Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER Applied Mathematics and Computation 166 (2005) 35–45
APPLIED MATHEMATICS AND COMPUTATION
www.elsevier.com/locate/amc

A robust (k,n) + 1 threshold proxy signature scheme based on factoring

RongXing Lu *, ZhenFu Cao, HaoJin Zhu

Department of Computer Science, Shanghai Jiao Tong University, 1954 Huashang Road, Shanghai 200030, People's Republic of China

Abstract

Proxy signature is an active cryptographic research area. Since Mambo et al. introduced the concept of proxy signature in 1996, many proxy signature schemes have been proposed. However, most of these previously proposed schemes are based on discrete logarithm problems. In this paper, we would like to propose a new robust (k,n) + 1 threshold proxy signature scheme based on factoring. In this scheme, generating a valid proxy signature needs not only any k or more members in n proxy signers but also a trusted *dealer* to cooperatively sign a message. To our best knowledge, this is the first (k,n) + 1 threshold proxy signature scheme based on factoring.

Keywords: Proxy signature; Threshold signature; Improved RSA scheme; Factoring

1. Introduction

In 1996, Mambo et al. first introduced the concept of the proxy signature [1,2]. In a proxy signature scheme, an original signer delegates her signing

* Corresponding author.
E-mail address: rxlu@cs.sjtu.edu.cn (RongXing Lu).

0096-3003/$ - see front matter
doi:10.1016/j.amc.2004.04.104

capability to a proxy signer, then the proxy signer can create a valid signature on behalf of the original signer. To verify a proxy signature, a verifier has to verify both the signature itself and original signer's agreement together. Generally, a proxy signature scheme should satisfies two basic security requirements:

- Verifiability: From a proxy signature, any verifier can be convinced of the original signer's agreement on the signed message.
- Unforgeability: Besides the original signer, only a delegated proxy signer can create a valid proxy signature on behalf of the original signer.

Since then, the proxy signature schemes have been widely researched, and many proxy signature schemes were proposed [3–12]. There are several kinds of proxy signature schemes, such as threshold proxy signature scheme, multi-proxy signature scheme and various variants of these schemes *etc*.

The threshold proxy signature schemes were proposed [3–7]. In a (k,n) threshold proxy signature scheme, the original signer can authorize n proxy signers, and only the cooperation of k or more proxy members can generate a valid proxy signature.

The multi-proxy signature scheme is a special case of the threshold proxy signature, which was first proposed in [8]. In a multi-proxy signature scheme, an original signer could authorize n proxy signers as her proxy agent. Only the cooperation of all of the signers can generate a valid proxy signature on behalf of the original signer.

Though the threshold proxy signature and the multi-proxy signature have been researched deeply, there still exist some issues that we should consider.

Any k or more proxy signers can cooperatively sign a message on behalf of the original signer in a (k,n) threshold proxy signature scheme. We can make sure that a valid signature is signed by some proxy signers, but we cannot judge whether a certain proxy signer has participated in signing a message. Therefore, if k or more dishonest proxy signers have signed a message, each of them can deny what he had done.

In a multi-proxy signature scheme, only all proxy signers participate in signing a message, a valid proxy signature could be generated. Therefore, any proxy signer cannot deny his signing behavior. However, when a proxy signer absents, other proxy signers could not produce a valid proxy signature.

To solve the above issues, in this paper, we would like to propose a new robust $(k,n)+1$ threshold proxy signature scheme based on factoring. In the scheme, we add a new role *dealer*, who is trusted by both the original signer and the proxy signers. The *dealer* takes charge of all proxy signature, that is to say, the *dealer* participates in all signature activities. Hence, he can judge who has taken part in signing a message. From this angle, our scheme not only keeps the merits of general threshold proxy signature schemes, but also has multi-proxy signature scheme's advantage. What's more, to our best knowl-

edge, our scheme is the first $(k,n)+1$ threshold proxy signature scheme based on factoring. Most previously proposed threshold proxy signature schemes are all based on discrete logarithm problems, and there still does not exist an indeed threshold proxy signature scheme based on factoring.

The rest of the paper is organized as follows. In Section 2, we review the related building technologies, such as improved RSA cryptosystem, improved RSA signature scheme. Then we propose our scheme in Section 3 and analyze its security in Section 4, respectively. Finally, concluding remarks are made in Section 5.

