Journal of Basic and Applied Engineering Research
Print ISSN: 2350-0077; Online ISSN: 2350-0255; Volume 1, Number 6; October, 2014 pp. 5-9

http://www.krishisanskriti.org/jbaer.html

A Study of Some Identity Based Proxy Signature Schemes

Vandani Verma

Amity Institute of Applied Sciences
Amity University -125, Noida, Uttar Pradesh - India

Abstract: **In a proxy signature scheme the original signer gives his signing rights to another person known as the proxy signer to generate a valid signature on behalf of him in his absence. Nominative proxy signature is a type of proxy signature in which the proxy signer generates a valid nominative signature on the original signer's behalf and only the nominee can check and if required proves its validity to another person. On the other hand in Self proxy signature the original signer provides himself with certain signing powers thus preventing the continuous exposure of his permanent private key. In this paper we propose a proxy signature scheme based on Bin Wang scheme and then construct nominative and self proxy scheme based on the proposed proxy signature scheme and check the computational effort required in constructing these schemes with Bing Wang.**

1. INTRODUCTION

ID based cryptography was proposed by Shamir in 1996 [8] in which the user's public key is developed from some public information related to the identity of the user such as his phone number, name, email-id, address etc. The corresponding secret key is generated by trusted third party called **Key generating center (KGC)** or **private key generator (PKG).** KGC uses his master key to generate secret key of the users and sends to them via a secure channel and keeps his master key secret. There is a certificate associated to its public value or identity. This certificate is of **$Cert_A$ (I_A|| μ_A ||P||S_{CA} (I_A|| μ ||P)** Where I_A is the Identity of user A (name, phone number etc.), $\mu_A = xP$, A chooses a secret value x and, makes x. P public, where P is the generator of additive group G_1, S_{CA} is the Signature of certifying authority on (I_A|| μ ||P), P is the public value as it is the generator of G_1. I_A and μ_A are concatenated.

The notion of digital signature called proxy signature was introduced by Mambo et al in 1996 [5]. A proxy signature scheme allows one entity called original signer to delegate his signing rights to one more entity called proxy signer. So proxy signer has the delegated power to sign messages in favor of the original signer. However the two signatures vary from each other. A verifier can easily test the authenticity and integrity of the proxy signature and get convinced of the original signer's agreement on the signed message.

Delegation of signing powers to the proxy signer can be classified as:-

- Full delegation: The original signer gives his secret signing key to the proxy signer as the proxy signing key. Thus, for a given message, signatures created between the original signer and the proxy signer is identical.
- Partial delegation: proxy signing key is obtained from the original signer's secret key. Also, it is infeasible for the proxy signer to figure and derive the original signer's private key. Moreover, the messages that a proxy signer can sign are not limited.
- Delegation by warrant: An original signer gives the proxy signer a particular message called warrant. The warrant authorizes that a proxy signer is valid and contains signer's identity, delegation span and the variety of messages on which proxy signer can sign.

On the basis of protection the proxy signature is further classified as:-

- Unprotected proxy signature: A proxy signature is developed by both the proxy signer and the original signer. In unprotected the verifier is not able to differentiate the identity of a signer.
- Protected proxy signature: It is developed by the proxy signature key of the original signer along with the private key of the proxy signer. Afterwards, a verifier confirms a proxy signature with the public keys of original signer and a proxy signer both.

H.-U. Park and I.-Y Lee [6] was the first to combine the idea of nominative signatures and proxy signature in 2001. In a nominative proxy signature scheme, an original singer gives his signing power to a proxy signer, who produces a nominative signature on behalf of the original signer. In this signature scheme, the nominee can only check the signature and if required, only the nominee can justify its validity to the third party.

Oded Goldreich et al [2] in 1998 introduced delegation schemes where a user provides certain rights to him. Self Proxy Signature (SPS) is a type of proxy signature in which an original signer delegates the signing rights to himself (Self Delegation), there by producing temporary public and secret key pairs for oneself. Thus, in SPS the user can avoid the risk of his secret key from continual use. ID based self proxy signature was proposed by S. Selvi [7] in 2010.

