An Analysis of Proxy Signatures: Is a Secure Channel Necessary?

Jung-Yeun Lee[1], Jung Hee Cheon[1], and Seungjoo Kim[2]

[1] IRIS (International Research center for Information Security)
ICU (Information and Communications Univ.), Korea
{bushman,jhcheon}@icu.ac.kr
[2] Korea Information Security Agency, Korea
skim@kisa.or.kr

Abstract. A proxy signature enables the original signer to delegate her signing capability to a proxy entity, who signs a message on behalf of the original signer. In this paper, we discuss the necessity of a secure channel in proxy signatures. Though establishing a secure channel has much influence on the efficiency of the scheme, to the best of our knowledge, this topic has not been discussed before. All known proxy signatures used a secure channel to deliver a signed warrant except one which used a 3-pass weak blind signature. However, the KPW scheme [2] appeared to be secure without the secure channel. We think that our result can contribute to designing more efficient proxy signature scheme.

1 Introduction

An employee in a company needs to go on a business trip to someplace which has no computer network access. During the trip he will receive e-mails, and may be expected to respond to some messages urgently. A solution for this situation is a *proxy signatures*. Before the trip, he delegates his signing capability to his secretary (called a proxy signer), and instructs his secretary to respond to the e-mails in place of him according to a prearranged plan. Then the secretary responds to the e-mails using the proxy signature.

In order to create this kind of signature securely, it should satisfy the following requirements [5, 3].

R1. Verifiability: From the proxy signature a verifier can be convinced of the original signer's agreement on the signed message.

R2. Strong unforgeability: A designated proxy signer can create a valid proxy signature for the original signer. But the original signer and other third parties who are not designated as a proxy signer cannot create a valid proxy signature.

R3. Strong identifiability: Anyone can determine the identity of the corresponding proxy signer from the proxy signature.

R4. Strong undeniability: Once a proxy signer creates a valid proxy signature of an original signer, he cannot repudiate the signature creation.

M. Joye (Ed.): CT-RSA 2003, LNCS 2612, pp. 68–79, 2003.

R5. Prevention of misuse: The proxy signer cannot use the proxy key for other purposes than generating a valid proxy signature. That is, he cannot sign, with the proxy key, messages that have not been authorized by the original signer.

The basic idea to implement a proxy signature scheme is that the original signer creates a signature (called a proxy) on the delegation information (the identity of the designated proxy signer, valid period, instruction for signing, or any warrant information) and then the proxy signer uses it to generate a proxy private key and signs on the delegated message. Since the proxy key pair is generated using the original signer's signature on delegation information, any verifier can check the original signer's agreement from the proxy signature.

We should note here that, in [5], Mambo *et al.* emphasized that the secure channel between the original signer and the proxy signer is necessary in the proxy delivery step. Otherwise, anyone who obtained the proxy can create the valid proxy signature. In this paper, we will take a closer look at the necessity of secure channel in the proxy signatures. Because establishing a secure channel has much influence on the efficiency of the scheme, it is desirable to construct a proxy signature scheme without secure channel. Furthermore, a proxy signature scheme without secure channel does not employ encryption. Thus we don't need to consider the current debate about cryptographic policy as to whether the law enforcement should be given when authorized surreptitious access to the plaintext of encrypted messages.

Related Works. A proxy signature (MUO scheme) was first introduced by Mambo *et al.* [5]. Since its warrant does not include the information of the proxy signer, a secure channel is required in the proxy delivery step. Petersen and Horster [7] proposed a proxy-protected scheme (PH scheme) using a 3-pass weak blind signature, where the proxy private key is an ordinary signature on the identity of the proxy signer using Schnorr's signature scheme. The KPW scheme [2] specified the warrant so as to contain the identity information and the limit of the signing capability of the proxy signer in order to prevent the misuse of the signing capability by the proxy signer. Lee *et al.* [3] proposed a scheme (LKK scheme) based on the KPW scheme, where the original signer and the proxy signer do not play the same role in the generation of a proxy signature and so the verifier can identify both of them without seeing the warrant information.

