Multi-Certification Signatures and Their Applications to Public Key Infrastructure

Byoungcheon Lee
Joongbu University,
San 2-25, Majon-Ri, Chuboo-Meon, Kumsan-Gun, Chungnam, 312-702, Korea
Email: sultan@joongbu.ac.kr
Kwangjo Kim
International Research center for Information Security (IRIS),
Information and Communications University (ICU),
58-4, Hwaam-dong, Yusong-gu, Daejeon, 305-732, Korea
Email: kkj@icu.ac.kr

ABSTRACT

As the application of digital signature is progressed in real life, the situation of using digital signatures tends to become more complex. Depending on applications a user may need to generate multiple signatures for the same message with his multiple signing keys. But the general approach of generating multiple independent signatures is not efficient. To solve this problem, we propose *multi-key signature* scheme in which a signer generates a single signature for a message using his multiple signing keys all together.

Traditionally a signature provides the authenticity of a message (linked to a key pair) and a certificate provides the authenticity of the key pair (linked to a signer, certified by a certification authority), and they are generated and verified independently. We propose a new digital signature scheme called *multi-certification signature* in which a signer generates a signature on a message using his signing key and related certification information together, and then a verifier can verify not only the signer's signature on the message, but also related certification information, in a highly combined manner.

Finally, we apply the proposed multi-certification signature scheme to public key infrastructure (PKI) and privilege management infrastructure (PMI), and show that signing and verification operations can be executed in very efficient manner.

KEY WORDS

Digital signature, multi-key signature, multi-certification signature, public key infrastructure, privilege management infrastructure

1. Introduction

Since the digital signature act provide legal support to the validity of digital signature, public key infrastructure (PKI) [5] industry is booming and digital signature technology is being adapted quickly in our real life. As the application of digital signature is progressed in real life, users may need to use multiple signing keys (and their certificates) for different kinds of application. For example, an active cyber user may need a certificate issued by the government for his social life, a certificate issued by a bank for electronic money transfer and electronic commerce, a certificate issued by a company for his daily job, *etc.* Also users may have multiple certifications for a public key. For example, in privilege management infrastructure (PMI) [6] a key pair can be shared for different kinds of application.

Depending on applications there are possibilities that a user needs to generate multiple signatures for a message with his multiple signing keys. For example, a company worker needs to sign an important contract document in his company job, but the other party requires multiple signatures on the same contract document with a certified key from the government for national identification of the signer, with a certified key from his company for identification in his company, and with a certified key from a bank for money transfer. As a first approach the signer can generate 3 independent digital signatures using his 3 different signing keys and provide them to the other party. But, if we consider the efficiency in signing and verification processes and the efficiency in signature size, there is a possibility of improvement. In this paper we propose a multi-key signature (MKS) scheme in which a signer generates a single signature for a message using his multiple signing keys all together, and then a verifier verifies the signature using corresponding public keys. This scheme can be used for many real world applications in which a user needs to generate multiple signatures for the same message with his multiple signing keys.

We also consider the efficiency in verification of a signature and validation of a certificate. Traditionally a signature provides the authenticity of a message (linked to a key pair) and a certificate provides the authenticity of the key pair (linked to a signer, certified by a certification authority), and they are generated and verified independently. To check the validity of a signature, a verifier has to verify not only the signature itself using the public key, but also the certificate for the public key. Moreover to verify the certificate, the verifier also needs to check certificate revocation list

(CRL) [2] and certification path from the signer to the root certification authority (RCA). Therefore, in the point of a verifier, the verification process of a digital signature is a burden and he should be very careful to check every required certifications. To solve this problem in more efficient way, we propose a new digital signature scheme called multi-certification signature (MCS) in which a signer generates a single signature on a message using his signing key and all related certification information together, and then a verifier can verify not only the signature on the message, but also all related certification information in a single process. This scheme is also very efficient in computation and communication. As a typical application of MCS, we apply the proposed schemes to PKI and PMI, and show that the efficiency in signing and verification operations are improved very much.

As we have shown above, the introduction of efficient digital signature schemes such as MKS and MCS is necessary in PKI and PMI environment. But, to the best of our knowledge there has been no similar work in the literature. MKS is analogous to the well-known multi-signature schemes [1, 3, 4] in the sense that a signature is generated using multiple signing keys. A difference is that MKS is a single party algorithm in which multiple signing keys are owned by a single signer, while multi-signature scheme is a multi-party protocol in which multiple signers cooperatively take part in signing protocol while keeping their signing keys secret.

