INFORMATICA, 2004, Vol. 15, No. 4, 455–464

An Efficient Proxy Signature Scheme with Revocation

Manik Lal DAS, Ashutosh SAXENA, Ved P. GULATI
Institute for Development and Research in Banking Technology
Castle Hills, Road No. 1, Masab Tank, Hyderabad, India
e-mail: {mldas, asaxena, vpgulati}@idrbt.ac.in

Received: October 2004

Abstract. A proxy signature allows a designated person, called a proxy signer, to sign the message on behalf of the original signer. Proxy signatures are very useful tools when one needs to delegate his/her signing capability to other party. A number of proxy signature schemes have been proposed and succeeded for proxy delegations, but the schemes are in defective in proxy revocations. In this paper, we propose two proxy signature schemes based on RSA cryptosystems. The proposed first scheme does not consider proxy revocation mechanism; however, it will help us to compare our protocol with the existing RSA-based schemes. The proposed second scheme provides an effective proxy revocation mechanism. The proposed schemes do not require any secure channel to proxy key delivery and support the necessary security requirements of proxy signature.

Key words: proxy signature, proxy key, partial delegation, warrant, timestamp, revocation.

1. Introduction

In 1996, Mambo *et al.* first introduced the concept of proxy signature (Mambo *et al.*, 1996).They classified the proxy signature on the basis of delegation, namely, full delegation, partial delegation and delegation by warrant. In full delegation, an original signer directly gives his secret key to a proxy signer, and the proxy signer signs documents on behalf of the original signer with it. The main drawback of full delegation is the absence of a distinguishability between the original signer and the proxy signer. In partial delegation, an original signer derives a proxy key from his secret key and hands it over to a proxy signer. In this case, the proxy signer can misuse the original signer's delegated rights because partial delegation does not restrict the proxy signer's signing capability. The weaknesses of full delegation and partial delegation are eliminated by adding an explicit warrant to the delegated rights, which is called partial delegation with warrant. A warrant consists of signers' identity, delegation period and the qualification of the message on which the proxy signer can sign. The revocation of delegated rights (i.e., proxy revocation) is an important issue of proxy signatures scheme. The proxy revocation is essential for the situation where signers' key is compromised or/and any misuse of the delegated rights is noticed. It may also happen that the original signer wants to terminate the delegated rights before the expiry of the delegated rights. So far, many proxy signature

schemes based on discrete logarithms (Diffie and Hellman, 1976) have been proposed in (Boldyreva *et al.*, 2003; Hwang and Chen, 2003; Lee *et al.*, 2001a; Lee *et al.*, 2001b; Li *et al.*, 2003; Lu and Huang, 2003; Kim *et al.*, 1997; Mambo *et al.*, 1996; Zhang, 1997); however, the proxy revocation mechanism is not addressed in these schemes. Sun (2000) proposed a scheme and claimed that the revocation problem can be solved by using a timestamp, but Lu and Huang (2003) showed that Sun's scheme is insecure. In contrast to the discrete logarithms based proxy signature schemes, few RSA-based proxy signature schemes have also been proposed in (Okamoto *et al.*, 1999; Lee *et al.*, 2001b; Shao, 2003), but none of them consider the proxy revocation mechanism. In this paper, we propose two proxy signature schemes based on RSA cryptosystems (Rivest *et al.*, 1978). The proposed first scheme does not consider proxy revocation mechanism; however, it will help us to compare our protocol with the existing RSA-based schemes (Lee *et al.*, 2001b; Okamoto *et al.*, 1999; Shao, 2003). The proposed second scheme provides an effective proxy revocation mechanism. The declaration of a valid delegation period in the warrant is of no use since no verifier can be assured of the exact time when a proxy signature was created. We use a timestamp in the proposed proxy revocation protocol by which the verifier can be assured of the exact time when a proxy signature was created. The rest of the paper is organized as follows. In the next section we present the related work. In Section 3 we propose the schemes. In Section 4 we analyze the security of the proposed schemes. In Section 5 we compare the efficiency of our schemes with the related schemes. The paper is concluded with Section 6.