Our Contribution. We analyze the necessity of the secure channel to deliver a proxy certificate for all known proxy signature schemes [5, 2, 6, 3] to make use of secure channel. We show that all schemes but the KPW scheme are insecure without the secure channel. Further we provide a heuristic proof for the security of the KPW scheme without the secure channel, and propose the modified schemes for the MUO scheme and the LKK scheme in order to be secure without the secure channel. Finally, we show that the LKK scheme does not satisfy the

strong unforgeability even with the secure channel which is different from the author's claim.

Organization. In Section 2, 3 and 5, we will review the proxy signature scheme the MUO scheme, the PH scheme and the LKK scheme respectively and analyze the functions of each scheme and the necessity of a secure channel. Then, in Section 4 we show that the KPW scheme is secure although we remove the secure channel in the proxy delivery step. In Section 6, We compare the functions and efficiency of each scheme and revise the MUO scheme and the LKK scheme so that their schemes are secure although we remove the secure channel. Finally, we conclude this paper in Section 7.

2 The MUO Scheme and Its Analysis

We review the proxy-protected proxy signature proposed in [5], which is based on the Discrete Logarithm Problem (DLP). Throughout this paper, p is a large prime with $2^{511} < p < 2^{512}$, q is a large prime with $q \mid p-1$ and g is a generator of a multiplicative subgroup of $\mathbb{Z}_p^*$ with order q. $h()$ denotes a collision resistant hash function. In addition, it is assumed that Alice is an original signer with a key pair $(x_A, y_A(= g^{x_A} \mod p))$ and Bob is a proxy signer with $(x_B, y_B(= g^{x_B} \mod p))$. We denote by m_w the warrant which contains the information of the proxy signer and by m_P the delegated message.

[Basic Protocol]

1. Generation of the proxy key: Bob gets the proxy key pair (x_P, y_P) through the following steps.

1. The original signer Alice generates a random number $k \in \mathbb{Z}_q^*$ and computes $K = g^k \mod p$. Then, she calculates
$$s_A = x_A + k \cdot K \mod q.$$
and sends (s_A, K) to Bob in a secure manner.
2. Bob checks
$$g^{s_A} \stackrel{?}{=} y_A \cdot K^K \mod p.$$
If it is passed, Bob computes the proxy private key as
$$x_P = s_A + x_B \cdot y_B.$$

2. Proxy signature generation: When Bob signs a document m_p for the sake of Alice, he executes the DLP-based ordinary signing operation with the proxy private key x_p. The created proxy signature σ is
$$(m_p, Sign_\sigma(m_p), K, y_A, y_B).$$

3. Verification: To verify the proxy signature σ, first the verifier computes the proxy public key as
$$y_P = g^{x_P} = y_A \cdot K^K \cdot y_B^{y_B}.$$
The verification is carried out by the same checking operation as in the original signature scheme.

This scheme is a proxy-protected one and satisfies all requirements except R5 of the security requirements. The requirement is not satisfied since the warrant (s_A, K) does not include the identity information and the limit of the capability of the designated proxy signer. The proxy signer also can transfer the warrant to someone else. To avoid these problems, one can manage the revocation list which represents the possession of the warrant and the limit of the signing capability of proxy signer.

As they mentioned, this scheme needs a secure channel. If we remove the secure channel, anyone who intercepts the warrant (s_A, K) can be the proxy signer of Alice. Furthermore, if he has a warrant (s_C, K') made by another user Charlie, he can change the original signer from Alice to Charlie as follows:
Let s_P be a proxy signature generated by Bob on behalf of Alice using Schnorr's scheme, i.e.,

$$\begin{aligned} s_P &= k_P + x_P \cdot h(m_p, r_P) \\ &= k_P + (x_B \cdot y_B + s_A) \cdot h(m_p, r_P) \\ &= k_P + x_B \cdot y_B \cdot h(m_p, r_P) + \underbrace{s_A \cdot h(m_p, r_P)}_{(1)}. \end{aligned}$$

Here, the underlined term (1) can be extracted from the proxy signature σ. Through the algebra

$$s_{P'} = s_P - (1) + s_C \cdot h(m_p, r_P),$$

the attacker can create the proxy signature $s_{P'}$ in which Charlie is the original signer and Bob is proxy signer. Hence, if this scheme is used without the secure channel, the proxy signer can repudiate that he/she generated the proxy signature on behalf of a specific person.