This paper is organized as follows: In Section 2 we introduce multi-key signature scheme and provide a typical implementation based on discrete logarithm problem (DLP). In Section 3 we define multi-certification signature scheme and show a typical implementation based on DLP. In Section 4 we apply the proposed MCS scheme to PKI and PMI environment where a signer can sign a message with his signing key and all the certification information together. Finally, we conclude in Section 5.

2. Multi-Key Signature

Multi-key signature (MKS) scheme is a variant of digital signature in which a signer generates a single signature for a message using his multiple signing keys all together, and then a verifier can verify the signature using corresponding public keys of the signer. During the signing and verification processes, signer's multiple signing keys should be kept secret.

In this section, we use the following notation.

- $\mathcal{S}$: a signer
- $\mathcal{V}$: a verifier
- $S_x(m)$: a signing algorithm on message m using a signing key x
- $V_y(s, m)$: a verification algorithm of a signature s using a public key y
- $h(), h_1(), h_2()$: collision resistant hash functions
- (x_i, y_i): signer's key pairs (signing key, public key)
- (x, y): signer's new key pair for MKS
- m: a message
- $\sigma = (r, s)$: signer's multi-key signature

2.1 Definition

Definition 1 (Multi-key signature) Let (x_1, y_1), $\ldots, (x_n, y_n)$ be signer's n certified key pairs where x_i and y_i are a signing key and the corresponding public key, respectively. Let $S()$ be a secure signing algorithm and let $V()$ be a verification algorithm. A signer $\mathcal{S}$ signs a message m with all the signing keys $(x_1, x_2, \ldots, x_n)$ and a verifier $\mathcal{V}$ verifies the signature with all the public keys $(y_1, y_2, \ldots, x_y)$. Multi-key signature scheme consists of the following three algorithms.

1. **Key generation algorithm** takes the n certified key pairs $(x_1, y_1), \ldots, (x_n, y_n)$ as input and outputs a new key pair (x, y)
$$x = f(x_1, x_2, \ldots, x_n), \quad y = g(y_1, y_2, \ldots, y_n)$$
where f, g are public algorithms.
2. **Signing algorithm** takes a message m, a new signing key x, and multiple public keys $(y_1, y_2, \ldots, y_n)$ as input and outputs a signature σ:
$$\sigma = S_x(m, y_1, y_2, \ldots, y_n).$$
3. **Verification algorithm** takes a message m, a signature σ, and multiple public keys $(y_1, y_2, \ldots,$ $y_n)$ as input and outputs binary value 0 (invalid) or 1 (valid):
$$V_{y_1,y_2,\ldots,y_n}(\sigma, m) \stackrel{?}{=} 1.$$

2.2 Typical Implementation Based on DLP

The multi-key signature scheme can be implemented easily using the DLP based cryptosystem when the system parameters of DLP cryptosystem are shared among multiple key pairs of the signer. We consider the Schnorr signature scheme as a primitive signature scheme.

Firstly we review Schnorr signature scheme 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 cryptographic hash function. Assume that a signer has a signing key x and the corresponding public key $y = g^x \bmod p$. To sign a message m, the signer chooses a random number $k \in_R Z_q^*$ and computes $r = g^k$, $s = x{\cdot}h(m, r){+}k$. Then the tuple (m, r, s) becomes a valid signed message. The validity of signature is verified by $g^s \stackrel{?}{=} y^{h(m,r)}r$. Note that the signing process requires one offline modular exponentiation and the verification of a signature requires two online modular exponentiations. This signature scheme has been proven to be secure under the random oracle model [7]. They have shown that existential forgery under the adaptively chosen message attack is equivalent to the discrete logarithm problem.

Now we assume that a signer $\mathcal{S}$ has n key pairs $(x_1, y_1), \ldots, (x_n, y_n)$ which share the same system parameters p and q. He wants to sign a message m with multiple signing keys $(x_1, x_2, \ldots, x_n)$. The multi-key signature scheme is given as follows.