2. Related Work

In 1999, Okamoto *et al.* proposed a proxy signature scheme (Okamoto *et al.*, 1999), where the proxy signature and verification process are based on RSA and done by a smart card. Afterwards, Lee *et al.* (2001b) showed that a secure mobile agent can be constructed using strong non-designated proxy signature, and they proposed RSA-based proxy signature scheme. Both the schemes (Okamoto *et al.*, 1999) and (Lee *et al.*, 2001b) are designed as proxy-unprotected notion (Kim *et al.*, 1997), where the original signer can forge the proxy signature by signing the message and then claim that the proxy signer has signed the message. Thus, these schemes do not meet the strong unforgeability property of proxy signatures. Moreover, Wang *et al.* (2003) showed the forgery attacks of (Lee *et al.*, 2001b). In 2003, Shao proposed a proxy signature scheme based on factoring. However, Shao's scheme does not present the security analysis of the scheme. In addition, both the schemes (Okamoto *et al.*, 1999) and (Shao, 2003) require a secure channel to deliver the proxy key. These schemes do not provide any proxy revocation mechanism, thereby the original signer cannot revoke his delegated rights if he wants to do so, which is a practical requirement (e.g., when the misuse of delegated rights is found) of the applications of proxy signatures.

3. Proposed Scheme

In this section, we propose two proxy signature schemes. Our first scheme is proxy signatures without revocation and the second scheme is proxy signatures with revocation. In both the schemes, the signature generation and verification process are based on RSA (Rivest *et al.*, 1978). The schemes are divided into five phases: setup parameters, proxy key generation, proxy key verification, proxy signature generation, and proxy signature verification. Throughout this article, the notation $h(.)$ is a one-way hash function (Schneier, 1996). In the rest of the paper, $h(m_1 \| m_2)$ denotes the hash of concatenation of two messages m_1 and m_2.

3.1. *Proxy Signature without Revocation*

The protocol works as follows:

[Setup Parameters]

(i) The original signer generates RSA public-secret key pair (e_o, d_o), where d_o is kept secret, (e_o , n_o) is the certified public key and n_o is the product of two large safe primes.

(ii) The proxy signer generates RSA public-secret key pair (e_p, d_p), where d_p is kept secret, (e_p, n_p) is the certified public key and n_p is the product of two large safe primes.

(iii) The original signer creates a signature on a warrant m_w and gives it to the proxy signer and then the proxy signer uses it to generate a proxy key. A warrant m_w is not a secret entity and consists of the delegation information, i.e., the identity of original signer and proxy signer, the qualification of the message on which the proxy signer can sign on behalf of the original signer, the validity period of delegation etc.

[Proxy Key Generation] The original signer does the following:

(i) Computes $s_o = h(m_w \| e_p)^{d_o} \bmod n_o$.

(ii) Sends (s_o, m_w) to the proxy signer over a public channel.

[Proxy Key Verification] The proxy signer checks whether $h(m_w \| e_p) = s_o^{e_o} \bmod n_o$. If it holds, the proxy signer accepts it as a valid proxy key; otherwise, rejects it.

[Proxy Signature Generation] To sign message m on behalf of the original signer, the proxy signer does the following:

(i) Computes $s_p = (s_o \oplus h(m \| m_w \| e_p))^{d_p} \bmod n_p$, where $\oplus$ is an exclusive-OR operation.

(ii) The proxy signature of message m is (m, m_w, s_p, e_o, e_p).

[Proxy Signature Verification] The verifier (typically the recipient of proxy signature) verifies whether $h(m_w \| e_p) = (s_p^{e_p} \bmod n_p \oplus h(m \| m_w \| e_p))^{e_o} \bmod n_o$. If it holds, he accepts it as a valid proxy signature; otherwise, rejects it.

