Controlled Delegation in e-cheques using Proxy Signatures

N.R.Sunitha
Dept. of CSE
SIT, Tumkur,
Karnataka, India.

B.B.Amberker
Dept. of CSE
NIT, Warangal,
Andra Pradesh, India.

Prashant Koulgi
Dept. of CSE
SIT, Tumkur,
Karnataka, India.

August 1, 2007

Abstract

A proxy signature scheme allows one user to delegate his/her signing capability to another user called a proxy signer in such a way that the latter can sign messages on behalf of the the former. After verification the verifier is convinced of the original signer's agreement on the signed message. We have come up with a scheme to control delegation of financial power to a proxy signer. The scheme satisfies the basic requirements of a secure proxy signature scheme. Using our scheme the proxy signer will be able to submit an e-cheque for only the amount he is entitled to by the original signer. Any cheating by the original signer or the proxy signer is identified by the verifier i.e. the bank. We have considered Forgery by the original signer, Impersonating and framing attack to prove the security of our scheme. Any discrete log based signature scheme can be used to sign the messages. Here we use the Digital Signature Algorithm(DSA).

Keywords : Digital Signature, Proxy Signature, Security, DSA.

1 Introduction

A proxy signature [9, 10] allows one user Alice, called the original signer, to delegate her signing capability to another user Bob, called the proxy signer. After that, the proxy signer Bob can sign messages on behalf of the original signer Alice. Upon receiving a proxy signature on some message, a verifier can validate its correctness by the given verification procedure. By this the verifier is convinced of the original signer's agreement on the signed message. Proxy signatures can be used in a number of applications like e-cash, electronic commerce, distributed shared object systems etc.

The basic working of most proxy signature schemes is as follows. The original signer Alice sends a specific message with its signature to the proxy signer Bob, who then uses this information to construct a proxy private key. With the proxy private key, Bob can generate proxy signatures by employing a specified standard signature scheme. When a proxy signature is given, a verifier first computes the proxy public key and then checks its validity according to the corresponding standard signature verification procedure.

Mambo, Usuda and Okamoto introduced the concept of proxy signatures and proposed several constructions in [9]. Based on the delegation type, they classified proxy signatures as full delegation, partial delegation and delegation by warrant schemes. In full delegation, Alice's private key is given to Bob so that Bob has the same signing capability as Alice. But such schemes are obviously impractical and insecure. In a partial delegation scheme, a proxy signer has a new key called proxy private key, which is different from Alice's private key. So, proxy signatures generated by using proxy private key are different from Alice'e standard signatures. However the proxy signer can sign any message of his choice i.e there is no limit on the range of messages he can sign. This limitation is eliminated in delegation by warrant schemes by adding a warrant that specifies what kind of messages are delegated and may contain the identities of Alice and Bob, the delegation period, etc.

According to whether the original signer knows the proxy private key, proxy signatures can be classified into proxy-unprotected and proxy-protected schemes. That is, in a proxy-protected scheme only the proxy signer can generate proxy signatures, while in proxy unprotected scheme either the proxy signer or the original signer can generate proxy signatures since both of them know the proxy private key. In many practical applications proxy-protected schemes are required to avoid potential disputes between the original signer and the proxy signer.

Any secure proxy signature scheme should satisfy the following five requirements :

1. Verifiability : From the proxy signature, a verifier can be convinced of the original signer's agreement on the signed message.

1541-7719/07 $25.00 © 2007 IEEE
DOI 10.1109/EDOC.2007.34

2. Strong unforgeability : Only the designated proxy signer can create a valid proxy signature on behalf of the original signer.

3. Strong Identifiability : Anyone can determine the identity of the corresponding proxy signer from the proxy signature.

4. Strong undeniability : Once a proxy signer creates a valid proxy signature on behalf of an original signer, he cannot repudiate the signature creation against anyone else.