3 The PH Scheme and Its Analysis

We review the proxy-protected proxy signature proposed in [7].

[Basic Protocol]

1. **Generation of the proxy key:** Bob gets a proxy private key x_P through the 3-pass weak blind signature protocol, where $x_P = s_A = k_A + x_A \cdot h(m_w, r_A) \mod q$. Namely, Bob uses Schnorr's signature on Bob's ID of Alice as the proxy private key.
2. **Proxy signature generation:** When Bob signs a document m_p for the sake of Alice, he executes the DLP-based ordinary signing operation with the proxy private key x_p. The created proxy signature σ is

$$(m_p, Sign_\sigma(m_p), ID_B, r_A)$$

3. Verification: To verify the proxy signature σ, first the verifier computes the proxy public key as

$$y_P = g^{x_P} = y_A^{h(ID_B, r_A)} \cdot r_A \mod p.$$

The verification of the proxy signature is carried out by the checking operation of the same signature scheme.

Alice does not know the proxy private key since they used a blind signature scheme at the time of the proxy signature generation. However, consequently the proxy private key is an ordinary Schnorr's signature on the identity ID_B of Bob. So it does not contain the private information of Bob. It makes several weaknesses of the PH scheme as follows:

- Since an attacker can generate Schnorr's signature on Bob's ID and then create the proxy signature in which Bob is a proxy signer regardless of Bob's will. Thus, this scheme does not satisfies the requirements R2 and R4 and is not proxy-protected.
- Like the MUO scheme, the warrant of this scheme does not contain the limit of the signing capability and so does not protect the misuse by the proxy signer [3].

We remark that the PH scheme does not use a secure channel, however it requires a 3-pass weak blind signature protocol, which increases the communication load.

4 The KPW Scheme and Its Analysis

Kim *et al.* [2] introduced the notion of the partial delegation performed by inserting the warrant into the proxy signature, i.e., the proxy signer generates proxy signatures using his private key and the warrant signed by the original signer.

[Basic Protocol]

1. Generation of the proxy key: To delegate the signing capability to proxy signer, the original signer Alice uses Schnorr's scheme to make the signed warrant m_w. If the following process is finished successfully, Bob gets a proxy key pair (x_P, y_P).
   1. Alice chooses $k_A \in \mathbb{Z}_q^*$ at random and computes $r_A = g^{k_A} \mod p$ and $s_A = k_A + x_A \cdot h(m_w, r_A) \mod q$, and then sends (m_w, r_A, s_A) to a proxy signer Bob secretly.
   2. Bob verifies the validity of the signature on m_w. If the signature is valid, Bob computes the proxy key pair (x_P, y_P) as $x_P = h(m_w, r_A) \cdot x_B + s_A$.

2. Proxy signature generation: Bob uses any signature scheme based on the difficulty of DLP with the key pair (x_P, y_P) and obtains a signature (r_P, s_P) for the delegated message m_P. The valid proxy signature will be the tuple

$$(m_P, r_P, s_P, m_w, r_A).$$

3. Verification: A recipient can verify the validity of the proxy signature by checking that both the proxy signer and the message conform to m_w and the verification with the proxy public key

$$y_P = (y_A \cdot y_B)^{h(m_w, r_A)} \cdot r_A \mod p.$$

This proxy signature scheme uses a warrant containing the identity information and the limit of the delegated signing capability and so satisfies the security requirements R1, R3 and R4. It is a proxy-protected one in the strict sense and satisfies the requirement R2 (strong unforgeability). They said that their scheme needs the secure channel in the proxy delivery step, but their scheme is still secure without the secure channel.

