A Novel Non-repudiable Threshold Proxy Signature Scheme with Known Signers

Samaneh Mashhadi

Department of Mathematics, Iran University of Science & Technology
Narmak, Tehran, 1684613114, Iran
(Email: smashhadi@iust.ac.ir)

(Received Apr. 20, 2011; revised and accepted July 13, 2011)

Abstract

In 2004, Yang et al. proposed an efficient (t, n) threshold proxy signature scheme with known signers. However, Maimani et al. showed that a malicious original signer can forge a valid proxy signature for any message in Yang et al.'s scheme and further proposed an improvement to remedy such an attack. In this paper, we will show that in Maimani et al.s improvement, a malicious original or proxy signer can forge a valid threshold proxy signature for any message by different ways. Furthermore, we propose a new threshold proxy signature scheme that remedies the weakness of Maimani et al.'s scheme.

Keywords: Non-repudiation, threshold proxy signature, unforgeability

1 Introduction

The concept of proxy signature was first proposed by Mambo et al. [4] in 1996. A proxy signature scheme allows a signer to delegate the signing capability to a designated person (call a proxy signer), the proxy signer can generate proxy signature of a message on behalf of the original signer [5, 8, 9]. A (t, n) threshold proxy signature scheme, which is a variant of the proxy signature scheme, the proxy signature key is shared among a group of n proxy signers delegated by the original signer. Any t or more proxy signers can cooperatively sign messages on behalf of the original signer, but $t-1$ or fewer proxy signers cannot. In order to create a secure (t, n) threshold proxy signature scheme, the following security requirements for proxy signature should be satisfied [10]:

Secrecy. The original signer's private key cannot be derived from any information, such as the parameters sent to the proxy signers. Even if the adversary corrupts all proxy signers, he cannot get the original signer private key.

Proxy protection. The partial proxy signature cannot be generated by others except the delegated proxy signer. Even if the adversary corrupts the original signer and t or more proxy signers, he cannot generate a valid partial proxy signature of a proxy signer he does not corrupt.

Unforgeability. The valid proxy signature can only be cooperatively generated by t or more delegated proxy signer. Even if the adversary corrupts the original signer and $t-1$ proxy signers, he cannot generate a valid proxy signature.

Non-repudiation. The proxy group cannot repudiate the proxy signatures they created, and the original signer cannot deny having delegated the power of signing messages to the proxy group.

Time constraint. The proxy signing keys can be only used in the authorized time.

Known signers. From a proxy signature, the identities of the actual signers can be determined.

In 2001, Hsu et al. [1] pointed out that Sun et al.'s threshold proxy signature scheme [7] was also vulnerable to collusion attack, especially, the proxy signers could change the parameter t in the process of cooperatively generating the proxy signature. To remedy the weaknesses, they gave a new improvement. In 2004, Yang et al. [10] proposed a new threshold proxy signature scheme, which was more efficient than Hsu et al.'s scheme [1]. Subsequently, during 2007–09, Shao [6], Maimani [3], and Hu [2] separately proposed some attacks to show that Yang et al.'s scheme [10] have serious security flaws. Via Maimani et al.'s attack [3], a malicious original signer can forge a valid proxy signature for any message in Yang et al.'s scheme. Further, Maimani et al. [3] proposed an improvement to remedied such an attack. In this paper, we will show that Maimani et al.'s scheme is also vulnerable to Hu attacks [2], that is a malicious original or proxy signer can forge a valid threshold proxy signature for any message. Moreover, we will show that [3] is also vulnerable to warrant

attack. That is, a malicious proxy signer can also change the content of warrant such as the identities of the original and proxy signers of the proxy group, the parameters t, n, the valid delegation time, etc. and sign the arbitrary message m by himself without the assistance of other proxy signers. Therefore, [3] cannot satisfy the properties of non-repudiation and unforgeability. Furthermore, we propose a new threshold proxy signature scheme that remedies the weakness of Maimani et al.'s scheme.