2. Preliminaries

2.1. Improved RSA cryptosystem

The improved RSA cryptosystem was proposed in [13]. Here, for convenience, we will repeat the scheme briefly as follow:

Choose two large primes p, q randomly, satisfying $p \equiv q \equiv 3 (\mathrm{mod}\, 4)$. Here p, q can be taken as security primes. Let $N = p \cdot q$, then $\phi(N) = (p-1)(q-1)$. Take a satisfying Jacobi symbol $\left(\frac{a}{N}\right) = -1$. Then choose $e \in \mathbb{Z}$ with

$$\gcd\left(e, \frac{1}{2}\phi(N)\right) = 1, \quad 1 < e < \frac{1}{2}\phi(N).$$

We can compute $d \in \mathbb{Z}$, satisfying $ed \equiv \frac{1}{2}\left(\frac{1}{4}\phi(N)+1\right)\left(\mathrm{mod}\, \frac{1}{2}\phi(N)\right)$, $1 < d < \frac{1}{2}\phi(N)$. And then open a, e, N as public key and keep d as secret key.

Encryption algorithm

Suppose that plaintext $x \in \mathbb{Z}_N$, $\gcd(x, N) = 1$. Then

$$E(x) = \begin{cases} x^{2e} \bmod N, & \text{if } \left(\frac{x}{N}\right) = 1, \\ (ax)^{2e} \bmod N, & \text{if } \left(\frac{x}{N}\right) = -1. \end{cases}$$

Ciphertext is $(E(x), c_1, c_2)$ where

$$c_1 = \begin{cases} 0, & x > \frac{N}{2}, \\ 1, & x < \frac{N}{2}, \end{cases} \qquad c_2 = \begin{cases} 0, & \text{if } \left(\frac{x}{N}\right) = 1, \\ 1, & \text{if } \left(\frac{x}{N}\right) = -1. \end{cases}$$

Decryption algorithm

If $c_2 = 0$, then $x^{2e} \equiv E(x) (\mathrm{mod}\, N)$. Compute

$$E(x)^d \equiv x^{2ed} \equiv x^{1+\frac{1}{4}\phi(N)} \equiv \pm x (\mathrm{mod}\, N).$$

Then plaintext x can be obtained from c_1.

If $c_2 = 1$, then $(ax)^{2e} \equiv E(x) (\mathrm{mod}\, N)$. Compute

$$E(x)^d \equiv (ax)^{2ed} \equiv (ax)^{1+\frac{1}{4}\phi(N)} \equiv \pm ax (\mathrm{mod}\, N).$$

That is, $x \equiv \pm a^{-1}(E(x))^d (\mathrm{mod}\, N)$. Therefore, plaintext x can be obtained from c_1.

For the detailed discussion about the improved RSA cryptosystem such as the parameters choosing and security prove, the interested reader can refer to [13,14].

In below, we will introduce the improved RSA signature scheme.

2.2. Improved RSA signature scheme

Signer **A** chooses system parameters p, q, N, a, e and d, which satisfying the same conditions as the improved RSA cryptosystem (for instance, p and q are two security primes). **A** publishes a, e, N as her public key and keeps d as her private key. Furthermore, an universal one-way hash function $H_0 : \{0,1\}^* \to \mathbb{Z}_N^*$ is published.

Signing algorithm

Suppose user **B** wants to get **A**'s signature on message m. **A** will run the following steps:

(1) Apply H_0 to get $H_0(m)$.
(2) Compute c_1, satisfying $\left(\frac{H_0(m)}{N}\right) = (-1)^{c_1}$.
(3) Compute $S = (a^{c_1} \mathrm{H}_0(m))^d (\mathrm{mod}\, N)$, where $\left(\frac{a^{c_1} H_0(m)}{N}\right) = \left(\frac{a^{c_1}}{N}\right)\left(\frac{H_0(m)}{N}\right) = ((-1)^{c_1})^2 = 1$.
(4) Send (m, S, c_1) as a signature on m to **B**.

Verification algorithm

After receiving (m, S, c_1), **B** checks its validity by the following equation:

$$S^{2e} \stackrel{?}{=} (a^{c_1} H_0(m))^{2ed} \equiv (a^{c_1} H_0(m))^{\frac{1}{4}\phi(N)+1} \equiv \pm a^{c_1} H_0(m) (\mathrm{mod}\, N).$$

If it holds, (m, S, c_1) will be accepted, otherwise, it will be rejected.

