Available at
www.ElsevierMathematics.com
POWERED BY SCIENCE DIRECT®

ELSEVIER Applied Mathematics and Computation xxx (2004) xxx–xxx

APPLIED MATHEMATICS AND COMPUTATION

www.elsevier.com/locate/amc

A new proxy signature scheme with revocation ☆

Eric Jui-Lin Lu *, Min-Shiang Hwang, Cheng-Jian Huang

Department of Information Management, Chaoyang University of Technology, 168 Gifeng E. Rd., Wufeng, Taichung County 413, Taiwan, ROC

Abstract

In current proxy signature schemes, an original signer delegates her/his signing capability to a proxy signer, and then the proxy signer can sign messages on behalf of the original signer. Although these schemes have succeeded in proxy delegations, they are in general defective in proxy revocations. In this paper, we propose a proxy signature scheme which allows the original signer to revoke delegations whenever necessary.

Keywords: Proxy signature; Delegation by warrant; Revocation; Time-stamp

1. Introduction

A digital signature is like a "hand-written" signature in an electronic form that can be used to authenticate the identity of the signer of a document. Traditionally, the signer uses her/his secret key to sign messages by using some signature scheme such as ElGamal or Schnorr signature scheme [1,8,9]. However, the signer may not be able to sign messages by herself/himself when, for example, the signer is on a business trip or on vacation. Therefore, the signer needs a proxy signer to sign messages on behalf of her/him.

☆ This research was partially supported by the National Science Council, Taiwan, ROC, under contract no.: NSC91-2213-E-324-015.

* Corresponding author.

E-mail address: jlu@mail.cyut.edu.tw (E.J.-L. Lu).

0096-3003/$ - see front matter © 2004 Published by Elsevier Inc.
doi:10.1016/j.amc.2003.12.039

Currently, there are three types of delegation, namely, full delegation, partial delegation, and delegation by warrant. In the full delegation, the proxy signer is given the same private key that the original signer has. However, this approach does not satisfy the security requirements of proxy signatures [5,6]. In the partial delegation schemes [3,5,6], the original signer delegates her/his signing capability to a proxy signer by giving the proxy signer a proxy key. With the proxy key, the proxy signer can sign messages on behalf of the original signer. In cases where the proxy signer abuses her/his delegated rights, the original signer needs to revoke the proxy signer's signing capability. Currently, the proxy revocation protocols can be classified into two approaches. One approach is to change the public key of the original signer. This approach is impractical because, once the public key of the original singer is changed, all signatures generated earlier by the original signer cannot longer be verified. Even if we can come up with some public key management scheme so that the verification is possible, the tasks to verify all versions of signatures will be extremely complex if the original signer has to revoke delegations from time to time. The other approach is to put r_A (i.e. a part of the proxy key generated by the original signer) to a public revocation list. Any verifier must ensure that r_A is not on the list before verifications. However, this approach has two serious drawbacks. One is, once r_A is posted, all valid proxy signatures generated earlier cannot longer be verified. This also results in difficulties in verifying the validness of proxy signatures generated by the proxy signer even after the proxy key is revoked because the proxy signer can argue that these proxy signatures were generated before the proxy key is revoked. The other drawback is that the size of the revocation list will keep growing unlimitedly as long as it is required to verify all proxy signatures at all times.

Delegation by warrant is another type of proxy signature schemes [2,7,10,11]. In the delegation by warrant schemes, a proxy warrant is given to the proxy signer to generate proxy signatures. The proxy warrant usually contains the identity of the proxy signer, the valid period of delegation, and possible other restrictions on the signing capability delegated to the proxy signer. Therefore, the proxy signer can sign messages on behalf of the original signer only in the valid delegation period. However, delegation by warrant has two major limitations. One limitation is that the declaration of a valid delegation period in the warrant is of no use because no verifier can be assured the exact time when a proxy signature was created. Although Sun [10] claimed that this problem can be solved in a time-stamped proxy signature scheme he proposed, Lu and Huang [4] showed that Sun's scheme is insecure. The other limitation is that, with the warrant, the delegation will be terminated after the delegation period has expired. However, sometimes the original signer must put an end to her/his delegation earlier than what was planned. Unfortunately, there is no known protocol which allows early termination of delegations.