2 Brief Review of Maimani et al.'s Scheme

In this section, we briefly review Maimani et al.'s scheme [3]. The scheme consists of four phases: the initialization, the proxy share generation, the proxy signature generation, and the proxy signature verification phases.

2.1 Initialization Phase

Let p be a large prime, q a prime divisor of $p-1$, g a generator in $\mathbb{Z}_p^*$ with order q, and $h(.)$ a secure one-way hash function. The parameters (p, q, g) are public. Suppose that P_0 is the original signer, and $G = \{P_1, P_2, \ldots, P_n\}$ is the proxy group of n proxy signers. The original signer P_0 determines its private key and public key as $x_0 \in \mathbb{Z}_q^*$, $y_0 = g^{x_0} \bmod p$ respectively. By the same way, each proxy signer $P_i \in G$ owns its private key $x_i \in \mathbb{Z}_q^*$ and public key $y_i = g^{x_i} \bmod p$, which are certified by the certificate authority (CA). Let m_W be a warrant which records the identities of the original and proxy signers of the proxy group, the parameters t, n, the valid delegation time, etc. Also $\|$ denotes the concatenation of strings.

2.2 Proxy Share Generation Phase

The original signer P_0 randomly chooses an integer $k \in \mathbb{Z}_q^*$ and computes $K = g^k \bmod p$. Then P_0 computes $\sigma = x_0 h(m_W \| K) + k \bmod q$ as the key of the proxy group and then broadcasts (σ, m_W, K) to the proxy signers of G. After receiving (σ, m_W, K), each proxy signer $P_i \in G$ checks whether the equation

$$g^{\sigma} = {y_0}^{h(m_W \| K)} K \bmod p$$

holds or not. If it holds, each P_i regards σ as its proxy key.

2.3 Proxy Signature Generation Phase

For convenience, let $D = \{P_1, P_2, \ldots, P_t\}$ be t actual proxy signers, ASID the identities of t proxy signers, C the receiver, and m the message to be signed. D, as a proxy group, performs the following steps:

1) Each $P_i \in D$ chooses random $k_i \in \mathbb{Z}_q^*$ and then computes and broadcasts $r_i = g^{k_i} \bmod p$;

2) After receiving $r_j (j = 1, 2, \ldots, t, j \neq i)$, each $P_i \in D$ computes $R = \prod_{j=1}^{t} r_j \bmod p$ and

$$S_i = k_i R + (t^{-1}\sigma + x_i K) h(R \| m \| ASID) \bmod q.$$

Next they send S_i to the designated receiver C via a secret channel.

3) After receiving S_i, the receiver C checks whether the following equation holds:

$$g^{S_i} = {r_i}^R \Big(\big(K {y_0}^{h(m_W \| K)} \big)^{t^{-1}} {y_i}^K \Big)^{h(R \| m \| ASID)} \bmod p.$$

If it holds, (r_i, S_i) is a valid partial proxy signature; then he computes $S = \Sigma_{i=1}^{t} S_i$. Therefore, $(R, S, K, m_W, ASID)$ is the threshold proxy signature of the message m.

2.4 Proxy Signature Verification Phase

From m_W, the verifier can get the threshold value t, and from ASID, he can get the number of actual proxy signers. The verifier checks the validity of proxy signature $(R, S, K, m_W, ASID)$ for the message m by the following equality:

$$g^S = R^R \Big(K {y_0}^{h(m_W \| K)} \big(\prod_{i=1}^{t} y_i \big)^K \Big)^{h(R \| m \| ASID)} \bmod p.$$

3 Attacks on Maimani et al.'s Scheme

We now describe some possible attacks on Maimani et al.'s scheme. In this section, we shall show that their scheme cannot resist the frame attack, public-key substitute attack, and warrant attack which shows that their scheme cannot achieve their claimed security requirements and satisfy the properties of non-repudiation and unforgeability.