3.2. *Proxy Signature with Revocation*

The proxy revocation mechanism can be categorized into two types. The first approach is to publish the public key of the original signer in a trusted server public revocation list. To validate a proxy signature, the verifier should first check whether the public key of the original signer is in the public revocation list. If the original signer's public key is published in the revocation list, the verifier will reject the signed message. The second approach is to use a timestamp along with the warrant in the signature generation phase, thereby enabling the verifier to validate the time of proxy signature generation on checking the validity period of warrant attached with the delegated rights. Moreover, from the timestamp-based approach, the verifier can be assured of the exact time when a proxy signature was created.

The proposed scheme comprises with four participants, namely, an original signer, a proxy signer, a trusted server (TS) and a verifier (typically the recipient of proxy signature). The TS is responsible to maintain a public key revocation list. The TS will issue timestamp to the proxy signer after verifying both the revocation list and warrant.

[Setup Parameters]

(i) The original signer generates RSA public-private key pair (e_o, d_o), where d_o is kept secret, (e_o, n_o) is the certified public key and n_o is the product of two large safe primes.

(ii) The proxy signer generates RSA public-private key pair (e_p, d_p), where d_p is kept secret, (e_p, n_p) is the certified public key and n_p is the product of two large safe primes.

(iii) The TS generates RSA public-private key pair (e_s, d_s), where d_s is kept secret, (e_s, n_s) is the certified public key and n_s is the product of two large safe primes.

(iv) The original signer creates a signature on a warrant m_w and gives it to the proxy signer and then the proxy signer uses it to generate a proxy key. A warrant m_w is not a secret entity and consists of the delegation information, i.e., the identity of original signer and proxy signer, the qualification of the message on which the proxy signer can sign on behalf of the original signer, the validity period of delegation etc.

[Proxy Key Generation] The original signer does the following:

(i) Computes $s_o = h(m_w \| e_p \| e_s)^{d_o} \bmod n_o$.

(ii) Sends (s_o, m_w) to both proxy signer and TS over the public channel.

[Proxy Key Verification] The proxy signer and TS check whether $h(m_w \| e_p \| e_s) = s_o^{e_o} \bmod n_o$. If it holds, they accept it; otherwise, reject it.

[Proxy Signature Generation]

(i) To sign message m, the proxy signer first requests a timestamp to the TS. For this, the proxy signer computes $R = h(m \| m_w \| e_o \| e_p)^{d_p} \bmod n_p$, and then sends (R, m, e_o, e_p) to the TS over the public channel.

(ii) The TS verifies whether $h(m\|m_w\|e_o\|e_p) = R^{e_p} \bmod n_p$. If it holds, the TS must ascertain the following conditions are true before issuing a timestamp:

– original signer's public key e_o is not in the TS public revocation list.

– m_w is not expired.

(iii) If (ii) holds, the TS computes $T_m = h(m\|m_w\|e_o\|e_p\|t)^{d_s} \bmod n_s$, where t denotes a timestamp. Then TS sends (T_m, t) to the proxy signer over a public channel.

(iv) The proxy signer verifies whether $h(m\|m_w\|e_o\|e_p\|t) = T_m^{e_s} \bmod n_s$.

(v) The proxy signer computes $s_p = (s_o \oplus h(m\|m_w\|e_p\|t))^{d_p} \bmod n_p$.

(vi) The proxy signature of message m is ($m, m_w, s_p, t, T_m, e_o, e_p$).

[Proxy Signature Verification]

(i) The verifier checks whether the original signer's public key e_o is in the TS public revocation list. If it holds, he rejects the signed message. Otherwise, he proceeds to step (ii).

(ii) The verifier checks whether $h(m\|m_w\|e_o\|e_p\|t) = T_m^{e_s} \bmod n_s$.

(iii) The verifier verifies whether $h(m_w\|e_p\|e_s) = (s_p^{e_p} \bmod n_p \oplus h(m\|m_w\|e_p\|t))^{e_o} \bmod n_o$. If (ii) and (iii) hold, he accepts it as a valid proxy signature; otherwise, rejects it.

4. Security Analysis

In the following, we show that the proposed schemes satisfy the security features, namely, strong unforgeability, strong undeniability, strong identifiability, verifiability and prevention of misuse.

