SCIS 2001 The 2001 Symposium on Cryptography and Information Security
Oiso, Japan, January 23-26, 2001
The Institute of Electronics, Information and Communication Engineers

Strong Proxy Signature and its Applications

Byoungcheon Lee * sultan@icu.ac.kr
Heesun Kim * sezsez@icu.ac.kr
Kwangjo Kim * kkj@icu.ac.kr

Abstract— Proxy signature is a signature scheme that an original signer delegates his/her signing capability to a proxy signer, and then the proxy signer creates a signature on behalf of the original signer. In this paper we show various attack scenarios against previous proxy signature schemes, which shows that proxy signature schemes should be designed very carefully. Based on these weaknesses, we provide new classifications of proxy signatures; strong vs. weak proxy signatures, designated vs. non-designated proxy signatures, and self-proxy signatures. We construct a simple and efficient strong non-designated proxy signature scheme and apply it to multi-proxy signature when plural delegations of multiple original signers exist. We also show that self-proxy signature can be applied to partially blind signatures.

Keywords: strong non-designated proxy signature, strong undeniability, prevention of misuse, multi-proxy signature, partially blind signature.

1 Introduction

Proxy signature is a signature scheme that an original signer delegates his/her signing capability to a proxy signer, and then the proxy signer creates a signature on behalf of the original signer. When a receiver verifies a proxy signature, he verifies the signature itself and original signer's delegation together. The basic methodology of proxy signature is that the original signer creates a signature on delegation information (ID of the proxy signer, or any warrant information) and gives it secretly to the proxy signer, and then the proxy signer uses it as a proxy private key or uses it to generate a proxy private key. Because the proxy key pair is generated from original signer's signature on delegation information, any verifier can check original signer's agreement from a proxy signature.

Mambo *et. al.* [MUO96] firstly introduced the concept of proxy signature. They classified proxy signatures based on delegation type as full delegation, partial delegation, and delegation by warrant. Partial delegation is further classified as proxy-unprotected and proxy-protected according to protection of proxy signer. They gave various constructions of proxy signature schemes and their security analysis. Kim *et. al.* [KPW97] extended them by using Schnorr signature and including warrant information in partial delegation schemes. Petersen and Horster [PH97] proposed to use weak blind signature to get proxy-protected proxy signature scheme.

Some security requirements of proxy signatures can be listed as follows [MUO96].

(i) Strong unforgeability A designated signer, called proxy signer, can create a valid proxy signature for the original signer. But the original signer and third parties who are not designated as a proxy signer cannot create a valid proxy signature.

(ii) Verifiability From proxy signature a verifier can be convinced of the original signer's agreement on the signed message either by a self-authenticating form or by an interactive form.

(iii) Strong identifiability Anyone can determine the identity of the corresponding proxy signer from a proxy signature.

(iv) Strong undeniability Once a proxy signer creates a valid proxy signature for an original signer, the proxy signer cannot repudiate his signature creation against anyone.

But the basic constructions of [MUO96] and the secure key issuing of [PH97] do not satisfy the strong undeniability property, i.e., the proxy signer can repudiate the fact that he has created the signature. It is because the proxy key pair does not contain any authentic information of the proxy signer. We will show various attacks against previous proxy signature schemes in section 2.

Based on these weaknesses, we classify proxy signature schemes into strong and weak ones according to undeniability property. Strong proxy signature represents both original signer's and proxy signer's signatures, while weak proxy signature represent only original signer's signature. We also classify proxy signature schemes into designated and non-designated ones according to designation of proxy signer. We will show that strong proxy signatures can be used without designation of proxy signer in the proxy key issuing stage because the resulting proxy signature has explicit authentic information of the proxy signer. This strong

* Information Security Group, Information and Communications Univ., 58-4 Hwaam-dong, Yusong-gu, Taejon, 305-732, Korea

non-designated proxy signature can be used for many applications in very flexible way.