3.1 Public-key Substitute Attacks

Recently, J. Hu and J. Zhang [2] presented an efficient public-key substitute attack on Yang et al.'s scheme [10]. A malicious attacker (consisting of the original signer and any proxy signer) can forge a valid proxy signature of any message by changing its public key.

In the similar manners, we will show that Maimani et al.'s scheme is also vulnerable to these attacks.

Suppose that a malicious original signer P_0 wants to forge a valid general proxy signature $(R', S', K', m_W, ASID)$ for his arbitrary chosen message m' and claim dishonestly that it is generated by other t proxy signers $D = \{P_1, P_2, \ldots, P_t\}$. Let $ASID$ be the identities of D.

For this purpose, P_0 can choose random integers $\alpha, \beta \in \mathbb{Z}_q^*$ and computes

$$\begin{aligned} R' &= g^{\beta} \bmod p, \\ y_0' &= \Big(K(\prod_{i=1}^{t} y_i)^{K} g^{\alpha}\Big)^{-h(m_W\|K)^{-1}} \bmod p. \end{aligned}$$

and requests CA to replace his public key with the above y_0'. Next he computes

$$S' = \beta R' - \alpha h(R'\|m\|ASID) \bmod q.$$

$(R', S', K', m_W, ASID)$ is the valid proxy signature of the message m. This is because

$$\begin{aligned} & g^{S'} = g^{\beta R' - \alpha h(R'\|m\|ASID)} \bmod p \\ = \; & R'^{R'}\Big(K {y_0'}^{h(m_W\|K)}(\prod_{i=1}^{t} y_i)^{K}\Big)^{h(R'\|m\|ASID)} \bmod p. \end{aligned}$$

In the verification stage, any verifier can verify the validity of the proxy signature so ASID may incorrectly record the identities as actual signers of the proxy group. In fact, $P_1, P_2, \ldots, P_t$ have never signed the message m, but they cannot deny.

Without loss of generality, suppose that a malicious proxy signer P_1 tries to forge a general proxy signature scheme of a message m'. For this purpose, P_1 chooses random $\alpha, \beta, \gamma \in \mathbb{Z}_q^*$, and computes

$$\begin{aligned} & R' = g^{\beta} \bmod p, \quad K' = g^{\gamma} \bmod p, \\ & S' = \beta R' + \alpha h(R'\|m\|ASID) \bmod q, \\ & y_1 = \Big({K'}^{-1} y_0^{-h(m_W\|K')}\big(\textstyle\prod_{i=2}^{t} y_i\big)^{-K'} g^{\alpha}\Big)^{K'^{-1}} \bmod p. \end{aligned}$$

Then he wants CA to replace his public key with the above y_1. $(R', S', K', m_W, ASID)$ is the valid proxy signature of the message m. This is because

$$\begin{aligned} & g^{S'} = g^{\beta R' + \alpha h(R'\|m\|ASID)} \bmod p = \\ & R'^{R'}\Big(K' y_0^{h(m_W\|K')}(\prod_{i=1}^{t} y_i)^{K'}\Big)^{h(R'\|m\|ASID)} \bmod p. \end{aligned}$$

3.2 Warrant Attack

Here, we will show that their scheme cannot also resist the warrant attack. That is, a malicious proxy signer P_1 can also change the content of warrant such as the identities of the original and proxy signers of the proxy group, the parameters t, n, the valid delegation time, etc. and sign the arbitrary message m by himself without the assistance of other proxy signers in D.