Rest of the paper is organized as follows: section 2 presents the model of the proposed proxy signature schemes, section 2 presents the proxy signature scheme, nominative proxy signature scheme and self proxy signature scheme based on Bin Wang [10] scheme, section 4 presents the results and discussions and finally conclude in section 5.

2. MODEL OF THE PROPOSED PROXY SIGNATURE SCHEMES

We discuss the models of our proposed schemes in the following i.e. what we are doing in each phase, which are basically the basic requirements of a digital signature scheme.

Model of Proxy Signature Scheme

- **Setup: -** This phase is run by key generating center (KGC) to find the public parameters **params as** $\langle$ (G_1,+), H_1, H_2, (G_2,
- mpk, e, q, P $\rangle$ where $\langle$ G_1, + $\rangle$ is a cyclic additive group having P as the generator, with an order of q which is a large prime number, $\langle$ G_2,
- $\rangle$ is a multiplicative cyclic group of the order q, and e: $G_1 \times G_1 \rightarrow G_2$ is a bilinear map. The public information that it provides is the master public key P_{pub} = sP, where 's' is the secret key of the key generating center (msk) and 'P' is G_1's generator respectively, H_1, H_2 are the hash functions being used by the signature scheme.
- **Key Generation: -** This is performed by KGC and is performed minimum one time for every user when they have registered with KGC. In this it takes as input the master secret key msk and the identity Q_A of user A and corresponding to the identity Q_A it computes the secret/private key S_A i.e. It takes as input ID_A and then computes Q_A=H_1 (ID_A) and S_A = s. Q_A. Then KGC sends S_A as the secret key to **A** through a protected medium. The correctness can be checked by the user by verifying e (P, S_A) = e (P_{pub} , Q_A).
- **Proxy Warrant Generation: -** The user **A** executes this algorithm, in this it takes as input the params, the user's identity Q_A, the user's secret key S_A, the message warrant m_w and a m the message and outputs a valid signature σ_A on message warrant m_w where the message warrant can be verified by anyone.
- **Proxy Warrant Verification: -** A verifier executes this algorithm to check the effectiveness of the message warrant m_w. It takes as input params, the identity Q_A of the signer, the message warrant m_w and the signature σ_A on m_w and computes the corresponding hash functions associated with the signature. If the signature σ_A on m_w is valid then the algorithm returns true or else it returns false.
- **Proxy Key Generation: -** This phase is executed by the user **B** in this it takes the signature or some public information or any random number or at times his secret key also in order to generate a valid proxy key. For providing the power of signing to B a proxy/delegated signer, **A** the real signer creates a warrant m_w containing the actual and the proxy signer's identities, the assignment period, message type on which the delegated signer can sign, etc.
- **Proxy Signature Generation: -** User **B** executes this algorithm to create a signature σ_B on message m taking input as the public parameters params, the proxy key generated by **B** and the message which is to be signed i.e. m.
- **Proxy Signature Verification: -** In this process the inputs are params, user's identity Q_A and Q_B, signature generated by **B** on m. It is run by any verifier wanting to check the trueness of σ_B on message m for this he should check whether the warrant is invalid; the verifier rejects the signature σ_B on m_w if it is invalid and if the signature σ_B on m is true the result is legal.

2.2 Model of Nominative Proxy Signature Scheme

In this method setup, key generation, proxy warrant generation, proxy warrant verification and proxy key generation are same as in proxy signatures except for the following given phases:

- **Nominative Proxy Signature Generation: -** User **B** executes this algorithm to generate a signature σ_B on m taking input as the public parameters params; the proxy key generated by **B**, the public parameter of **C** i.e. its hash function and the message to be signed i.e. m.
- **Nominative Proxy Signature Verification: -** This phase is executed by **C**; the inputs here are params, user's identity Q_A, Q_B and Q_C, the signature generated by **B** on message m. Only **C** can check σ_B's trueness on m. The result is valid if σ_B is a legal signature on m or else the result is false. For this **C** checks the validity of σ_B on m for this he should check whether the warrant is invalid;

the verifier rejects the signature σ_B on m_w if it's invalid and if σ_B is a legal signature on m then the result is true.