We also consider a possibility of self-proxy signature in which original signer creates a proxy key pair for herself, i.e., original signer and proxy signer are the same party. Abe and Okamoto [AO00] proposed partially blind signatures which are an extension of blind signature schemes that allow a signer to explicitly include necessary information in the resulting signature under some agreement with the receiver. We show that self-proxy signature can be used very efficiently for this application.

In section 2, we briefly review some previous proxy signature schemes, show various attack scenarios against these schemes, and suggest some countermeasures for them. Based on these weaknesses, we provide new classifications of proxy signatures in section 3. In section 4, we show a simple and efficient construction of strong non-designated proxy signature scheme and its application to multi-proxy signature when plural delegations of multiple original signers exist. In section 5, we show an application of self-proxy signature to partially blind signature. We conclude in section 6.

2 Review on Proxy Signature Schemes

In this section we briefly review some selected proxy signature schemes and show possible attack scenarios against them. These schemes are commonly based on discrete logarithm problem and Schnorr signature, so we firstly review Schnorr signature briefly.

Let p and q be large primes with $q|p-1$. Let g be a generator of a multiplicative subgroup of Z_p^* with order q. $h(\ )$ denotes a collision resistant hash function. Assume that a signer A has a private key x_A and the corresponding public key $y_A = g^{x_A}$. To sign a message m, A chooses a random number $k \in_R Z_q^*$ and computes $r = g^k$, $s = x_A h(m,r) + k$. Then the tuple (m, r, s) becomes a valid signed message. The validity of signature is verified by $g^s \stackrel{?}{=} y_A^{h(m,r)} r$.

This signature scheme has been proven to be secure under the random oracle model [PS96]. They have shown that existential forgery under the adaptive chosen message attack is equivalent to the discrete logarithm problem.

2.1 Petersen and Horster's Scheme

[PH97] proposed self-certified keys and used them for proxy signature schemes. They used basic key issuing (proxy-unprotected) and secure key issuing (proxy-protected) protocols to issue proxy key pair. Assume that an original signer A has certified key pair (x_A, y_A) and tries to issue a proxy key pair (x_P, y_P) to a proxy signer B.

Basic key issuing protocol: In this protocol A generates a proxy key pair (x_P, y_P) by herself and sends it secretly to B.

A chooses a random number $k_A \in_R Z_q^*$ and computes $r_A = g^{k_A}$, $s_A = x_A h(ID_B, r_A) + k_A$ where ID_B is B's identity. The tuple (r_A, s_A) is A's valid signature on ID_B. A sends (r_A, s_A) to B secretly. Then B uses $x_P \equiv s_A$ as a proxy private key and $y_P \equiv g^{x_P}$ as the corresponding proxy public key if the verification

$$y_P \stackrel{?}{=} y_A^{h(ID_B, r_A)} r_A \tag{1}$$

holds. This is a proxy-unprotected protocol because A also knows the proxy private key x_P.

Secure key issuing protocol: In this protocol A and B execute the following 3-pass weak blind signature protocol for A to issue a proxy key pair (x_P, y_P) to B.

1. A chooses $\tilde{k}_A \in_R Z_q^*$, computes $\tilde{r}_A = g^{\tilde{k}_A}$, and sends $\tilde{r}_A$ to B.

2. B chooses $b \in_R Z_q^*$, computes $r_A = \tilde{r}_A g^b$, and sends (ID_B, r_A) to A.

3. A computes $\tilde{s}_A = x_A h(ID_B, r_A) + \tilde{k}_A$ and sends $\tilde{s}_A$ to B.

4. B computes his proxy private key as $x_P \equiv \tilde{s}_A + b$. The tuple (r_A, x_P) is A's valid signature on ID_B. He verifies the validity of A's signature by
$$y_P \equiv g^{x_P} \stackrel{?}{=} y_A^{h(ID_B, r_A)} r_A. \tag{2}$$
Then the proxy public key is y_P.

This is a proxy-protected protocol because the proxy private key x_P is hidden from the original signer A.