In this paper, we shall propose a proxy signature scheme which fulfills all the security requirements of proxy signatures stated in [5,6]. Also, all the drawbacks and limitations stated above are resolved by the proposed proxy revocation protocol. The rest of the paper is organized as follows. The proposed proxy signature scheme is described in Section 2, and the analysis of the proposed scheme is discussed in Section 3. Finally, we conclude this paper in Section 4.

2. The proposed proxy signature scheme

The proposed proxy signature scheme is based on the discrete logarithm problem. There are three participants in our scheme: an original signer, a proxy signer, and an authentication server (AS). The AS is a server who is responsible for verifying proxy warrants and issuing time-stamps for messages. The AS is a trusted third party and can identify the identities of the original signer and proxy signer with some general authentication scheme.

2.1. Notations

In the proposed proxy signature scheme, we shall use the following notations.

x_A	the private key of the original signer
y_A	the public key of the original signer
x_B	the private key of the proxy signer
y_B	the public key of the proxy signer
x_S	the private key of the AS
y_S	the public key of the AS
p	a large prime number
g	a generator for Z_p^*

2.2. Basic protocol

The details of the proposed proxy signature scheme are described as follows:

1. (*Proxy generation*) The original signer generates a random number $k_A \in_R Z_{p-1}$ and computes the followings:

$$r_A = g^{k_A} \bmod p, \quad \text{and}$$

$$\sigma_A = k_A + x_A h(m_w, r_A) \bmod p - 1.$$

2. (*Proxy delivery*) The original signer sends (m_w, r_A, σ_A) to both the proxy signer and authentication server through a secure channel.

3. (*Verification and alteration of the proxy*) The proxy signer checks the validity of (m_w, r_A, σ_A) by verifying whether or not the following equation holds.

$$g^{\sigma_A} = r_A y_A^{h(m_w,r_A)} \bmod p.$$

Similarly, the verification of the proxy also has to be carried out by the AS. If the verification is successful, the proxy signer then computes an alternative proxy private/public key pair σ_p and y'_p, respectively, such that

$$\sigma_p = \sigma_A + x_B h(m_w, r_A) \bmod p - 1, \quad \text{and}$$
$$y'_p = g^{\sigma_p} \bmod p.$$

4. (*Proxy validation*) For signing a message m, the proxy signer must first request a time-stamp for the message m from the AS. To obtain a time-stamp for the message m, the proxy signer transmits her/his identity and (σ_A, m) securely to the AS. The AS then searches for the tuple (m_w, r_A, σ_A) that was received in the "proxy delivery" step. With the tuple (m_w, r_A, σ_A), the AS must ascertain the following conditions are true before the time-stamp is issued.

(a) It is still in the valid period of proxy delegation specified in m_w.
(b) The r_A is not in the public revocation list maintained by the AS. If r_A is in the public revocation list, it means the delegation had been revoked.

5. (*Time-stamp generation*) The AS generates a random number k_S and computes $r_S = g^{k_S} \bmod p$ as well as $T_S = k_S + x_S h(m, t, r_S) \bmod p - 1$ where t denotes the time and date.
6. (*Time-stamp delivery*) The AS sends the time-stamp (r_S, t, T_S) to the proxy signer.
7. (*Time-stamp verification*) The proxy signer verifies the time-stamp by computing $g^{T_S} = r_S y_S^{h(m,t,r_S)} \bmod p$.
8. (*Signature generation*) The proxy signer uses σ_p to execute an ordinary signing operation. The proxy signature on the message m is then

$$(m, m_w, r_A, Sign_{\sigma_p}(m), r_S, t, T_S).$$

9. (*Verification of the proxy signature*) Any verifier first uses the same verification procedures of the original signing scheme to check $Sign_{\sigma_p}(m)$. Furthermore, the verifier has to check whether or not the following equations hold.

$$y'_p = r_A (y_A y_B)^{h(m_w,r_A)} \bmod p, \quad \text{and}$$
$$g^{T_S} = r_S y_S^{h(m,t,r_S)} \bmod p.$$

The calculations are to ascertain that the validness of the public proxy key and time-stamp.