In order to show that the KPW scheme satisfies the second requirement without the secure channel, we classify the security problem as follows:

1. [Attack for the role of the original signer]
   (a) An attacker creates the signature of some user on the warrant information and so impersonate the original signer.
   (b) An attacker replaces the signed warrant by other valid warrant signed by different user. Therefore, he change the original signer.
2. [Attack for the role of proxy signer] An attacker converts a normal signature into a proxy signature.

The first problem 1-(a) is overcome easily by using the signature scheme which is secure against the existential forgery. Since the KPW scheme uses Schnorr's signature scheme whose security is proved, we ignore this problem. Next we show that the KPW scheme is secure for the two cases 1-(b) and 2.

We noted that the previous schemes have several weaknesses when the warrants are revealed. the one of those weaknesses results from the divisibility of the roles of original signer and proxy signer in the proxy signature. Namely, the attacker can remove the part of original signer from that of the proxy signature and insert the warrants created by another original signer. Therefore, he changes the original signer. In order to protect this problem, the proxy signature scheme must include a part which binds the private key of proxy signer with the public parameter used in the signing process on the warrant by original signer. Fortunately, the proxy signature by the KPW scheme has those parts and so their scheme does not need a secure channel.

More precisely, we assume that an attacker Charlie wants to change the original signer from Alice to Charlie himself and Charlie has a signed warrant (s_A, r_A) and a proxy signature (m, r_P, s_P, m_w, r_A), where

$$s_P = \underbrace{k_P + x_B \cdot h(m_p, r_P) \cdot h(m_w, r_A)}_{(1)} + \underbrace{s_A \cdot h(m_p, r_P)}_{(2)}.$$

Charlie must modify the term (2).

First, let's consider the situation that the modification of the term (2) has no effect on the term (1). In this case, he must sign on the warrant information with the same random number r_A as s_A. In order to do that, he must compute the integer k_A such that $r_A = g^{k_A} \mod n$, i.e., the DLP itself. Hence, he can not use this method.

Next, we consider the situation that Charlie chooses a random number k_C independently from r_A and signs on the warrant information m'_w. Then, Charlie must modify the first term (1) and consequently obtains the $k_P + x_B \cdot h(m_p, r_P) \cdot h(m'_w, r_C)$. In order to get this term, he has two approaches. One is to use the term (1) and another is to compute directly by himself. We know that the difficulty of the latter case is equal to solving DLP through the same way as [8].

Let $A = k_P + x_B \cdot h(m_p, r_P) \cdot h(m_w, r_A)$ and $B = k_P + x_B \cdot h(m_p, r_P) \cdot h(m_w, r_C)$. Suppose that he obtained B from A. Let $H_1 = h(m_w, r_A) \cdot h(m_P, r_P)$ and $H_2 = h(m'_w, r_C) \cdot h(m_P, r_P)$. To obtain B from the algebra

$$\frac{H_2}{H_1} \cdot A = \frac{H_2}{H_1} \cdot k_P + H_2 \cdot x_B,$$

the following equation must be satisfied

$$h(m_P, g^{\frac{H_2}{H_1} \cdot k_P}) \cdot h(m'_w, r_C) = H_2.$$

However, since h is a collision resistant hash function, it is computationally infeasible to find H_2 satisfying the above equation.

5 The LKK Scheme and Its Analysis

It is based on the KPW scheme implementing the delegation with the warrant, with the difference that the warrant information signed by the original signer need not explicitly include either his identity or the identity of the proxy signer since the original signer and proxy signer do not play the same role in the generation of a proxy signature. Thus, the verifier can identify the roles of them just from the signature.