SR1. In our schemes, the proxy signature is created with the proxy signer's secret key d_p and delegated proxy key s_o. The proxy key is binding with the original signer's secret key d_o. No one (including the original signer) can construct the proxy signature without having the knowledge of the secret keys d_p and d_o. Obtaining these secret keys by any other party is as difficult as breaking RSA, which is believed to be a hard problem (Boneh, 1999). Moreover, the verification of $h(m_w\|e_p)$ with the signed message prevents the dishonest party from the creation of forged proxy signatures. Therefore, any party including the original signer cannot forge a valid proxy signature and thus, the proposed schemes satisfy the strong unforgeability property.

SR2. From a proxy signature of the proposed scheme, the involvements of both original signer and proxy signer are ascertained by

– the warrant m_w,

– the connection of the public keys e_p and e_o in the verification process.

Thus, the proxy signer and the original signer cannot deny their involvement in a valid proxy signature, that is, the schemes satisfy the strong undeniability property.

SR3. The verification process of the proposed schemes requires proxy signer's public key e_p and warrant m_w. Any verifier can determine the identity of the proxy signer

from the signed message, because the signed message is computed as $s_p = (s_o \oplus h(m\|m_w\|e_p\|t))^{d_p} \bmod n_p$, where s_o is a signed warrant by the original signer. Therefore, in the verification process any verifier can determine the identity of the proxy signer from m_w.

SR4. The proxy signatures of the proposed schemes consist of an explicit warrant m_w and a signed warrant s_o delegated by the original signer. Any verifier can be convinced of the signers' agreement and validity of the proxy signature of the transmitted message from m_w.

SR5. Both the proxy signer and the original signer's misuse are prevented in our schemes. The proxy signer cannot forge the delegated rights. In case of the proxy signer's misuse, the responsibility of the proxy signer is determined from the warrant m_w. The original signer's misuse is also prevented because he cannot compute a valid proxy signature against the name of the proxy signer, which is the strong unforgeability property (**SR1**) of our schemes. Furthermore, if the proxy signer's misuse of delegated rights is noticed to the original signer, the original signer immediately revokes his public key with the help of TS.

In addition to the above security properties, the proposed scheme with revocation uses a trusted server, who verifies the validity of the original signer and the proxy signer from the request ($R = h(m\|m_w\|e_o\|e_p)^{d_p} \bmod n_p$) sent by the proxy signer, and then issues the timestamp.

5. Performance Analysis

In order to analyze the performance of our scheme, we compare the computational complexity of our schemes with the existing RSA-based proxy signature schemes (Okamoto *et al.*, 1999; Lee *et al.*, 2001b) and (Shao, 2003). It is noted that the existing schemes and our first scheme do not provide the proxy revocation mechanism, but, in the following, we show that our first scheme is efficient than the existing schemes. Our second scheme solves the proxy revocation problems and provides effective proxy revocation mechanism, which is an important requirement for many practical situation (e.g., when the secret key of the signers' is compromised or any misuse of the delegation rights is occurred). For simplicity, we neglect exclusive-OR operation time of the schemes. The notations used in the Table 1 are as follows:

t_e: computation time for an exponentiation operation;

t_m: computation time for a multiplication operation;

t_o: computation time for a modular operation;

h: computation time for a hash operation.

For simplicity, we neglect the concatenation and exclusive-OR operational time. The computation time of different phases of the schemes is given in Table 1.

It is important to note that the computation time for a valid proxy signature falls into two parts. The first part consists of the time taken for the setup parameters, proxy key generation and proxy key verification process, which are a one-time computation and

