World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:3, No:5, 2009

A Proxy Multi-Signature Scheme with Anonymous Vetoable Delegation

Pei-yih Ting, Dream-Ming Huang, and Xiao-Wei Huang

Open Science Index, Computer and Information Engineering Vol:3, No:5, 2009 publications.waset.org/14356/pdf

Abstract—Frequently a group of people jointly decide and authorize a specific person as a representative in some business/poitical occasions, e.g., the board of a company authorizes the chief executive officer to close a multi-billion acquisition deal. In this paper, an integrated proxy multi-signature scheme that allows anonymously vetoable delegation is proposed. This protocol integrates mechanisms of private veto, distributed proxy key generation, secure transmission of proxy key, and existentially unforgeable proxy multi-signature scheme. First, a provably secure Guillou-Quisquater proxy signature scheme is presented, then the "zero-sharing" protocol is extended over a composite modulus multiplicative group, and finally the above two are combined to realize the GQ proxy multi-signature with anonymously vetoable delegation. As a proxy signature scheme, this protocol protects both the original signers and the proxy signer. The modular design allows simplified implementation with less communication overheads and better computation performance than a general secure multi-party protocol.

Keywords— GQ proxy signature, proxy multi-signature, zero-sharing protocol, secure multi-party protocol, private veto protocol

I. Introduction

As communication, networking, and computing technologies advance faster and faster, electronic commerce has gained sufficient momentum as the common business infrastructure. A digital signature scheme, being the primary authentication method for digital documents, provides not only functionalities of a handwritten signature but also enhanced security and great flexibility in the design of practical business protocols. It appears in many applications, from simple to sophisticated ones, with various appearances to provide desired authentication and certification. For our concern, when a person is absent from his post and cannot authorize an operation or when shareholders of a company make the executive officer as the fully authorized representative, a digital proxy signature scheme [1] can be employed to accomplish the goals with plenty of security merits.

A proxy signature scheme allows an original signer delegating the proxy signers to sign in place of him at occasions specified by the delegation warrant. If the original signer and the proxy signer cannot meet face-to-face, an authenticated secure channel should be used to carry out the delegation. Such a proxy signature scheme usually provides the following characteristics: the existential unforgeability of the proxy signatures, the dependency of the proxy key on the private key of the original signer, the verifiability of the delegation, the distinguishability between a proxy signature and an original signer's signature, the identifiability of the proxy signer, the protection over the original signer, the protection over the proxy signer, and the non-repudiation of a proxy signature.

In e-commerce or e-government applications, sometimes a group of people have to make collective decisions and jointly authorize a certain representative to examine and sign a document for them. For example, when the Minister of Foreign Affairs signs a bilateral commerce agreement or the Minister of Defence authorizes some military operations, it would be essential to obtain the authorizations from the Prime Minister and the Cabinet in order to prevent from dictatorship. If one several authorities does not consent with the motion, he might face apparently pressure from the others. In this situation, mechanisms of private anonymous veto can be employed to ensure the practice of independent judgement. There are still other applications in which many people transfer their signing capacity to one single representative. Currently, a direct way to handle this problem is to use a secure voting protocol followed by a proxy multi-signature scheme [2], [3], in which a multi-signature scheme [4] addresses the case that multiple signers jointly examine and sign a document.

In this paper, a GQ proxy multi-signature scheme is proposed to fulfil the above requirements. First, we design a provably secure GQ proxy signature scheme. Then all the original signers and the proxy signer jointly execute a "zero-sharing" protocol [5], which is extended over a composite modulus multiplicative group to obtain a secret share of unity without any trusted third party involved. Each original signer then multiplies his delegation proxy key with his secret share and publishes the result over a public channel. Finally, the proxy signer obtains the delegated proxy key through his secret share when all original signers grant the delegation unanimously. This integrated proxy multi-signature scheme provides good protection for both the original signers and the proxy signer.

The remaining part of this paper is organized as follows: First we summarize the Guilliou-Quisquater digital signature scheme [6] and a corresponding proxy signature scheme in Section II. Section III summaries the "zero-sharing" protocol

P.-Y. Ting is with the Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung 202, Taiwan (phone:886-2-24622192; fax:886-2-24623249; e-mail: pyting@mail.ntou.edu.tw).

D.-M. Huang was with the Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung 202, Taiwan (e-mail: m94570028@mail.ntou.edu.tw).