5. Proxy signer's deviation : A proxy signer cannot create a valid proxy signature not detected as a proxy signature.

Followed by the first constructions given in [9, 10], a number of new schemes and improvements have been proposed [4, 14, 15, 8, 3, 11, 5, 6, 12, 7, 13, 2]; however, most of them do not fully meet the above listed security requirements. In [4], Kim, Park and Won introduced the concept of partial delegation by warrant, and proposed a threshold proxy signature, in which the original signing power is shared among a delegated group of n proxy signers such that only t or more of them can generate proxy signatures cooperatively. In [8], Lee et al. pointed out some weaknesses in Zhangs threshold proxy signatures [14, 15]. Later, some different opinions on their attacks are commented in [3]. In [5], Lee, Kim and Kim proposed non-designated proxy signature in which a warrant does not designate the identity of a proxy signer so any possible proxy signer can respond this delegation and become a proxy signer. Furthermore, their scheme is used to design secure mobile agents in electronic commerce setting [6]. One-time proxy signatures are studied in [1, 13]. In [7], Lee, Cheon, and Kim investigated whether a secure channel for delivery of a signed warrant is necessary in existing schemes. Their results show that if the secure channel is not provided, the MUO scheme [9] and the LKK scheme [5, 6] all are insecure. To remove the requirement of a secure channel and overcome some other weaknesses, they revised the MUO and LKK schemes. In contrast to the above mentioned schemes, which all are based on discrete logarithm cryptosystems, several RSA-based proxy signature schemes are proposed in [11, 6]. In particular, the OTO scheme [11] and the LKK RSA-based scheme [6] are proved as secure as RSA signatures in the sense of polynomial-time reducibility.

In section 2 for sake of completeness we describe the DSA algorithm. In section 3 we describe our proxy signature scheme. In section 4 we give the implementation model to control delegation of financial power in a Conference organising system. In section 5 we discuss the security of our system and in section 6 we conclude.

2 Digital Signature Algorithm

The Digital Signature Algorithm (DSA) is a United States Federal Government standard or FIPS (Federal Information Processing Standard) for digital signatures. It was proposed by the National Institute of Standards and Technology (NIST) in August 1991 for use in their Digital Signature Standard (DSS), specified in FIPS. This scheme is a digital signature scheme which is based on the difficulty of computing discrete logarithms

2.1 Key Generation:

- Choose a 160 bit prime q.

- Choose a L-bit prime p, such that $p = qz + 1$ for some integer z.

- Choose h, where $1 < h < p - 1$ such that $g = h^z \bmod p > 1$.

- Choose x where $0 < x < q$.

- Calculate $y = g^x \bmod p$.

- Public key is (p, q, g, y). Private key is x.

2.2 Signature Generation:

- Generate a random per message value k where $0 < k < q$

- Calculate $r = (g^k \bmod p) \bmod q$

- Calculate $s = (k^{-1}(SHA - 1(m) + x * r)) \bmod q$ where SHA-1(m) is the SHA-1 hash function applied to the message m

- The signature is (r, s)

2.3 Signature Verification:

- Calculate $w = s^{-1} \bmod q$.

- Calculate $u1 = (SHA - 1(m) * w) \bmod q$

- Calculate $u2 = r * w \bmod q$

- Calculate $v = ((g^{u1} * y^{u2}) \bmod p) \bmod q$

- The signature is valid if $v = r$

3 Our Scheme

3.1 Proxy key generation:

Let p, q be two large primes such that $q/(p-1)$ and $G_q =< g >$ is a q-order multiplicative subgroup of Z_p^* generated by an element $g \epsilon Z_p^*$.

The original signer is Alice with a certified key pair (x_A, y_A) where $y_A = g^{x_A} \bmod p$ and Bob is a proxy signer with a certified key pair (x_B, y_B) where $y_B = g^{x_B} \bmod p$.

Alice chooses a random number $k_A \epsilon Z_q^*$, computes

$$K = g^{k_A} \bmod p \tag{1}$$

$$S_A = k_A.y_B + x_A.h(ma) \tag{2}$$

where h is a collision resistant hash function and ma is the message. As both the secret key, x_A, of the original signer(Alice) and the public key, y_B, of the proxy signer(Bob) is used to calculate S_A, Alice cannot deny that Bob is the proxy signer.

