Proxy Blind Signature Scheme

Sunder Lal* **and Amit Kumar Awasthi**

Institute of Basic Sciences,
Khandari, Agra – 282002(UP) – INDIA
Hindustan College of Science and Technology,
Farah Mathura – 281122(UP) – INDIA
E-mail: *awasthi_hcst@yahoo.com*

Abstract:

Blind signature is the concept to ensure anonymity of e-coins. Untracebility and unlinkability are two main properties of real coins, which require mimicking electronically. Whenever a user is permitted to spend an e-coin, he is in need to fulfill above requirements of blind signature. This paper proposes a proxy blind signature scheme with which a proxy is able to make proxy blind signature which verifier is able to verify in a way similar to proxy signature schemes.

1. Introduction

D. Chaum [1] introduced the concept of a blind signature scheme in1982. Using this scheme a user A can obtain the signature of B on any given message, without revealing any information about the message or its signature. Apart from unforgeability, the scheme ensures untracebility and unlinkability. A lot of work has been done in field of blind signature schemes since Chaum. [1-3, 8-10]

In production of coins, the user makes the bank blindly sign a coin using blind signature schemes. The user is in possession of a valid coin such that the bank itself cannot recognize nor link with the user. Whenever a user goes through a valid branch to withdraw a coin, he needs the branch to make proxy blind signature on behalf of the signee bank. This application leads to the need of proxy blind signature schemes.

In 1996 Mambo et al [5, 6] introduced the concept of 'proxy signature'. In this scheme an original signer delegates his signing authority to another (proxy) signer in such a way that the proxy signer can sign any message on behalf of the original signer and the verifier can verify and distinguish between normal (original) signature and proxy signature. He also elaborated the two types of scheme: proxy unprotected (proxy and original signer both can generate a valid *proxy* signature) and proxy protected (only proxy can generate a valid *proxy* signature). These schemes ensures among other things, non-repudiation and unforgeability.

Recently Tan et al. [7] introduced a proxy blind signature scheme, which ensures security properties of both the schemes, viz., the blind signature schemes and the proxy signature schemes. The scheme is based on Schnorr blind signature scheme. In this paper we introduce a new proxy blind signature scheme, which is based on Mambo et al., Our scheme is computationally more efficient than that of Tan et al. We also discuss a few attacks on the Tan et al scheme and show that these can be overcome using our proposed scheme.

* Research is partially supported by U. G. C. Grant No. 8 – 9/98 (SR-I)

2. Proposed Scheme

In the proposed scheme the system parameters and some notations are

p : a large prime number .

q : a large prime factor of $(p-1)$.

g : an element of Z_p^* of order q.

x_A : the secret key of the original signer A ,

y_A : the public key of the original signer A, where $y_A = g^{x_A} \bmod p$

$h(.)$: a secure one way hash function

2.1 *For* ***proxy unprotected*** *case our protocol runs as f*ollows:

2.1.1 Proxy phase

1. (*Proxy Generation*) The original signer A randomly chooses $k \in Z_q^*$, $k \neq 1$ and computes

$$r = g^k \bmod p,$$
$$s = x_A + k\,r \bmod q, \text{ and}$$
$$y_p = g^s \bmod p$$

2. (*Proxy Delivery*) The original signer sends (s, r) to a proxy signer B in a secure way and makes y_p public.

3. (*Proxy Verification*) After receiving the secret key (s, r) the proxy signer B checks the validity of the secret key with the following congruence

$$y_p = g^s = y_A\, r^r \bmod p$$

If (s, r) satisfies this congruence, he accepts it as a valid proxy, otherwise rejects it. In the later case he either requests for another key, or simply stops the protocol.

2.1.2 Signing phase

1. B chooses a random number $K \in Z_q^*$, $k \neq 1$, computes

$$R = g^K \bmod p$$

and sends it to the receiver C.