X.-W. Huang is with the Department of Computer Science and Engineering, National Taiwan Ocean University, Keelung 202, Taiwan (e-mail: m94570010@mail.ntou.edu.tw).

Open Science Index, Computer and Information Engineering Vol:3, No:5, 2009 publications.waset.org/14356/pdf

and presents the design of the extended "zero sharing" protocol over a composite modulus multiplicative group. Section IV presents the integrated design of the GQ proxy multi-signature with anonymously vetoable delegation. The security are analyzed in Section V. Section VI summarizes our works.

II. Guillou-Quisquater digital signature and proxy signature

Key generation: Pick two large prime numbers p and q, calculate the modulus $n = p \cdot q$, and choose an encryption exponent $e \in \mathbb{Z}^*_{\phi(n)}$, where $\phi(n) = (p-1) \cdot (q-1)$ is the Euler totient function. Pick a random number $x \in \mathbb{Z}^*_n$ and calculate $y \equiv x^{-e} \pmod{n}$. The private signing key of the signer is x and the corresponding public key is (n, e, y).

Signing: Let $h(\cdot)$ be a collision-resistant cryptographic hash function. A signer calculates the signature (c, r) for a message M as follows:

1) Pick a random number $u \in \mathbb{Z}^*_n$, calculate the commitment $a \equiv u^e \pmod{n}$, and compute the hash value $c = h(M, n, e, y, a)$.
2) Calculate $r \equiv u \cdot x^c \pmod{n}$.

Verification: Given the message M, the signature (c, r), and the signer's public key (n, e, y), a verifier calculates $a' \equiv r^e \cdot y^c \pmod{n}$ and checks if c equals $h(M, n, e, y, a')$.

Based on the intractability assumption of breaking the RSA function, the GQ digital signature is proven existentially unforgeable under chosen-message attack in the random oracle model [7].

Next, we present the design of a GQ proxy signature scheme. Basically, this follows the design methodology of Mambo's [1] proxy signature scheme and Kim's [8] proxy signature scheme in converting the Schnorr's signature scheme to its proxy signature version. In this scheme, the original signer sends the proxy key to the proxy signer through a secure channel and publishes the corresponding verification key together with the delegation warrant. The original signer is responsible for the proxy signatures issued by a proxy signer subject to the limitations stated on the delegation warrant. The scheme includes five stages: key generation, delegation/authorization, proxy key verification, proxy signing, and proxy signature verification.

Key generation: Pick two large prime numbers p and q, calculate the modulus $n = p \cdot q$, and choose an encryption exponent $e \in \mathbb{Z}^*_{\phi(n)}$. Pick a random number $x \in \mathbb{Z}^*_n$ as the private signing key, calculate $y \equiv x^{-e} \pmod{n}$, and set (n, e, y) as the corresponding public key.

Delegation/authorization: Let W be the delegation warrant, specifying the authorization extents and restrictions. The original signer calculates the GQ signature (c, r) of the warrant W, uses r as the proxy key, and publishes (n, e, y, a, c) as the corresponding verification key:

1) Pick a random number $u \in \mathbb{Z}^*_n$, calculate $a \equiv u^e \pmod{n}$, and calculate $c = h(W, n, e, y, a)$.
2) Calculate the proxy key $r \equiv u \cdot x^c \pmod{n}$ and send r to the proxy signer through a secure channel.

Proxy key verification: The proxy signer calculates $a' \equiv r^e \cdot y^c \pmod{n}$ and verifies if c equals $h(W, n, e, y, a')$.

Proxy signing: The proxy signer calculates the proxy signature (f, s) of a message M as follows:

1) Choose a random number $\nu \in \mathbb{Z}^*_n$, calculate $b \equiv \nu^e \pmod{n}$, and calculate $f = h(M, n, e, y, a, c, b)$.
2) Calculate $s \equiv \nu \cdot r^f \pmod{n}$.

Proxy signature verification: When presented with the delegation warrant W, the public key (n, e, y, a, c), a message M, and its proxy signature (f, s), a verifier calculates $b' \equiv s^e \cdot y^{c \cdot f} \cdot a^{-f} \pmod{n}$ and checks if f equals $h(M, n, e, y, a, c, b')$ and c equals $h(W, n, e, y, a)$.

In the above procedure, the delegation and proxy-key transmission are carried out through a secure channel, usually in a face-to-face manner. If a physically secure channel is not available, an authenticated and encrypted channel should be used instead.

