J Syst Sci Complex (2011) 24: 816–824

THRESHOLD PROXY RE-SIGNATURE*

Piyi YANG · Zhenfu CAO · Xiaolei DONG

DOI: 10.1007/s11424-011-8370-3
Received: 20 October 2008 / 27 February 2009

Abstract The focus of this paper is to design an efficient and secure solution addressing the semi trusted issue in proxy re-signature schemes, i.e., the proxy knows the re-signature key from user A to user B, so he is able to translate any signatures made by user A to user B, which damages the essential requirement ("non-repudiation" property) of proxy re-signature schemes. In this paper, the authors first define the security model for threshold proxy re-signature scheme, and then propose two threshold proxy re-signature schemes based on Ateniese-Hohenberger's and Shao-Cao-Wang-Liang's approach.

Key words Proxy re-signature, robust, threshold, unforgeable.

1 Introduction

Proxy re-signature scheme, introduced by Blaze, Bleumer, and Strauss[1] at Eurocrypt'98, enables a semi-trusted proxy given some information to transform Alice's signature on a message m into Bob's signature on m, but the proxy cannot, on its own, generate signatures for either Alice or Bob. Although Blaze, Bleumer, and Strauss proposed the idea of a proxy re-signature scheme in 1998, no construction that was both efficient and secure was found until recently, when the work of Ateniese, Hohenberger[2] was published in 2005.

And recently, Shao, et al.[3] have proposed a bidirectional proxy re-signature scheme without random oracle. Libert, et al.[4] have proposed a multi-use unidirectional proxy re-signature scheme. Besides proxy re-signature scheme, a renewed interest of research community in proxy re-encryption[5−7] has been seen in recent years.

Due to the transformation function, proxy re-signature schemes are very useful and can be applied in many applications, including simplifying key management[1], providing a proof for a path that has been taken, managing group signatures, simplifying certificate management[2], constructing a digital rights management (DRM) interoperable system[8].

However, there are some drawbacks in these schemes. The most criticism against these schemes, called semi trusted issue, is that the proxy knows the re-signature key from user A to user B, so he is able to translate any signatures made by user A to user B.

To address the semi trusted issue, we use secret sharing, which is firstly introduced by Shamir[9]. Using this technique, signatures can be translated by a group of proxies rather than

Piyi YANG · Zhenfu CAO · Xiaolei DONG
Department of Computer Science and Engineering, Shanghai Jiaotong University, Shanghai 200240, *China*.
Email: yang.piyi@gmail.com; zfcao@cs.sjtu.edu.cn; xldong@sjtu.edu.cn.
*This research is supported in part by the National Natural Science Foundation of China under Grant Nos. 61033014, 60970110, 60972034, and the National 973 Program under Grant No. 2007CB311201.
◇ *This paper was recommended for publication by Editor Xiao-Shan GAO.*

by one party. In order to translate a valid re-signature on a given signature, the number of the participant players must attain the given threshold value, the re-signature can be created.

1.1 Motivations and Contribution

The aforementioned discussion suggests the first motivation of our proposal, i.e., to construct proxy re-signature scheme under the general threshold scenario. Another motivation is that the general threshold scenario itself is independently interesting, even if the inherent semi trusted problem is not under consideration. To the authors' knowledge, there is no threshold proxy re-signature scheme that has been formally presented yet.

1.2 Organization

Section 2 describes some of the existing tools that we use in our solutions. Section 3 introduces the model and definitions for threshold proxy re-signature scheme. Section 4 shows our two provably secure threshold proxy re-signature schemes and the proofs. Section 5 discusses the efficiency of our threshold proxy re-signature scheme. Finally, Section 6 gives the conclusion.

2 Preliminaries

2.1 Bilinear Pairings and Assumptions

Let us consider two multiplicative group $\mathbb{G}$ and $\mathbb{G}_T$ of the same prime order q. A bilinear pairing is a map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ with the following properties[10]:

Bilinear: $e(aP, bQ) = e(P, Q)^{ab}$, where $P, Q \in \mathbb{G}$, and $a, b \in \mathbb{Z}_q^*$.