The improved RSA signature scheme is secure, which security is based on factoring problem. In Section 4, we will prove it in the Random Oracle model.

2.3. Description of Shoup protocol

In a threshold signature scheme, any sub signature should be verified before combining the whole signature. Thus, we have to consider such a verification protocol. In 2000, Shoup proposed an effective verification protocol in [15]. The protocol has been largely applied in secret share and threshold scheme, such as in [16–18]. Therefore, we briefly review the protocol in below:

Assume $m_1, m_2 \in \mathbb{Z}_{\frac{1}{4}\phi(N)}$, signer S owns private key d, and $h_1, h_2 \in \mathbb{Z}_n$ such that

$$h_1 \equiv m_1^d (\mathrm{mod}\, N), \qquad h_2 \equiv m_2^d (\mathrm{mod}\, N).$$

By running the following protocol, S will convinces verifier V that she indeed owns the secret d, but not expose it.

Let H_s be a secure hash function. To illustrate clearly, we denote the protocol as $Shoup(m_1, m_2, h_1, h_2, d)$.

(1) S selects $w \in_R \mathbb{Z}_{\frac{1}{4}\phi(N)}$, computes a_1 and a_2

$$a_1 \equiv m_1^w (\mathrm{mod}\, N), \qquad a_2 \equiv m_2^w (\mathrm{mod}\, N)$$

and computes $c = H_s(m_1, m_2, h_1, h_2, a_1, a_2)$ and $r = dc + w$, then sends (r, a_1, a_2) as the proof of knowing the secret d.

(2) V computes $c = H_s(m_1, m_2, h_1, h_2, a_1, a_2)$, and then checks

$$m_1^r \equiv h_1^c \cdot a_1 (\mathrm{mod}\, N), \qquad m_2^r \equiv h_2^c \cdot a_2 (\mathrm{mod}\, N).$$

If they both hold, V can be convinced that S indeed own the secret d.

$Shoup(m_1, m_2, h_1, h_2, d)$ is an efficient protocol, the security can refer to [15].

3. The proposed scheme

In this section, we will propose our robust $(k, n) + 1$ threshold proxy signature scheme based on factoring. To illustrate clearly, we divide it into five phases: *System initialization, Proxy share generation, Proxy share verifying, Signing phase and Verifying phase.*

3.1. System initialization

An original signer p_0 chooses two distinct secure primes p, q randomly, satisfying $p \equiv q \equiv 3 (\mathrm{mod}\, 4)$. Let $N = p \cdot q$, then $\phi(N) = (p-1)(q-1)$. Take a satisfying Jacobi symbol $\left(\frac{a}{N}\right) = -1$. Then choose $e \in \mathbb{Z}$ with

$$\gcd\left(e, \frac{1}{2}\phi(N)\right) = 1, \quad 1 < e < \frac{1}{2}\phi(N).$$

Compute $d \in \mathbb{Z}$, satisfying $ed \equiv \frac{1}{2}\left(\frac{1}{4}\phi(N) + 1\right)\left(\mathrm{mod}\, \frac{1}{2}\phi(N)\right)$, $1 < d < \frac{1}{2}\phi(N)$. Finally, p_0 publishes a, e, N as her public key and keeps d as her secret key.

Let $H_1 : \{0,1\}^* \to \mathbb{Z}^*_{\frac{1}{4}\phi(N)}$, $H_2 : \{0,1\}^* \to \mathbb{Z}^*_{\frac{1}{4}\phi(N)}$ be public hash functions.

3.2. Proxy share generation

Suppose that the original signer p_0 wants to delegete her signing capacity to n proxy signers $p_1, p_2, \ldots, p_n$ and a trusted *dealer* p_d in such a way that a proxy signature can be created by any subset of k or more members from n proxy

signers together with the trusted *dealer*. To meet this requirement, she should run the following steps:

(1) The original signer p_0 first makes a warrant m_w, which records the delegation policy including limits of authority, valid periods of delegation and the identifiers of the original signer, trusted *dealer* and proxy signers *etc*. Then she computes $H_1(m_w)^{-1}$, satisfying $H_1(m_w)^{-1} \cdot H_1(m_w) \equiv 1 \pmod{\frac{1}{4}\phi(N)}$, and publishes the warrant m_w.