2.3 Model of Self Proxy Signature Scheme

- **Setup and Key Generation:** - Same as in proxy signatures
- **Temporary Key Generation: -** User **A** for different time creates a temporary secret/public key pairs. This phase as input takes params and creates a non-permanent key pair which is private and public.
- **Self Proxy Warrant Generation: -** The user **A** executes this algorithm, in this it takes as input the params, the user's identity Q_A, the user's secret key S_A, the message warrant m_w and a message m, the temporary secret key and outputs a standard signature σ_A on the message warrant m_w .
- **Self Proxy Warrant Verification:** - same as in proxy signatures
- **Self Proxy Signature Generation: -** In this algorithm we take as input params, the non-permanent secret key of proxy U_A, original message i.e. m which is to be signed. The signer **A** runs this phase to produce signature σ on message m utilizing his non-permanent secret key.
- **Self Proxy Signature Verification: -** The inputs here are params, user's identity Q_A, non-permanent public key corresponding to **A** and σ_A the signature on m. Any verifier wishing to check σ_A trueness on m runs this phase ; for this he should check whether the warrant is invalid; the verifier rejects the signature σ_B on m_w if its invalid and if σ_B is a legal signature on m then the result is true.

3. CONSTRUCTION OF THE PROPOSED SCHEMES

3.1 Construction of Proxy Signature Scheme Based on Bin Wang [10] Scheme

- **Setup:-** We take k as the security parameter of the system and take as input. and $\langle G_1, + \rangle$ is a cyclic additive group having P as the generator, with an order of large prime q, $\langle G_2, \bullet \rangle$ is a cyclic multiplicative group also of the same order, and let $e : G_1 \times G_1 \rightarrow G_2$ be a bilinear map. The key generating center (KGC) performs the following operations:
  Picks a random number $s \in Zq^*$ and sets the master or secret key pair $\langle P_{pub}, s \rangle$ Selects two secure one way hash functions H_1, H_2 defined as follows:-H_1: $\{0, 1\}^* \rightarrow G_1$, H_2: $\{0, 1\}^* \times G_1 \rightarrow Z_q^*$,Also $P_{pub} = s. P$, where P is the generator of G_1 and s is the secret key of KGC. Fixes the parameters params as $\langle (G_1, +), (G_2, \bullet), P_{pub}, e, P, H_1, H_2, q \rangle$
- **User key generation: -** This phase takes ID_A as input and then KGC computes $Q_A=H_1 (ID_A)$ and $S_A = s.Q_A$. KGC sends S_A as the secret key to **A** via a secure channel. The user can check the correctness by $e (S_A, P) = e (Q_A, P_{pub})$
- **Proxy Warrant Generation: -** The user **A** Computes $U_1 = H_2 (Q_A, m, m_w)$ and $U_2= U_1.S_A$ in order to sign a message m. The signature on m is $\sigma_A = \langle U_2 \rangle$ and sends to **B**,.
- **Proxy Warrant Verification: -** User **B** computes $H_2 (Q_A, m, m_w)$ and verifies

$$
\begin{aligned}
e (P, U_2) &= e (P, U_1 S_A) \\
&= e (P, U_1.s.Q_A) \\
&= e (P_{pub}, U_1.Q_A).
\end{aligned}
$$

If the signature σ_A on m_w is valid then the algorithm returns true or else it returns false

- **Proxy Key Generation: -** The user **B** takes the signature σ_A send by **A** and his secret key generated by KGC i.e. S_B to compute a valid proxy key. Hence, User B computes $V_1 = U_2+S_B$.
- **Proxy Signature Generation: -** User **B** computes $W_1 = H_2 (Q_A, Q_B, m, m_w). P$ and $W_2 = W_1+ V_1$

  The proxy signature on m is $\sigma_B = \langle W_2 \rangle$.