Non-degeneracy: There exists $g \in \mathbb{G}$ such that $e(g, g)$ has order p in $\mathbb{G}_T$. In other words, $e(g, g)$ is a generator of $\mathbb{G}_T$, whereas g generates $\mathbb{G}$.

Computability: It is efficient to compute $e(P, Q)$ for all $P, Q \in \mathbb{G}$.

2.2 Threshold Secret Sharing Schemes

Secret sharing schemes were introduced by Shamir[9]. A (n, k) threshold secret sharing scheme distributes a secret s among a set of players $\mathcal{P} = \{R_1, R_2, \cdots, R_n\}$ of n players by a dealer. Each player R_i will privately receive s_i as a share of the secret by the dealer. Then, those subsets with at least k players could recover the secret, while other subsets containing less than k players couldn't gain any information about the secret.

Shamir's solution[9] uses polynomial interpolation. Let $GF(q)$ be a finite field with $q \geq n$ elements, and let $s \in GF(q)$ be the secret to be shared. The dealer randomly picks a polynomial $f(x)$ of degree $k-1$, and the constant of $f(x)$ is s. So $f(x)$ has the form $f(x) = s + \sum_{j=1}^{k-1} a_j x^j$.

If we assign every player R_i with a unique field element α_i. Then the dealer sends the secret share $s_i = f(\alpha_i)$ to R_i through a private channel. Now, if the set of players $A \subset \mathcal{P}$ such that $|A| \geq k$, then they could recover the secret $s = f(0)$ by using the following formula:

$$f(\alpha_j) = \sum_{R_i \in A} \lambda_{ij}^A f(\alpha_i) = \sum_{R_i \in A} \lambda_{ij}^A s_i,$$

where

$$\lambda_{ij}^A = \prod_{R_l \in A, l \neq i} \frac{\alpha_j - \alpha_l}{\alpha_i - \alpha_l}.$$

On the other hand, it can be proved that if the subset $B \subset \mathcal{P}$ such that $|B| < k$ couldn't get any information about the polynomial $f(x)$.

3 Definitions

3.1 Threshold Proxy Re-Signature

We extend the definition of proxy re-signature scheme[2] to define threshold proxy re-signature scheme.

Definition 1 A threshold proxy re-signature scheme is a tuple of (possibly probabilistic) polynomial time algorithms (KeyGen, ShareRekey, Sign, ShareResign, Combine, Verify), where:

(KeyGen, Sign, Verify) form the standard key generation, signing, and verification algorithms.

On input $(pk_A, sk_A^*, pk_B, sk_B)$, the share re-signature key generation algorithm, ShareRekey, outputs n keys $rk_{A\to B}^i$ for the proxies $P_1, P_1, \cdots, P_n$. The input marked with a '$*$' is optional.

On input a public key pk_A, a signature σ_A, and a message m, the proxy P_i runs ShareResign with $rk_{A\to B}^i$ to get the re-signature share $\sigma_{B,i}$.

On input a public key pk_A, a signature σ_A, a message m, and k re-signature share $\sigma_{B,i}$, the re-signature share combine function, combine, outputs σ_B if $\text{Verify}(pk_A, m, \sigma_A) = 1$ and $\perp$ otherwise.

3.2 Security Model

Definition 2 The (k, n) threshold proxy re-signature scheme is unforgeable (denoted by UF-THPRS-CMA) if no polynomial bounded adversaries $A_i (i = 1, 2, 3, 4)$ has a non-negligible advantage in the following game. In the definition, A_1 means an outside attacker, A_2 means malicious proxies, A_3 means malicious proxies with dishonest delegator, A_4 means malicious proxies with dishonest delegatee.

To simulate these different kinds of adversaries, we define the following oracles:

$\mathcal{O}_{U\text{KeyGen}}$: Obtain a new key pair $(pk, sk) \leftarrow \text{KeyGen}(1^k)$. The adversary is given pk.

$\mathcal{O}_{C\text{KeyGen}}$: Obtain a new key pair $(pk, sk) \leftarrow \text{KeyGen}(1^k)$. The adversary is given (pk, sk).