Assume that P_1 wants to forge a threshold proxy signature of any message m and claims that is generated by t' proxy signers $D' = \{P_1, P_2, \ldots, P_{t'}\}$, while the proxy group D' don't know it. Let $ASID'$ be the identities of group D'. Firstly, P_1 generates the warrant m_W' as he wants, chooses random integers $\alpha, \beta, \gamma \in \mathbb{Z}_q^*$, and computes

$$R' = g^{\beta} \bmod p, \qquad K' = g^{\gamma} \bmod p.$$

Then he computes

$$y_1 = \Big({K'}^{-1} y_0^{-h(m_W'\|K')}(\prod_{i=2}^{t'} y_i)^{-K'} g^{\alpha}\Big)^{K'^{-1}} \bmod p,$$

and requests CA to replace his public key with the above y_1. Next he computes

$$S' = \beta R' + \alpha h(R'\|m\|ASID') \bmod q.$$

$(R', S', K', m_W', ASID')$ is the valid proxy signature of the message m. This is because

$$\begin{aligned} g^{S'} &= g^{\beta R' + \alpha h(R'\|m\|ASID')} \bmod p \\ &= R'^{R'}\Big(K' y_0^{h(m_W'\|K')}(\prod_{i=1}^{t'} y_i)^{K'}\Big)^{h(R'\|m\|ASID')} \bmod p. \end{aligned}$$

4 Improvement of Maimani et al.'s Scheme

In this section, we will modify the Maimani et al.'s scheme to remedy the weakness as described in Section 3.
In the proxy signature generation phase, we replace the partial signature S_i with

$$S_i = k_i + (\sigma y_i + x_i y_0 K) h(R\|m\|m_W\|ASID) \bmod q$$

and replace the partial signature verification equation with

$$g^{S_i} = r_i\Big(\big(K y_0^{h(m_W\|K)}\big)^{y_i} y_i^{K y_0}\Big)^{h(R\|m\|m_W\|ASID)} \bmod p.$$

Also, in the proxy signature verification stage, the verifier checks the validity of proxy signature $(R, S, K, m_W, ASID)$ for the message m by the following equality:

$$\begin{aligned} & R \Big(\big(K y_0^{h(m_W\|K)}\big)^{\sum_{i=1}^{t} y_i}(\prod_{i=1}^{t} y_i)^{K y_0}\Big)^{h(R\|m\|m_W\|ASID)} \\ = \; & g^{S} \bmod p. \end{aligned}$$

If it does, $(R, S, K, m_W, ASID)$ is the valid proxy signature of message m. This is because

$$\begin{aligned} & R \Big(\big(K y_0^{h(m_W\|K)}\big)^{\sum_{i=1}^{t} y_i}\big(\textstyle\prod_{i=1}^{t} y_i\big)^{K y_0}\Big)^{h(R\|m\|m_W\|ASID)} \\ = \; & g^{\sum_{i=1}^{t} k_i + (\sigma y_i + x_i y_0 K) h(R\|m\|m_W\|ASID)} \\ = \; & g^{\sum_{i=1}^{t} S_i} = g^{S} \bmod p. \end{aligned}$$

Table 1: Security comparison of threshold schemes with proposed scheme

Security features	Yang [10]	Shao [6]	Hu [2]	Maimani [3]	Proposed
Secrecy.	Yes	Yes	Yes	Yes	Yes
Proxy protection.	Yes	Yes	Yes	Yes	Yes
Unforgeability.	No	No	Yes	No	Yes
Non-repudiation.	No	No	Yes	No	Yes
Time constraint.	No	Yes	Yes	No	Yes
Known signers.	No	No	Yes	No	Yes
Secure channel.	No	No	No	No	No
Scheme can resist public-key substitute attacks.	No	No	Yes	No	Yes
Scheme can resist warrant attacks.	No	Yes	Yes	No	Yes
At least t proxy signers can generate proxy signature.	No	No	Yes	No	Yes

5 Discussions

Before examining the security of our improved scheme, we give the following theorems.

Theorem 1 *If*

$$y_0 = \Big((\prod_{i=1}^{t} y_i)^{y_0' K'} K'^{-1+\sum_{i=1}^{t} y_i} \Big)^{-\left(h(m_W \| K') \sum_{i=1}^{t} y_i\right)^{-1}},$$

then $(R', S', K', m_W, ASID)$ *given by*

$$\begin{aligned} R' &= g^{\beta} \bmod p, \\ K' &= g^{\alpha} \bmod p, \quad \textit{and} \\ S' &= \beta + \alpha h(R' \| m' \| m_W \| ASID) \bmod q, \end{aligned}$$

is a valid proxy signature.