Table 1

Computation time of the schemes

Phases	OTO Scheme (Okamoto *et al.*, 1999)	LKK Scheme (Lee *et al.*, 2001b)	Shao's Scheme (Shao, 2003)	Our Scheme without revocation	Our Scheme with revocation
Setup Parameters	$t_e+t_m+t_o$	$2t_e+2t_m+2t_o$	$t_e+t_m+t_o$	$2t_e+2t_m+2t_o$	$3t_e+3t_m+3t_o$
Proxy Key Generation	$2t_e+t_m+t_o+h$	t_e+t_o+h	t_e+t_o+h	t_e+t_o+h	t_e+t_o+h
Proxy Key Verification	$t_e+t_m+t_o+h$	t_e+t_o+h	$t_e+t_m+t_o+h$	t_e+t_o+h	$2t_e+2t_o+2h$
Signature Generation	$2t_e+3t_m+2t_o+h$	$3t_e+3t_o+2h$	$2t_e+t_m+2t_o+h$	t_e+t_o+h	$5t_e+5t_o+5h$
Signature Verification	$2t_e+2t_m+t_o+h$	$3t_e+3t_o+2h$	$2t_e+t_m+t_o+2h$	$2t_e+2t_o+2h$	$3t_e+3t_o+3h$

remain fixed for the entire delegation period. We call the sum of the setup parameters, proxy key generation, proxy key verification computation time as *initial computation time*. The second part consists of the computation time of proxy signature generation and proxy signature verification process, which are dynamic operations as and when required and we call the sum of these computation time as *dynamic operation time*.

As initial computation time is a one-time computation and remains fixed for the entire delegation period, the efficiency of the scheme depends primarily on the dynamic operation time. The schemes (Okamoto *et al.*, 1999; Lee *et al.*, 2001b; Shao, 2003) and our first scheme do not consider proxy revocation mechanism. It is observed from Table 1 that for a proxy signature without revocation our scheme saves at least t_e or t_o time unit in comparisons to others. Our second proposed scheme provides a proxy revocation mechanism but with additional computational cost as compared with others' schemes.

6. Conclusion

In this article, we have proposed two proxy signature schemes based on RSA cryptosystems. Our first scheme does not consider proxy revocation mechanism, but it is efficient than the existing RSA-based schemes (Okamoto *et al.*, 1999; Lee *et al.*, 2001b) and (Shao, 2003). Our second scheme provides an effective proxy revocation mechanism, in which the proxy signer takes a timestamp from the TS and signs message on behalf of the original signer. With the proxy revocation protocol, the proxy signer cannot create a valid proxy signature after the expiry of the delegated rights or once the original signer's public key is published in the TS public revocation list. Moreover, the timestamp in the proxy revocation protocol will assure the verifier about the exact time when a proxy signature was created. The proposed schemes satisfy the necessary security requirements of proxy signatures and do not require any secure channel to deliver the proxy key, whereas,

a secure channel is must for the schemes (Okamoto *et al.*, 1999; Shao, 2003). The TS can remove the original signer's public key from the public revocation list once the delegation period has expired, and thus the public revocation list will not grow unlimitedly.

Acknowledgements

We are thankful to the anonymous reviewers for their valuable comments. We also thank Rashmi Dev for manuscript improvements. This research was supported in part by the Ministry of Communications and Information Technology, Govt. of India, under the grant No. DIT/R&D/Coord/1(6)/2003.

References

Boldyreva, A., A. Palacio and B. Warinschi (2003). Secure signature schemes for delegation of signing rights. *IACR ePrint Report no. 96*. `http://eprint.iacr.org/2003/96`

Boneh, D. (1999). Twenty years of attacks on the RSA cryptosystem. *Notices of the American Mathematical Society*, **46**(2), 203–213.

Diffie, W., and M.E. Hellman (1976). New directions in cryptography. *IEEE Transactions on Information Theory*, **IT-22**(6), 644–654.

Hwang, S., and C. Chen (2003). Cryptanalysis of nonrepudiable threshold proxy signature schemes with known signers. *Informatica*, **14**(2), 205–212.

Lee, B., H. Kim and K. Kim (2001a). Strong proxy signature and its applications. In *Proceedings of Symposium on Cryptography and Information Security*, Japan. pp. 603–608.

Lee, B., H. Kim and K. Kim (2001b). Secure mobile agent using strong non-designated proxy signature. In *Proceedings of ACISP'01, LNCS*, 2119, Springer–Verlag. pp. 474–486.