Now B can create a proxy signature for a message m on behalf of A, and any verifier can check the validity of proxy signature as follows.

Signing by proxy signer: B generates a proxy signature $\sigma = S(x_P, m)$ on message m using his proxy private key x_P where $S(\ )$ is a general signature generation algorithm. The tuple $(m, \sigma, ID_B, r_A, y_A)$ is a valid proxy signature.

Verification of a proxy signature: A verifier checks the validity of a proxy signature by

$$V(y_A^{h(ID_B, r_A)} r_A, m, \sigma) \stackrel{?}{=} true$$

where $V(\ )$ is a general signature verification algorithm.

2.2 Kim, Park and Won's Scheme

The above secure key issuing protocol requires 3-pass interaction between A and B to issue a proxy-protected key pair. [MUO96] suggested a non-interactive proxy-protected key issuing protocol using B's authentic key pair (x_B, y_B). But in this scheme original signer's signature parameter does not contain any information on proxy signer's identity or any warrant, so there are possibilities of misuse as will be shown later. [KPW97] extended it by using Schnorr signature and warrant m_w. In this scheme A chooses a random number $k_A \in_R Z_q^*$ and computes signature parameter as $r_A = g^{k_A}$, $s_A =$

$x_A h(m_w, r_A) + k_A$. The tuple (r_A, s_A) is A's valid signature on m_w. A sends (m_w, r_A, s_A) to B secretly. Then B verifies

$$g^{s_A} \stackrel{?}{=} y_A^{h(m_w, r_A)} r_A.$$

If this verification holds, B computes his proxy key pair as

$$x_P = s_A + h(m_w, r_A) x_B,$$
$$y_P \equiv g^{x_P} = (y_A y_B)^{h(m_w, r_A)} r_A. \tag{3}$$

This is a proxy-protected signature scheme because the proxy private key x_P can be computed only by B who knows the private key x_B.

2.3 Attacks against Proxy Signature Schemes

The approach of [PH97] that uses weak blind signature to get proxy-protected proxy signature scheme does not seem to work. Although A issues a proxy key pair to B using the secure key issuing protocol, the following types of attacks are possible.

1) Proxy signer's repudiation: Since a proxy signature $(m, \sigma, ID_B, r_A, y_A)$ does not contain any authentic information of B, the proxy signer B can repudiate his signature creation later and argue that it was created by A. From the point of third verifiers, basic key issuing and secure key issuing protocols are indistinguishable, i.e., a third verifier cannot determine whether the proxy key pair was issued by using basic or secure key issuing (equation (1) and (2) are same). When B repudiates his proxy signature creation, a verifier cannot determine who is misbehaving.

2) Proxy signer's misuse: Another attack scenario is that B gets a proxy key pair (x_P, y_P) from A and registers it to CA as his own key pair. He can use it as his own key pair for any purpose. When he did something wrong, he can repudiate his signature creation by showing that his signature is actually A's proxy signature. He can argue that he has not created it or at least he can share his responsibility with A.

3) Original signer's misuse: Original signer A can do similar attack. She can create a proxy key pair with ID_B (and restricted warrant information if it is used) without B's agreement, and then register it to CA as her another key pair. When she did something wrong with the new key pair, she can repudiate her signature creation by showing that her signature is actually a proxy signature created by B which does not conform to the warrant.

In [MUO96] an original signer A gives her signature parameter secretly to a proxy signer B which does not contain any information on proxy signer's identity or warrant. B can generate a proxy key pair using his certified key pair. If B gives A's signature parameter to another party C, he can also generate a valid proxy key pair in the same way. Because A does not include any information on proxy signer in her signature parameter, a verifier cannot determine whether C is a valid proxy signer or not.

[KPW97] has solved this problem because its signature parameter is a Schnorr signature on warrant m_w. But m_w should state the delegation relation explicitly because there are possibilities that any valid signature of A can be used to generate a proxy key pair. Moreover the roles of original signer and proxy signer are symmetric as shown in equation (3). If m_w does not specify the roles of participants explicitly, a proxy signature can be argued to be created by the original signer on behalf of the proxy signer.