[Basic Protocol]

1. Generation of the proxy key: Bob gets a proxy key pair (x_P, y_P) through the following steps.

1. Alice chooses $k_A \in \mathbb{Z}_q^*$ at random, computes $r_A = g^{k_A} \mod p$ and $s_A = k_A + x_A \cdot h(m_w, r_A) \mod q$, and then sends (m_w, r_A, s_A) to a proxy signer Bob secretly.
2. Bob verifies the validity of the signature on m_w. If the signature is valid, Bob computes the proxy key pair (x_P, y_P) as $x_P = x_B + s_A$ and $y_P = g^{x_P} \mod p$.

2. Proxy signature generation: The proxy signer Bob signs on the delegated message m_P with the proxy private key x_P using Schnorr's signature scheme.

3. Verification: A recipient checks if the proxy signer and the message conform to m_w and then verifies the validity of the proxy signature with the proxy public key

$$y_P = {y_A}^{h(m_w, r_A)} \cdot r_A \cdot y_B \mod p.$$

If both verifications hold, the proxy signature is valid. Any verifier can check the original signer's agreement on m_w, identify the proxy signer from the proxy public key, and check the validity of the proxy signer's signature on m_P.

Because this proxy signature scheme uses the warrant containing the identity information of the proxy signer and the limit of the signing capability, it overcomes the weaknesses of the MUO scheme. However, this scheme is not a proxy-protected one and either does not provide R2 (strong unforgeability) in the strict sense.

When Lee *et al.* [3] analyze the security of their scheme, they only consider the situation that first the original signer delegates the signing capability and then the proxy signer signs the delegated message. However, we found the fact that the proxy signature can also be generated by reversing the signing order. That is, first the proxy signer Bob signs on the message m_P and then the original signer Alice adds some factor $s_A \cdot h(m_P, r_P)$ to the signature and therefore created the proxy signature. Let's see the algebra:

$$\begin{aligned} s_P &= k_P + x_P \cdot h(m_P, r_P) \\ &= k_P + (x_B + s_A) \cdot h(m_P, r_P) \\ &= k_P + x_B \cdot h(m_P, r_P) + s_A \cdot h(m_P, r_P) \\ &= s_B + s_A \cdot h(m_P, r_P), \end{aligned}$$

where s_B is a signature on the message m_P by Bob and the second term is generated by the original signer by herself. Using this method, every signature generated by the Schnorr's signature scheme can be converted into a proxy signature in which the signer is regarded as the proxy signer by the verifier. In addition, the original signer can remove the same term from the valid proxy signature and obtain a plain Schnorr's signature. Consequently, this scheme is not a proxy-protected one. See Fig. 1 for the more detail scenario.

Now we discuss the necessity of a secure channel. From the above approach, we can see why the LKK scheme needs the secure channel to deliver the original signer's signature on the warrant to the proxy signer. If the signed warrant is delivered through an insecure channel, an attacker who intercepts the delegation information can convert Bob's signature on any message conforming the warrant into the proxy signature in which Bob is the proxy signer regardless of Bob's will. Conversely, he can get a proxy signer's valid signature on the message m_P by removing the last term of the proxy signature. As another reason to establish the secure channel, if some attacker intercepts all the signed warrant delivered to Bob and then changes the term $(h(m_P, r_P) \cdot s_A)$ into the one among the intercepted warrants, he can change the original signer.