1. **Key generation:** A signer $\mathcal{S}$ computes a new signing key pair (x, y) as follows.
   - Computes a new signing key
   $$x = x_1 + x_2 + \cdots + x_n.$$
   - Computes a new public key
   $$y = y_1 y_2 \cdots y_n.$$
2. **Signing:** A signer $\mathcal{S}$ computes a multi-key signature as follows.
   - Chooses a random number $k \in_R Z_q^*$ and computes
   $$r = g^k, \quad s = x \cdot h_1(m, r) + k \cdot h_2(y_1, y_2, \ldots, y_n)$$
   where $h_1()$ and $h_2()$ are two hash functions.
   - Gives (r, s) as a multi-key signature on message m.
3. **Verification:** A verifier $\mathcal{V}$ checks the validity of (r, s) as follows.
   - Computes a new public key $y = y_1 y_2 \cdots y_n$.
   - Verify (r, s) using y by
   $$g^s \stackrel{?}{=} y^{h_1(m,r)} r^{h_2(y_1, y_2, \ldots, y_n)}.$$

If the verification holds, it means that the multi-key signature (r, s) is valid with regard to multiple public keys $(y_1, y_2, \ldots, y_n)$. This scheme is analogous to the well-known multi-signature schemes [1, 3, 4] in the sense that a signature is generated using multiple signing keys. A difference is that MKS is a single party algorithm in which multiple signing keys are owned by a single signer, while multi-signature scheme is a multi-party protocol in which multiple signers cooperatively take part in signing protocol while keeping their signing keys secret.

2.3 Efficiency

To compare the efficiency of the proposed MKS scheme, we consider a general approach that a signer generates n independent signatures using n signing keys, respectively, and a verifier verifies n signatures independently. We show the comparison result in Table 1.

Table 1. Efficiency of MKS scheme in computation and communication.

	General approach	MKS
No. of Exp. in signing	n	1
No. of Exp. in verification	$2n$	3
Signature size	$n(\|p\| + \|q\|)$	$\|p\| + \|q\|$

In the general approach signing requires n signature generations (n offline exponentiations) and n signature verifications ($2n$ online exponentiations), while in the proposed MKS scheme signing requires 1 signature generations (1 offline exponentiation) and 1 signature verifications (3 online exponentiations).

In signature size general approach uses n independent signatures ($n(|p| + |q|)$) while MKS requires a single signature ($|p| + |q|$). Therefore MKS scheme is n times efficient than the general multiple signature approach in computation and communication.

2.4 Applications

As the application of digital signature is progressed in real life, users may need to use multiple signing keys (and their certificates) for different kinds of application. Each certificate is certified by different authorities and each signing key is used only for specific purpose. When a signing message is important, a receiver can require multiple signatures on the same message with different qualifications. For example, when a user tries to open a banking account, the bank will require his signatures on numerous documents, possibly with different qualifications. Hand-written signatures are hard to have different qualifications, but digital signatures can have different qualifications depending on key pairs using certificates. The proposed MKS scheme is a perfect solution in this case. We expect that MKS scheme can be used for many applications in real life in which users need to generate multiple signatures for the same message with his multiple signing keys.

3. Multi-Certification Signature

Traditionally a signature provides the authenticity of a signed message (linked to a key pair) and a certificate provides the authenticity of the key pair (linked to a signer, certified by a certification authority), and they are generated and verified independently. To check the validity of a signature, a verifier has to verify not only the signature itself using the public key, but also the certificate related with the public key. Moreover to verify the certificate, the verifier also needs to check CRL and the certification path from the signer to RCA. Therefore, in the point of a verifier, the verification process of a digital signature is a burden and he should be very careful to check every required certifications.

To solve this problem in more efficient way, we propose a new digital signature scheme called multi-certi-
fication signature (MCS) in which a signer generates a signature on a message using his signing key and all related certification information together, and then a verifier can verify not only the signature on message, but also all related certification information. During the signing and verification processes, signer's private signing key should be kept secret, but related certification information can be published safely. Actually public key certificate, CRL, certification path are public information. This is a difference compared with MKS scheme.

In this section, we use the following notation.

- $\mathcal{S}$: a signer
- $\mathcal{V}$: a verifier
- $\mathcal{A}_i$: i-th authority
- $S_x(m)$: a signing algorithm on message m using a signing key x
- $V_y(s, m)$: a verification algorithm of a signature s using a public key y
- $h(), h_1(), h_2()$: collision resistant hash functions
- (x_0, y_0): signer's key pair (signing key, public key)
- (x_i, y_i): $\mathcal{A}_i$'s key pair (signing key, public key)
- $c_i = (r_i, s_i)$: certificate, issued by authority $\mathcal{A}_i$, for the public key y_0 and the signer $\mathcal{S}$
- CI_i: certification information, prepared by authority $\mathcal{A}_i$, for the public key y_0 and the signer $\mathcal{S}$
- (x, y): signer's new key pair for MCS
- m: a message
- $\sigma = (r, s)$: signer's multi-certification signature