2.4 Countermeasures against these Attacks

Since proxy key pair can be used for other purposes as shown above, proxy signature schemes should be designed very carefully. It is very difficult to assume trustedness of original signer, proxy signer and the proxy key issuing protocol between them.

The simplest countermeasure against these attacks is using proxy signer's authentic key pair to generate proxy key pair as shown in the proxy-protected signature schemes of [MUO96] and [KPW97]. Based on the basic assumption of public key cryptography that the private key of authentic key pair is kept secretly by the legitimate user, valid proxy key pair can be computed only by the proxy signer. So any deviation of the protocol can be determined as proxy signer's responsibility.

Another countermeasure is using explicit warrant information. It should state the delegation information explicitly such that any deviation of the protocol cannot happen.

So we need additional security requirement of proxy signature schemes.

(v) Prevention of misuse It should be confident that proxy key pair cannot be used for other purposes. In the case of misuse, the responsibility of proxy signer should be determined explicitly.

3 New Classifications of Proxy Signature Schemes

As shown above, proxy signature schemes should be designed carefully for the proxy key pair not to be used for other purposes. In this section we provide new classifications of proxy signature schemes.

According to the undeniability property, we classify proxy signature schemes into strong and weak.

- Strong proxy signature: It represents both original signer's and proxy signer's signatures. Once a proxy signer creates a valid proxy signature, he cannot repudiate his signature creation against anyone.

- Weak proxy signature: It represents only original signer's signature. It does not provide non-repudiation of proxy signer.

Weak proxy signature cannot be used in real world because there can be many deviations as shown above. Strong proxy signature can be used widely without relying on any trustedness assumption.

If proxy key pair contains authentic information of the proxy signer as [KPW97], original signer does not need to designate proxy signer in proxy key issuing stage. In this sense we further classify proxy signature schemes into designated and non-designated proxy signatures according to designation of proxy signer in proxy key issuing stage.

- Designated proxy signature: In this scheme original signer specify a proxy signer in proxy key issuing stage. ID of proxy signer is included in original signer's signature parameter.
- Non-designated proxy signature: In this scheme original signer does not specify a proxy signer in proxy key issuing stage. Anyone who has original signer's signature parameter can generate own proxy key pair non-interactively. He can create proxy signature on behalf of original signer if the message conforms to the warrant information.

One more possibility is self-proxy signature in which the original signer issues a proxy key pair for herself, i.e., original signer and proxy signer are the same party. In later section we will show that self-proxy signature can be used for partially blind signature.

- Self-proxy signature: In this scheme the original signer issues a proxy key pair for herself and uses it as her new key pair.

4 Strong Non-Designated Proxy Signature and its Application to Multi-Proxy Signature

Based on above classifications, [KPW97] is a strong designated proxy signature. But there are many applications that original signer cannot determine in proxy key issuing stage who will be proxy signer. [KBLK01] shows an example of applying proxy signature scheme to mobile agent. When a customer prepares a mobile agent (proxy key issuing stage), he does not know which shopping mall will propose conforming deal to his requirement. Proxy signatures without designation of proxy signer can be used in many applications in more flexible and efficient way.

To construct a strong non-designated proxy signature, we modify [KPW97] slightly such that m_w does not include identity of proxy signer and asymmetry of roles is achieved.