Proof. It is a valid proxy signature of the message m' because

$$\begin{aligned} & g^{S'} = g^{\beta} g^{\alpha h(R' \| m' \| m_W \| ASID)} \bmod p \\ = \; & R' \big(K' K'^{\sum_{i=1}^{t} y_i} K'^{-\sum_{i=1}^{t} y_i}\big)^{h(R' \| m' \| m_W \| ASID)} \\ & \Big(\big(\prod_{i=1}^{t} y_i\big)^{-K' y_0} \big(\prod_{i=1}^{t} y_i\big)^{K' y_0} \Big)^{h(R' \| m' \| m_W \| ASID)} \\ = \; & R' \big(K' {y_0}^{h(m_W \| K')}\big)^{\sum_{i=1}^{t} y_i h(R' \| m' \| m_W \| ASID)} \\ & \big(\prod_{i=1}^{t} y_i\big)^{K' y_0 h(R' \| m' \| m_W \| ASID)} \bmod p. \end{aligned}$$

□

Theorem 2 *If*

$$y_1 = \Big(\big(K {y_0}^{h(m_W \| K)}\big)^{\sum_{i=1}^{t} y_i} (\prod_{i=2}^{t} y_i)^{y_0 K} g^{\alpha} \Big)^{-(K y_0)^{-1}},$$

then $(R', S', K, m_W, ASID)$ *given by*

$$\begin{aligned} R' &= g^{\beta} \bmod p, \\ S' &= \beta - \alpha h(R' \| m \| m_W \| ASID) \bmod q, \end{aligned}$$

is a valid proxy signature.

Proof. It is a valid proxy signature of the message m because

$$\begin{aligned} & g^{S'} = g^{\beta - \alpha h(R' \| m \| m_W \| ASID)} \\ = \; & g^{\beta} \Big(\big(\prod_{i=2}^{t} y_i\big)^{y_0 K} \big(\prod_{i=2}^{t} y_i\big)^{-y_0 K} g^{-\alpha} \Big)^{h(R' \| m \| m_W \| ASID)} \\ & \big(K K^{-1} (y_0 y_0^{-1})^{h(m_W \| K)}\big)^{\sum_{i=1}^{t} y_i h(R' \| m \| m_W \| ASID)} \\ = \; & R' \big(K {y_0}^{h(m_W \| K)}\big)^{\sum_{i=1}^{t} y_i h(R' \| m' \| m_W \| ASID)} \\ & \big(\prod_{i=1}^{t} y_i\big)^{K y_0 h(R' \| m' \| m_W \| ASID)} \bmod p. \end{aligned}$$

□

Now, we examine the security of our scheme.

5.1 Public-key Substitute Attacks

Consider the scenario of a public-key substitute attack made by original signer. Assume that a malicious original signer P_0 wants to forge a valid general proxy signature $(R', S', K', m_W, ASID)$ for his arbitrary chosen message m' and claim dishonestly that it is generated by other t proxy signers $D = \{P_1, P_2, \ldots, P_t\}$. Let $ASID$ be the identities of D.

For this purpose, P_0 can choose random integers $\alpha, \beta \in \mathbb{Z}_q^*$ and computes $K' = g^{\alpha} \bmod p$, and $R' = g^{\beta} \bmod p$. Now, P_0 should determine

$$y_0' = \Big((\prod_{i=1}^{t} y_i)^{y_0' K'} K'^{-1+\sum_{i=1}^{t} y_i} \Big)^{-\left(h(m_W \| K') \sum_{i=1}^{t} y_i\right)^{-1}} \bmod p,$$

and

$$S' = \beta + \alpha h(R' \| m' \| m_W \| ASID) \bmod q.$$

Then he should want CA to replace his public key with the above y_0'. However, P_0 should solve the discrete logarithm problem $\prod_{i=1}^{t} y_i = g^c \bmod p$ in order to compute the above y_0' as follows:

$$y_0' = g^{-\left(c y_0' K' - \alpha + \sum_{i=1}^{t} y_i\right)\left(h(m_W \| K') \sum_{i=1}^{t} y_i\right)^{-1}}.$$

Thus P_0 cannot forge a valid general proxy signature of any message m', which generated by D.

