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

Information Processing Letters 85 (2003) 137–143

Information Processing Letters

www.elsevier.com/locate/ipl

Proxy signature schemes based on factoring

Zuhua Shao

Department of Computer and Electronic Engineering, Zhejiang University of Science and Technology, Hangzhou, Zhejiang, 310012, PR China

Received 2 January 2002; received in revised form 5 June 2002

Communicated by F.Y.L. Chin

Abstract

The proxy signature schemes allow proxy signers to sign messages on behalf of an original signer, a company or an organization. However, most of existing proxy signature schemes are based on the discrete logarithm problem. In this paper, the author would like to propose two efficient proxy signature schemes based on the factoring problem, which combine the RSA signature scheme and the Guillou–Quisquater signature scheme. One is a proxy-unprotected signature scheme that is more efficient. No matter how many proxy signers cooperatively sign a message, the computation load for verifiers would remain almost constant. The other is a proxy-protected signature scheme that is more secure. Finally, to protect the privacy of proxy signers, the author proposes a proxy-protected signature scheme with anonymous proxy signers.

Keywords: Cryptography; Proxy signature; Multisignature; Factoring; Discrete logarithm

1. Introduction

For electronic commercial applications, evidence of possession of documents is especially important. A digital signature is analogous to an ordinary handwritten signature and establishes both of signer authenticity and data integrity assurance. In a signature scheme, each signer randomly chooses a private key and publishes a public key. To generate a signature for some message, the signer uses his private key to compute the signature while any other can verify this signature with the signer's public key. Without the knowledge of the private key, anyone cannot generate a valid signature for any message. Hence it is necessary for the signer to keep his private key secret.

When the signer cannot sign messages for some reasons, he maybe has to consign somebody to sign messages on behalf of him. For instance, when the president of a company is going on a vacation, he maybe needs to authorize his secretary to sign messages on behalf of him. However, it is dangerous for him to give his private key to his secretary. A proxy signature scheme provides a method, by which the original signer authorizes a designated person, called proxy signer, to sign messages on behalf of the original signer, a company or an organization.

In 1996, Mambo, Usuda and Okamoto first introduced the concept of proxy signature [1]. There are, so far, three types of delegation, full delegation, partial delegation and delegation by warrant. In the full delegation, a proxy signer is given the same private key as the original signer has, and computes the

E-mail address: zhshao_98@yahoo.com (Z. Shao).

0020-0190/02/$ – see front matter
PII: S0020-0190(02)00367-8

same signatures as the original signer does. Hence, the original signer should bear all responsibility for messages signed by his proxy signer. In the partial delegation [2], the original signer uses his private key to create a proxy signature key and sends it to the proxy signer in a secure manner. The proxy signer uses the proxy signature key to compute proxy signatures on behalf of the original signer. For the security reason, it must be computationally infeasible to compute the original signer's private key from the proxy signer's proxy signature key. In the delegation by warrant [3], the original signer gives the proxy signer a warrant, which certifies that the proxy signer is legal. Delegation by warrant is performed by the consecutive execution of signing of the public key signature scheme.

Therefore, among these three types of delegation, the partial delegation is the best choice to use, since the full delegation cannot provide enough security and the delegation by warrant cannot provide faster processing speed.

Since Diffie and Hellman invented the concept of public-key cryptography [4], there have been two kinds of most popular public key cryptosystems: one is based on the factoring problem [5], the other is based on the discrete logarithm problem [6]. In the existing proxy signature schemes [7–12], most are based on the discrete logarithm problem. It is desirable to design proxy signature schemes based on the factoring problem.

Depending on whether the original signer can generate the same proxy signatures as the proxy signers do, there are two kinds of proxy signature schemes:

(1) *Proxy-unprotected*: The proxy signer generates proxy signatures only with the proxy signature key given by the original signer. So the original signer can also generate the same proxy signatures. If there are some disputes between the original signer and his proxy signers, outsiders have not any method to identify who indeed signs the messages on behalf of the original signer. Verifiers validate proxy signatures only with the public key of the original signer and pay attention to legality of the warrant. They do not mind the actual signer, which maybe is anonymous to them. Therefore the original signer must consign trustworthy enough persons as his proxy signers and only permit these persons to proxy sign less important documents on behalf of him, since the original signer would bear all responsibility for proxy signatures.