$\mathcal{O}_{\text{ShareRekey}}$: On input (pk, sk^*, pk', sk') by the adversary, where pk, sk^*, pk', sk' were generated before by KeyGen and the input marked with a '$*$' is optional, return the re-signature key share $rk_{pk\to pk'}^i = \text{ShareRekey}(pk, sk^*, pk', sk')$, where sk, sk' are the secret keys that corresponds to pk, pk'. Notice that for A_3, sk' couldn't be the secret key of the delegatee, which means A_3 can't get the re-signature key from any user to the delegatee.

$\mathcal{O}_{\text{ShareResign}}$: On input $(pk, pk', rk_{pk\to pk'}^i, m, \sigma)$, where pk, pk' were generated before by KeyGen. The adversary is given the re-signed signature share

$$\sigma_i' = \text{ShareResign}(rk_{pk\to pk'}^i, pk, pk', m, \sigma).$$

$\mathcal{O}_{\text{Combine}}$: Performs the same as the scheme.

$\mathcal{O}_{\text{Sign}}$: On input a public key pk, a message m, the adversary is given the corresponding signature $\sigma = \text{Sign}(sk, m)$, where sk is the secret key corresponding to pk. Notice that in A_3, the sk is the delegatee's secret key and in A_4, the sk is the delegator's secret key.

Init The adversary outputs a set $S \subset \{1, 2, \cdots, n\}$ of k–1 re-signature proxies to corrupt.

Query phase The adversary adaptively issues queries $\mathcal{O}_{U\text{KeyGen}}, \mathcal{O}_{C\text{KeyGen}}, \mathcal{O}_{\text{ShareReKey}}$, $\mathcal{O}_{\text{ShareResign}}, \mathcal{O}_{\text{Sign}}$.

Forgery The adversary outputs a message m^*, a public key pk^*, and a string σ^*. The adversary succeeds if the following hold true:

$\text{Verify}(pk^*, m^*, \sigma^*) = 1$.

For A_3, pk^* is the public key of delegatee. For A_4, pk^* is the public key of delegator and σ^* is first-level signature.

pk^* is not from $\mathcal{O}_{C\text{KeyGen}}$.

(pk^*, m^*) is not a query to $\mathcal{O}_{\text{Sign}}$.

$(\Diamond, pk^*, m^*, \blacklozenge)$ does not exist or it is not a query to $\mathcal{O}_{\text{ShareResign}}$ for the uncorrupted proxies S, where $\Diamond$ denotes any public key, and $\blacklozenge$ denotes any signature.

The advantage of adversaries $A_i (i = 1, 2, 3, 4)$ in the above game is defined to be $Adv_{A_i} = \texttt{Pr}[A_i \text{ succeeds}]$, where the probability is taken oven all coin tosses made by the challenger and the adversaries.

Definition 3 (Robustness) A (k, n) threshold proxy re-signature scheme THPRS is said to be robust if it computes a correct output even in the presence of a malicious attacker that makes the corrupted signature generation servers deviate from the normal execution.

3.3 Relationship Between UF-PRS-CMA and UF-THPRS-CMA

We use Gennaro et al.'s[11] methodology for proving the security of threshold proxy re-signature, we define simulatability of THPRS as follows.

Definition 4 (Simulatability of THPRS) Let THPRS=(KeyGen, ShareRekey, ShareResign, Combine, Sign, Verify) be a (k, n) threshold proxy re-signature scheme. The scheme THPRS is said to be simulatable if the following conditions hold.

ShareRekey is simulatable: There exists a simulator $SIM_{\text{ShareRekey}}$ that, given two public keys (pk_A, pk_B), can simulate the view of the attacker on an execution of ShareRekey of THPRS.

ShareResign is simulatable: There exists a simulator $SIM_{\text{ShareResign}}$ that, given two public key pk_A and pk_B, two signatures σ_A and σ_B, k–1 shares of the re-signature key $rk^i_{A\to B}$, and a message m, can simulate the view of the attacker on an execution of ShareResign of THPRS.