- **Proxy Signature Verification: -** The verifier gets Q_B, Q_A from m_w and then computes $U_1= H_2 (Q_A, m, m_w)$ and checks whether $e (P, W_2) = e (P, W_1+ V_1)$
- **Correctness: -**

$$
\begin{aligned}
e (P, W_2) &= e (P, W_1+ V_1) \\
&= e (P, W_1). e (P, V_1) \\
&= e (P, W_1). e (P, U_2 + S_B) \\
&= e (P, W_1). e (P, U_1. S_A + S_B) \\
&= e (P, W_1). e (P, s. (U_1. Q_A + Q_B)) \\
&= e (P, W_1). e (P_{pub}, U_1 Q_A + Q_B).
\end{aligned}
$$

If σ_B is a legal signature on m then the result is valid or else it is false.

3.2 Construction of Nominative Proxy Signature Scheme Based on the Proposed Scheme

To propose the nominative proxy signature scheme we have done necessary changes only in the nominative proxy signature generation and nominative proxy signature verification phase rest of the phases are same as in 3.1

- **Nominative Proxy Signature Generation: -** User B selects any number $k \in Z_q^*$ and computes $W_1 = Q_C.k$, $W_2 = H_2(Q_A, Q_B, Q_C, m, m_w).P_{pub}$, $W_3 = k^{-1} (W_2+ V_1)$, the proxy signature on m is $\sigma_B = \langle W_1, W_3 \rangle$ and send it to **C** through a protected medium.

- **Nominative Proxy Signature Verification: -** C on receiving the σ_B computes the hash functions $H_2(Q_A, Q_B, Q_C\ m, m_w)$ and $U_1 = H_2(Q_A, m, m_w)$.

  Checks whether $e(W_1, W_3) = e(W_1, k^{-1}(W_2 + V_1))$

- **Correctness: -** Only verifier C can check

  $e(W_1, W_3)$
  $= e(W_1, k^{-1}(W_2 + V_1))$
  $= e(K.\ Q_C, k^{-1}(W_2 + V_1))$
  $= e(Q_C, (W_2 + V_1))$
  $= e(Q_C, W_2).\ e(Q_C, V_1)$
  $= e(Q_C, H_2(Q_A, Q_B, Q_C, m, m_w).\ P_{pub}).\ e(Q_C, U_1.S_A + S_B)$
  $= e(S_C, H_2(Q_A, Q_B, Q_C, m, m_w).\ P + U_{1}.Q_A + Q_B)$

  If σ_B is a legal signature on m then the result is valid or else it is false.

3.3 Construction of Self proxy signature scheme based on the proposed scheme

- **Setup: -** Same as 3.1
- **User Key Generation: -** Same as 3.1
- **Temporary Key Generation: -** User **A** for different times creates a temporary secret/public key pairs. Input here is params and creates a non-permanent key pair which is private and public for signing m the message, now **A** Computes $U_1 = H_2(Q_A, m, m_w)$ and $U_2 = U_1.S_A$. The temporary key is $\langle U_2 \rangle$ now user **A** uses this key for performing various tasks.
- **Self Proxy Warrant Generation: -** User A selects a random number $k_1 \in Z_q^*$ and computes $V_1 = k_1.\ U_2$ and $V_2 = k_1.\ U_1.\ Q_A$ and sends the signature as $\sigma_A = \langle V_1, V_2 \rangle$, the message warrant can be verified by anyone.
- **Self Proxy Warrant Verification:** To verify the warrant A computes $H_2(Q_A, m, m_w)$ checks if
  $e(P, V_1)$
  $= e(P, k_1.U_2)$
  $= e(P, k_1.U_1.\ S_A)$
  $= e(P_{pub}, V_2)$
- **Self Proxy Signature Generation: -** In this algorithm user **A** selects a random number $k_2 \in Z_q^*$ and computes $W_1 = k_2.\ U_2$, $W_2 = k_1.U_1.Q_A$ and $W_3 = k_2^{-1}(W_1 + P)$. The signature on m is $\sigma_A' = \langle W_3, W_2 \rangle$
- **Self Proxy Signature Verification: -** to verify the self proxy signatures any verifier can compute $H_2(Q_A, m, m_w)$ and check $e(P, W_2) = e(Ppub, W_2).\ e(P, k_2^{-1}.P)$
- **Correctness: -** Any verifier checks,