(2) *Proxy-protected*: The proxy signer generates proxy signatures not only with the proxy signature key given by the original signer but also with the private key of himself. Anyone else, including the original signer, cannot generate the same proxy signatures. Verifiers validate the proxy signatures with both the public key of the proxy signer and the public key of original signer. The proxy signer cannot deny his signatures and the original signer cannot disavow his warrant. Both the original signer and the actual proxy signer must bear responsibility for the proxy signatures together. Therefore, proxy-protected signature scheme can provide more security level.

When the president of a company entrusts his secretary to sign receipts of letters on behalf of him, they can use the first kind of proxy signature schemes. However, when a bank approves his customers to write bank's checks, they must use the second kind of proxy signature schemes.

In this paper, the author would like to propose two proxy signature schemes based on the factoring problem, which combine the RSA signature scheme and the Guillou–Quisquater signature scheme [13]. One is a proxy-unprotected signature scheme that is more efficient. The other is a proxy-protected signature scheme that is more secure.

Finally, to protect the privacy of proxy signers, the author proposes a proxy-protected signature scheme with anonymous proxy signers by use of anonymous public key certificates [14].

2. The proxy-unprotected signature scheme

In the public key cryptosystems based on factoring problem, each user should choose his RSA private key. The signer randomly chooses two large safe primes p and q, and computes a public modulus $n = pq$. Then the signer chooses a pair of integers e and d satisfying the properties $ed = 1 \pmod{\phi(n)}$ and d is large enough, where $\phi(n)$ is the Euler–Totient function. The signer chooses a public one-way hash function $h()$. The private key (p, q, d) is kept secret by the signer,

while the public key of the signer is (n, e), which is certified by a CA.

2.1. Proxy generation

Suppose that the original signer p_0 has private key (p_0, q_0, d_0) and public key (n_0, e_0), where e_0 should be larger than the output of the one-way hash function $h()$.

For each proxy user p_i, the original signer computes a proxy signature key

$$v_i = h(m_w, SN_i)^{-d_0} \pmod{n_0}.$$

Then he sends (v_i, m_w, SN_i) to the proxy user p_i in a secure manner. Here m_w is a warrant, which records the delegation policy including limits of authority, valid periods of delegation and proxy signature, and the identity and the public key of the original signer. SN_i is an inner serial number.

Each proxy signer p_i can verify the proxy signature key by checking the following equation:

$$v_i^{e_0} h(m_w, SN_i) = 1 \pmod{n_0}.$$

2.2. Proxy-unprotected mono-signature

Assume that according to the limit of authority, the proxy signer p_i has right to proxy sign a message m on behalf of the original signer. He does the following steps:

(1) The proxy signer randomly chooses an integer $t \in [1, n_0]$ and computes $r = t^{e_0} \pmod{n_0}$.
(2) The proxy signer computes $k = h(m, r)$.
(3) The proxy signer computes $y = t v_i^k \pmod{n_0}$.

The proxy signature of the message m is (m, m_w, SN_i, y, k). The verifier can verify the proxy signature as follows:

(1) The verifier computes

$$r' = y^{e_0} h(m_w, SN_i)^k \pmod{n_0}.$$

(2) The verifier checks equation $h(m, r') = k$.

Because

$$v_i^{e_0} = h(m_w, SN_i)^{-1} \pmod{n_0},$$

$$r' = t^{e_0} v_i^{k e_0} h(m_w, SN_i)^k = t^{e_0} = r \pmod{n_0}.$$

So,

$$h(m, r') = h(m, r) = k.$$

2.3. Proxy-unprotected multi-signature scheme

Assume that according to the limit of authority, some important message m should be proxy signed by at least s proxy signers on behalf of the original signer. Without loss of generality, assume that a group of proxy signers $p_1, p_2, \ldots, p_s$ have proxy signature keys $v_1, v_2, \ldots, v_s$, respectively. They cooperatively sign the message m on behalf of the original signer.