Then the tuple (ma, K, S_A) is sent to the proxy signer Bob, who checks its validity by

$$g^{S_A} \equiv y_A^{h(ma)}.K^{y_B} \bmod p. \tag{3}$$

Notice that since

$$\begin{aligned} RHS &= y_A^{h(ma)}.K^{y_B} \bmod p \\ &= g^{x_A \cdot h(ma)}.g^{k_A \cdot y_B} \bmod p \\ &= g^{x_A \cdot h(ma) + k_A \cdot y_B} \bmod p \\ &= g^{S_A} \\ &= LHS \end{aligned}$$

the tuple (ma, K, S_A) sent by an honest signer will be accepted.

If the above verification is correct, Bob sets his proxy key pair (x_p, y_p) as follows :

$$x_p = S_A + x_B.y_A \bmod p \tag{4}$$

$$y_p = g^{x_p} \bmod p \tag{5}$$

3.2 Proxy signature generation:

With the proxy key pair (x_p, y_p), Bob can use any DLP based signature scheme to generate proxy signature on any message m. The resulting proxy signature is the tuple $(sign(m, x_p), K, m, y_A, y_B)$.

This signature also helps to identify the original signer and the proxy signer. Once the verification of this signature for a given message passes with the computation of proxy public key given by equation (6), the identity of the original signer and the proxy signer is confirmed. Thus the third requirement, Strong identifiability, of a secure proxy signature is satisfied.

We observe in equation(4) that the proxy private key x_p used to generate the signature is computed using the private key of the proxy signer and the public key of the original signer. Thus the proxy signer is creating a valid proxy signature on behalf of the original signer. He therefore cannot repudiate the signature against anyone else. Thus the fourth requirement, Strong undeniability, of a secure proxy signature is satisfied.

3.3 Proxy signature verification:

The verifier computes the proxy public key y_p as follows :

$$y_p = y_A^{h(m)}.K^{y_B}.y_B^{y_A} \bmod p \tag{6}$$

We observe that the above computation also corresponds to the same public key computed by the proxy signer i.e.

$$\begin{aligned} LHS &= g^{x_p} \bmod p \\ &= g^{S_A + x_B \cdot y_A} \bmod p \\ &= g^{x_A \cdot h(ma) + k_A \cdot y_B + x_B \cdot y_A} \bmod p \\ &= g^{x_A \cdot h(ma)}.g^{k_A \cdot y_B}.g^{x_B \cdot y_A} \bmod p \\ &= y_A^{h(ma)}.K^{y_B}.y_B^{y_A} \bmod p \\ &= RHS. \end{aligned}$$

Finally, the verifier checks whether $(sign(m, x_p))$ is a valid signature of message m with respect to the proxy public key y_p (given by equation(6)) in the corresponding DLP (Discrete Log Problem) based signature scheme. If this check passes, the verifier is convinced of the original signer's agreement on the signed message as the public key used to verify the signature is calculated using the public key and the security parameter K of original signer. Thus the first requirement, Verifiability, of a secure proxy signature is satisfied.

Also, the proxy signature is identified as a proxy signature and not as an ordinary signature as it is verified only by the proxy public key (y_p) and not by the public key of the proxy signer (y_B). Thus the fifth requirement, that a proxy signer cannot create a valid proxy signature not detected as a proxy signature, of a secure proxy signature is satisfied.

4 Implementation Details

We have considered the scenario of organising a conference. The chairman is given the financial power to distribute the funds to various committees. Generally the following methods are used:
On the requisition (based on budget) of the committee members

1. The chairman gives cheques in the name of the committee member for a specified amount.

2. The chairman transfers the amount to the account of committee member.

We have come up with a scheme in which the chairman can delegate his signing capability to the committee members who use proxy signatures to sign e-cheques. Here chairman has a controlled delegation i.e. the chairman decides for what amount each member is entitled to spend. The member can only draw the amount for which he is entitled to. The model is shown in figure (1).

The chairman requests the bank to open an account in the name of the conference and puts all the funds in that account. The chairman sends the signed security parameter K_i and the amount that each member is entitled to spend m_i to the bank. He also sends the signed triplet (SA_i, K_i, m_i) to each of the i^{th} member. K_i and SA_i are computed as follows :

$$K_i = g^{k_{Ai}} \bmod p \tag{7}$$

$$SA_i = k_{Ai}.y_i + x_A.h(m_i) \tag{8}$$

where $k_{Ai} \epsilon Z_q^*$. y_A is the public key and x_A is the private key of the of the chairman(original signer) . y_i is the public key and x_i is the private key of the i^{th} member.