The computation and delegation of GQ proxy signature require only modular multiplications and exponentiations. This property is used to design the proxy multi-signature scheme in Section IV. Also, this property allows an efficient integration with a private veto protocol. In the next section, we will describe a modularized secure multi-party protocol - the "zero sharing" protocol [5] and extended it in order to carry out the distributed proxy key generation, multiple proxy delegations, and secure proxy key transmission at the same time.

III. Non-interactive verifiable "zero-sharing" protocol

"Zero sharing" protocol [5] is a modular secure multi-party protocol, in which each of the m participants $\{B_i\}_{i=1,\ldots,m}$ independently contributes a secret random value as input and obtains upon the finish of the protocol a secure share z_i, which depends on all the above private inputs and satisfies $\prod_{i=1}^{m} z_i \equiv 1 \pmod{p'}$. Conceptually, each participant gets a secret value $z_i \equiv h^{t_i} \pmod{p'}$ such that $\sum_{i=1}^{m} t_i \equiv 0 \pmod{q'}$, where p' and q' are both large prime numbers satisfying $q'|p'-1$. It is thus called "zero-sharing". During the execution of the protocol, no interaction between pairs of participants is required, only communications with a non-trusted server are required. The protocol is summarized as follows:

1) All participants jointly choose a large prime number p' such that there exists a large prime number q' satisfying $q'|p'-1$. Choose jointly two independent generators g and $h \in \mathbb{G}_{q'}$, where $\mathbb{G}_{q'} \subset \mathbb{Z}^*_{p'}$ is the unique order q' subgroup. The intractability assumption of the discrete

Open Science Index, Computer and Information Engineering Vol:3, No:5, 2009 publications.waset.org/14356/pdf

log problem in $\mathbb{G}_{q'}$ guarantees that $\log_g h$ is not known to anybody.

2) Each participant B_i chooses $m-1$ random values $\{s_{i,j}\}_{j=1,\ldots,m-1}$ in $\mathbb{Z}_{q'}$ and calculates $s_{i,m} \equiv -\sum_{j=1}^{m-1} s_{i,j} \pmod{q'}$.

3) Each participant B_i calculates and publishes $R_{i,j} \equiv g^{s_{i,j}} \pmod{p'}$, which satisfy $\prod_{j=1}^{m} R_{i,j} \equiv 1 \pmod{p'}$.

4) Each participant B_i chooses randomly an integer α_i in $\mathbb{Z}^*_{q'}$, calculates and publishes $h_i \equiv h^{\alpha_i} \pmod{p'}$, calculates and publishes $R'_{i,j} \equiv h_j^{s_{i,j}} \pmod{p'}$, and publishes a non-interactive zero knowledge proof (NIZKP) [9], [10] to show that $\log_g R_{i,j} = \log_{h_j} R'_{i,j}$ in the group $\mathbb{G}_{q'}$.

5) Server (or equivalently every participant) calculates $h_j^{t_j} \equiv \prod_{i=1}^{m} R'_{i,j} \pmod{p'}$, where $t_j \equiv \sum_{i=1}^{m} s_{i,j} \pmod{q'}$ satisfies $\sum_{j=1}^{m} t_j \equiv 0 \pmod{q'}$.

6) Each participant B_j calculates his secret share $z_j \equiv h^{t_j} \equiv \left(h_j^{t_j}\right)^{\alpha_j^{-1}} \pmod{p'}$, where α_j^{-1} satisfies $\alpha_j^{-1} \cdot \alpha_j \equiv 1 \pmod{q'}$.

After completing the above protocol, each participant B_j obtains his secret share z_j that satisfies $\prod_{j=1}^{m} z_j \equiv h^{\sum_{j=1}^{m} t_j} \equiv 1 \pmod{p'}$. In the next section, we need a secure multi-party protocol that gives every participant a secret share z_j satisfying the relation $\prod_{i=1}^{m} z_i \equiv 1 \pmod{n}$ with a composite modulus $n = p \cdot q$, where p, q are both large prime numbers.