The first proxy signer p_1 randomly chooses $t_1 \in [1, n_0]$, computes $r_1 = t_1^{e_0} \pmod{n_0}$, and sends r_1 to the second proxy signer p_2.

The second proxy signer p_2 randomly chooses $t_2 \in [1, n_0]$, computes $r_2 = r_1 \cdot t_2^{e_0} \pmod{n_0}$, and sends r_2 to the third proxy signer p_3.

...

The last proxy signer p_s randomly chooses $t_s \in [1, n_0]$, computes $r = r_s = r_{s-1} \cdot t_s^{e_0} \pmod{n_0}$, and broadcasts r.

Each proxy signer computes $k = h(m, r)$.

The first proxy signer p_1 computes $y_1 = t_1 v_1^k \pmod{n_0}$ and sends y_1 to p_2.

The second proxy signer p_2 can verify y_1 by checking the following equation:

$$y_1^{e_0} h(m_w, SN_1)^k = r_1 \pmod{n_0}.$$

Then, p_2 computes $y_2 = y_1 t_2 v_2^k \pmod{n_0}$ and sends y_2 them to p_3.

The third proxy signer p_3 can verify y_2 by checking the following equation:

$$y_2^{e_0} \big(h(m_w, SN_1) h(m_w, SN_2)\big)^k = r_2 \pmod{n_0}.$$

...

The last proxy signer p_s can verify y_{s-1} by checking the following equation:

$$y_{s-1}^{e_0} \big(h(m_w, SN_1) h(m_w, SN_2) \cdots h(m_w, SN_{s-1})\big)^k = r_{s-1} \pmod{n_0}.$$

Then, p_s computes $y = y_s = y_{s-1} t_s v_s^k \pmod{n_0}$ and broadcasts $(m, m_w, y, k, SN_1, \ldots, SN_s)$ as the proxy

signature of the message m. The verifier can verify the proxy signature as follows:

(1) The verifier computes

$$r' = y^{e_0}\big(h(m_w, SN_1)h(m_w, SN_2)\cdots h(m_w, SN_s)\big)^k \pmod{n_0}.$$

(2) The verifier checks equation $h(m, r') = k$.

Each proxy signer can verify the proxy signature by checking the verification equation. If the equation does not hold, they can look for the error by a binary search. Let $i = \lceil s/2 \rceil$. First, p_i validates the data sent by his predecessor. If the data is valid, then the error comes from $p_i, p_{i+1}, \ldots, p_s$, otherwise the error comes from $p_1, p_2, \ldots, p_{i-1}$. Hence, they can find the error by $\log_2 t$ verifications.

This scheme is very efficient. No matter how many proxy signers cooperatively sign a message, the computation load for verifiers would almost remain constant.

3. The proxy-protected signature scheme

3.1. Proxy generation

Suppose that the original signer p_0 has private key (p_0, q_0, d_0) and public key (n_0, e_0), where e_0 should be larger than the output of the one-way hash function $h()$.

For each proxy user p_i with identity ID_i and public key (n_i, e_i), the original signer computes proxy signature key

$$v_i = h(m_w, ID_i)^{-d_0} \pmod{n_0},$$

$$u_i = \lfloor v_i/n_i \rfloor \quad \text{and} \quad w_i = v_i^{e_i} \pmod{n_i}.$$

Then he sends (w_i, m_w, u_i) to the proxy user p_i. Here m_w is a warrant, which records the delegation policy including limits of authority, valid periods of delegation and proxy signature, and the identity and the public key of the original signer.

