INFORMATICA, 2005, Vol. 16, No. 2, 261–274
© 2005 *Institute of Mathematics and Informatics, Vilnius*

Efficient Pairing-Based Threshold Proxy Signature Scheme with Known Signers

Haifeng QIAN, Zhenfu CAO [*], Qingshui XUE
Department of Computer Science and Engineering, Shanghai Jiao Tong University
1954 Huashan Road, Shanghai 200030, the People's Republic of China
e-mail: ares@sjtu.edu.cn, cao-zf@cs.sjtu.edu.cn, xue-qsh@cs.sjtu.edu.cn

Received: November 2003

Abstract. Since threshold proxy signature has been proposed, all threshold proxy signature schemes are based on the discrete logarithm problems in the modular multiplicative group which has an element g with a large prime order. Nowadays this kind of threshold proxy signature schemes become more and more complex and time-consuming for security and specific requirement. In this paper, we propose a (bilinear) pairing-based threshold proxy signature scheme with known signers, analyze its security and check the following properties the proposed scheme has: *non-repudiation, unforgeability, identifiability, distinguishability, verifiability, prevention of misuse of proxy signing right*, etc. Moreover, we point out that the proposed scheme is of great efficiency by comparing it with Sun's and Hsu *et al.*'s scheme.

Key words: cryptography, digital signatures, proxy signature, threshold proxy signature, bilinear pairing.

1. Introduction

The proxy signature scheme (Mambo *et al.*, 1996a; Mambo *et al.*, 1996b) a variation of ordinary digital signature schemes, which enables a proxy signer to sign messages on behalf of the original signer, has many applications in mobile agent environment and electronic transaction. There are, so far,three types of delegation: *full delegation, partial delegation, and delegation by warrant*. In the full delegation, a proxy signer is given the same private key as the original signer has, and computes the same signatures as the original signer does. In the partial delegation (Mambo *et al.*, 1996b), the original signer uses his private key to create a proxy signature key and sends it to the proxy signer in a secret way. The proxy signer uses the proxy signature key to compute proxy signatures on behalf of the original signer. For the security reason, it must be computationally infeasible to compute the original signer's private key from the proxy signer's proxy signature key. In the delegation by warrant (Neuman, 1993), the original signer gives the proxy signer a warrant, composed of a message part and a public signature key, which certifies that the proxy signer is legal. Then the proxy signer use the corresponding private key to sign the message on behalf of the original signer.

[*] Corresponding author.

Following the development of proxy signature scheme (Hsu *et al.*, 2001; Hwang *et al.*, 2000; Hwang and Chen, 2003; Li and Cao, 2002; Li *et al.*, 2002; Li *et al.*, 2003a; Li *et al.*, 2003b; Mambo *et al.*, 1996a; Mambo *et al.*, 1996b; Neuman, 1993; Sun, 1999; Sun *et al.*, 1999; Zhang and Kim, 1997; Zhang and Kim, 2003), the threshold proxy signature was also widely studied in (Hwang and Chen, 2003; Hsu *et al.*, 2001; Hwang *et al.*, 2000; Li and Cao, 2002; Sun, 1999; Zhang and Kim, 1997; Zhang and Kim, 2003). In a (t, n) threshold proxy signature scheme, the original signer authorizes a proxy group with n proxy members. Any t or more proxy signers can cooperatively employ the proxy signature keys to sign messages on behalf of an original signer, but $t - 1$ or fewer proxy signers cannot. Threshold proxy signature with known signers is proposed by Sun (Sun, 1999) in 1999, which has the property that the t proxy signers' identity who cooperate to generate the proxy signature can be verified in the equation of verification. After that, Hwang and Sun *et al.* (Hwang *et al.*, 2000) pointed out that Sun's scheme was insecure against collusion attack. By the collusion, any $t - 1$ proxy signers among the t proxy signers can cooperatively obtain the secret key of the remainder one. Then they also proposed an improved scheme which can guard against the collusion attack. However, Sun's scheme is vulnerable against conspiracy attack for another weakness. That is, any t malicious proxy signers can collusively derive the secret keys of the other proxy signers in the group and can impersonate some other proxy signers to generate proxy signatures.