We now state and prove the following theorem regarding the relationship between the security of threshold proxy re-signature (THPRS) and that of proxy re-signature (PRS). The implication of the theorem is that if we have a UF-PRS-CMA secure proxy re-signature scheme, we can use it as a building block for a UF-THPRS-CMA secure threshold proxy re-signature scheme by ensuring simulatability. The reader can refer to Ateniese, et al.'s paper[1] for details of UF-PRS-CMA security of proxy re-signature scheme.

Theorem 1 *If the THPRS scheme is simulatable and the PRS scheme associated with the THPRS scheme is UF-PRS-CMA secure, then the THPRS scheme is UF-THPRS-CMA secure. We have the following bound:*

$$Adv^{UF-THPRS-CMA}_{THPRS}(t_{CMA}, q_{Rekey}, q_{Resign}) \le Adv^{UF-PRS-CMA}_{PRS}(t'_{CMA}, q'_{Rekey}, q'_{Resign}),$$

where $t'_{CMA} = t_{CMA} + t_{SIM_{\text{ShareRekey}}} + t_{SIM_{\text{ShareResign}}}$. *Here,* $t_{SIM_{\text{ShareRekey}}}$ *and* $t_{SIM_{\text{ShareResign}}}$ *denote the running time of the simulators* $SIM_{\text{ShareRekey}}$ *and* $SIM_{\text{ShareResign}}$, *respectively.*

Proof Let A_{THPRS} denote the attacker of the threshold proxy re-signature scheme and A_{PRS} denote the attacker of the proxy re-signature scheme.

We show how an A_{THPRS} could help A_{PRS} to break the underlying proxy re-signature scheme, under the assumption that THPRS is simulatable.

First, we use A_{PRS}'s common parameter as A_{THPRS}'s common parameter. We then do the following.

Whenever A_{THPRS} issues a $\mathcal{O}_{U\text{KeyGen}}$ or $\mathcal{O}_{C\text{KeyGen}}$ query, we intercept it and forward to A_{PRS}'s challenger. The challenger returns the resulting public key or public/secret key pair. We simply send it back to A_{THPRS}.

Whenever A_{THPRS} issues a $\mathcal{O}_{\text{ShareRekey}}$ query on target pk_B which is generated by $\mathcal{O}_{U\text{KeyGen}}$. We run $SIM_{\text{ShareRekey}}$ taking pk_A, pk_B as input to simulate the view of A_{THPRS}.

Whenever A_{THPRS} issues a $\mathcal{O}_{\text{ShareRekey}}$ query on other pk'_B which is not the target and is generated by $\mathcal{O}_{C\text{KeyGen}}$. We run ShareRekey using pk_A, pk'_B, sk_A, sk'_B, and return the result to

A_{THPRS}.

Whenever A_{THPRS} issues a $\mathcal{O}_{\mathrm{ShareResign}}$ query on target pk_B which is generated by $\mathcal{O}_{U\mathrm{KeyGen}}$. We run $SIM_{\mathrm{ShareResign}}$ taking σ_A, σ_B, m and k–1 re-signature key shares as input to simulate the view of A_{THPRS}.

Whenever A_{THPRS} issues a $\mathcal{O}_{\mathrm{ShareResign}}$ query on other pk'_B which is not the target and is generated by $\mathcal{O}_{C\mathrm{KeyGen}}$. We run ShareRekey using $pk_A, sk^*_A, pk'_B, sk'_B$ to get n shares of re-signature key, run $ShareResign$ using $rk^i_{A\to B}, \sigma_A$ and return the result to A_{THPRS}.

If A_{THPRS} outputs $\tilde{pk}, \tilde{M}, \tilde{\sigma}$, we intercept it and return it as A_{PRS}'s forgery. Since $THPRS$ is simulatable, the A_{THPRS}'s view from the simulation is identical to its view in the real attack game. Considering the running time and the number of queries, we obtain the bound in the theorem statement. ∎

4 Threshold Proxy Re-Signature Schemes

In this section, we extend two proxy re-signature schemes to two threshold proxy re-signature schemes: Schemes 1 and 2. Furthermore, we give their security proof in this section.

4.1 Scheme 1

In this subsection, we propose our first threshold proxy re-signature scheme which is extended from [3]. It is bidirectional.

We assume that the length of the message is n_m.