If we execute the above protocol with $|p'| \geq m \cdot |n|$, where $|\cdot|$ denotes the bit length of an integer, each participant can still obtain z_j such that multiplication modulo n is not affected by any modulo p' operation. However, the computation overhead is enormous, e.g., when $|p| = |q| = 1024$ and $m = 10$, the zero sharing protocol has to operate with $|p'| = 10240$. In the following, we present an extension to deal with this problem. Basically, we replace the modulus p' by $n = p \cdot q$ and replace the modulus q' by $\phi(n)$ in the above protocol. However, there are three apparent difficulties in the protocol design. We describe them and modify the protocol as follows:

1) A trusted third party chooses two large prime numbers p and q, where $gcd((p-1)/2, (q-1)/2) = 1$, calculates $n = p \cdot q$, chooses randomly a generator h in the cyclic subgroup of quadratic residues, $\mathbb{Q}_n$, picks a random number β in $\mathbb{Z}^*_{\phi(n)/4}$, calculates $g \equiv h^{\beta} \pmod{n}$, and publishes n, h, g, and β to all participants.

2) Every participant B_i chooses $m-1$ random numbers $\{s_{i,j}\}_{j=1,\ldots,m-1}$ in $\mathbb{Z}_{n/4}$, calculates $s_{i,m} = -\sum_{j=1}^{m-1} s_{i,j}$. (Note that since $\phi(n)$ is unknown to B_i, B_i cannot calculate $s_{i,m} \equiv -\sum_{j=1}^{m-1} s_{i,j} \pmod{\phi(n)}$ at this step.)

3) Every participant B_i calculates $R_{i,j} \equiv g^{s_{i,j}} \pmod{n}$. Note that $R_{i,j}$ still satisfies $\prod_{j=1}^{m} R_{i,j} \equiv 1 \pmod{n}$. B_i publishes $\{R_{i,j}\}_{j=1,\ldots,m}$. Everybody calculates $(R_j)^2 \equiv \prod_{i=1}^{m} (R_{i,j})^2 \equiv g^{2t_j} \pmod{n}$, where $t_j \equiv \sum_{i=1}^{m} s_{i,j} \pmod{\phi(n)}$ and $\sum_{j=1}^{m} t_j \equiv 0 \pmod{\phi(n)}$. (Each player will use $(R_j)^2$ to calculate his secret share at step 6.)

4) Each participant B_i chooses a random number α_i in $\mathbb{Z}_{n/4}$ such that $gcd(\alpha_i, \beta) = 1$, calculates and publishes $h_i \equiv h^{\alpha_i} \pmod{n}$ and $(R'_{i,j})^2 \equiv (h_j)^{2s_{i,j}} \pmod{n}$, and publishes an NIZKP for $\log_g (R_{i,j})^2 = \log_{h_j} (R'_{i,j})^2$ in the subgroup $\mathbb{Q}_n$ [11].

5) The server calculates and publishes $(R'_j)^2 \equiv \prod_{i=1}^{m} (R'_{i,j})^2 \equiv (h_j)^{2t_j} \pmod{n}$.

6) Because α_j was chosen such that $gcd(\alpha_j, \beta) = 1$, B_j uses the extended Euclidean algorithm to compute c_1 and c_2 such that $c_1\beta + c_2\alpha_j = 1$. Using c_1 and c_2, he computes the secret share $z_j \equiv \left((R_j)^2\right)^{c_1} \cdot \left((R'_j)^2\right)^{c_2} \; equiv \left(g^{2t_j}\right)^{c_1} \cdot \left({h_j}^{2t_j}\right)^{c_2} \equiv \left(h^{2t_j}\right)^{c_1\beta + c_2\alpha_j} \equiv h^{2t_j} \pmod{n}$.

Note that at step 1, if a distributed key generation algorithm like [12] is used to generate p and q, they should satisfy additionally $p = 2p' + 1$ and $q = 2q' + 1$, where p' and q' are also large prime numbers. In this situation, all participants jointly choose random numbers h in $\mathbb{Q}_n$ and β in $\mathbb{Z}_{n/4}$. Although p and q are unknown, the probability that $ord_n(h) = p' \cdot q'$ and $ord_n(g) = p' \cdot q'$ is asymptotically close to 1 since p and q are both Sophi-Germain prime numbers.

Note also that B_i proves $\log_g (R_{i,j})^2 = \log_{h_j} (R'_{i,j})^2$ at step 4 instead of $\log_g R_{i,j} = \log_{h_j} R'_{i,j}$ since $R_{i,j}$ and $R'_{i,j}$ are calculated privately and might not be in $\mathbb{Q}_n$.