Up to the present, all threshold proxy signature schemes are still based on the discrete logarithm problems in the multiplicative group Z_p^* where p is a large prime. This kind of threshold proxy signature schemes become more and more complex and time-consuming for security and specific requirement. Since the GDH signature (short signature scheme) in (Boneh *et al.*, 2001) has been proposed by Bonel et al many cryptosystems based on bilinear, non-degenerate, efficiently computable mappings (called pairings) over certain groups have been widely studied. So in this paper we propose a new kind of threshold proxy signature with known signers based on pairings which could be built from Weil pairing or Tate pairing on an elliptic curve or a supersinglar elliptic curve.

At first we will introduce some related work about the bilinear pairings, then state the proposed threshold proxy signature scheme based on bilinear pairings. Next we analyze the security of the proposed scheme. After that, we will compare the proposed scheme with Sun's and Hsu *et al.*'s scheme in terms of computational complexities in some cases. Finally, we will draw a conclusion on the whole paper.

2. Background and Related Work

Here we summarize some concepts of bilinear pairings using similar notations as in (Zhang and Kim, 2003).

2.1. *Bilinear Pairings*

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be additive and multiplicative groups of the same prime order q, respectively. Let P be a generator of $\mathbb{G}_1$. Assume that the discrete logarithm problems in both

$\mathbb{G}_1$ and $\mathbb{G}_2$ are hard to solve. Let $\hat{e}: \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ be a pairing which satisfies the following properties:

1. *Bilinear*: $\hat{e}(aP, bP^{'}) = \hat{e}(P, P^{'})^{ab}$ for all P, $P^{'} \in \mathbb{G}_1$ and all $a, b \in \mathbb{Z}$.
2. *Non-degenerate*: If $\hat{e}\ (P, P^{'}) = 1, \forall P^{'} \in \mathbb{G}_1$ then $P = \mathcal{O}$.
3. *Computable*: There is an efficient algorithm to compute $\hat{e}(P,P^{'})$ for any $P,P^{'} \in \mathbb{G}_1$.

To construct the bilinear pairing, we can use the Weil pairing or revised Tate pairing associated with supersinglar elliptic curves.

With such a group $\mathbb{G}_1$, we can define the following hard cryptographic problems:

- **Discrete Logarithm (DL) Problem:** Given $P, P^{'} \in \mathbb{G}_1$, find an integer n such that $P = nP^{'}$ whenever such integer exists.
- **Computational Diffie–Hellman (CDH) Problem:** Given a triple $(P, aP, bP) \in \mathbb{G}_1^3$, for $a, b \in \mathbb{Z}_q^*$, find the element abP.
- **Decision Diffie–Hellman (DDH) Problem:** Given a quaternion $(P, aP, bP, abP) \in \mathbb{G}_1^4$, for $a, b, c \in \mathbb{Z}_q^*$, decide whether $c = ab\ (mod\ q)$ or not.
- **Gap Diffie–Hellman (GDH) Problem:** A class of problems where the CDH problem is hard but the DDH problem is easy.

Groups where the CDH problem is hard but the DDH problem is easy are called Gap Diffie–Hellman (GDH) groups Details about them can be seen in (Boldyreva, 2003; Boneh *et al.*, 2001; Boneh and Franklin, 2001; Boneh *et al.*, 2003; Joux and Nguyen, 2001).

2.2. *A GDH Signature Scheme*

A signature scheme S consists three algorithms. A randomized *key generation* algorithm $\mathcal{K}$ takes a global information I and outputs a pair (sk, pk) of a secret and a public keys. A randomized *signature generation* algorithm $\mathcal{S}$ takes a message M to sign and global information I and a secret key sk and outputs M and a signature σ. A deterministic *verification* algorithm $\mathcal{V}$ takes a public key pk, and a message and a signature σ and output 1 (accepts) if the signature is valid and 0 (rejects) otherwise.