KeyGen Chooses two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of prime order q, from which an admissible pairing e: $\mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ can be constructed. Let g be a generator of $\mathbb{G}_1$. Furthermore, it selects a random a from $\mathbb{Z}_p$, and $n_m + 2$ random number $(g_2, u', u_1, \cdots, u_{n_m})$ from $\mathbb{G}_1$, and output the key pair $pk = g_1 = g^a$ and $sk = a$, the public parameters $(\mathbb{G}_1, \mathbb{G}_2, e, g_2, u', u_1, \cdots, u_{n_m})$.

ShareRekey On input two secret keys $sk_A = a, sk_B = b$, the dealer performs as follows:

a) Chooses a random polynomial $f(x)$ of degree $k-1$ as Section 2.2:

$$f(x) = \sum_{i=0}^{k-1} a_i x_i^i,$$

such that $f(0) = b$.

b) Broadcast $A_i = g^{\frac{f(i)}{a}}$ and $B_i = g^{f(i)}$ for $i = 0, 1, 2, \cdots, n$. Notice that $B_0 = pk_B$.

c) Compute the shares $rk^i_{A\to B} = \frac{f(i)}{a}$ for $i = 1, 2, \cdots, n$ and sends $rk^i_{A\to B}$ to Proxy P_i by a secret channel.

d) After receiving $rk^i_{A\to B}$ from the dealer, each proxy P_i randomly chooses A_{k_j}'s from $A_k (k = 0, 1, \cdots, i-1, i+1, \cdots, n)$. Let $K = \{k_j\}$. Check

$$g^{rk^i_{A\to B}} = A_i \stackrel{?}{=} \prod_{j=0}^{k-1} A_{k_j}^{\lambda^K_{i,k_j}},$$

where λ^K_{i,k_j}'s are the Lagrange interpolation coefficients defined in Section 2.2. If it does not hold, proxy P_i broadcasts a complaint against the dealer.

If more than k proxies report complaints about the dealer, he is clearly bad and disqualified. Otherwise, the dealer distributes the share of re-signature key $rk_{A\to B} = \frac{b}{a}$ to n proxies.

Sign On input a secret key $sk = a$ and an n_m bit message

$$m = (l_1 \cdots l_i \cdots l_{n_m}),$$

l_i is 0 or 1, output $\sigma = (U, V) = (g_2^a \cdot w^r, g^r)$, where r is randomly chosen from $\mathbb{Z}_q$, and $w = u' \prod_{i=1}^{n_m} u_i^{l_i}$.

Verify On input a public key pk, an n_m bit message m, and a signature $\sigma = (U, V)$, output 1 if $e(pk, g_2)e(V, w) = e(U, g)$ and 0 otherwise.

ShareResign On input a public key pk_A, a signature σ_A, and an n_m bit message

$$m = (l_1 \cdots l_i \cdots l_{n_m}),$$

l_i is 0 or 1, the ith proxy which holds the rekey share $rk^i_{A\to B}$ check that Verify$(pk_A, m, \sigma_A) = 1$. If σ_A does not verify, output $\perp$. Otherwise, output the re-signature share $\sigma_{B,i} = \sigma_A^{rk^i_{A\to B}} = (U_i, V_i) = (g_2^{f(i)} \cdot w^{\frac{rf(i)}{a}}, g^{\frac{rf(i)}{a}})$.

Combine Each proxy P_j verifies the shares (U_i, V_i) he received from the other proxies. For each $i = 1, 2, \cdots, n$, P_j checks if

$$e(B_i, g_2)e(V_i, w) \stackrel{?}{=} e(U_i, g). \tag{1}$$

If the check fails for an index i, P_j broadcasts a complaint against P_i. Each proxy marks as disqualified any party that received more than k complaints. Each proxy then build the set of non-disqualified parties $QUAL$. If $|QUAL| >= k$, we could compute the re-signature

$$\begin{aligned}\sigma_B &= \textstyle\prod_{i\in QUAL} \sigma_{B,i}^{\lambda_{0,i}^{QUAL}} \\ &= \Big(\textstyle\prod_{i\in QUAL} g_2^{\lambda_{0,i}^{QUAL} f(i)} \cdot w^{\lambda_{0,i}\frac{rf(i)}{a}}, \prod_{i\in QUAL} g^{\lambda_{0,i}^{QUAL}\frac{rf(i)}{a}} \Big) \\ &= (g_2^b w^{\frac{rb}{a}}, g^{\frac{rb}{a}}).\end{aligned}$$