Each proxy signer recovers the proxy signature key by $v_i = u_i \times n_i + (w_i^{d_i} \bmod n_i)$, which can be verified by checking the following equation:

$$v_i^{e_0} h(m_w, ID_i) = 1 \pmod{n_0}.$$

3.2. Proxy-protected mono-signature

Assume that according the limit of authority, the proxy signer p_i has right to proxy sign a message m on behalf of the original signer. He does the following steps:

(1) The proxy signer randomly chooses an integer $t \in [1, n_0]$ and computes $r = t^{e_0} \pmod{n_0}$.
(2) The proxy signer computes $k = h(m, r)$ and $u = k^{d_i} \pmod{n_i}$.
(3) The proxy signer computes $y = tv_i^k \pmod{n_0}$.

The proxy signature of the message m is (m, m_w, y, u, ID_i). The verifier can verify the proxy signature as follows:

(1) The verifier computes $k' = u^{e_i} \pmod{n_i}$.
(2) The verifier computes

$$r' = y^{e_0} h(m_w, ID_i)^{k'} \pmod{n_0}.$$

(3) The verifier checks equation $h(m, r') = k'$.

Because

$$v_i^{e_0} = h(m_w, ID_i)^{-1} \pmod{n_0},$$

$$k' = u^{e_i} = k^{d_1 e_i} \pmod{n_i} = k,$$

$$r' = t^{e_0} v_i^{ke_0} h(m_w, ID_i)^k = t^{e_0} = r \pmod{n_0}.$$

So,

$$h(m, r') = h(m, r) = k = k'.$$

3.3. Proxy-protected multi-signature scheme

Assume that according the limit of authority, some important message m should be proxy signed by at least s proxy signers on behalf of the original signer. Without loss of generality, assume that a group of proxy signers $p_1, p_2, \ldots, p_s$ have proxy signature keys $v_1, v_2, \ldots, v_s$ and public keys $n_1, e_1, n_2, e_2, \ldots, n_s, e_s$, respectively, with $n_1 < n_2 < \cdots < n_s$. They cooperatively sign the message on behalf of the original signer.

The first proxy signer p_1 randomly chooses $t_1 \in [1, n_0]$, computes $r_1 = t_1^{e_0} \pmod{n_0}$, and sends r_1 to the second proxy signer p_2.

The second proxy signer p_2 randomly chooses $t_2 \in [1, n_0]$, computes $r_2 = r_1 \cdot t_2^{e_0} \pmod{n_0}$, and sends r_2 to the third proxy signer p_3.

...

The last proxy signer p_s randomly chooses $t_s \in [1, n_0]$, computes $r = r_s = r_{s-1} \cdot t_s^{e_0} \pmod{n_0}$, and broadcasts r.

Each proxy signer computes $k = h(m, r)$.

The first proxy signer p_1 computes $y_1 = t_1 v_1^k \pmod{n_0}$, $k_1 = k^{d_1} \pmod{n_1}$ and sends (y_1, k_1) to p_2.

The second proxy signer p_2 can verify (y_1, k_1) by checking the following equations:

$$y_1^{e_0} h(m_w, ID_1)^k = r_1 \pmod{n_0} \quad \text{and}$$

$$k_1^{e_1} \pmod{n_1} = k.$$

Then, p_2 computes $y_2 = y_1 t_2 v_2^k \pmod{n_0}$, $k_2 = k_1^{d_2} \pmod{n_2}$ and sends (y_2, k_2) them to p_3.

The third proxy signer p_3 can verify (y_2, k_2) by checking the following equations:

$$y_2^{e_0} \big(h(m_w, ID_1) h(m_w, ID_2)\big)^k = r_2 \pmod{n_0} \quad \text{and}$$

$$(k_2^{e_2} \bmod n_2)^{e_1} \pmod{n_1} = k.$$

...