2.3. Revocation protocol

When the original signer delegates her/his signing power to a proxy signer, the valid delegation period and other constraints on the signing capability are specified in the proxy warrant m_w. In general, the delegation will be terminated after the valid delegation period expires. However, if the original signer must revoke the delegation before the specified delegation period, the original signer then asks the AS to put the r_A in a public revocation list.

Upon being requested for a time-stamp by the proxy signer for a message m, the AS will checks both the valid delegation period specified in the proxy warrant m_w and the r_A in the public revocation list. If it is still in the delegation period and no matching r_A is found, the AS will issue a time-stamp for a message. If the delegation period is still valid but a matching r_A can be found, it means the delegation has been revoked, and then the AS will refuse the proxy signer's request. Also, if the delegation period has expired, the AS will refuse the proxy signer's request.

Note that the r_A in the public revocation list can be removed once the delegation period has expired. Therefore, the size of the public revocation list will not grow unlimited, unlike other revocation protocols that use the concept of the revocation list. More importantly, even when r_A is removed, the proxy signatures generated before the r_A was posted can still be verified. Also, since a proxy signature cannot be created without a time-stamp issued by the AS, it can be assured that no invalid proxy signatures can be generated by the proxy signer after either the delegation period has expired or the delegation was revoked.

3. Analysis

In this section, we shall analyze that the proposed proxy signature scheme satisfies not only all the security requirements of proxy signatures stated in [5,6], but also provides an effective revocation mechanism.

(i) *Strong unforgeability*: Strong unforgeability requires that only the proxy signer can create a valid proxy signature. Even the original signer cannot create a valid proxy signature. Because the generation of a proxy signature requires the private key of the proxy signer in the proposed scheme, no one can forge a valid proxy signature.

(ii) *Verifiability*: Any verifier can be convinced of the agreement of the original signer on the signed message from its corresponding proxy signature. In the proposed scheme, the proxy signature consists of (m, m_w, r_A, $Sign_{\sigma_p}(m)$, r_S, t, T_S). From the warrant m_w and r_A generated by original signer, any verifier can be convinced of the original signer's agreement on the signed message.

(iii) *Proxy signer's deviation*: This requirement imposes that all valid proxy signatures created by a proxy signer can be detected as her/his signatures, and the proxy signer cannot create a valid signature for some signer found in a public key list. Because the generations of proxy signatures need the proxy signer's private key, all valid proxy signatures created by a proxy signer can be detected as her/his signatures. Additionally, since all private keys are kept secret by their owners, the proxy signer cannot create a valid signature for some signer found in a public key list.

(iv) *Distinguishability*: It is required that the valid proxy signatures created by the proxy signer and the ordinary signatures created by the original signer can be distinguished. Because the verification of these signatures utilizes different congruences, the proposed proxy signature scheme satisfies the distinguishability.

(v) *Strong identifiability*: Anyone can determine the identity of the proxy signer from the proxy signatures created by her/him. In the proposed scheme, anyone can identify the identity of the proxy signer from the m_w.

(vi) *Secret-keys' dependence*: The secret-key's dependence requires that the proxy signature is generated from a new secret computed from the private key of the original signer. This condition is satisfied because the proxy secret σ_A is computed from the original signer's private key x_A in the proposed scheme.

(vii) *Strong undeniability*: This requirement states that the proxy signer cannot repudiate the valid proxy signatures created by her/him. This is also called "non-repudiation" in some literatures. Since a proxy signature is created by using the proxy signer's private key x_B, the proxy signer cannot disavow the proxy signature she/he created.