$e(P, W_2)$
$= e(P, k_2^{-1}.(W_1 + P))$
$= e(P, k_2^{-1}.W_1)\ e(P, k_2^{-1}.P)$
$= e(P, k_2^{-1}.\ k_2.U_2).\ e(P, k_2^{-1}.P)$
$= e(P, k_1\ U_1.S_A).\ e(P, k_2^{-1}.P)$
$= e(Ppub, W_2).\ e(P, k_2^{-1}.P)$

4. RESULTS AND DISCUSSIONS

In this section we compare our proposed proxy signatures scheme, nominative proxy signature scheme and self proxy signature scheme with Bin Wang scheme on the basis of the computational aspects such as hash function, bilinear pairing, multiplier, inverse and exponentiation in the Proxy Signature Generation and Proxy Signature Verification phase.

Table 1: Proxy signature generation (PSG) comparison table of proposed scheme and Bin Wang scheme

Schemes → / operations ↓	Bin Wang	Proposed Proxy Scheme	Proposed Nominative Proxy Scheme	Proposed Self Proxy Scheme
Hash	2	2	2	2
Pairing	3	2	2	2
Exponential	0	0	0	0
Multiplier	$4\lvert Z_q \rvert$	$2\lvert Z_q \rvert$	$4\lvert Z_q \rvert$	$4\lvert Z_q \rvert$
Inverse	0	0	$\lvert Z_q \rvert$	$\lvert Z_q \rvert$

Table 2: proxy signature verification (PSV) comparison table of proposed scheme and Bin Wang scheme

Schemes → / operations ↓	Bin Wang	Proposed Proxy Scheme	Proposed Nominative Proxy Scheme	Proposed Self Proxy Scheme
Hash	2	2	2	1
Pairing	4	3	2	3
Exponential	0	0	0	0
Multiplier	$4\lvert G_2 \rvert$	$3\lvert G_2 \rvert$	$2\lvert G_2 \rvert\lvert$	$3\lvert G_2 \rvert\lvert$
Inverse	0	0	$\lvert Z_q \rvert$	$\lvert Z_q \rvert$

Table 3: Total number of computations used in proposed scheme and Bin Wang scheme in PSG and PSV

Phases	Bin Wang	Proposed Proxy Scheme	Proposed Nominative Proxy Scheme	Proposed Self Proxy Scheme
PSG	2H + 3P + 4$\lvert Z_q \rvert$	2H + 2P + 2$\lvert Z_q \rvert$	2H + 3P + 4$\lvert Z_q \rvert$	2H + 3P + 4$\lvert Z_q \rvert$
PSV	2H + 4P + 4$\lvert G_2 \rvert$	2H + 3P + 3$\lvert G_2 \rvert$	2H + 2P + 2$\lvert G_2 \rvert$ + $\lvert Z_q \rvert$	1H + 3P + 3$\lvert G_2 \rvert$ + $\lvert Z_q \rvert$
TOTAL (PSG+PSV)	4H + 7P +4$\lvert Z_q \rvert$ +4$\lvert G_2 \rvert$	4H + 5P +2$\lvert Z_q \rvert$ +3$\lvert G_2 \rvert$	4H + 5P + 5$\lvert Z_q \rvert$ +2$\lvert G_2 \rvert$	3H + 6P +5$\lvert Z_q \rvert$ +3$\lvert G_2 \rvert$

H = Hash, M = Multiplication, E = Exponential, P = Pairing, I = Inverse.