The widely-accepted notion of security for signature schemes is unforgeability under chosen-message attacks, the notion adjusted to the random oracle model is given in (Boneh *et al.*, 2001). Now we introduce the GDH signature scheme in (Boneh *et al.*, 2001). Let $\mathbb{G}_1$ be a GDH group. Let $[\{0,1\}^* \rightarrow \mathbb{G}_1^*]$ be a hash function family, each member of which maps arbitrary long strings to group $\mathbb{G}_1^*$ and H be a random member of this family. The global information I contain the generator P of $\mathbb{G}_1$, prime order q and a description of H. The algorithms $(\mathcal{K}, \mathcal{S}, \mathcal{V})$ of the GDH group signature scheme $GS[\mathbb{G}_1]$ are defined as follows.

- $\mathcal{K}(I)$: Parse I as (P, q, H). Pick random $x \leftarrow Z_q^*$ and compute $Y \leftarrow xP$. Return $(pk = (P, q, H, Y), sk = x)$.
- $\mathcal{S}(I, sk, M)$: Parse I as (P, q, H). Compute $\sigma = xH(M)$. Return (M, σ).
- $\mathcal{V}(M, pk, \sigma)$: Parse pk as (P, q, H, Y). If $\hat{e}\ (P, \sigma)\ = \hat{e}(Y, H(M))$, then return 1 else return 0.

In (Boneh *et al.*, 2001) the authors state and prove the following result.

Theorem 1. *Let $\mathbb{G}$ be a GDH group, Then $GS[\mathbb{G}]$ is a secure signature scheme in the random oracle model.*

3. Proposed Scheme

In this section we propose a partial delegation threshold proxy signature scheme with warrant m_ω which records the identities of the original signer and the proxy signers of the proxy group, parameters t and n, the valid delegation time, etc. It is also a proxy-protected threshold proxy signature scheme.

The system parameters are the same as those in the GDH signature scheme assuming that $\mathbb{G}_1$ and $\mathbb{G}_2$ are additive and multiplicative groups of the same prime order q, P is a generator of $\mathbb{G}_1$, $\hat{e}\colon \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ is a secure bilinear pairing(Boneh *et al.*, 2003), H_1: $\{0,1\}^* \times \mathbb{G}_1 \rightarrow \mathbb{Z}_q^*$, H_2: $\{0,1\}^* \rightarrow \mathbb{G}_1 \backslash \{1\}$ are two cryptographic hash functions and the original signer has a secret key $sk = x_o$ randomly chosen from Z_q^* and a public key $pk = Y = x_oP$ which is certified by CA (Certificate Authority). Let $p_1, p_2, p_3, \cdots, p_n$ be the n proxy signers. Each proxy signer has a secret key $sk = x_i$ randomly chosen from Z_q^* and a public key $pk = Y = x_iP$ which is certified by CA as well. Let $ASID$ (Actual Signers'ID) denotes the identities of the actual signers. Our scheme mainly consists of three protocols: Proxy share generation protocol $\mathcal{TPK}$, Generation of the proxy signature without revealing shares $\mathcal{TPS}$, and Proxy signature verification protocol $\mathcal{TPV}$.

3.1. *Proxy Share Generation Protocol $\mathcal{TPK}$*

Proxy share generation protocol makes use of Verifiable Secret Sharing (VSS) proposed by Pederson (Pedersen, 1991). To delegate the signing capability to proxy signers, the original signer Alice uses the Schnorr signature scheme to make the warrant m_ω signed (since the Schnorr signature scheme is known to be provably-secure (Pointcheval and Stern, 1996) in the random-oracle model). There is an explicit description of the delegation relation in the warrant m_ω. If the following process is finished successfully, each proxy signer will get his or her proxy share key.

Step 1. The original signer picks a random number $r \in Z_q^*$, computes $U = rP$. Let $h = H_1(m_\omega, U)$, computes $v = (r + hx)$. The signature of m_ω is $\sigma = (U, v)$, then Alice (the original signer) sends σ and m_ω to each proxy signer.

Step 2. Each proxy signer verifies the validity of the signature on m_ω by checking whether the following equation sounds or not.

$$vP = U + hY, \tag{1}$$

and accept σ if and only if the above equation sounds. If the signature σ is valid, proxy signer p_i picks up a random number k_i, broadcasts k_iP and computes $s_i \equiv n^{-1}v + x_i + k_i, mod\ q$ with his own secret key.

