$$D = b^w mod \phi(N)$$

Original signer's private key d_0 remains secret with original signer and it is not shared with others in any phase in the proposed scheme.

2) **Proxy Protected:** Proxy protection requires that only a designated proxy signer S_i can generate his partial proxy signature s_i. However, in Hwang's scheme it was found that partial proxy signature key k_i was just a share of D generated by the original signer S_0 and it not derived from private key d_i of proxy signer S_i. Hence, k_i is known to original signer. Therefore, he can compute s_i which is partial signature of i^{th} proxy signer. In Hwang's scheme, original signer needs to be trusted one for sending signing key to proxy signer S_i. The proposed scheme does not require a share of D from trusted original signer.
In the proposed scheme, no one else can generate valid partial proxy signature s_i because it is computed using k_i as $s_i = m^{L_i * k_i} mod N$, which remains secret with respective proxy signer S_i. In our scheme partial proxy signature key k_i is derived by i^{th} proxy signer and k_i remains unknown to original signer. The proposed scheme require $n(n-1)+n$ additional transmission for generation of k_i. In Hwang's scheme, n transmission are required for generation of k_i. Hence, original signer can not forge the partial proxy signature of $S_i$in our scheme.

3) **Unforgeability:** A valid proxy signature is generated by the cooperation of t or more proxy signers. A non designated third party can not forge proxy signature S as he can not derive D. In the Hwang's scheme, original signer S_0 (who knows D), can forge a signature on message m as $S = m^D mod N$. Verifier will accept a proxy signature S just by checking whether it satisfies the equation $S^E mod N = m^{DE} mod N = m$ or not.
In the proposed scheme, the verifier V needs s_i for verification of the partial proxy signature of the corresponding proxy signer at verification stage.

ISBN: 978-1-61208-427-5

Hence, scheme collects all partial proxy signature s_i at verification stage from a combiner. The combiner need not be trusted party. The verifier at later stage can find out if the combiner has modified or delete any partial signature. Original signer can not compute partial signature s_i because he does not know k_i to calculate $s_i = m^{k_i*L_i}$, which is required at verification stage.

4) **Non Repudiation:** Each proxy's partial signature s_i is encrypted by its private key d_i which is computed by proxy signer S_i. Hence he can not deny the existence of proxy signature on signed message. Original signer can not deny delegating the signing right as he publishes warrant w in the form $(w\|E\|a|\psi)^{d_0}modN_0$ which is encrypted using its private key d_0.

5) **Known Signer:** For internal auditing, combiner need not be trusted in the proposed scheme. It combines partial signatures of proxy signers. If combiner tries to modify or delete s_i, it will be detected by verifer at verification phase, as it is encrypted with the proxy's private key.

6) **Time Constraint:** Original signer publishes warrant w with proxy verification key E in the form $(w\|E\|a|\psi)^{d_0}modN_0$. Verifer will check the validity of E in the delegation period provided in warrant w to check whether E is the valid proxy verification key.

VI. PERFORMANCE COMPARISON

The performance comparison of the proposed scheme with the Hwang's scheme is shown in Table II. It can be noted that proposed scheme has merits of proxy protection since only a designated proxy signer S_i can generate his partial proxy signature s_i; non designated third party can not forge proxy signature This also achievers non-repudiation. In addition, the proposed scheme does not require trusted original signer and trusted combiner. However, the proposed scheme requires additional initialization phase. Thus, it can be seen that the proposed scheme has numerous cryptoanalytical merits as compared to that of Hwang's method.

TABLE II. COMPARISION OF PROPOSED SCHEME WITH HWANG'S SCHEME

S.No	Security Requirement	Hwang's Scheme	Our Scheme
1	Secrecy	No	Yes
2	Proxy Protected	No	Yes
3	Unforgeabilty	No	Yes
4	Non Repudiation	No	Yes
5	Time Constraint	Yes	Yes
6	Known Signer	Yes	Yes
7	Trusted Combiner	Yes	No
8	Secure Channel	No	No

VII. CONCLUSION

In this paper, we have proposed a new threshold proxy signature scheme based on RSA, which uses the Boneh and Franklin protocol and Hguyens scheme for generation of shared RSA keys. It has been shown that the proposed scheme significantly removes the weaknesses found by Wang in the Hwang's scheme which is also based on RSA. The marginal increase in computational complexity in the proposed scheme is compensated by reduction in the cost of trusted combiner required during signature generation phase. Therefore, the proposed scheme can be used as potential candidate for implimentation in proxy signature applications.