Each member checks the validity of the information received from the chairman using the following equation

$$g^{SA_i} \equiv y_A^{h(m_i)}.K_i^{y_i} \bmod p. \tag{9}$$

Each member creates a proxy key pair using the following equations:

$$x_{pi} = SA_i + x_i.y_A \bmod p \tag{10}$$

$$y_{pi} = g^{x_{pi}} \tag{11}$$

and signs the e-cheque for the amount m_i using the proxy secret key x_{pi}. The signature scheme can be any discrete log based scheme. We have chosen the widely used DSA scheme. The proxy signature is $(sign(m_i, x_{pi}), m_i, K_i, y_A, y_B)$. $sign(m_i, x_{pi})$ is generated as discussed in section 2.2. The remaining parameters specified in the signature help the verifier(bank) to compute the proxy public key y_{pi}. In other words each of the i^{th} member requests the verifier to verify the signature using the proxy public key y_{pi} and not his public key y_i.

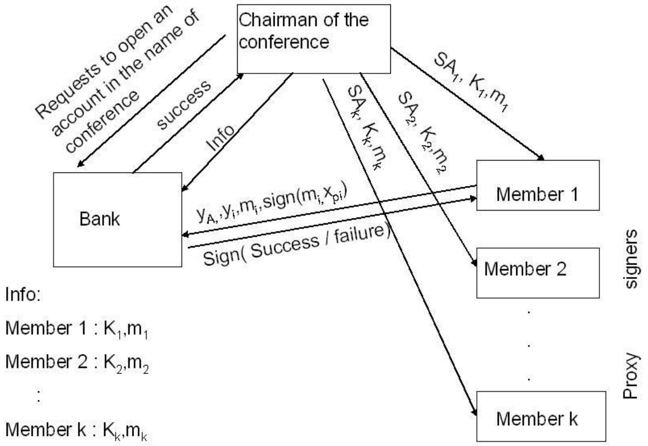
On receiving the proxy signature the bank computes the proxy public key using the following equation :

$$y_{pi} = y_A^{h(m_i)}.K_i^{y_i}.y_i^{y_A} \bmod p \tag{12}$$

The signature is verified using the proxy public key computed with the help of the verification equation (see section 2.3) of DSA scheme.

If there is change in m_i i.e. the amount for which the proxy signer is entitled to, or in any other parameters, the proxy public key computed by proxy signer will be different from that computed by the verifier. By this the verification equation of the DSA scheme will not hold good and the cheating is identified.

Figure 1: Model for e-cheque processing with proxy signers



5 Security of our scheme

1. Forgery by the Original Signer: The original signer can generate the proxy public key using equation (6). But he cannot generate the proxy private key as it is a discrete log based problem given by equation (5). Thus the original signer is unable to sign like the proxy signer. Therefore forgery by original signer is computationally not possible.

2. Impersonating attack: Let us assume that Bob is not designated as a proxy signer by the original signer Alice. Though Bob can generate a proxy key pair (x'_p, y'_p) with K' and m' satisfying equation (6) and sign a message on behalf of Alice, the verifier who also computes the proxy public key y_p using (K, m) sent by original signer will be able to identify that the proxy public key y_p is not equal to y'_p. By this the verification equation of the DLP based signature scheme fails. Thus Bob cannot become the proxy signer unless he is designated by the original signer Alice.

3. Framing attack: In this attack, a third party Charlie forges a proxy private key and then generates valid proxy signatures such that the verifier believes that these proxy signatures were signed by the proxy signer Bob on behalf of the original signer Alice. When such a proxy signature is presented, Alice cannot deny that she is the original signer of the proxy signer Bob. The result is that Alice and Bob will be framed.