4.1 Strong Non-Designated Proxy Signature

Proxy key issuing: Original signer A chooses a random number $k_A \in_R Z_q^*$ and computes $r_A = g^{k_A}$, $s_A = x_A h(m_w, r_A) + k_A$ where m_w is warrant information. m_w does not contain proxy signer's ID, but should state application dependent delegation information explicitly. The tuple (m_w, r_A, s_A) is A's signature on m_w. A sends (m_w, r_A, s_A) to B secretly (or publishes it in restricted group of concerned parties). Then B verifies

$$g^{s_A} \stackrel{?}{=} y_A^{h(m_w, r_A)} r_A.$$

If this verification holds, B computes his proxy key pair as

$$x_P = s_A + x_B, \;\; y_P \equiv g^{x_P} = y_A^{h(m_w, r_A)} r_A y_B. \qquad (4)$$

Although the original signer has not specified a proxy signer, anyone can generate his own proxy key pair secretly and create a proxy signature on behalf of the original signer if his message m conforms to the warrant information m_w. A verifier can identify the proxy signer from the proxy public key. Note that y_A and y_B are used in different ways in equation (4), which gives the asymmetry of roles of A and B.

Signing by proxy signer: If a message m conforms to m_w, proxy signer B can create a proxy signature on m as $\sigma = S(x_P, m)$ using his proxy private key x_P. The tuple $(m, \sigma, m_w, r_A, y_A, y_B)$ is a valid proxy signature.

Verification of proxy signature: A recipient can verify $(m, \sigma, m_w, r_A, y_A, y_B)$ by checking $m \in \{m_w\}$ and

$$V(y_A^{h(m_w, r_A)} r_A y_B, m, \sigma) \stackrel{?}{=} true. \qquad (5)$$

We show that the proposed strong non-designated proxy signature scheme satisfies all the security requirements listed above.

Theorem 1 *Proposed strong non-designated proxy signature scheme satisfies all the listed security requirements of* $(i) \sim (v)$.

Proof:

(i) Strong unforgeability: Anyone except the proxy signer B cannot generate a valid proxy key pair under the name of B because it contains proxy signer's private key x_B. Only the legitimate proxy signer can create a valid proxy signature.

(ii) Verifiability: Original signer's agreement on the warrant information m_w can be confirmed in verification equation (5). If message m conforms to m_w, the proxy signature is valid.

(iii) Strong identifiability: Identity information of a proxy signer is included explicitly in a valid proxy signature as a form of public key y_B. So anyone can determine the identity of the corresponding proxy signer.

(iv) Strong undeniability: Once a proxy signer creates a valid proxy signature, he cannot repudiate it because the proxy key pair can be computed only by himself.

(v) Prevention of misuse: If a proxy signer uses the proxy key pair for other purposes that are not specified in m_w, it is his responsibility because he is the only person who can generate it. So the scenario of proxy signer's misuse is not possible. Original signer's misuse is also prevented because she cannot compute a valid proxy key pair under the name of the proxy signer.

□

Note that the proposed proxy signature scheme satisfies strong undeniability although original signer did not specify proxy signer in proxy key issuing stage. It is because proxy signer uses his authentic key pair to generate a proxy key pair.

Including proxy signer's signature in proxy signature scheme is very natural approach. Proposed strong proxy signature represents both original signer's and proxy signer's signatures. Computational overhead for this additional functionality is just one exponentiation.

4.2 Application to Multi-Proxy Signature

Because proposed strong non-designated proxy signature scheme does not specify proxy signer in proxy key issuing stage, it can be applied to many real applications in more flexible and efficient ways. It can be applied to multi-proxy signature in which multiple original signers delegate their signing capabilities to unspecified proxy signers. Let's consider following scenarios.

In a company there will be many departments such as personnel, financial, business, and general affair. An employee wants to get signatures for some typical message from these departments. If the message is very typical, those departments can delegate their signing capabilities to employees with explicit warrant. If employee's message conforms to the warrant, the employee can create multiple proxy signatures by himself.

Alternatively, an employee wants to get signatures for a proposal from his section manager, department manager, director, and president. He already reviewed his proposal with them several times and his final proposal has minor corrections. If his plural bosses delegate their signing capabilities to him with the specification of corrections, he can create multiple proxy signatures by himself without any further communication with them. The scenario of plural delegations without specifying proxy signer is very common in real applications.