Step 3. Proxy signer p_i picks up randomly a polynomial $f_i(z)$ of degree $t-1$ in Z_q such that $f_i(0) = s_i = a_{i,0}$. That is

$$f_i(z) = s_i + a_{i,1}z + a_{i,2}z^2 + \cdot + a_{i,t-1}z^{t-1}, \tag{2}$$

then p_i computes and broadcasts $a_{i,j}P$ for $j = 1, 2, 3, \cdots, t-1$, doesnot need to broadcast $a_{i,0}P$ for $a_{i,0}P = n^{-1}vP + Y_i + k_iP$; sends $f_i(j)$ secretly to each proxy signer p_j for $j = 1, 2, 3, \cdots, n;\ j \neq i$.

Step 4. Proxy signer p_i after receiving $f_j(i)$ from p_j, $j = 1, 2, 3, \cdots, n;\ j \neq i$, verifies $f_j(i)$ by checking

$$f_j(i)P = \sum_{k=0}^{t-1} i^k \cdot a_{j,k}P. \tag{3}$$

If the check fails, p_i broadcasts a complaint against p_j. Assume that none of the proxy signers has a complaint. Then the proxy signer p_i computes the secret proxy share $x'_i = \sum_{k=1}^{n} f_k(i)$, and computes the public proxy share $Y'_i = x'_iP$.

In this protocol if we let $f(z) = \sum_{i=1}^{n} f_i(z)$, we will get the secret proxy share $x'_i = f(i)$ in fact. The public proxy share Y'_i must be $f(i)P$.

3.2. *Generation of the Proxy Signature $\mathcal{TPS}$*

Let m be a message to be signed. Without loss of generality, we assume that $p_1, p_2, p_3, \cdots, p_t$ are the t proxy signers who want to cooperate to sign a message m on behalf of the original signer Alice.

Setp 1. Each proxy signer p_i for ($i = 1, 2, \cdots, t$) uses his or her secret proxy share x'_i to sign the message m. Referring to the signature scheme in (Boneh *et al.*, 2001) each proxy signer p_i computes $\omega_i = \prod_{j \neq i}^{j \in \{1,2,\cdots,t\}} \frac{j}{j-i}$, gets the partial signature of the message $\sigma_i = (x'_i\omega_i + x_i)H_2(m)$.

Setp 2. The t proxy signers after gathering σ_i, verify σ_i by checking

$$\hat{e}(P, \sigma_i) = \hat{e}\big(\omega_i Y'_i + Y_i, H_2(m)\big). \tag{4}$$

If the above equation doesn't hold, They will know p_i does not send the correct partial signature or p_i is not honest one, we may ask another one or p_i to do Step 1 again.

Now we assume the equation holds, they can compute the proxy signature $\sigma' = \sum_{i=1}^{t} \sigma_i$ and $K = \sum_{m=1}^{n} k_mP$, record the t proxy singers' ID on $ASID$. So the complete valid proxy signature will be the tuple $< m, U, m_\omega, \sigma', K, ASID >$.

REMARK 1. In Step 2 we may designate one of the t proxy signers or a clerk who is assumed honest to check the correctness of the partial signature and generate the complete signature. If we use $\{i_1, i_2, \cdots, i_t\}$ to represent the t proxy signers' ID which is a subset of $\{1, 2, \cdots, n\}$, we may use $\sum_{k=1}^{t} 2^{i_k}$ represent $ASID$, So $ASID$ is only an n-bit-long string.

3.3. *Proxy Signature Verification Protocol* $\mathcal{TPV}$

Receiving the threshold proxy signature $< m, U, m_\omega, \sigma', K, ASID >$ of m, any verifier can confirm the validity of the proxy signature and identify the actual signers. The steps of the phase are stated as follows:

Setp 1. The verifier can identify the original signer and the proxy signers from m_ω and $ASID$, and get their public keys from the CA. Besides, he/she can also identify the actual proxy signers.

Setp 2. A recipient can verify the validity of the proxy signature by checking if the following equation holds or not.

$$\hat{e}(P, \sigma') = \hat{e}\Bigg(U + \big(H_1(m_\omega, U)\big)Y + K + \sum_{i=1}^{n} Y_i + \sum_{i=1}^{t} Y_i, H_2(m)\Bigg). \tag{5}$$

If it holds, the recipient accepts the signature, otherwise rejects.