Lemma 1 *The scheme* 1 *is simulatable.*

Proof To prove the simulatability of ShareRekey, the simulator first chooses k–1 random v_i from $\mathbb{Z}_p$, and computes k–1 pieces of g^{v_i}. Using k–1 pieces of g^{v_i} and the public key $pk_B = g^b$, the simulator constructs $g^{f(x)}$ that $g^{f(0)} = g^b = pk_B$ using Lagrange interpolation. Then it computes $g^{\frac{f(x)}{a}} = g^{f(x)} \cdot (g^a)^{-1} = g^{f(x)} \cdot pk_A^{-1}$. Since $A_i = g^{\frac{f(i)}{a}}, B_i = g^{f(i)}$, we have proved ShareRekey is simulatable.

To prove the simulatability of ShareResign, the simulator use Lagrange interpolation to compute the shares of re-signature

$$\sigma_{B,i} = \left\{ U_B^{\lambda_{i,0}^{\Phi}} \prod_{j=1}^{k-1} g_2^{a\lambda_{i,j}^{\Phi} f(j)/a} \cdot w^{\lambda_{i,j}^{\Phi} rf(j)/a}, V_B^{\lambda_{i,0}^{\Phi}} \prod_{j=1}^{k-1} g^{\lambda_{i,j}^{\Phi} rf(j)/a} \right\},$$

by $\sigma_B = \{U_B, V_B\} = \{g_2^b w^{rb/a}, g^{rb/a}\}$ and k–1 shares of re-signature key $rk^j_{A\to B} = f(j)/a$ for $i = 1, 2, \cdots, n$. Φ is the set that holds 0 and the index of k–1 signature shares. Notice that the k–1 corrupted players can pass the Equation (1). ∎

Combining Theorem 1 and Lemma 1, we have the following theorem.

Theorem 2 *Our scheme* 1 *is unforgeable, if the proxy re-signature scheme*[3] *is unforgeable.*

Robustness The following theorem can be easily proven by inspection of the scheme 1.

Theorem 3 *Our scheme* 1 *is* (k, n) *robust, if* $n \geq 2k - 1$.

Proof In the process of ShareRekey, we assume the dealer is malicious. Then both the broadcast value A_i, B_i and the rekey shares will be altered. The modification of A_i, B_i and rekey shares could be discovered easily by any proxy using Lagrange interpolation. In the

process of combination, without loss of generality, we assume proxy P_i intend to deviate the output signature. Therefore, the shares (U_i, V_i) generated by P_i contains the wrong value. Since there are at least k honest proxies in the system, we will receive more than k complaints against P_i. As a result, it will be removed from the qualified set $QUAL$. ∎

4.2 Scheme 2

In this subsection, we propose our second threshold proxy re-signature scheme which is extended from [2]. This scheme is unidirectional.

KeyGen On input the security parameter 1^k, chooses two groups $\mathbb{G}_1$ and $\mathbb{G}_2$ of prime order $q = \Theta(2^k)$, from which an admissible pairing e: $\mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ can be constructed. Let g, h be a generator of $\mathbb{G}_1$. The global parameters are $(e, q, \mathbb{G}_1, \mathbb{G}_2, g, h, H)$, where H is a hash function from arbitrary strings to elements in $\mathbb{Z}_q$.

Select a random $a \in \mathbb{Z}_q$, and output the key pair $pk = (g^a, h^{1/a})$ and $sk = a$.