The last proxy signer p_s can verify (y_{s-1}, k_{s-1}) by checking the following equations:

$$y_{s-1}^{e_0} \big(h(m_w, ID_1) h(m_w, ID_2) \cdots h(m_w, ID_{s-1})\big)^k = r_{s-1} \pmod{n_0},$$

$$\big((k_{s-1}^{e_{s-1}} \bmod n_{s-1})^{e_{s-2}} \cdots \bmod n_2\big)^{e_1} \pmod{n_1} = k.$$

Then, p_s computes $y = y_s = y_{s-1} t_s v_s^k \pmod{n_0}$, $u = k_s = k_{s-1}^{d_s} \pmod{n_s}$ and broadcasts $(m, m_w, y, u, ID_1, \ldots, ID_s)$ as the proxy signature of the message m. The verifier can verify the proxy signature as follows:

(1) The verifier computes $k' = ((u^{e_s} \bmod n_s)^{e_{s-1}} \cdots \bmod n_2)^{e_1} \pmod{n_1}$.
(2) The verifier computes $r' = y^{e_0} (h(m_w, ID_1) \times h(m_w, ID_2) \cdots h(m_w, ID_s))^{k'} \pmod{n_0}$.
(3) The verifier checks equation $h(m, r') = k'$.

Each proxy signer can verify the proxy signature by checking the verification equation. If the equation does not hold, they can look for the error by a binary search. Let $i = \lceil s/2 \rceil$. First, p_i validates the data sent by his predecessor. If the data is valid, then the error comes from $p_i, p_{i+1}, \ldots, p_s$, otherwise the error comes from $p_1, p_2, \ldots, p_{i-1}$. Hence, they can find the error by $\log_2 t$ verifications.

3.4. Proxy-protected signature scheme with anonymous proxy signers

In some application cases, proxy signers want to protect the privacy of personal information, such as identities, locations and work units. They do not wish that verifiers can know that they are in the employ of somebody from proxy signatures.

To prevent verifiers from knowing the proxy signer's real identity, the proxy signer applies to a CA for an anonymous public key certificate. The anonymous certificate includes a new public key and a pseudonym. The CA issuing the anonymous certificate can establish a link from the pseudonym to the real identity of the proxy signer.

Assume that a proxy signer p_i has an anonymous public key (n_i, e_i). He sends it to the original signer. The original signer computes proxy signature key $v_i = h(m_w, n_i, e_i)^{-d_0} \pmod{n_0}$. Then the proxy signer uses the proxy signature key together with his private key d_i to compute proxy signatures on behalf of the original signer. Verifiers use the anonymous public keys (n_i, e_i) and the public key of the original signer to verify the proxy signatures. The detailed operation is similar to those of Sections 3.2, 3.3.

Certainly, the original signer must know the real identity of his proxy signers. The anonymous public key certificate is not anonymous for the original signer.

4. Security and performance

4.1. Security discussion

The proposed proxy-unprotected signature scheme is indeed a generalization of the Guillou–Quisquater signature scheme. The original signer plays the role of the Trusted Authority TA. There has not been any security flaw found in the Guillou–Quisquater signature scheme, though it has not been proven secure.

However, the original signer can generate the same proxy signature for any message as the proxy signers create. Outsiders have no method to identify who indeed signs the message between the original signer and the proxy signers. If the original signer is ready to bear all responsibility for proxy signatures, it is enough for the verifiers to validate both the proxy signature and the warrant. As long as proxy signatures are valid and the original signer would not disavow the responsibility for delegation, the verifiers would not mind who indeed signs the message.

The proposed proxy-protected signature scheme is the combination of the Guillou–Quisquater signature scheme and the RSA signature scheme. Without the knowledge of the private keys of proxy signers, any attacker, including the original signer, is not able to compute the partial signature u from k. The proxy signers cannot generate proxy signatures without the proxy signature keys given by the original signer either. Hence any outsider can not only validate proxy signatures but also identify the actual signers. The actual signers must bear their responsibility for the proxy signatures either.

The proxy-protected signature scheme with anonymous proxy signers is a generalization of proxy-protected signature scheme with known signers.

4.2. Performance

Most of existing proxy signature schemes would require some additional computation, communication and storage compared with the corresponding ordinary signature schemes. However, the proposed proxy-unprotected signature scheme is not so. Moreover, no matter how many proxy signers cooperatively sign a message, the computation load for verifiers would almost remain constant.