As stated earlier, we believe a proxy signature scheme should also provide an effective revocation mechanism. In the proposed scheme, because (1) each valid proxy signature requires a time-stamp from the AS, (2) the AS will issue the time-stamp only when the delegation has not been revoked and the delegation period has not expired, and (3) the r_A can be removed from the public revocation list once the delegation period is expired, the following benefits are achieved:

- the verification of proxy signatures will not fail because of the posting of r_A in the public revocation list,
- the proxy signer cannot create a valid proxy signature after the delegation period has expired or the delegation is revoked and then later argue the proxy signature is created before the revocation, and
- the size of the public revocation list will not increase unlimitedly.

Also, note that, because the T_S is calculated from the t and the message m and signed by the AS, the proxy signer can only use the T_S for the message m. It is not allowed for the proxy signer to use the T_S for any other message.

4. Conclusion

In current partial proxy signature schemes, an original signer cannot delegate her/his signing capability to a proxy signer within a pre-specified delegation period. To overcome this problem, proxy signature schemes with delegation by warrant have been proposed. In general, for these proxy signature schemes with delegation by warrant, the delegation is terminated after the delegation period, specified in the warrant, is expired. However, there is no way to revoke the delegation before the specified delegation date.

In this paper, we proposed a new proxy signature scheme with effective revocation protocol. In the proposed scheme, it satisfies all the security requirements of proxy signatures. Also, by deploying a trusted third party called the authentication server, the proposed scheme provides an effective revocation protocol such that it require the proxy signer to get a time-stamp from the authentication server to sign messages on behalf of the original signer. Also, the AS will not issue the time-stamp unless the delegation has not been revoked or the delegation period specified in the warrant has not expired. Furthermore, the AS can remove the r_A from the public revocation list once the delegation has expired, and thus the public revocation list will not grow unlimitedly.

References

[1] M.S. Hwang, I.C. Lin, E.J.L. Lu, A secure nonrepudiable threshold proxy signature scheme with known signers, Informatica 11 (2) (2000) 1–8.

[2] Seungjoo Kim, Sangjoo Park, Dongho Won, Proxy signatures, revisited, in: Proceedings of International Conference on Information and Communications Security, Lecture Notes in Computer Science, vol. 1334, 1997, pp. 223–232.

[3] Byoungcheon Lee, Heesun Kim, Kwangjo Kim, Strong proxy signature and its applications, in: Proceedings of the 2001 Symposium on Cryptography and Information Security (SCIS 2001), Shirahama, 2001.

[4] Eric Jui-Lin Lu, Cheng-Jian Huang, Cryptanalysis of a time-stamped proxy signature scheme, Int. J. Informatica, in press.

[5] Mambo Masahiro, Keisude Usuda, Eiji Okamoto, Proxy signatures: delegation of the power to sign messages, IEICE Trans. Fundament. E79-A (9) (1996) 1338–1354.

[6] Mambo Masahiro, Keisuke Usuda, Eiji Okamoto, Proxy signatures for delegating signing operation, in: Proceedings of 3rd ACM conference on Computer and Communications Security, 1996, New Delhi, pp. 48–57.

[7] B. Clifford Neuman, Proxy-based authorization and accounting for distributed systems, in: Proceedings of the 13th International Conference on Distributed Computing Systems, 1993, Pittsburgh, PA, pp. 283–291.

[8] Bruce Schneier, Applied Cryptography, Wiley, New York, 1996.

[9] Claus-Peter Schnorr, Efficient signature generation by smart cards, J. Cryptol. 4 (1991) 161–174.

[10] Hung-Min Sun, Design of time-stamped proxy signatures with traceable receivers, IEE Proc. Comp. Digital Techn. 147 (6) (2000) 462–466.

[11] Vijay Varadharajan, Phillip Allen, Stewart Black, An analysis of the proxy problem in distributed systems, in: Proceedings of 1991 IEEE Computer Society Symposium on Research in Security and Privacy, 1991, pp. 255–275.