ShareRekey On input a public key $pk_A = (g^a, h^{1/a})$ and a secret key $sk_B = b$, the dealer performs as follows:

a) Chooses a random polynomial $f(x)$ of degree k–1 as Section 2.2:

$$f(x) = \sum_{i=0}^{k-1} a_i x_i^i$$

such that $f(0) = b$.

b) Broadcast $A_i = g^{f(i)}$ for $i = 0, 1, \cdots, n$.

c) Compute the shares $rk^i_{A\to B} = h^{\frac{f(i)}{a}}$ for $i = 1, 2, \cdots, n$ and sends $rk^i_{A\to B}$ to Proxy P_i by a secret channel.

d) After receiving $rk^i_{A\to B}$ from the dealer, each proxy P_i randomly chooses A_{k_j}'s from $A_k (k = 0, 1, \cdots, i-1, i+1, \cdots, n)$. Let $K = \{k_j\}$. Check

$$e(rk^i_{A\to B}, g) = e(h^{f(i)/a}, g) = e(A_i, h^{1/a}) \stackrel{?}{=} e\left(\prod_{j=0}^{k-1} A_{k_j}^{\lambda^K_{i,k_j}}, h^{1/a}\right),$$

where λ^K_{i,k_j}'s are the Lagrange interpolation coefficients defined in Section 2.2. If it does not hold, proxy P_i broadcasts a complaint against the dealer.

If more than k proxies report complaints about the dealer, he is clearly bad and disqualified. Otherwise, the dealer distributes the share of re-signature key $rk_{A\to B} = h^{b/a}$ to n proxies.

Sign On input a secret key $sk = a$ and a message m, select a random $k \in \mathbb{Z}_q$, set $r = h^k, s = a(H(m||r) + k) (\text{mod } q)$; output the pair $\sigma = (r, s)$. We call a signature of this form a first-level signature.

ShareResign On input a public key pk_A, a signature σ_A, and a message m, the ith proxy which holds the rekey share $rk^i_{A\to B}$ check that $\text{Verify}(pk_A, m, \sigma_A) = 1$. If σ_A does not verify, output $\perp$. Otherwise, output the re-signature share $\sigma_{B,i} = (r, (rk^i_{A\to B})^s) = (r, s_i) = (h^k, h^{f(i)(H(m||r)+k)})$.

Combine Each proxy P_j verifies the shares (r, s_i) he received from the other proxies. For each $i = 1, 2, \cdots, n$, P_j checks if

$$e(g, s_i) \stackrel{?}{=} e(A_i, rh^{H(m||r)}).$$

If the check fails for an index i, P_j broadcasts a complaint against P_i. Each proxy marks as disqualified any party that received more than k complaints. Each proxy then build the set of non-disqualified parties $QUAL$. If $|QUAL| \geq k$, we could compute the re-signature

$$\sigma_B = \left(r, \prod_{i \in QUAL} s_i^{\lambda_{0,i}^{QUAL}}\right) = \left(r, \prod_{i \in QUAL} h^{\lambda_{0,i}^{QUAL} f(i)(H(m||r)+k)}\right) = (r, h^{b(H(m||r)+k)}).$$

Verify On input a public key $pk = (pk^{(1)}, pk^{(2)})$, a message m, and a signature $\sigma = (r, s)$ (if σ is a *first-level* signature, set $s = h^s$), output 1 if $e(g, s) = e(pk^{(1)}, rh^{H(m||r)})$ and 0 otherwise.

Lemma 2 *The scheme* 2 *is simulatable.*

Proof To prove the simulatability of ShareRekey, the simulator first chooses k–1 random v_i from $\mathbb{Z}_p$, and computes k–1 pieces of g^{v_i}. Using k–1 pieces of g^{v_i} and the public key $pk_B = g^b$, the simulator constructs $g^{f(x)}$ that $g^{f(0)} = g^b = pk_B$ using Lagrange interpolation. Since $A_i = g^{f(i)}$, we have proved ShareRekey is simulatable.

To proof the simulatability of ShareResign, we need $\sigma_B = \{r_B, s_B\}$ and k–1 re-signature shares to construct the other shares of the re-signatures by Lagrange interpolation as follows:

$$s_{B,i} = s_B^{\lambda_{i,0}^{\Phi}} \prod_{j=1}^{k-1} h^{\lambda_{i,j}^{\Phi} f(j)(H(m||r)+k)}.$$

Φ is the set that holds 0 and the index of k–1 signature shares. ∎

Combining Theorem 1 and Lemma 2, we have the following theorem.