Let A_i $(i = 1, ..., n)$ denote plural original signers who have certified key pairs (x_i, y_i) and warrant informations m_i. They try to delegate their signing capabilities to unspecified proxy signers. Let B be a proxy signer who has certified key pair (x_B, y_B). He is willing to create a proxy signature on behalf of $\{A_1, ..., A_n\}$ under warrants $\{m_1, ..., m_n\}$.

In proxy key issuing stage plural original signers A_i choose random numbers $k_i \in_R Z_q^*$ and compute $r_i = g^{k_i}$, $s_i = x_i h(m_i, r_i) + k_i$. The tuple (m_i, r_i, s_i) is A_i's valid signature on m_i. A_i sends (m_i, r_i, s_i) to B secretly (or publishes it in restricted group of concerned parties). Then B verifies

$$g^{s_i} \stackrel{?}{=} y_i^{h(m_i, r_i)} r_i.$$

If B wants to create a proxy signature on behalf of $\{A_1, ..., A_n\}$ under warrants $\{m_1, ..., m_n\}$, he generate a proxy key pair (x_P, y_P) as

$$x_P = s_1 + \cdots + s_n + x_B, \quad y_P = g^{x_P}.$$

If his message m conforms to $\{m_1, ..., m_n\}$, B can create a proxy signature on m as $\sigma = S(x_P, m)$. The tuple

$$(m, \sigma, m_1, r_1, y_1, ..., m_n, r_n, y_n, y_B)$$

is a valid proxy signature.

A verifier can check the validity of proxy signature by checking $m \in \{m_1, ..., m_n\}$ and

$$V(y_P, m, \sigma) \stackrel{?}{=} true$$

where $y_P = y_1^{h(m_1, r_1)} r_1 \cdots y_n^{h(m_n, r_n)} r_n y_B$.

4.3 Comparison with multiple signatures

As stated in [MUO96], proxy signature schemes of partial delegation are more efficient than those of delegation by warrant. Consider a traditional approach of multiple independent signatures that original signers A_i publish their signatures (m_i, r_i, s_i) and the proxy signer B just signs on m with his certified key pair (x_B, y_B). The proposed multi-proxy signature scheme is more efficient than the traditional approach of multiple independent signatures in the following sense.

- Message size is reduced by $n|q|$ because $(s_1, ..., s_n)$ are not necessary in proposed scheme.
- A valid signature can be created by the proxy signer himself without any interaction with original signers, while traditional scheme requires n communications with original signers.
- Verification of signature is more efficient because proposed scheme requires only $n + 2$ exponentiations (one signature verification and n exponentiations) while traditional scheme requires $2(n+1)$ exponentiation for $n + 1$ signature verifications. Moreover, simultaneous multiple exponentiation with distinct bases can be computed very efficiently [MOV97].

Proposed scheme can be used in very flexible way because proxy signer can choose different combinations of delegations by himself depending on the property of his message.

5 Self-Proxy Signature and its Application to Partially Blind Signature

5.1 Self-Proxy Signature

Self-proxy signature is a scheme in which original signer issues a proxy key pair for herself, i.e., original signer and proxy signer are the same party. In this scheme original signer issues a proxy key pair which contains self-delegation information and uses it as her new key pair. A verifier checks the validity of new key pair by verifying the self-delegation relation.

This scheme can be applied to user key renewal. Based on a certified key pair (x, y), user can generate new key pairs (x_i, y_i) without any interaction with CA. In this section we show that self-proxy signature can be used for partially blind signature.

5.2 Application to Partially Blind Signature

Blind signature, firstly introduced by Chaum [Cha82], allows a receiver to get a signature without giving the signer any information about the message or the resulting signature. This blindness property plays a central role in applications such as electronic cash where anonymity is of prime concern. But one shortcoming is that the signer has no control over the attributes except for those bound by the public key. Another shortcoming can be seen in a simple electronic cash scheme where a bank issues blind signatures as electronic coins. Since the bank cannot inscribe the coin values on the blindly issued coins, it has to use different public keys for different coin values.