Consider the scenario of a public-key substitute attack made by proxy signers.

Without loss of generality, suppose P_1 tries to forge a threshold proxy signature $(R', S', K', m_W, ASID)$ by the public-key substitute attack and claim dishonestly that is generated by the proxy signers of $D = \{P_1, P_2, \ldots, P_t\}$. For this purpose, P_1 chooses random $\alpha, \beta, \gamma \in \mathbb{Z}_q^*$ and computes

$$R' = g^{\beta} \bmod p, \quad K' = g^{\gamma} \bmod p,$$
$$S' = \beta - \alpha h(R' \| m \| m_W \| ASID) \bmod q,$$
$$y_1 = \left(\left(K' {y_0}^{h(m_W \| K')} \right)^{\sum_{i=1}^{t} y_i} \left(\prod_{i=2}^{t} y_i \right)^{y_0 K'} g^{\alpha} \right)^{-(K' y_0)^{-1}}.$$

Then he wants CA to replace his public key with the above y_1; However, P_0 should solve the discrete logarithm problems $y_0 = g^{x_0} \bmod p$ and $\prod_{i=2}^{t} y_i = g^d \bmod p$ in order to compute the above y_1 as follows:

$$y_1 = g^{\left(\gamma x_0 h(m_W \| K') \sum_{i=1}^{t} y_i + d y_0 K' + \alpha\right)(-K' y_0)^{-1}}.$$

Therefore, a proxy signer P_i can't forge a valid threshold proxy signature by the public-key substitute attack.

5.2 Warrant Attacks

Without loss of generality, suppose P_1 tries to sign the arbitrary message m by himself without the assistance of other proxy signers in D. For this purpose, P_1 can choose m'_W, t' and random integers $\alpha, \beta, \gamma \in \mathbb{Z}_q^*$ and compute

$$R' = g^{\beta} \bmod p; \qquad K' = g^{\gamma} \bmod p.$$

Then, he should calculate new y_1 and S' as follows:

$$y_1 = \left(\left(K' {y_0}^{h_1(m'_W \| K')} \right)^{\sum_{i=1}^{t'} y_i} \left(\prod_{i=2}^{t'} y_i \right)^{y_0 K'} g^{\alpha} \right)^{-(K' y_0)^{-1}} \bmod p,$$
$$S' = \beta - \alpha h(R' \| m \| m'_W \| ASID') \bmod q.$$

Then he wants CA to replace his public key with the above y_1. But P_1 cannot compute y_1, because of the difficulty of solving discrete logarithm problems $y_0 = g^{x_0} \bmod p$ and $\prod_{i=2}^{t'} y_i = g^a \bmod p$. Therefore, any proxy signer P_i can't forge a valid threshold proxy signature by this warrant attack.

Shao et al. [6] presented an warrant attack on Yang et al.'s scheme [10]. As Shao has analyzed, this security leak is caused by the fact that the individual signature S_i is independent of the warrant m_W. Thus, the adversary can easily substitute the proxy certificate and frame the innocent proxy signers. In our improved scheme, the warrant m_W is a part of $h_1(m_W \| K)$ in individual signature

$$S_i = k_i + (\sigma y_i + x_i y_0 K) h(R \| m \| m_W \| ASID) \bmod q.$$

Thus, after intercepting a valid proxy signature $(R, S, K, m_W, ASID)$, it is impossible for anyone to replace (m_W, σ) by (m'_W, σ'), and in the same time the following equality holds:

$$h(R \| m \| m_W \| ASID) = h(R \| m \| m'_W \| ASID).$$

This is because $h(\cdot)$ is a collision resistant hash function. Hence, our scheme can resist this warrant attack.