   To accomplish this attack, Charlie needs to generate Bob's proxy key pair (x_p, y_p) with K and m satisfying equation (6). y_p is not publicly announced by the proxy signer, Bob, but instead computed by the verifier just before verification. Even if this key is made available, Charlie cannot generate the proxy private key as it is a discrete log based problem given by equation (5).

   Thus our scheme withstands the above attacks. By this we can say that only the designated proxy signer can create a valid proxy signature on behalf of the original signer. In other words, the original signer and other third parties who are not designated as proxy signer cannot create a valid signature. Thus the second requirement, Strong unforgeability, of a secure proxy signature is satisfied.

6 Conclusion

We have come up with a scheme to control delegation of financial power to a proxy signer. The scheme satisfies the basic requirements of a secure proxy signature scheme. We have considered Forgery by the original signer, Impersonating and framing attack to prove the security of our scheme. This concept of proxy signatures can be used in any discrete log based signature scheme. Here we have applied it for Digital Signature Algorithm(DSA).

References

[1] M. Ai-Ibrahim and A. Cerny. Proxy and threshold one-time signatures. In: Proc. of the 11th International Conference Applied Cryptography and Network Security (ACNS03), LNCS 2846, Springer-Verlag, 2003.

[2] A. Boldyreva, A. Palacio, and B. Warinschi. Secure proxy signature schemes for delegation of signing rights. Available at http://eprint.iacr.org/2003/096

[3] H. Ghodosi and J. Pieprzyk. Repudiation of cheating and non-repudiation of Zhangs proxy signature schemes. In: Information Security and Privacy (ACISP99), LNCS 1587, pp. 129-134. Springer-Verlag, 1999.

[4] S. Kim, S. Park, and D. Won. Proxy signatures, revisited. In: Information and Communications Security (ICICS97), LNCS 1334, pp. 223-232. Springer-Verlag, 1997.

[5] B. Lee, H. Kim, and K. Kim. Strong proxy signature and its applications. In: Proceedings of the 2001 Symposium on Cryptography and Information Security (SCIS01), Vol. 2/2, pp. 603-608. Oiso, Japan, Jan. 23-26, 2001.

[6] B. Lee, H. Kim, and K. Kim. Secure mobile agent using strong non-designated proxy signature. In: Information Security and Privacy (ACISP01), LNCS 2119, pp. 474-486. Springer-Verlag, 2001.

[7] J.-Y. Lee, J. H. Cheon, and S. Kim. An analysis of proxy signatures: Is a secure channel necessary ? In: Topics in Cryptology - CT-RSA 2003, LNCS 2612, pp. 68-79. Springer-Verlag, 2003.

[8] N.-Y. Lee, T. Hwang, and C.-H. Wang. On Zhangs nonrepudiable proxy signature schemes. In: Information Security and Privacy (ACISP98), LNCS 1438, pp. 415-422. Springer-Verlag, 1998.

[9] M. Mambo, K. Usuda, and E. Okamoto proxy signature: Delegation of the power to sign messages. IEICE Trans. Fundamentals, Sep. 1996, Vol. E79-A, No. 9, pp. 1338-1353.

[10] M. Mambo, K. Usuda, E. Okamoto. Proxy signatures for delegating signing operation. In: Proc. of 3rd ACM Conference on Computer and Communications Security (CCS96), pp. 48-57. ACM Press, 1996

[11] T. Okamoto, M. Tada, and E. Okamoto. Extended proxy signatures for smart cards. In: Information Security Workshop (ISW99), LNCS 1729, pp. 247-258. Springer-Verlag, 1999.

[12] H.-U. Park and I.-Y. Lee. A digital nominative proxy signature scheme for mobile communications. In: Information and Communications Security (ICICS01), LNCS 2229, pp. 451-455. Springer-Verlag, 2001.

[13] H. Wang and J. Pieprzyk. Efficient one-time proxy signatures. In: Asiacrypt03. Springer-Verlag, 2003.

[14] K. Zhang. Threshold proxy signature schemes. In: Information Security Workshop (ISW97), LNCS 1396, pp. 282-290. Springer-Verlag, 1997.

[15] K. Zhang. Nonrepudiable proxy signature schemes. Manuscript, 1997. Available at http://citeseer.nj.nec.com/360090.html