Li, L.H., S.F. Tzeng and M.S. Hwang (2003). Generalization of proxy signature-based on discrete logarithms. *Computers & Security*, **22**(3), 245–255.

Lu, E.J., and C. Huang (2003). Cryptanalysis of a time-stamped proxy signature scheme. Accepted and to appear in the *International Journal of Computational and Numerical Analysis and Applications*.

Lv, J., J. Liu and X. Wang (2003). Further cryptanalysis of some proxy signature schemes. *IACR ePrint Report no. 111*. `http://eprint.iacr.org/2003/111`

Kim, S., S. Park and D. Won (1997). Proxy signatures revisited. In *Proceedings of ICICS'97, LNCS*, 1334, Springer–Verlag. pp. 223–232.

Mambo, M., K. Usuda and E. Okamoto (1996). Proxy signatures: Delegation of the power to sign messages. *IEICE Transactions on Fundamentals*, **E79-A**(9), 1338–1353.

Okamoto, T., M. Tada and E. Okamoto (1999). Extended proxy signatures for smart card. In *Proceedings of Information Security Workshop, LNCS*, 1729, Springer–Verlag. pp. 247–258.

Rivest, R.L., A. Shamir and L. Adleman (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, **21**(2), 120–126.

Schneier, B. (1996). *Applied Cryptography* (2nd edition). John Wiley & Sons Inc.

Shao, Z. (2003). Proxy signature schemes based on factoring. *Information Processing Letters*, **85**(3), 137–143.

Sun, H., and B. Hsieh (2003). On the security of some proxy signature schemes. *IACR ePrint Report no. 68*. `http://eprint.iacr.org/2003/68`

Sun, H. (2000). Design of time-stamped proxy signatures with traceable receivers. *IEE Proceedings of Computers and Digital Techniques*, **176**(6), 462–466.

Wang, G., F. Bao, J. Zhou and R.H. Deng (2003). Security analysis of some proxy signatures. *IACR ePrint Report no. 196*. `http://eprint.iacr.org/2003/196`

Zhang, K. (1997). Threshold proxy signature schemes. In *Proceedings of Information Security Workshop, LNCS*, 1396, Springer–Verlag. pp. 191–197.

M.L. Das received his M. Tech. Degree in 1998. He is currently pursuing his PhD work in K. R. School of Information Technology, Indian Institute of Technology – Bombay, India. He is a member of Cryptology Research Society of India and Indian Society for Technical Education. His research interests include Cryptography and Information Security.

A. Saxena received his PhD degree in computer science. He is an associate professor with Institute for Development and Research in Banking Technology, Hyderabad, India. He is a member of Cryptology Research Society of India and Computer Society of India. His research interests include authentication technologies, smart cards, key management and security issues in banking.

V.P. Gulati received his PhD degree from Indian Institute of Technology – Kanpur, India. He is a director of Institute for Development and Research in Banking Technology, Hyderabad, India. He is a member of IEEE, Computer Society of India and vice president of Cryptology Research Society of India. His research interests include payment systems, security technologies, financial networks and banking applications.

Įgalioto parašo su atšaukimu efektyvus metodas

Manik Lal DAS, Ashutosh SAXENA, Ved P. GULATI

Straipsnyje pasiūlyti du įgaliotojo parašo sukūrimo metodai. Pirmajame metode nėra parašo atšaukimo galimybės, tačiau jis yra efektyvesnis už kitus metodus, naudojančius RSA kriptoschemas. Antrasis metodas turi parašo įgaliojimo atšaukimo mechanizmą, kai įgaliotasis asmuo gauna laiko žymę iš patikimo serverio ir dokumentą pasirašo įgaliojančiojo asmens vardu. Tačiau įgaliotasis asmuo negali pasirašyti po to, kai pasibaigia parašo įgaliojimo laikas arba jei įgaliojančiojo asmens viešasis raktas yra patikimo serverio atšauktųjų raktų sąraše. Įgaliotajam raktui persiųsti nereikalingas slaptasis ryšio kanalas, o pasiūlyti metodai tenkina būtinus įgaliotojo parašo saugumo reikalavimus.