5.3 Security Properties

Based on discrete logarithm problem, it is virtually impossible to obtain the original signer's private key x_0 from the corresponding public key y_0. Also, according to the Schnorr signature scheme, it is very difficult for anyone to obtain x_0 from σ. Therefore, the original signature's private key can be kept secretly and be reused during the span of the system. Hence, Therefore, the property of *secrecy* is fulfilled in our scheme.

Similarly, the original signer cannot obtain the proxy signer's private key x_i and masquerade as a proxy signer to create a partial proxy signature. This protects the authority of the proxy signer. Hence, Therefore, the property of *proxy protection* is fulfilled in our scheme.

An intruder may try to derive a forged proxy signature by using the previous attacks. But, we have shown that all attacks fail on our scheme. Therefore, the proposed scheme ensures that a valid proxy signature would be generated only when t or more proxy signers can cooperatively sign the message which satisfies the properties of *unforgeability*.

The property of *non-repudiation* is that both the original signer and the actual proxy signers cannot deny generating the valid proxy signature. Any valid proxy signature $(R, S, K, m_W, ASID)$ of a message m should be generated by t or more proxy signers. This is because only P_i has the private key x_i, from the corresponding public key y_i. Thus, P_i cannot deny signing the partial proxy signature. Moreover, the warrant m_W is created by the original signer. The original signer cannot deny delegating the power of signing messages to the proxy signers. Therefore, the valid proxy signature was signed on behalf of the original signer. Hence, both the original signer and the actual proxy signers cannot deny generating the valid proxy signature.

Time constraint denotes the valid time period of the delegation of the signing power. In our scheme, the warrant m_W which records the stipulated period of this proxy is created only by the original signer and it is impossible for anyone to change m_W. In the verification stage, the verifier checks whether or not the warrant has expired. Therefore, our scheme satisfies the property of time constraint.

Finally, from $ASID$, the verifier can know who the actual signers are. In our scheme, any receiver is able to identify the actual signers in the proxy group. Furthermore, the adversary cannot replace $ASID$ by $ASID'$ satisfying $h(R \| m \| m_W \| ASID) = h(R \| m \| m_W \| ASID')$, since $h(\cdot)$ is a collision resistant hash function; it is computationally infeasible to get such an $ASID'$. Therefore, our scheme satisfies the property of *known signers*.

From what have been analyzed above, we are fully certain that the necessary requirements of (t, n) threshold proxy signature scheme are fulfilled in our scheme.

6 Performance

In this section, in terms of computational complexity, we compare the new proxy signature scheme with threshold proxy signature schemes proposed in [1, 2, 3, 6, 7, 10] and summarize the result in Table 2. For convenience, the following notations are used to analyze the computational complexity.

T_e**:** The time for one exponentiation computation.

T_m**:** The time for one modular multiplication computation.

Table 2: Comparison of computational complexity previous schemes with proposed scheme

Scheme	Share generation	Signature generation	Verification
Sun [7]	$(5n+2t+1)T_e$ $+(nt+2t)T_m+T_H$	$(4t^2-t)T_e+(t^2-t)T_i$ $+(10t^2-14t)T_m+2T_H$	$4T_e+(t+3)T_m+2T_H$
Hsu[1]	$(5n+2t+1)T_e$ $+(nt+2t)T_m+T_H$	$(t^2+4t+1)T_e+(t^2-1)T_i$ $+(4t^2+2t)T_m+2T_H$	$4T_e+(t+3)T_m+2T_H$
Yang [10]	$3T_e+2T_m+T_H$	$(4t+2)T_e+(t^2+4t+1)T_m+2T_H$	$4T_e+(t+2)T_m+2T_H$
Shao [6]	$3T_e+2T_m+T_H$	$(4t+2)T_e+(t^2+5t-1)T_m+2T_H$	$4T_e+(t+3)T_m+2T_H$
Hu [2]	$(4n+t+1)T_e$ $+(nt+n+2)T_m+T_H$	$(4t+2)T_e+(3t+3)T_m+2T_H$	$5T_e+(t+2)T_m+2T_H$
Maimani [3]	$3T_e+2T_m+T_H$	$(4t+2)T_e+(t^2+5t+1)T_m+2T_H$	$5T_e+(t+2)T_m+2T_H$
Proposed	$3T_e+2T_m+T_H$	$(4t+2)T_e+(t^2+3t+1)T_m+2T_H$	$5T_e+(t+3)T_m+2T_H$