[AO00] suggested a concept of partially blind signature which is an extension of blind signature that allows a signer to explicitly include necessary information (expiration date, collateral conditions, or whatever) in the resulting signature under some agreement with the receiver. They proposed secure and efficient schemes based on the technique of witness indistinguishable protocols.

In this section we suggest that the same functionality can be achieved more efficiently by using self-proxy signature and blind signature sequentially. Assume that the bank has a certified key pair (x, y) and n coin values $c_1, ..., c_n$. It can create n key pairs (x_i, y_i) for coin values c_i and whatever necessary information using the proxy key issuing with self-delegation information. Then it can issue electronic coins to customers by using blind signature with (x_i, y_i) as new authentic key pairs for coin values c_i. Customers can verify the validity of coins through the blind signature and coin values through the self-delegation relation. Using this technique, the bank can create coins of different values with a single certified key pair.

The security of this scheme depends on those of proxy signature and blind signature schemes. More detailed constructions and analysis will be given in full paper.

6 Conclusion

We have presented some attacks against previous proxy signature schemes. Proxy signer can repudiate his signature creation later if a proxy signature does not contain any authentic information of the proxy signer. There are also possibilities of misuse that original signer or proxy signer try to use the proxy key pair for other purposes. To resist against these attacks, the proxy key pair should be computed from proxy signer's private key, which guarantees strong undeniability property. Any misuse of proxy key pair is determined to be proxy signer's responsibility. The warrant information m_w should state the delegation information explicitly.

Based on these arguments, we have provided new classifications of proxy signature schemes; strong vs. weak proxy signature, designated vs. non-designated proxy signature, and self-proxy signature. We have provided a simple and efficient construction of strong non-designated proxy signature and applied it to multi-proxy signature when plural delegations of multiple original signers exist. We also have introduced self-proxy signature and shown that it can be used as another solution for partially blind signatures.

Proxy signatures are very useful tools when one needs to delegate his/her signing capability to other party. But in distributed environment it is very difficult to assume the trustedness of original signer, proxy signer, and the proxy key issuing protocol between them. Delegating signing capability to others can be risky. So proxy signature schemes should be designed carefully such that proxy signer's responsibility is determined explicitly and any possibility of misuse is prevented.

References

[AO00] M. Abe, T. Okamoto, "Provably secure partially blind signatures", In *Advances in Cryptology: Crypto'2000*, pages 271–299, 2000.

[Cha82] D. Chaum, "Blind signatures for untraceable payments", In *Advances in Cryptology: Crypto'82*, pages 199 - 204, Prenum Publishing Corporation, 1982.

[HMP95] P. Horster, M. Michels, H. Petersen "Hidden signature schemes based on the discrete logarithm problem and related concepts", In *Proc. Communications and Multimedia Security'95*, pages 162 - 177, Chapman & Hall, 1995.

[KBLK01] H. Kim, J. Baek, B. Lee, and K. Kim, "Computing with secrets for mobile agent using one-time proxy signature", to appear in *SCIS'2001*.

[KPW97] S. Kim, S. Park, and D. Won, "Proxy signatures, revisited", In *Proc. of ICICS'97, International Conference on Information and Communications Security*, Springer, Lecture Notes in Computer Science, LNCS 1334, pages 223-232, 1997.

[MOV97] A. Menezes, P. Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*, pages 617 - 618, CRC Press, 1997.

[MUO96] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signatures: Delegation of the power to sign messages", In *IEICE Trans. Fundamentals*, Vol. E79-A, No. 9, Sep., pages 1338–1353, 1996.

[PH97] H. Petersen and P. Horster, "Self-certified keys – Concepts and Applications", In *Proc. Communications and Multimedia Security'97*, pages 102 - 116, Chapman & Hall, 1997.

[PS96] D. Pointcheval and J. Stern, "Security proofs for signatures", In *Advances in Cryptology: Eurocrypt'96*, pages 387 - 398, Springer, 1996.