3.4. *Correctness*

The verification of the signature is justified by the following equations:

$$\begin{aligned}
\hat{e}(P, \sigma') &= \hat{e}\Bigg(P, \sum_{i=1}^{t} \sigma_i\Bigg) \\
&= \hat{e}\Bigg(P, \sum_{i=1}^{t} (x_i' \omega_i + x_i) H_2(m)\Bigg) \\
&= \hat{e}\Bigg(P, \Bigg(f(0) + \sum_{i=1}^{t} x_i\Bigg) H_2(m)\Bigg) \\
&= \hat{e}\Bigg(P, \Bigg(\sum_{i=1}^{n} f_i(0) + \sum_{i=1}^{t} x_i\Bigg) H_2(m)\Bigg) \\
&= \hat{e}\Bigg(P, \Bigg(nn^{-1}v + \sum_{i=1}^{n} (x_i + k_i) + \sum_{i=1}^{t} x_i\Bigg) H_2(m)\Bigg) \\
&= \hat{e}\Bigg(P, \Bigg(r + H_1(m_\omega, U)x + \sum_{i=1}^{n} (x_i + k_i) + \sum_{i=1}^{t} x_i\Bigg) H_2(m)\Bigg) \\
&= \hat{e}\Bigg(\Bigg(r + H_1(m_\omega, U)x + \sum_{i=1}^{n} (x_i + k_i) + \sum_{i=1}^{t} x_i\Bigg) P, H_2(m)\Bigg) \\
&= \hat{e}\Bigg(U + \big(H_1(m_\omega, U)\big)Y + K + \sum_{i=1}^{n} Y_i + \sum_{i=1}^{t} Y_i, H_2(m)\Bigg).
\end{aligned} \tag{6}$$

So the correctness of verification protocol is proved.

4. Security Analysis of the Proposed Scheme

In the following section, we will prove that the proposed scheme can resist all kinds of known attack including the forgery attack, conspiracy attack, public key substitution attack etc.

Like the general proxy signature, our proposed signature scheme satisfies the requirements stated in abstract as well.

Distinguishability: This is obvious, because there is a warrant m_ω in a valid proxy signature, at the same time, this warrant m_ω and the public keys of the original signer and proxy signer must occur in the verification equation of proxy signature.

Verifiability: The valid proxy signature for message m will be the tuple $< m, U, m_\omega, \sigma', K, ASID >$. From the construction of $< U, \sigma', K >$ and the verification phase, the verifier can be convinced that the proxy signer has the original signer's signature on the warrant m_ω. In general the warrant m_ω contains the identity information and the limit of the delegated signing capacity etc, so our scheme satisfies the verifiability.

Strong non-forgeability: First, the third adversary who wants to forge the proxy signature of message m' for the proxy signers and original signer must have the original signer's signature σ on the warrant m_ω, but cannot forge this since Schnorr signature scheme is secure. And we can see even third adversary knows the signature σ sent by the original signer he cannot make a forgery signature on any other message m'_ω, so he cannot make a forgery proxy signature on m' either.

Second, the original signer cannot create a valid proxy signature, since the proxy signature is obtained by the proxy signers using the GDH signature scheme (a secure signature scheme) and the proxy signers' secret proxy shares $\{x'_i\}$ which contain the private key $\{x_i\}$ of each proxy signer. And also the original signer doesn't know $\sum_{i=1}^{n}(x_i + k_i)$ and $\sum_{i=1}^{t} x_i$, so the original signer cann't forge a valid proxy signature. Now we can see the proposed scheme is a *proxy protected* one.

Third, proxy signer can't forge valid proxy signatures. From the proxy signature $< m, U, m_\omega, \sigma', K, ASID >$, any proxy signer can't obtain the private keys of other proxy signers. He/she can't get $k_1, k_1, \cdots, k_n$ randomly chosen by the proxy signers and $\sum_{i=1}^{t}(x'_i + x_i)$ either because of difficult Discrete Logarithm problems. Therefore, proxy signatures can't be forged by any proxy signer.