Theorem 4 *Our scheme* 2 *is unforgeable, if the proxy re-signature scheme*[2] *is unforgeable.*

Robustness The following theorem can be easily proven by inspection of the scheme 2.

Theorem 5 *The scheme* 2 *is* (k,n) *robust, if only* $n \geq 2k - 1$.

Proof In the process of ShareRekey, we assume the dealer is malicious. Then both the broadcast value A_i and the rekey shares will be altered. The modification of A_i and rekey shares could be discovered easily by any proxy using Lagrange interpolation. In the process of combination, without loss of generality, we assume proxy P_i intend to deviate the output signature. Therefore, the shares (r, s_i) generated by P_i contains the wrong value. Since there are at least k honest proxies in the system, we will receive more than k complaints against P_i. As a result, it will be removed from the qualified set $QUAL$. ∎

5 Performance Analysis

In this section, in terms of computational complexity, we show that even though we address the key escrow problem of the proxy re-signature scheme, the cost is still affordable compared with the original proxy re-signature scheme. Note that in order to resign a signature, the threshold proxy re-signature scheme needs to perform ShareResign and Combine.

The performance evaluation notations are defined as Tables 1 and 2, where

T_{exp}: time for an exponetiation computation,

T_{pair}: time for a bilinear pairing.

Table 1 Scheme 1 Performance evaluation

	Original Proxy Re-Signature Scheme	Threshold Proxy Re-Signature Scheme
Resign	$2T_{\text{exp}}$	$(2k+2)T_{\text{exp}} + 3T_{\text{pair}}$

Table 2 Scheme 2 Performance evaluation

	Original Proxy Re-Signature Scheme	Threshold Proxy Re-Signature Scheme
Resign	$1T_{\text{exp}}$	$(k+1)T_{\text{exp}} + 2T_{\text{pair}}$

6 Conclusion

In this paper, we define the security model for threshold proxy re-signature scheme and proposed two threshold proxy re-signature schemes based on Ateniese-Hohenberger's[2] and Shao-Cao-Wang-Liang's[3] approach. Additionally, we develop the relationship between simulatable threshold proxy re-signature scheme and the underlying proxy re-signature scheme. The proposed schemes manage to limit the re-signature proxy's power, to reduce the risk of single point failure, and to enhance the system's robustness.

References

[1] M. Blaze, G. Bleumer, and M. Strauss, Divertible protocols and atomic proxy cryptography, *EUROCRYPT, LNCS*, 1998, **1403**: 127–144.
[2] G. Ateniese and S. Hohenberger, Proxy re-signatures: New definitions, algorithms, and applications, 12*th ACM Conference on Computer and Communications Security*, New York, 2005.
[3] J. Shao, Z. F. Cao, L. C. Wang, and X. H. Liang, Proxy re-signature schemes without random oracles, *INDOCRYPT, LNCS*, 2007, **4859**: 197–209.
[4] B. Libert and D. Vergnaud, Multi-use unidirectional proxy re-signatures, 15*th ACM Conference on Computer and Communications Security*, New York, 2008.
[5] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, Improved proxy re-encryption schemes with applications to secure distributed storage, *ACM Trans. Inf. Syst. Secur.*, 2006, **9**(1): 1–30.
[6] R. Canetti and S. Hohenberger, Chosen-ciphertext secure proxy re-encryption, 14*th ACM Conference on Computer and Communications Security*, New York, 2007.
[7] M. Green and G. Ateniese, Identity-based proxy re-encryption, *Applied Cryptography and Network Security, LNCS*, 2007, **4521**: 288–306.
[8] G. Taban, A. A. C'ardenas, and V. D. Gligor, *Towards a Secure and Interoperable DRM Architecture*, ACM DRM, New York, 2006.
[9] A. Shamir, How to share a secret, *Communications of the ACM* 1979, **22**(11): 612–613.
[10] D. Boneh and M. Franklin, Identity-based encryption from the weil pairing, *SIAM Journal of Computing*, 2003, **32**(3): 586–615.
[11] R. Gennaro, S. Halevi, and T. Rabin, Secure hash-and-sign signatures without the random oracle, *EUROCRYPT, LNCS*, 1999, **1592**: 123–139.