3.1 Definition

Definition 2 (Multi-certification signature) Let (x_0, y_0) be a signer's certified key pair where x_0 and y_0 are a signing key and the corresponding public key, respectively. Let $(c_1, c_2, \ldots, c_n)$ be n certificates related with the public key y_0. Let $S()$ be a secure signing algorithm and let $V()$ be a verification algorithm. A signer $\mathcal{S}$ signs a message m with his signing key x_0 and n certificates $(c_1, c_2, \ldots, c_n)$, and a verifier $\mathcal{V}$ verifies the signature with the public key y_0 and n certificates $(c_1, c_2, \ldots, c_n)$ together. Multi-certification signature scheme consists of the following three algorithms.

1. **Key generation algorithm** takes signer's key pair (x_0, y_0) and n certificates $(c_1, c_2, \ldots, c_n)$ as input and outputs a new key pair (x, y)

   $$x = f(x_0, c_1, c_2, \ldots, c_n), \quad y = g(y_0, c_1, c_2, \ldots, c_n)$$

   where f, g are public algorithms.

2. **Signing algorithm** takes a message m, a new signing key x, and n certificates $(c_1, c_2, \ldots, c_n)$ as input and outputs a signature σ:

   $$\sigma = S_x(m, c_1, c_2, \ldots, c_n).$$

3. **Verification algorithm** takes a message m, a signature σ, a public key y_0, and n certificates $(c_1, c_2,$
   $\ldots, c_n)$ as input and outputs binary value 0 (invalid) or 1 (valid):

   $$V_{y_0, c_1, c_2, \ldots, c_n}(\sigma, m) \stackrel{?}{=} 1.$$

3.2 Typical Implementation Based on DLP

The MCS scheme can be implemented easily using the DLP based cryptosystem when the system parameters are shared among the key pair and n certificates. Similar with the MKS case, we consider the Schnorr signature scheme as a primitive signature scheme.

We assume that a signer $\mathcal{S}$ has a certified key pair (x_0, y_0) where $y_0 = g^{x_0}$ and n certificates $(c_1, c_2, \ldots, c_n)$ related with it. We also assume that the same system parameters p and q are shared among the key pair and n certificates.

The certificate c_i is a signature on some certification information CI_i related with the public key y_0 and is provided by an authority $\mathcal{A}_i$ to $\mathcal{S}$. Let (x_i, y_i) be $\mathcal{A}_i$'s key pair where $y_i = g^{x_i}$. Then c_i is a Schnorr signature of the authority $\mathcal{A}_i$ on certification information CI_i. $\mathcal{A}_i$ chooses $k_i \in_R Z_q^*$ and computes

$$c_i = (r_i, s_i) = (g^{k_i}, x_i \cdot h(CI_i, r_i) + k_i).$$

It's validity can be verified by $g^{s_i} \stackrel{?}{=} y_i^{h(CI_i, r_i)} r_i$. $\mathcal{A}_i$ have issued $c_i = (r_i, s_i)$ to $\mathcal{S}$ as a certificate.

Now the multi-certification signature scheme is given as follows.

1. **Key generation:** A signer $\mathcal{S}$ computes a new signing key pair (x, y) as follows.

   - Computes a new signing key

     $$x = x_0 + s_1 + \cdots + s_n.$$

   - Computes a new public key

     $$y = y_0 y_1^{h(CI_1, r_1)} r_1 \cdots y_n^{h(CI_n, r_n)} r_n.$$

2. **Signing:** A signer $\mathcal{S}$ computes a multi-certification signature on message m and certification information $(CI_1, r_1, \ldots, CI_n, r_n)$ as follows.

- Chooses a random number $k \in_R Z_q^*$ and computes a signature as $r = g^k$ and

$$s = x \cdot h_1(m, r) + k \cdot h_2(CI_1, r_1, \ldots, CI_n, r_n)$$

where $h_1()$ and $h_2()$ are two hash functions.

- Gives $\{(r, s), CI_1, r_1, \ldots, CI_n, r_n\}$ as a MCS on message m.

3. **Verification:** A verifier $\mathcal{V}$ checks the validity of $\{(r, s), CI_1, r_1, \ldots, CI_n, r_n\}$ as follows.