At the end of the above protocol, the secret shares $\{z_j\}_{j=1,\ldots,m}$ satisfy $\prod_{j=1}^{m} z_j \equiv 1 \pmod{n}$. This protocol can be used when a group of participants want to compute the product of their individual secrets, while keeping their secrets private under the factorization intractability assumption. For example, in a scenario with three players B_1, B_2, and B_3, each player has a private value x_1, x_2, and x_3, respectively. First, they complete the above composite modulus zero-sharing protocol and obtain secret shares $z_1 \equiv h^{2t_1} \pmod{n}$, $z_2 \equiv h^{2t_2} \pmod{n}$, and $z_3 \equiv h^{2t_3} \pmod{n}$. Second, each player B_i computes privately the product of both secret values, $x_i \cdot z_i \pmod{n}$, and publishes the product. At last, everyone can calculate $\prod_{i=1}^{3} x_i \cdot z_i \equiv (x_1x_2x_3) \cdot (z_1z_2z_3) \pmod{n}$. Because the product of secret shares $\{z_1, z_2, z_3\}$ is 1, we have $\prod_{i=1}^{3} x_i \cdot z_i \equiv x_1x_2x_3 \pmod{n}$, which is the desired product of all private values. The value x_i of B_i remains private unless all other parties collaborate to deduce z_i.

Extending the above example to the GQ proxy signature of previous section, we obtain the integrated GQ proxy multi-signature scheme over a public channel. By the way, the original signers also enjoy the capability of anonymous veto such that the authorization decision can be truthful according to his own utility. For example, let B_1 and B_2 be original

Open Science Index, Computer and Information Engineering Vol:3, No:5, 2009 publications.waset.org/14356/pdf

signers. B_i has a GQ proxy key x_i. Let B_3 be the proxy signer. He chooses a secret random value x_3 and completes the above product protocol to obtain $\prod_{i=1}^{3} x_i \cdot z_i \equiv x_1 x_2 x_3 \pmod{n}$. Thus, only he can compute the authorized joint proxy key $x_1 \cdot x_2 \pmod{n}$ since x_3 is his secret. The complete protocol is presented in the next section.

IV. GQ Proxy Multi-Signature Subject to Anonymous Veto

The original signers $B_1, B_2, \ldots, B_m$ jointly execute the following protocol over a public channel to delegate B_{m+1} as the proxy signer.

Key generation: A trusted third party picks two large prime numbers p and q, where $gcd((p-1)/2, (q-1)/2) = 1$, calculates the modulus $n = p \cdot q$, chooses an encryption exponent $e \in \mathbb{Z}^*_{\phi(n)}$, and chooses the parameters h, β, and g for the composite modulus zero-sharing protocol. Each original signer B_i picks a random number $x_i \in \mathbb{Z}^*_n$ as his private signing key and calculates $y_i \equiv {x_i}^{-e} \pmod{n}$, where (n, e, y_i) is his corresponding public key. The group private key is defined as $x \equiv \prod_{i=1}^{m} x_i \pmod{n}$, although not used explicitly, and the group public key is defined as $y \equiv \prod_{i=1}^{m} y_i \, (\equiv \prod_{i=1}^{m} {x_i}^{-e}) \pmod{n}$

Group delegation subject to anonymous veto:

1) Each B_i chooses a random number u_i in $\mathbb{Z}^*_n$, calculates and publishes $a_i \equiv {u_i}^e \pmod{n}$

2) B_i calculates the common commitment $a \equiv \prod_{i=1}^{m} a_i \pmod{n}$, and calculates $c = h(W, n, e, y, a)$, where W is the delegation warrant specifying the extents and restrictions of the group delegation.

3) $B_1, B_2, \ldots, B_{m+1}$ participate the modulo-n zero-sharing protocol and obtain the secret shares $z_1, z_2, \ldots, z_{m+1}$, respectively, such that $\prod_{i=1}^{m+1} z_i \equiv 1 \pmod{n}$.

4) Each B_i calculates a secret proxy key $r_i \equiv u_i \cdot {x_i}^c \pmod{n}$, calculates $\hat{r}_i \equiv r_i \cdot z_i \pmod{n}$, and publishes it.

5) Finally, the proxy signer B_{m+1} calculates the joint proxy key $r \equiv (\prod_{i=1}^{m} \hat{r}_i) \cdot z_{m+1} \equiv \prod_{i=1}^{m} r_i \cdot \prod_{i=1}^{m+1} z_i \equiv \prod_{i=1}^{m} r_i \pmod{n}$.