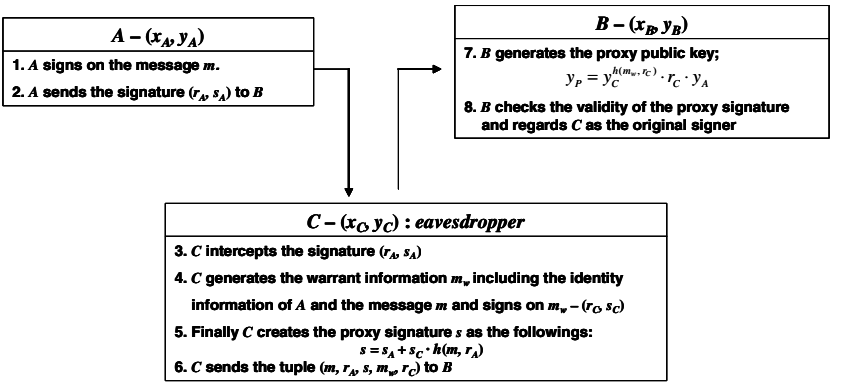

Fig. 1. Attack scenario against proxy protection

6 Comparison and Revisions

We compare the above four schemes and the OTO scheme [6] with respect to the providing functions and efficiency in Table 1. The providing functions mean the security requirements satisfied by each scheme and the efficiency means the necessity of the secure channel. In the MUO scheme the proxy signer can misuse the unlimited signing capability and transfer to others since the warrant does not include the identity information of the proxy signer or the limit of the delegated signing capability. The PH scheme uses a 3-pass blind signature scheme to protect the proxy private key from the original signer. However, since the proxy private key is the ordinary signature on the identity of the proxy signer using Schnorr's scheme, a malicious one can generate the proxy signature where the original signer is himself and the proxy signer is anyone who he wants. Thus their scheme does not satisfy the requirements R2 and R4 and so is not proxy-protected. The proxy signer can also misuse the signing capability. the OTO scheme follows the same process as the PH scheme, i.e., the proxy signer uses an ordinary signature as a proxy private key, with a difference that it uses a secure channel and the warrant contains the limit of the delegated rights. The KPW scheme is a proxy-protected one and satisfies all the security requirements in the strict sense as mentioned above. Furthermore, in case that we remove the secure channel, their scheme has no effect on the security while other schemes break out several problems. Actually, the removal of the secure channel improves the efficiency greatly. The LKK scheme is not a proxy-protected one since an attacker is able to change any valid signature into the proxy signature and the malicious original signer can obtain the valid signature from the proxy signature.

We propose the revisions of the above schemes [5, 3] to prevent the weaknesses arising in the situation removing the secure channel and provide the same functions with the original scheme with the same computational complexity.

Table 1. Comparison of proxy signatures

Features	MUO	PH	KPW	OTO	LKK
Secure Channel	Need	Not Need	Not Need	Need	Need
Proxy-Protected	O	X*	O	X	X*
R1 Verifiability	O	O	O	O	O
R2 Unforgeability	O	X	O	X	X*
R3 Identifiability	O	O	O	O	O
R4 Undeniability	O	X	O	X	O
R5 Prevention of Misuse	X	X	O	O	O
Non-Transferability	X	O	O	O	O

* Indicates the different assertion from the author's claim.

6.1 Revision of the MUO Scheme

In order that the original signer designates the proxy signer in advance, we modify the proxy key generation stage as follows:

1'. Generation of the proxy key: Bob gets a proxy key pair (x_P, y_P) through the following steps.

1. An original signer Alice generates a random number $k \in \mathbb{Z}_q^*$ and computes $K = g^k \mod p$. After that, she calculates
$$s_A = x_A + k \cdot y_B \mod q,$$
and then sends (s_A, K) to a proxy signer Bob.
2. Bob checks
$$g^{s_A} \stackrel{?}{=} y_A \cdot K^{y_B} \mod p.$$
If it is passed, Bob computes the proxy private key as
$$x_P = s_A + x_B \cdot y_A \mod q.$$

The proxy public key, which is used in the verification stage, is generated as follows:

$$y_P = g^{x_P} = y_A \cdot K^{y_B} \cdot y_B^{y_A}.$$

In this revision having no secure channel, the original signer can designate the proxy signer and so once the proxy signature is generated by the proxy signer anyone cannot change the original signer.

6.2 Revision of the LKK Scheme

The weakness of the LKK scheme results from the characteristic of the the proxy private key and Schnorr's signature scheme. We revise the scheme at the proxy key generation stage as follows:

1'. Generation of the proxy key: Bob gets a proxy key pair (x_P, y_P) through the following steps.