   - Computes a new public key

$$y = y_0 y_1^{h(CI_1, r_1)} r_1 \cdots y_n^{h(CI_n, r_n)} r_n.$$

   - Verify (r, s) using y by

$$g^s \stackrel{?}{=} y^{h_1(m,r)} r^{h_2(CI_1, r_1, \ldots, CI_n, r_n)}.$$

If the verification holds, it means that the signature of the signer is valid and n certification information are also confirmed. Note that in the computation of s message is used in the first hash function and certification information are used in the second hash function.

3.3 Efficiency

To compare the efficiency of the proposed MCS scheme, we consider a general approach that the signer just generates a signature on the message m with his signing key x_0, and then the verifier has to verify $n+1$ signatures (a signature of the signer and n certification information) independently. We show the comparison result in Table 2.

In the general approach signing requires 1 signature generation (1 offline exponentiation) and $n+1$ signature verifications ($2(n+1)$ online exponentiations), while in the proposed MCS scheme signing requires 1 signature generation (1 offline exponentiation) and 1 signature verification together with n exponentiations ($n+3$ online exponentiations). In signature size general approach uses $n+1$ independent signatures ($(n+1)(|p|+|q|)$) while MCS requires a single signature and $r_1, \ldots, r_n$ ($(n+1)|p|+|q|$). (Note that if the signer sends certificates themselves as certification information to the verifier, communication size will not be changed.) Therefore MCS scheme is more efficient than the general approach in computation and communication.

If we consider the case that n certification information are somewhat fixed and the verifier has verified them all in advance, then the verifier in MCS scheme can also compute the new public key y in advance and use it repeatedly. Then the amount of computation are the same.

We can consider another efficiency point. In MCS scheme a signer provides a verifier with a highly combined digital signature which is an unforgeable combination of digital signature and all the relevant certification information. If the signature cannot pass the verification process because of certification information, it will not be considered as a valid signature. Therefore, a signer has to provide correct certification information to a verifier. Then the verifier does not need to locate certification information by himself.

Table 2. Efficiency of MCS scheme in computation and communication.

	General approach	MCS
No. of Exp. in signing	1	1
No. of Exp. in verification	$2(n+1)$	$n+3$
Signature size	$(n+1) \cdot(\|p\|+\|q\|)$	$(n+1)\|p\| +\|q\|$

4. PKI using Multi-Certification Signature

To check the validity of a signature, a verifier should check various certifications related with the public key. For example, the verifier needs to check signer's public key certificate (PKC), CRL [2] and certification path from the signer to RCA. Therefore, in the point of a verifier, the verification process of a digital signature is a burden and he should be very careful to check every required certifications. The proposed MCS scheme is very useful under the PKI [5] and PMI [6] environment. Using MCS scheme a digital signature and multiple certification information can be verified together in an efficient manner.

Recently, attribute certificate (AC) and PMI are becoming an issue. Since the PKC provide authentication only for the public key and is used for relatively long period of time, it is not suitable to authenticate short term attributes of signer (such as access control, role, authorization, *etc.*) which are used for relatively short period of time. For these applications attribute authority (AA) issues AC to a signer to certify signer's specific attribute. PMI is an infrastructure to manage AC.

AC does not use an independent key pair, but has a link to a PKC, therefore same key pair is shared among PKC and AC. When a signer signs a message with the key pair and asserts both certifications of PKC and AC, a verifier has to verify both signatures of PKC and AC. MCS scheme is very useful to verify both PKC and AC.

System Set-up: To apply the proposed MCS scheme, we assume that the whole social environment of PKI and PMI use the same system parameters p and q of DLP based cryptosystem. In real situation, a verifier may have to verify various kinds of certification information related with the public key of the signer, but we simplify the case for the ease of description. We assume that a signer $\mathcal{S}$ signs a message m with his signing key and asserts the certifications of PKC and AC, and a verifier $\mathcal{V}$ has to verify $\mathcal{S}$'s PKC, AC, and CRL. Now we consider the following case.