In the above group delegation procedure, the proxy signature verification key corresponding to the proxy key r is (n, e, y, a, c). Note that a_i is mode public such that each signer can compute a common commitment a. If B_i is not willing to delegate his signing capacity to B_{m+1}, he chooses and publishes a random number in $\mathbb{Q}_n$ as $\hat{r}_i$. In that case, the proxy signer would not be able to obtain the valid proxy key r.

Proxy key verification: The proxy signer B_{m+1} calculates $a' \equiv r^e \cdot y^c \pmod{n}$ and verifies if c equals $h(W, n, e, y, a')$.

Proxy signing: The proxy signer B_{m+1} calculates the proxy signature (f, s) of a message M as follows:

1) B_{m+1} chooses a random number ν in $\mathbb{Z}^*_n$, calculates $b \equiv \nu^e \pmod{n}$ and $f = h(M, n, e, y, a, c, b)$.
2) B_{m+1} calculates $s \equiv \nu \cdot r^f \pmod{n}$.

Proxy signature verification: Given the delegation warrant W, the public key (n, e, y, a, c), the message M, and the proxy signature (f, s), a verifier calculates $b' \equiv s^e \cdot y^{c \cdot f} \cdot a^{-f} \pmod{n}$ and checks whether the following two equations hold: $f = h(M, n, e, y, a, c, b')$ and $c = h(W, n, e, y, a)$.

In an ideal proxy multi-signature scheme, every original signer would like to make independent decision based on his own discretion. A naive system could be implemented if every original signer B_i sends his delegation proxy key r_i to the proxy signer B_{m+1} directly. In this way, the delegation operates non-anonymously and B_i's is under considerable pressure in making his decision. Another potential problem is that the proxy key r_i not only can be used to sign the m-to-1 proxy multi-signature, but also can be used to sign B_i's 1-to-1 proxy signature. This could cause serious problem if the delegation warrants are not checked carefully at the verification stage.

In the proposed protocol, the proxy signer B_{m+1} verifies the proxy key r through $a \equiv r^e \cdot y^c \pmod{n}$. Upon success, all the original signers agree with this authorization of proxy signing capacity. If the verification fails, there is at least one original signer disagrees with the authorization based on the specified terms. The proxy signer cannot identify the person who opposes the authorization. Hence, the free will to authorize is protected.

V. Security Analysis

The correctness and security of the proposed scheme are analyzed in the following four aspects: the first is the existential forgeability of the GQ signature; the second includes those features the GQ proxy signature provides [1], including the unforgeability of the proxy signature, the dependency of the proxy key on the original signer's private key, the verifiability of the authorization, the distinguishability of the proxy signature and the original signer's signature, the identifiability of the proxy signer, the protection of the original signer, the protection of the proxy signer, and the non-repudiation property of the proxy signature; the third aspect covers the features provided by the proxy multi-signature, including signing under the consensus of the whole group, and the unforgeability of any proxy signature of arbitrary subset of original signers; the last is related to the anonymous group authorization protocol implemented with the zero-sharing, including the anonymity when an original signer vetoes and the secrecy of the private key of an original signer.

First, Pointcheval and Stern [7] pointed out that the GQ signature is existentially unforgeable under the adaptive chosen message attack (EUF-CMA) in the random oracle model.

Open Science Index, Computer and Information Engineering Vol:3, No:5, 2009 publications.waset.org/14356/pdf

Second, the security of a GQ proxy signature is analyzed as follows:

1) **Unforgeability:** The proxy signer receives the proxy key r through a secure channel. Since the proxy key r is itself a standard GQ signature signed by the original signer, the EUF-CMA property of the GQ signature guarantees that only the specified proxy signer and the original signer know the proxy key. Nobody else can produce a valid and verifiable proxy key corresponding to the announced public key of the original signer and, thereby, produce valid proxy signatures. Furthermore, since the proxy signature is also a standard GQ signature, the EUF-CMA property again assures that without the valid proxy key r, nobody can forge successfully a proxy signature.