2. **(a)** C chooses randomly $\alpha, \beta \in Z_q^*$ and computes

$$r' = R\, g^{-\alpha} y_p^{-\beta} \bmod p$$

If $r' = 0$, he chooses another set of α and β; otherwise computes

$$e = h(r' \oplus m) \bmod q$$

and $e' = e + \beta \bmod q$

C sends e to B

3. After receiving e, B computes

$$s' = K - s\,e \bmod q$$

and sends it to C.

4. Now C computes

$$S_p = s' - \alpha \bmod q$$

The tuple (m, S_p, e') is the proxy blind signature.

2.1.3 Verification Phase

The verifier or recipient of the proxy blind signature computes

$$e'' = h(g^{S_P} y_p^{e'} \pmod p \oplus m) \bmod q$$

where y_p is the public value of step 1 in 2.1.1..

Here $e'' = e'$, if and only if the tuple (m, S_p, e') is a valid proxy signature.

2.2 *Proxy protected*

If we want only proxy signer to generate a valid proxy signature, we modify the proxy phase 2.1.1 of the previous protocol as follows:

Proxy phase

1. (*Proxy Generation*) The original signer A randomly chooses $k \in Z_q^*$, $k \neq 1$ and computes

$$r = g^k \bmod p,$$

$$\sigma = x_A + k\, r \bmod q, \text{ and}$$

$$y_p = g^{\sigma} y_B \bmod p,$$

where $y_B = g^{x_B} \bmod p$, is the public key of B.

2. (*Proxy Delivery*) The original signer sends (σ, r) to a proxy signer B in a secure way and makes y_p public.

3. (*Proxy verification and key alteration*)

After confirming the validity of the pair (σ, r) B alters the proxy key as

$$s = \sigma + x_B \bmod q.$$

The Signing Phase and *The Verification Phase* are same as in the proxy unprotected case.

3. Efficiency

In this section we show the efficiency of our scheme over that of Tan et al. Let E, M and I respectively denote the computational load for exponentiation, multiplication and inversion. Then following table shows the comparison of computational load of our scheme vs. Tan et al.

Scheme ↓	**Phase**			**Total**
	Proxy generation	*Signature generation*	*Signature verification*	
Tan et al	3E+2M	8E+7M+4I	3E+3M+I	14E+12M+5I
Proposed Scheme	4E+2M	3E+3M+2I	2E+M	9E+6M+2I

Each phase in our scheme has less computational load except in proxy generation phase, where it is one exponential computation more than tan et al. This computational load may be adjusted with compromise that some computational load in verification phase increases. In some applications digital information is signed once but verified more than once. In such situation the efficiency of our scheme increases with the number of times verification is done.
Further the total computation cost in our scheme is 9E+6M+2I as compared to Tan et al which is 14E+12M+5I. Thus, our scheme has computational advantage over that of Tan et al.

4. Security Analysis

(i) In signature verification phase we use different congruence to check the validity of the original signatures and the proxy signatures. So the *original* signature is *distinguishable* from the *proxy* signature.

(ii) To put a valid proxy signature s (in case proxy protected x_B too) is required. It is impossible to create a valid signature with out knowing x_B or s or both. Thus proxy signature cannot be forged. Furthermore, though original signer creates s, also have no knowledge about x_B in case of proxy protected. Thus the proxy signer cannot deny the proxy signature that he has created.

(iii) The public key y_p is computed from the original signer's public key y_A. thus the original signer cannot deny his agreement. Proxy signer's public key is also involved in the public key (in case proxy protected). Therefore the proxy signer can be identified from the signature.