- CA has a key pair (x_c, y_c) and issues a PKC (r_c, s_c) to the signer $\mathcal{S}$ on the certification information CI_s as follows:
$$r_c = g^{k_c}, \quad s_c = x_c \cdot h(CI_s, r_c) + k_c.$$
- AA has a key pair (x_a, y_a) and issues an AC (r_a, s_a) to the signer $\mathcal{S}$ on the attribute message Att_s as follows:
$$r_a = g^{k_a}, \quad s_a = x_a \cdot h(Att_s, r_a) + k_a.$$
- CA also issues a CRL (r_l, s_l) on the revoked list as follows:
$$r_l = g^{k_l}, \quad s_l = x_c \cdot h(CRL, r_l) + k_l.$$

Signing: Assume that a signer $\mathcal{S}$ has a certified key pair (x_s, y_s). He wants to sign a message m and assert the certifications of PKC and AC. He also wants to show that his certificate was not revoked yet. The signer computes a MCS on message m and certification information as follows.

1. Computes a new signing key
$$x = x_s + s_c + s_a + s_l.$$
2. Chooses a random number $k \in_R Z_q^*$ and computes a signature as $r = g^k$ and
$$s = x \cdot h_1(m, r) + k \cdot h_2(CI_s, r_c, Att_s, r_a, CRL, r_l).$$
3. Gives $\{(r, s), CI_s, r_c, Att_s, r_a, CRL, r_l\}$ as a MCS on message m.

Verification: A verifier checks the validity of $\{(r, s), CI_s, r_c, Att_s, r_a, CRL, r_l\}$ as follows.

1. Computes a new public key
$$y = y_s y_c^{h(CI_s, r_c)} r_c y_a^{h(Att_s, r_a)} r_a y_c^{h(CRL, r_l)} r_l.$$
2. Verify (r, s) using y as
$$g^s \stackrel{?}{=} y^{h_1(m,r)} r^{h_2(CI_s, r_c, Att_s, r_a, CRL, r_l)}.$$

If the verification holds, it means that the signature of the signer is valid and related certification information are also valid.

As shown in Section 3, the signing and verification processes are much more efficient than the general approach of independent multiple signatures. If a signer wants to demonstrate multiple certifications related with his public key together with a signature, he organizes all relevant certifications to generate new key pair and computes a single signature with it. This can be considered as an additional service of the signer for the convenience of a verifier. Then a verifier can verify the signature itself and multiple certifications the signer had asserted all together.

The proposed MKS and MCS schemes can be used in combined manner. If a signer has to sign a message with multiple signing keys and wants to assert multiple certifications related with the signing keys, MKS and MCS schemes can be used in combined way.

5. Conclusion

In this paper we have considered the complex real situations of using digital signatures in PKI and PMI environment and derived the necessity of new digital signature schemes. First, we have shown the necessity of signing a message with multiple signing keys of a signer and proposed an efficient multi-key signature scheme. We also have shown the necessity of signing a message with a signing key and multiple certification information related with the public key together, and proposed an efficient multi-certification signature scheme. Finally, we have applied the proposed multi-certification signature scheme to PKI and PMI, and shown that signing and verification operations can be executed very efficiently.

As the application of digital signature is progressed in real life, the situation of using digital signatures tends to become more complex. Users need to use multiple signing keys (and their certificates) for different kinds of applications. To verify a signature, a verifier should be very careful to check all relevant certifications together with the signature itself. Therefore the proposed MKS and MCS schemes are very useful to improve the overall efficiency of using digital signature technology in PKI and PMI environment.

References

[1] M. Burmester, Y. Desmedt, H. Doi, M. Mambo, E. Okamoto, M. Tada, and Y. Yoshifuji, "A structured ElGamal type multisignature scheme", *Public Key Cryptography 2000*, LNCS 1751, Springer-Verlag, pages 466–482, 2000.

[2] RFC 2459, Internet X.509 Public Key Infrastructure Certificate and CRL Profile, IETF, 1999, http://www.ietf.org/html.charters/pkix-charter.html

[3] K. Ohta and T. Okamoto, "Multi-signature schemes secure against active insider attacks", *IEICE Transactions of Fundamentals*, Vol. E-82-A, No. 1, 1999.

[4] K. Ohta and T. Okamoto, "Generic construction method of multi-signature schemes", *Proc. of the 2001 Symposium on Cryptography and Information Security*, SCIS01-2B, Jan. 23-26, 2001.

[5] Public-Key Infrastructure (X.509) (pkix),

http://www.ietf.org/html.charters/pkix-charter.html

[6] Request for Comments, An Internet Attribute Certificate Profile for Authorization (RFC 3281), IETF, 2002.

[7] D. Pointcheval and J. Stern, "Security Proofs for Signatures", *Advances in Cryptology: Eurocrypt'96*, pages 387 - 398, Springer, 1996.