(2) The original signer p_0 selects d_t randomly, satisfying $\gcd\left(d_t, \frac{1}{2}\phi(N)\right) = 1$. Meanwhile let $d_0 \equiv d \cdot d_t \cdot H_1(m_w)^{-1} \pmod{\frac{1}{2}\phi(N)}$, where $1 < d_t < \frac{1}{2}\phi(N)$, $1 < d_0 < \frac{1}{4}\phi(N)$. Here d_0 is called the *shadow* of the original signer's private key d.

(3) The original signer p_0 computes d_t^{-1}, satisfying $d_t^{-1} \cdot d_t \equiv 1 \pmod{\frac{1}{2}\phi(N)}$.

(4) The original signer p_0 randomly chooses a polynomial $f(x) \in \mathbb{Z}_{\frac{1}{4}\phi(N)}[x]$ of degree $k-1$ with $d_0 = f(0)$. Also, she chooses at random $x_1, x_2, \ldots, x_n \in \mathbb{Z}_{\frac{1}{4}\phi(N)}$ satisfying

$$\gcd\left(x_i - x_j, \frac{1}{4}\phi(N)\right) = 1 \quad (i \neq j). \tag{1}$$

So $f(x) = d_0 + c_1 x + \cdots + c_{k-1}x^{k-1}$.

(5) The original signer p_0 can compute $y_i = f(x_i)$ over $\mathbb{Z}_{\frac{1}{4}\phi(N)}$, $i = 1, 2, \ldots, n$. Clearly, if y_{i_l} $(l = 1, 2, \ldots, k)$ are known, then from the interpolation formula we can get

$$d_0 \equiv f(0) \equiv \sum_{1 \leqslant l \leqslant k} y_{i_l} \prod_{1 \leqslant w \leqslant k, w \neq l} (-x_{i_w})(x_{i_l} - x_{i_w})^{-1} \left(\mathrm{mod}\ \frac{1}{4}\phi(N)\right). \tag{2}$$

From Eq. (1), we can see that $(x_{i_l} - x_{i_w})^{-1} \pmod{\frac{1}{4}\phi(N)}$ exists and so Eq. (2) is computable.

(6) The original signer p_0 chooses the following parameters for users: N, e, a, (i, x_i) $(i = 1, 2, \ldots, n)$. And compute

$$z_i \equiv y_i b^{-1} \left(\mathrm{mod}\ \frac{1}{4}\phi(N)\right) \quad (i = 1, 2, \ldots, n), \tag{3}$$

where $b = \prod_{1 \leqslant j < i \leqslant n}(x_i - x_j)$, b^{-1} satisfying $b \cdot b^{-1} \equiv 1 \pmod{\frac{1}{4}\phi(N)}$. From Eq. (1), $b^{-1} \pmod{\frac{1}{4}\phi(N)}$ is computable.

(7) The original signer p_0 chooses $g_t, g_1, g_2, \ldots, g_n \in \mathbb{Z}_{\frac{1}{4}\phi(N)}$ at random, and computes $G_t \equiv g_t^{d_t^{-1}} \pmod{N}$, and $G_i \equiv g_i^{z_i} \pmod{N}$ $(i = 1, 2, \ldots, n)$.

(8) The original signer p_0 secretly sends (i, z_i), $i = 1, 2, \ldots, n$ to n proxy signers p_i $(i = 1, 2, \ldots, n)$, also sends d_t^{-1} to the trusted *dealer* secretly.

(9) Finally, The original signer p_0 publishes all g_i, G_i $(i = 1, 2, \ldots, n)$ and g_t, G_t.

3.3. Proxy share verifying

Each proxy signers p_i ($i = 1, 2, \ldots, n$) can verify his key z_i by checking the following equation:

$$g_i^{z_i} \equiv G_i (\mathrm{mod}\, N). \tag{4}$$

If Eq. (4) holds, the key belongs to him is valid. Otherwise, it is invalid. The trusted *dealer* also can use $g_t^{d_t^{-1}} \equiv G_t (\mathrm{mod}\, N)$ to verify d_t^{-1}.

3.4. Signing phase

Assume a message m should be signed. For simplicity, we will only discuss the situation when $\left(\frac{H_2(m)}{N}\right) = 1$. Any m, satisfying $\left(\frac{H_2(m)}{N}\right) = -1$, can easily be turned into m', where $\left(\frac{m'}{N}\right) = 1$, by multiplying a factor a, satisfying $\left(\frac{a}{N}\right) = -1$.