2) **Key dependency:** The proxy key r is derived from the private key x_i of each original signer in an m-to-1 delegation scenario or in a 1-to-1 delegation scenario. Since each private key x_i is independent of private keys of other players, if one has a proxy key delegated by a subset of original signers (e.g. $r_1 \cdot r_2 \cdot r_3$ from the subset $\{B_1, B_2, B_3\}$), it is not possible to derive the proxy key for a different subset of original signers (e.g. $r_1 \cdot r_2 \cdot r_3 \cdot r_4$ for the subset $\{B_1, B_2, B_3, B_4\}$). Since the proxy key involves a random number u_i and the hash value c, which is related to the current warrant W (i.e. $r_i \equiv u_i \cdot {x_i}^c \pmod n$), if a proxy signer has a delegation received in the past (e.g. $r_1 \equiv u_1 \cdot {x_1}^c \pmod n$), unless he can extract the term ${x_1}^c$ from r_1, there is no chance to deduce ${x_1}^{c'}$ for another hash value c', and obtain a different delegation for other set of original signers (e.g. $r_1' \equiv u_1' \cdot {x_1}^{c'} \pmod n$).

3) **Proxy verifiability:** Because the proxy key is itself a standard GQ signature, the verification of the proxy key is through the standard verification equation. Also, the EUF-CMA property of the GQ signature guarantees that nobody can forge a valid proxy key without the private key of the original signer. Thus, the proxy signer is assured that the original signer has authorized him. A verifier of the proxy signature should verify $c = h(W, n, e, y, a)$ according to the protocol. Thus, by the content of the warrant W, he is assured that the original signer has delegated the signing capacity to the proxy signer.

4) **Signature distinguishability:** Since the public verification key (n, e, y, a, c) of the proxy signature differs from the public verification key (n, e, y) of original signer's signature, a verifier can easily distinguish a signature of the proxy signer from a signature of the original signer.

5) **Proxy signer identifiability:** A verifier can learn the identity of the proxy signer from the delegation warrant W. If this information is tampered, the verification of $c = h(W, n, e, y, a)$ would fail. If c in the proxy signature verification key is modified together, verification of $f = h(M, n, e, y, a, c, b')$ and $b' \equiv s^e \cdot y^{c \cdot f} \cdot a^{-f} \pmod n$ would fail.

6) **Proxy signer deviation:** In the case of a 1-to-1 delegation, a proxy signer gets only a and r, where $a \equiv u^e \pmod n$ and $r \equiv u \cdot x^c \pmod n$. The RSA assumption assures that nobody can calculate in polynomial time the value of u from the first equation. Thus, the value $x^c \pmod n$ cannot be extracted from r, not to mention the value x be extracted from $x^c \pmod n$ which requires the knowledge of $\phi(n)$. In the case of an m-to-1 delegation, the proxy signer can only obtain the product of all proxy keys $\{r_i\}$ if the zero-sharing protocol is secure. To obtain the individual proxy key r_i of a certain original signer, the proxy signer need to collaborate with all other original signers. Even if we have the individual proxy key r_i, the difficulty now is the same as the case of a 1-to-1 delegation, i.e. an adversary needs to break the RSA problem in order to obtain the private key x_i of B_i. Hence, forging a proxy key $r_i \equiv u_i \cdot {x_i}^c \pmod n$, where $c = h(W, n, e, y, a)$, for a different c is difficult.

7) **Proxy signer protection:** The proxy signature scheme in Section IV does not protect the proxy signer since the original signers can issue the proxy signature themselves. However, the scheme can be modified slightly to protect the proxy signer by combining the proxy signer's self delegated proxy key in the proxy key: the group public key y becomes the product of each players' public keys (including the proxy signer's public key), i.e. $y \equiv \prod_{i=1}^{m+1} y_i \left(\equiv \prod_{i=1}^{m+1} {x_i}^{-e} \right) \pmod n$; the proxy key r is the product of each players' proxy key, i.e. $r \equiv \prod_{i=1}^{m+1} r_i \pmod n$k, where r_{m+1} is the proxy key issued by the proxy signer to himself. Corresponding signing and verification of the proxy multi-signatures can be derived easily and are omitted here. Because original signers do not have the private key x_{m+1} of the proxy signer, they can no longer issue the proxy signature by themselves.

8) **Non-repudiation:** The delegation warrant W specifies the identities of the original signers as well as the proxy signer. Nobody can modify this information without being detected since the hash function $h(\cdot)$ is assumed collision resistant. Only the specified proxy signer has the valid proxy key. Hence, if a proxy signature passes the verification equations, the existential unforgeability assures that both the proxy signer and the original signers cannot deny the responsibility.