T_H: The time for hash function computation.

T_i: The time for one inverse computation.

From Table 2, we can see that our scheme can reduce computation costs, and it is the most efficient and the most secure non-repudiable threshold proxy signature scheme with known signers.

7 Conclusions

In order to some practical application, Yang et al. proposed a threshold proxy signature scheme. But their scheme has some weaknesses. Maimani et al.'s showed that a malicious original signer can forge a valid proxy signature for any message in Yang et al.'s scheme and further proposed an improvement to remedied such an attack. However, Maimani et al.'s scheme has some similar weaknesses. In this paper, we have pointed out the security leakage of Maimani et al.'s scheme and further proposed a novel non-repudiable threshold proxy signature scheme with known signers, which not only keeps the previous schemes's merits but also overcomes the security weaknesses.

References

[1] C. L. Hsu, T. S. Wu, and T.C. Wu, "New nonrepudiable threshold proxy signature scheme with known signers," *The Journal of Systems and Software*, vol. 58, no. 2, pp. 119–124, 2001.

[2] J. Hu and J. Zhang, "Cryptanalysis and improvement of a threshold proxy signature scheme," *Computer Standards & Interfaces*, vol. 31, pp. 169–173, 2009.

[3] H. R. Kakayi, H. R. Maimani, and M. Bagheri, "A method for improvement of some threshold proxy signature schemes," in *Proceedings of 4th Iranian Society of Cryptology Conference*, pp. 133–140, Tehran, Iran, 2007.

[4] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signature: delegation of the power to sign messages," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E79-A, no. 1, pp. 1338–1353, 1996.

[5] S. Mashhadi, "A novel secure self proxy signature scheme," *International Journal of Network Security*, vol. 14, no. 6, pp. 22–26, 2012.

[6] J. Shao and R. Lu Z. Cao, "Improvement of yang et al.s threshold proxy signature scheme," *The Journal of Systems and Software*, vol. 80, no. 6, pp. 172–177, 2007.

[7] H. M. Sun, "An efficient nonrepudiable threshold proxy signature scheme with known signers," *Computer Communications*, vol. 22, no. 8, pp. 717–722, 1999.

[8] Z. W. Tan, "Improvement on nominative proxy signature schemes," *International Journal of Network Security*, vol. 17, no. 8, pp. 175–180, 2008.

[9] M. Tian and L. Huang, "Breaking a proxy signature scheme from lattices," *International Journal of Network Security*, vol. 14, no. 8, pp. 320–323, 2012.

[10] C. H. Yang, S. F. Tzeng, and M.S. Hwang, "On the efficiency of nonrepudiable threshold proxy signature scheme with known signers," *The Journal of Systems and Software*, vol. 73, no. 3, pp. 507–514, 2004.

Samaneh Mashhadi was born in Tafresh, Iran, on March 27, 1982. She received the B.Sc. and M.Sc. degrees with honors in pure mathematics from Iran University of Science and Technology (IUST), and Amirkabir University of Technology (AUT) in 2003 and 2005, respectively. She received her Ph.D. with honors in mathematics (Cryptography) in 2008 from IUST. She is currently an assistant professor in department of mathematics of IUST. She is a member of IMS as well. Her research interests include the analysis, design, and application of digital signatures, secret sharing schemes, and security protocols, etc.