The trusted *dealer* takes part in the whole signing activity. He is up to collecting all valid sub signatures and combining the whole signature. The detail steps as follows:

(1) Every proxy signer p_i ($i = 1, 2, \ldots, n$) computes $s_i \equiv H_2(m)^{z_i} (\mathrm{mod}\, N)$, and sends s_i to the trusted *dealer*.
(2) The trusted *dealer* receives s_i, he can run $Shoup(g_i, H_2(m), G_i, s_i, z_i)$ with the proxy signer p_i to verify s_i is valid.
(3) In such a way, *dealer* can obtain any k valid sub messages s_{i_l} ($l = 1, 2, \ldots, k$), then he can form the whole signature.
(4) From x_{i_l} and N published by the original signer p_0, *dealer* can compute b'' and b_l ($l = 1, 2, \ldots, k$).
Hence

$$b' = \prod_{1 \leqslant l, w \leqslant k, w < l} (x_{i_l} - x_{i_w}), \quad b'' = \frac{b}{b'}, \quad b_l = \frac{b' \prod_{1 \leqslant w \leqslant k, w \neq l} (-x_{i_w})}{\prod_{1 \leqslant l, w \leqslant k, w \neq l} (x_{i_l} - x_{i_w})}. \tag{5}$$

(5) Then, *dealer* can compute sign:

$$\begin{aligned} \mathrm{sign} &\equiv \prod_{l=1}^{k} s_{i_l}^{b_l \cdot b''} \equiv H_2(m)^{\sum_{l=1}^{k} z_{i_l} \cdot b_l \cdot b''} \equiv H_2(m)^{b'' \sum_{l=1}^{k} z_{i_l} \cdot b_l} \\ &\equiv H_2(m)^{b'' \cdot b^{-1} \sum_{l=1}^{k} y_{i_l} \cdot b_l} \equiv H_2(m)^{b'' \cdot b^{-1} \cdot b' \sum_{l=1}^{k} y_{i_l} \frac{\prod_{1 \leqslant w \leqslant k, w \neq l} (-x_{i_w})}{\prod_{1 \leqslant l, w \leqslant k, w \neq l} (x_{i_l} - x_{i_w})}} \\ &\equiv H_2(m)^{b'' \cdot b^{-1} \cdot b' \cdot d_0} \equiv H_2(m)^{d_0} (\mathrm{mod}\, N). \end{aligned} \tag{6}$$

(6) From d_t^{-1}, *dealer* can compute sign′:

$$\mathrm{sign}' \equiv \mathrm{sign}^{d_l^{-1}} \equiv H_2(m)^{d_0 \cdot d_l^{-1}} \equiv H_2(m)^{d \cdot H_1(m_w)^{-1}} (\mathrm{mod}\, N). \tag{7}$$

(7) Finally, *dealer* sends (sign', m) as a signature of message m to a verifier.

3.5. Verifying phase

When a verifier receives a signature (sign', m), he should run the following steps to verify its validity.

(1) Check the proxy warrant m_w, and compute $H_1(m_w)$.
(2) Verify the following equation:

$$(\text{sign}')^{2eH_1(m_w)} \equiv H_2(m)^{2eH_1(m_w)\cdot d\cdot H_1(m_w)^{-1}} \equiv H_2(m)^{2ed} \equiv H_2(m)^{1+\frac{1}{4}\phi(N)} \equiv \pm H_2(m) (\text{mod}\, N). \tag{8}$$

If it holds, the proxy signature (sign', m) is accepted, otherwise, it will be rejected.

4. Security analysis

In this section, we mainly discuss the security of the proposed scheme.

Here, we first prove that the improved RSA signature scheme in Section 2.2 is secure in Random Oracle Model. The main proof idea is similar to FDH signature scheme's proof [19]. Here, we denote $Cost(\cdot)$ as the main cost of reduction and assume $c_1 = 0$ for proving theorem clearly.