REFERENCES

[1] M. Mambo, K. Usuda and E. Okamoto , " Proxy signature: delegation of power to sign message" IEICE Transaction on Fundamental, vol E79-A, pp.1338-1353, 1996.

[2] S. Kim, S. Part and D. Won, " Proxy signature, revisited,"International Conference on Information and Communication Security ICICS'97, China Beijing, pp. 223-232, 1997.

[3] T. Elgamal, " A public key cryptosystem and signature scheme based on discrete logarithms," IEEE Transaction on Information Theory, Vo1.31, pp. 469-472, 1985.

[4] C. Schnoor, " Efficient signature generation by smart card," Cryptography, vol.4, pp. 161-174, 1991.

[5] K. Zang, " Threshold proxy signature schemes", Information Security Workshop ISW97, California, USA, pp.282-290, 1997.

[6] H. M. Sun, " Threshold proxy signatures," Computer and Digital Techniques, vol. 146, pp.259-263, IEE preceedings, 1999.

[7] M. S. Hwang, I. C. Lin and J. L. Lu," A secure nonrepudiable threshold proxy signature scheme with known signers", International Journal Informatica, Vol. 11, pp. 1-8, 2000.

[8] Z. Shao, " Improved threshold proxy signature schemes", Computer Standards and Interfaces, vol. 27, pp. 53-59 , 2004.

[9] M. S. Hwang, J. L. Lu and I. C. Lin, " A practical (t, n) threshold proxy signature scheme based on the RSA cryptosystem", IEEE Transactions on Knowledge and Data Engineering, Vol. 15, pp.1552-1560. 2003.

[10] G. Wang, F. Bao, J. Zhou and R. H. Deng," Comments on a practical (t, n) threshold proxy signature scheme based on the RSA cryptosystem," IEEE Transactions on Knowledge and Data Engineering, Vol. 16, pp.1309-1311, 2004.

[11] Y. J. Geng, H. Tian and F. Hong, " A modified and practical threshold proxy signature scheme based on RSA", *Advance Communication Technology, 9th conference* Vol. 3, pp. 1958-1960, 2014.

[12] Y. Zhang, D. W. Yue and H. Zhang, " An improved (t, n) threshold proxy signature scheme with fault tolerance based on RSA", JICIC, Vol 6, pp. 3205-3218 2010.

[13] Z. W. Tan and Z. J. Liu, " Cryptanalysis and Improvement on a Threshold Proxy Signature Scheme" Journal of Information Science and Engineering, vol. 25, pp. 619-631 2009.

[14] Samaneh Mashhadi, "A Novel Non-repudiable Threshold Proxy Signature Scheme with Known Signers", International Journal of Network Security, Vol.15, No.4, pp.274-279, July 2013.

[15] Raman Kumar, Harsh Kumar Verma and Renu Dhir, " Analysis and Design of Protocol for Enhanced Threshold Proxy Signature Scheme Based on RSA for Known Signers" Wireless Personal Communications, vol. 80 pp. 1281-1345. 2014.

[16] R.L. Rivest, A. Shamir and L.M. Adleman," A method for obtaining digital signatures and public-key cryptography" Communication of the ACM, Vol. 21, pp. 120-126, 1978.

[17] D. Boneh and Matthew Franklin. ,"Efficient generation of shared RSA keys,", Advances in Cryptography-CRYPTO '97, Springer- Verlag 1233 pp: 425-439, 1997.

[18] A. Shamir, "How to share a secret?" Communication of the ACM, Vol. 22, pp. 612-613, 1979.

[19] Ben-Or, M. Goldwasser and Wigderson , " Completeness theorems for noncryptographic fault-tolerant distributed computation," Proceeding of the 20th Annual ACM Symposium on Theory of Computing Chicago, I11, May 2-4, Newyork, pp. 1-10, 1998.

[20] H. L. Hguyen," RSA threshold cryptography" Dept. of Computer Science, University of Bristol, UK, 2005

[21] N. Koblitz, " A Course in Number Theory and Cryptography", Springer-Verlag, 1994.