1. Alice chooses at random $k_A \in \mathbb{Z}_q^*$ and computes $r_A = g^{k_A} \bmod p$ and $s_A = k_A + x_A \cdot h(m_w, r_A) \bmod q$. Then she sends (m_w, r_A, s_A) to Bob.
2. Bob verifies the validity of the signature on m_w. If the signature is valid, Bob chooses a random number k_P and computes his proxy key pair (x_P, y_P) such that $x_P = r_P \cdot x_B + s_A$ and $y_P = g^{x_P} \bmod p$ for $r_P = g^{k_P} \bmod p$.

Then the proxy public key is generated as follows:

$$y_P = {y_A}^{h(m_w, r_A)} \cdot r_A \cdot {y_B}^{r_P}.$$

In order to show that this scheme overcomes the weakness of the LKK scheme, we must show that Alice and Bob cannot create the proxy signature by the reversing-order method. Let's see the following two equations:

1. $s_1 = k_P + x_B \cdot h(m_P, r_P)$
2. $s_2 = k_P + r_P \cdot x_B \cdot h(m_P, r_P)$

where the conditions for each parameter are the same as the above scheme. s_1 is the Schnorr's signature on the message m_P and we cannot obtain s_1 in the polynomial time [8]. Through the comparison between the first equation and the second, we can know that finding s_2 is as difficult as the first and anyone cannot induce s_2 from s_1. Here, in order that an attacker creates the proxy signature generated by our revised scheme from the valid signature generated by the Schnorr's signature scheme, the attacker should induce the second equation from the first. Thus, the proxy signature is not generated by the reversing-order method.

Consequently, our revised scheme overcomes the weakness of the previous schemes and we do not require the secure channel any more for the delivery of the signed warrant.

7 Conclusion

In this paper, we analyzed and compared several proxy signature schemes. The comparison was given in Table 1. We also discussed the necessity of secure channel in proxy signatures. All known proxy signatures used a secure channel to deliver a proxy certificate except one which used a 3-pass weak blind signature. However, one of them appeared to be secure without the secure channel. As a further work, it would be interesting to devise a security model on proxy signatures and give a rigorous proof based on this.

References

[1] Javier Herranz and German Saez "Fully Distributed Proxy Signature Schemes", *http://eprint.iacr.org/*, 2002.

[2] S. Kim, S. Park, and D. Won, "Proxy signatures, revisited", In *Pro. of ICICS'97, International Conference in Information and Communications Security,* Springer, Lecture Notes in Computer Science, LNCS1334, pages 223-232, 1997. 68, 69, 72

[3] Byoungcheon Lee, Heesun Kim, Kwangjo Kim, "Strong Proxy Signature and its Applications", SCIS2001, vol 2/2 pp 603-608, Jan.23 26, 2001. 68, 69, 72, 75, 76

[4] Byoungcheon Lee, Heesun Kim, and Kwangjo Kim "Secure Mobile Agent using Strong Non-designated Proxy Signature", Proc. of ACISP2001, LNCS, Springer Verlag Vol.2119, pp.474-486, 2001.

[5] M. Mambo, K. Usuda, and E. Okamoto, "roxy signature: Delegation of the power to sign messages", In *IEICE Trans. Fundamentals*, Vol. E79-A, No. 9, Sep., pp. 1338-1353, 1996. 68, 69, 70, 76

[6] Takeshi Okamoto, Mitsuru Tada, and Eiji Okamoto, "Extended Proxy Signatures for Smart Cards", In *Pro. of ISW'99* Springer, Lecture Notes in Computer Science, LNCS 1729, pp. 247-258, 1999. 69, 76

[7] H. Petersen and P. Horster, "Self-certified keys - Concepts and Applications", In *Proc. Communications and Multimedia Security'97*, pages 102-116, Chapman and Hall, 1997. 69, 71

[8] D. Pointcheval and J. Stern, "Security proofs for signatures", In *Advances in Cryptoloy: Eurocrypt'96*, pages 387-398, Springer, 1996. 74, 78