Theorem 1. *Suppose RSA* [1] *is a* (τ', ϵ')*-secure. Then, for any* q_{sig}, q_{hash}, *the improved RSA signature scheme is* $(\tau, q_{\text{sig}}, q_{\text{hash}}, \epsilon)$*-secure, where*

$$\tau = \tau' - (q_{\text{sig}} + q_{\text{hash}} + 1) \cdot Cost(\cdot),$$

$$\epsilon = (q_{\text{sig}} + q_{\text{hash}} + 1) \cdot \epsilon'.$$

Proof. Suppose $\mathscr{A}$ is a forger, who can $(\tau, q_{H_p}, q_{\text{sig}}, \epsilon)$-break the improved RSA signature. we can construct a algorithm $\mathscr{S}$ which takes N, e, y as input and can compute x, satisfying $x^{2e} \equiv y (\text{mod}\, N)$ in t' steps and ϵ' probability where

$$\tau' = \tau + (q_{H_p} + q_{\text{sig}} + 1) \cdot Cost(\cdot), \tag{9}$$

[1] Here, RSA denotes the improved RSA cryptosystem.

$$\epsilon' = \frac{\epsilon}{(q_{\text{sig}} + q_{\text{hash}} + 1)}. \tag{10}$$

Algorithm $\mathscr{S}$ is given as input(N, e, y) where N, e, d were obtained by running the generator $RSA(1^k)$, k is a secure parameter, and y was chosen at random from $\mathbb{Z}_N^*$. It is trying to find a x such that $x^{2e} \equiv y (\text{mod}\, N)$.

Algorithm $\mathscr{S}$ simulates a run of an improved RSA scheme to the forge $\mathscr{A}$. Also, $\mathscr{S}$ should answer $\mathscr{A}$'s hash function queries and signature oracle queries. For simplicity, we assume that if $\mathscr{A}$ makes sign m then it has already made hash oracle query m. It is easily seen to be wlog. Let $q = q_{\text{sig}} + q_{\text{hash}}$. Algorithm $\mathscr{S}$ picks an integer j from $\{1, \ldots, q\}$ at random. Then, we can describe how $\mathscr{S}$ answers oracle queries. Here i is a counter, initially 0.

Suppose $\mathscr{A}$ makes hash oracle query m. Algorithm $\mathscr{S}$ increments i and sets $m_i = m$. If $i = j$ then it sets $y_i = y$ and return y_i. Else it picks r_i at random in $\mathbb{Z}_N^*$, sets $y_i \equiv r_i^{2e} (\text{mod}\, N)$, and returns y_i.

Alternatively, suppose $\mathscr{A}$ makes signing m. By assumption, there was already a hash query of m, so $m = m_i$ for some i. Let $\mathscr{S}$ return the corresponding r_i as the signature.

Eventually, $\mathscr{A}$ returns an attempted forgery (m, x), and $\mathscr{S}$ outputs x. Without loss of generality, we may assume that $m = m_i$ for some i. In this case, if (m, x) is valid forgery, then with probability at least $\frac{1}{q}$, we have $i = j$ and $x \equiv y_i^d \equiv y^d (\text{mod}\, N)$.

The main cost of algorithm $\mathscr{S}$ is that of running the original signer $\mathscr{A}$, hash function queries and signature oracle queries. Thus we can add these values and give the running time in Eq. (9). □

We have proved the improved RSA signature scheme is secure in Theorem 1. Then, let us discuss the important role *dealer* in our scheme. The *dealer* has the following characteristics:

- The *dealer* is trusted by both the original signer and the proxy signers.
- The *dealer* participates in all signing activities and takes charge of collecting sub signatures and combining the whole signature.
- The *dealer* cannot generate a valid proxy signature without interacting with the k or more members in n proxy signers. From Eqs. (6) and (7), he could not get d_0 from $H_2(m)^{d_0}$.

With above characteristic, the *dealer* can become a witness of all valid proxy signatures and our scheme will be more convenient and more security.

From Theorem 1 and characteristic of the *dealer*, in below, we will discuss the proposed $(k, n) + 1$ threshold proxy signature scheme satisfies all the security requirements.

- *Verifiability:* From Eqs. (1)–(8), it is obvious that the proposed scheme satisfies verifiability.
- *Unforgeability:* From Theorem 1 and the role of *dealer*, anyone except the original signer cannot generate a valid proxy key. Unforgeability is satisfied.
- *Undeniablity:* Once a proxy signer p_i participated in signing a message, he cannot repudiate it, because the *dealer* can identify him. On the other hand, though the original signer p_0 can also generate the same proxy signature for any message as the proxy signers create, the *dealer* can identifies whether the signature is signed by the proxy signer or not as his participation in the signing. Therefore, in this sense, the proposed scheme satisfies undeniablity.
- *Robustness:* Assume at least k proxy signers ally to attack the original signer's private key, they may get $d_0 \pmod{\frac{1}{4}\phi(N)}$. But without the trusted *dealer*'s participation, they could not get the original signer's private key d. Hence the proposed scheme satisfies robustness.