Kiayias [5] proved for the zero-sharing protocol that any participant B_i can only obtain his secret share z_i and nothing about other's secret share. Also, the value of z_i is independently distributed in $\mathbb{Q}_n$. When B_i calculates $\hat{r_i} \equiv z_i \cdot r_i \pmod n$ in the proposed protocol presented in Section IV, r_i is masked perfectly by z_i. Nobody except B_i himself can extract the secret r_i out of $\hat{r_i}$. Therefore, B_i can transmit the proxy key r_i over a public channel while maintaining its secrecy.

In the GQ proxy multi-signature scheme, the "key dependency" feature assures that all original signers must approve the delegation before the proxy signer can sign legally. Furthermore, without the knowledge of any individual r_i^* (as guaranteed by the the privacy feature of the "zero-sharing" protocol), the proxy signer cannot obtain the proxy key $r' \equiv \prod_{i=1, i\neq i^*}^{m} r_i \pmod{n}$ corresponding to a strict subset of the original signers $\{B_i\}_{i=1,\ldots,m}$ and $_{i\neq i^*}$ and thus cannot issue any proxy signature for the subset.

If an original signer B_i does not wish to authorize B_{m+1}, he can pick a random number in $\mathbb{Q}_n$ to replace $\hat{r}_i$. Since $\hat{r}_i$ in the protocol is uniformly distributed in $\mathbb{Q}_n$, the proxy signer has no way to distinguish a random number from a real $\hat{r}_i$ by the privateness property [5] of the zero sharing protocol. Thus, the veto executed by the original signer is anonymous.

VI. Conclusions

Proxy signature is an indispensable mechanism in the modern e-business and e-government infrastructures. Most of the past researches on this topic focused on the proxy signature scheme itself. However, in many common applications, e.g., a company board authorizes chief executive officer to make decision at a joint venture, a group of authorities make a joint decision and delegate their capacity to a single person. This would have been dealt with by two separate protocols: voting and proxy delegation. In this paper, we propose an integrated mechanism that provides the functions of anonymous private veto, distributed proxy key generation, secure transmission of proxy key, unforgeable proxy mechanism, and proxy multi-signature scheme. This joint mechanism could protect suitably both the original delegators and the proxy signer. The modular design saves both computation and communication than a general purpose multi-party protocol.

References

[1] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signatures for delegating signing operation," Proc. 3rd ACM conference on Computer and Communications Security, CCS'96, 1996.
[2] L. Yi, G. Bai, and G. Xiao, "Proxy multi-signature scheme: a new type of proxy signature scheme," Electronic Letters, Vol. 36, No. 6, pp.527-528, 2000.
[3] Z. Liu, Y. Hu, and H. Ma, "Secure proxy multi-signature scheme in the standard model," Provable Security 2008, pp.127-140, 2008.
[4] C. Boyd, "Digital multisignature," Coding and Cryptography, 1986, pp.15-17, 1986.
[5] A. Kiayias and M. Yung, "Non-interactive zero-sharing with applications to private distributed decision making," Financial Cryptography 2003, pp.303-32, 2003.
[6] L. C. Guillou and J.-J Quisquater, "A paradoxical identity-based signature scheme resulting from zero-knowledge," Advances in Cryptology - Crypto'88, pp.216-23, 1988.
[7] D. Poincheval and J. Stern, "Security proofs for signature schemes," Advances in Cryptology - Eurocrypt'96, pp.387-398, 1996.
[8] S. Kim, S. Park, and D. Won, "Proxy signatures, revisited," ICICS'97, pp.223-232, 1997.
[9] A. Fiat and A. Shamir, "How to prove yourself: practical solution to identification and signature problems," Advances in Cryptology - Crypto'86, pp.186-194, 1986.
[10] D. Chaum and T. P. Pedersen, "Wallet databases with observer," Advances in Cryptology - Crypto'92, pp.89-10, 1992.
[11] V. Shoup, "Practical threshold signatures," Advances in Cryptology - Eurocrypt'00, pp.207-220, 2000.
[12] D. Boneh and M. Franklin, "Efficient generation of shared RSA keys," Advances in Cryptology - Crypto'97, pp.425-439, 1997.

Open Science Index, Computer and Information Engineering Vol:3, No:5, 2009 publications.waset.org/14356/pdf