To achieve the traceability property and the responsibility of the actual signers, the proposed proxy-protected signature scheme would require some more computation, communication and storage as the RSA signature scheme and the Guillou–Quisquater signature scheme. The computations for both generation and verification of proxy signatures are all proportional to the number of actual signers. The length of proxy signatures become longer too.

Therefore the proposed proxy-protected signature scheme is less efficient than the proposed proxy-unprotected signature scheme.

To protect the privacy of proxy signers, the author has proposed a proxy-protected signature scheme with anonymous proxy signers. By use of anonymous public key certificates, outsiders can verify proxy signatures with the public key of the original signer and the anonymous public keys. Though the actual signers are unknown for outsiders, the original signer can identify them.

5. Conclusion

Based on factoring problem, the author has proposed two proxy signature schemes. One is direct generalization of the Guillou–Quisquater signature scheme, which inherits the advantage of the Guillou–Quisquater signature scheme and does not need additional computation, communication and storage. The other is a combination of the Guillou–Quisquater signature scheme and the RSA signature scheme. Though it needs some more computation, communication and storage, any outsider can identify the actual signers who indeed proxy sign the message. This property avoids possible disputes who should be responsible for the proxy signature besides the original signer.

Finally, to protect the privacy of proxy signers, the author has proposed a proxy-protected signature scheme with anonymous proxy signers by use of anonymous public key certificates.

References

[1] M. Mambo, K. Usuda, E. Okamoto, Proxy signatures: Delegation of the power to sign messages, IEICE Trans. Fundam. E79-A (9) (1996) 1338–1354.
[2] M. Mambo, K. Usuda, E. Okamoto, Proxy signatures for delegating signing operation, in: Proc. 3rd ACM Conference on Computer and Communications Security, New Dehli, India, ACM Press, New York, 1996, pp. 48–57.
[3] B.C. Neuman, Proxy-based authorization and accounting for distributed systems, in: Proc. 13th International Conference on Distributed Systems, 1993, pp. 283–291.
[4] W. Diffie, M.E. Hellman, New directions in cryptography, IEEE Trans. IT-22 (1976) 644–654.
[5] R.L. Rivest, A. Shamir, L. Adelman, A method for obtain digital signatures and public-key cryptosystem, Comm. ACM 21 (2) (1978) 120–126.

[6] T. ElGamal, A public key cryptosystem and a signature scheme based on discrete logarithms, IEEE Trans. IT-31 (1985) 469–472.
[7] H.-M. Sun, N.-Y. Lee, T. Hwang, Threshold proxy signatures, IEE Proc.—Comput. Digit. Tech. 146 (5) (1999) 259–263.
[8] H.-M. Sun, An efficient nonrepudiable threshold proxy signature scheme with known signers, Comput. Comm. 22 (1999) 717–722.
[9] H.-M. Sun, Design of time-stamped proxy signatures with traceable receivers, IEE Proc.—Comput. Digit. Tech. 147 (6) (2000) 462–466.
[10] K. Zhang, Threshold proxy signature schemes, in: 1997 Information Security Workshop, Japan, September 1997, pp. 191–197.
[11] S. Kim, S. Park, D. Won, Proxy signatures, revisited, in: ICICS'97, in: Lecture Notes in Comput. Sci., Vol. 1334, Springer, Berlin, 1997, pp. 223–232.
[12] L. Yi, G. Bai, G. Xiao, A new type of proxy signature scheme, Electron. Lett. 36 (6) (2000) 527–528.
[13] L.G. Guillou, J.-J. Quisquater, A 'paradoxical' identity-based signature scheme resulting from zero-knowledge, in: Advances in Cryptology—CRYPTO'88 Proceedings, Springer, Berlin, 1990, pp. 216–231.
[14] N. Zhang, Q. Shi, M. Merabti, Anonymous public-key certificates for anonymous and fair document exchange, IEE Proc.—Comput. Digit. Tech. 147 (6) (2000) 345–350.