5. Conclusions

In this paper, we have proposed the first robust (k,n) + 1 threshold proxy signature scheme based on factoring. The scheme not only inherits the general threshold proxy signature schemes' merits, but also has the multi-proxy signature schemes' advantage. Furthermore, our scheme also satisfies robustness. However, there is a new role *dealer* employed in our scheme. As *dealer* is trustable in all signing activities, therefore, in our future work, we would pay more attention on the *dealer*'s duties.

Acknowledgments

This research is partially supported by the National Natural Science Foundation of China under Grant no. 60072018, the National Natural Science Foundation of China for Distinguished Young Scholars under Grant no. 60225007 and the National Research Fund for the Doctoral Program of Higher Education of China under Grant no. 20020248024.

References

[1] M. Mambo, K. Usuda, E. Okmamoto, Proxy signatures: delegation of the power to sign messages, IEICE Trans. Functional E79-A (9) (1996) 1338–1353.
[2] M. Mambo, K. Usuda, E. Okmamoto, Proxy signatures for delegation signing operation, in: Proceedings of the Third ACM Conference on Computer and Communication Security, New Delhi, India, January 1996, pp. 48–57.

[3] H. Zhang, Threshold proxy signature schemes, 1997 Information Security Workshop, Japan, September 1997, pp. 191–197.
[4] H.-M. Sun, An efficient nonrepudiable threshold proxy signature scheme with known signers, Computer Communications 22 (1999) 717–722.
[5] H.-M. Sun, N.-Y. Lee, T. Hwang, Threshold proxy signatures, IEE Proceedings – Computers and Digital Techniques 146 (5) (1999) 259–263.
[6] C.-L. Hsu, T.-S. Wu, T.-C. Wu, New nonrepudiable threshold proxy signature scheme wit known signers, The Journal of System and Software 58 (2001) 119–124.
[7] M.-S. Hwang, L.-C. Lin, J.-L.L.U. Eric, A secure nonrepudiable threshold proxy signature scheme with known signers, Information 11 (2) (2000) 137–144.
[8] S.J. Hwang, C.-H. Shi, A simple multi-proxy signature scheme, in: Proceedings of the Tenth National Conference on Information Security, Hualien, Taiwan, ROC, 2000, pp. 134–138.
[9] L. Yi, G. Bai, G. Xiao, Proxy multi-signature scheme: a new type of proxy signature scheme, Electronics Letter 36 (6) (2000) 527–528.
[10] S.J. Hwang, C.-H. Shi, The specifiable proxy signature, National Computer symposium 1999, Taipei, Taiwan, ROC, December 1999, pp. 190–197.
[11] H.-M. Sun, B.-T. Hsieh, Time-stamp proxy signatures with traceable receivers, Proceedings of the Ninth National Conference on Information Security, Taichung, Taiwan, ROC, 1999, pp. 247–253.
[12] S.J. Hwang, C.-C. Chen, New multi-proxy multi-signature schemes, Applied Mathematics and Computation 147 (2004) 57–67.
[13] Z. Cao, A threshold key escrow scheme based on public key cryptosystem, Science in China (Series E) 44 (4) (2001) 441–448.
[14] Z. Cao, Two classes of robust threshold key escrow schemes, Journal of Software 14 (6) (2003) 1164–1171.
[15] V. Shoup, Practical threshold signatures, in: Eurocrypt'2000, LNCS, 1807, Springer-Verlag, 2000, pp. 207–220. http://www.shoup.net/papers.
[16] D. Boneh, M. Franklin, Efficient generation of shared RSA keys, Journal of the ACM 48 (4) (2001) 702–722.
[17] G. Wang, S. Qing, M. Wang, Improvement of Shoup's threshold RSA signature scheme 405, Computer Research and Development 39 (9) (2002) 1046–1050 (in Chinese).
[18] G. Wang, S. Qing, M. Wang, Z. Zhou, Threshold undeniable RSA signature schemes, in: S. Qing, T. Okamoto, J. Zhou (Eds.), Information and Communications Security (ICICS 2001), LNCS, 2229, Springer-Verlag, Berlin, 2001, pp. 221–232.
[19] M. Bellare, P. Rogaway, The exact security of digital signatures – how to sign with RSA and Rabin, in: U. Maurer (Ed.), Proceedings of Eurocrypt 1996, May 1996, LNCS, 1070, Springer-Verlag, 1996, pp. 399–416.