Some security attacks that work in Tan's scheme, have also been removed in our scheme. These attacks are as follows: -

The verification equation in Tan et al's scheme is

$$e = h(g^S y_B^{-e} y_A^e u \,||\, m) \bmod q$$

which ensure the participation of both the signers A and B and hence the tuple (*m, u, s, e*) [3.4 of Tan et al.] is a valid proxy blind signature by B on behalf of A. Here involvement of both signers public key is the only way to recognize that it is a proxy blind signature of signer B for A. Here a forgery by R may be possible in Tan et al scheme. The receiver R may prove that (*m, u, s, e*) is a valid proxy blind signature of some other signer F although F might have not given his signing authority to any one. It may happen as follows
When the receiver R interacts with 'B'. he computes

$$u' = (r\ y_A^{\bar{r}})^{-e+b} y_F^{-e} \bmod q$$

instead of

$$u = (r y_A^{\bar{r}})^{-e+b} y_A^{-e} \bmod q$$

No other equation would be affected by this forgery. Now the receiver may prove that the tuple (*m, u', s, e*) is a valid proxy blind signature of signer F by the similar verification equation

$$e = h(g^S y_B^{-e} y_F^e u \,||\, m) \bmod q$$

which ensure the participation of the signer F in that blind signature.

In Second case, the receiver may prove that a signer D had produced a valid proxy blind signature on behalf of A during verification. For this he computes

$$u'' = (r \; y_A^{\;\bar{r}})^{-e+b} y_A^{\;-e} y_B^{\;-e} y_D^{\;e} \bmod q$$

and thus modification equation changes as

$$e = h(g^S y_D^{\;-e} y_A^{\;e} u \| m) \bmod q$$

which ensure that tuple (*m, u″, s, e*) is a valid proxy blind signature by the signer D on behalf of signer A, although A never delegated his proxy key to D.

To overcome these forgeries either freeness of the '*u*' should be restricted or it should be removed from the computation altogether. In our equation *u* does not appear and hence our scheme is secure for these type of forgeries.

5. Conclusion

In this paper we propose a proxy blind signature scheme with which a proxy user is able to make proxy blind signature and verifier may verify it very similar to proxy signature schemes. Our scheme is based on Mambo et al's protocols. Its computational load is less than that of a recent scheme by Tan et al. In this paper, we also had discussed some possible attacks on Tan's scheme and which our proposed scheme is free from.

6. References

1. D. Chaum. Blind signatures for untraceable payments. *Advances in Cryptology Crypto ' 82*, LNCS, pp. 199-203.
2. D. Chaum, Blind signature system, *Proceedings of Crypto'83*, Plenum, pp. 153.
3. D. Chaum, A. Fiat and M Naor, Untraceable electronic cash, *Proceedings of Crypto'88*, LNCS #403, Springer-Verlag, pp. 319-327.
4. M Mambo, K Usuda, E Okamoto, Proxy signature: Delegation of the power to sign messages, *IEICE Trans. Fundamentals*, E79 – A: 9, 1996, 1338 – 1353
5. M Mambo, K Usuda, E Okamoto, Proxy signatures for delegating signing operation, Proc. *3rd ACM Conference on Computer and Communication Security*, 1996, pp 48 – 57.
6. S. Kim, S. Park, D. Won, Proxy signatures: Revisited, *ICICS'97*, LNCS #1334, Springer– Verlag, pp 223 – 232
7. W B Lee, C Y Chang, Efficient Proxy–protected proxy signature scheme based on discrete logarithm, *Proceedings of 10th Conference on Information Security*, Hualien, Taiwan, ROC, 2000 pp 4-7.
8. Z Tan, Z Liu and C Tang, Digital proxy blind signature schemes based on DLP and ECDLP, MM Research Preprints, No. 21, December 2002, MMRC, AMSS, Academia, Sinica, Beijing pp 212–217
9. A. Lysyanskaya and Z. Ramzan. Group blind digital signatures: A scalable solution to electronic cash. *In Financial Cryptography (FC '98)*, LNCS #1465, Springer-Verlag, 1998, pp.184--197.
10. M Bellare, C Namprempre, D Pointcheval and M Semanko The One-More-RSA-Inversion Problems and the Security of Chaum's Blind Signature Scheme, Financial Cryptography '01, LNCS #2339, Springer-Verlag, 2002. pp 319-338