From the comparative table we came to the conclusion that our proxy scheme is of less computation as compared to Bin Wang scheme as our scheme has two less pairing and one less multiplier in G_2 and two less multipliers in Zq than Bin Wang's scheme i.e. our pairing is 5 and multiplier is 3|G_2|+2|Zq| compared to Bin Wang's which is 7 and 4|G_2|+|4|Zq| respectively. And in case of nominative proxy signature scheme we see that by taking four more multipliers of Zq and one less multiplier of G_2 we get one less pairing from our proposed proxy scheme and can convert it into a nominative proxy signature scheme i.e. 4 pairings and 2|G_2|+6|Zq| multiplier compared to 5 pairing and 3|G_2|+2|Zq| multiplier of nominative and proxy scheme respectively. And from the table we come to the conclusion that from our proposed Proxy scheme we get one less hash function in the self proxy scheme but we had to take four more multipliers in |Zq| in comparison to proxy where the hash function is 4 and multiplier is 3|G_2|+2|Zq| and can change a proxy scheme to self proxy scheme.

5. CONCLUSION

We have constructed the proxy protected schemes with partial delegation by warrant based on Bin Wang scheme and tried to compare the schemes with Bin Wang and with our proposed proxy schemes in computational aspects in terms of Hash function, bilinear pairing, exponential, multiplier and inverse. However, the security aspects of the schemes are to be done. We will study our schemes in different security model discussed in literature. The schemes we had proposed have numerous applications in electronic voting [3], distributes computing, electronic commerce [1], [4], application for mobile agent, etc. In mobile communication [9] there method is useful as it gives the customers invisibility by using nominative signature thus reduces mobile user's estimation value with proxy/delegated signature. In today's world, various on-line works like home transactions, internet premiums, net banking etc. depend on public key cryptography. This thus leads to hijacking the private key or password by duplicating or reproducing them, so in such a situation self proxy scheme enables the user to create a temporary private key in order to avoid his permanent private key from exposure.

REFERENCES

[1] J.-Z Dai, X.-H. Yang, and J.-X. Dong, "Designated-receiver proxy signature scheme for electronic commerce," Proc. of IEEE International Conference on Systems, Man and Cybernetics, Vol. 1, pp.384-389. Oct.5-8, 2003. IEEE, 2003.

[2] Oded Goldreich, Birgit Pfitzmann, and Ronald L. Rivest. Self delegation with controlled propagation- or- what if you loose your laptop. In Advances incryptogy-CRYPTO 1998, volume 1462 of Lecture Notes in Computer Science, pages 153-168. Springer, 1998.

[3] B. Lee, H. Kim, K. Kim, "Strong proxy signature and its applications", in Proceedings of ICICSí97, International Conference on Information and Communication Security, pp.603-608, 2001.

[4] Younho Lee, Heeyoul Kim, Yongsu Park, and Hyunsoo Yoon. A new proxy signature scheme providing self-delegation. In Information Security and Cryptology - ICISC 2006, volume 4296 of Lecture Notes in Computer Science, pages 328–342. Springer, 2006.

[5] M. Mambo, K. Usuda, E. Okamoto, "Proxy signatures for delegating signing operation", in Proceedings of 3rd ACM Conference on Computer and Communications Security , ACM Press, pp.48-57, 1996.

[6] H.-U. Park and I.-Y. Lee, "A digital nominative proxy signature scheme for Mobile communication," Proc. of ICICS 2001, International Conference on Information and Communications Security, Springer-Verlag, Lecture Notes in Computer Science 2229, pp.451-455, 2001.

[7] S. Sharmila Deva Selvi, S. Sree Vivek, S.Gopinath, C. Pandu Rangan, "IdentityBased Self Delegated Signature - Self Proxy Signatures", https://eprint.iacr.org/2010/359.

[8] Adi Shamir. Identity-based cryptosystems and signature schemes. In CRYPTO- 1984, pages 47–53. Springer, 1984.

[9] S.-H Seo and S.-H. Lee, "New nominative proxy signature scheme for mobile communication," Proc. of SPI'2003, Security and Protection of Information, ISBN: 80-85960-50-8, pp.149-154, 2003.

[10] Bin Wang. "A new identity based proxy signature scheme", https://eprint.iacr.org/2008/323.