Fourth, the designated proxy signer or clerk in *Step 2* of protocol $\mathcal{TPS}$ (Generation of the proxy signature) can't forge the proxy signatures either. From the partial signature σ_i the clerk can't get the knowledge $x'_i\omega_i + x_i$ of because of difficult Discrete Logarithm problems. Of course, the clerk is unable to obtain the knowledge of x'_i or x_i either. From the equation $\sigma' = \sum_{i=1}^{t} \sigma_i$, the clerk can't get $\sum_{i=1}^{t}(x'_i\omega_i + x_i)$ either because of the same reason. Therefore, the proxy signature can't be forged by the designated one.

Identifiability: The valid signature contains the warrant m_ω, so any one can determine the identities of the corresponding proxy signers from the warrant. As the verifier also receives $ASID$ from the valid proxy signature $< m, U, m_\omega, \sigma', K, ASID >$ which records the identity of the actual t proxy signers who cooperate in generating the proxy signature. So the proposed threshold proxy signature is identifiable.

Strong nonrepudiation: As the identifiability, the valid signature contain the warrant m_ω and $ASID$, which must be verified in the process of verification, it cannot be modified by the proxy signers. Thus once proxy signers creates a valid proxy signature for the original signer, he cannot repudiate the signature creation. In the verification phase the verifier also takes in the public keys of the proxy signers including actual signer's identity ($ASID$) and original signer, so the signers cannot repudiate the signature creation either.

Prevention of misuse: In our proposed proxy signature scheme, using the warrant m_ω, We had determined the limit of delegated signing capacity in the warrant m_ω. we can conclude that any one who even knows the signature v on m_ω can't sign any other message on behalf of the original signer since Schnorr signature scheme is secure. So our proposed signature yields the property of prevention of misuse.

Next we will show that even when $t-1$ proxy signers are corrupted (who have the warrant m_ω), the proposed threshold proxy signature will still be secure. So we can conclude our scheme is a threshold proxy signature scheme.

Theorem 2. *Even there exists an adversary who can corrupt $t-1$ proxy signers among n proxy signers, The $\mathcal{TPK}$ and $\mathcal{TPS}$ protocols still complete successfully.*

Proof. In the $\mathcal{TPK}$ protocol we use the technique of VVS, when each proxy signer receives v he must use his private key to generate a polynomial $f_i(z)$ of degree $t-1$ in Z_q such that $f_i(0) = s_i = a_{i,0} = n^{-1}v + x_i + k_i$ in Step 3 of the $\mathcal{TPK}$ protocol. And in Step 4 each proxy signer p_i will check each $f_i(j)$, So the $t-1$ proxy signers cannot do anything to cheat or forge.

In the $\mathcal{TPS}$ protocol every partial signature σ_i is verified by the corresponding public proxy share Y'_i in the equation (4) of Step 2. Even at most $t-1$ signers can be corrupted, the adversary still needs to get one partial signature from the other signers (which the adversary can't forge) to form t valid signature shares. Only with t valid signature shares, the adversary can produce a valid signature.

What we want to point out next is that our threshold proxy signature can avoid **conspiracy attack** in (Hsu *et al.*, 2001) which says t malicious proxy signers can impersonate some other proxy signers to generate valid proxy signatures even may know the secret keys of the other signers for misuse. In our scheme if t malicious proxy signers want to impersonate some other t proxy signers to generate valid proxy signatures, they must use $\{x'_i\}$, $\{x_i\}$ $(i = 1, 2, \cdots, n)$ or $\sum_{i=1}^{t}(x'_i\omega_i + x_i)$, each proxy signer randomly chooses

k_i and publishes k_iP that makes it impossible to know other proxy signers' secret keys by facing difficult Discrete Logarithm problem.

At last, the scheme can resist the **public key substitution attack** from the original signer or any proxy signer. In the scheme, CA (Certificate Authority) is need. If the original signer or any proxy signer wants to substitute a new public key for the original public key, he/she must know the corresponding private key. In the public key substitution attack, generally speaking, the attacker doesn't know the corresponding private key. Thus, the attacker can't change its public key in the system public directory which is managed by CA. So the public key substitution attack doesn't work, either.

Through the analysis of security what we want to point out is that our proxy signature scheme does not need secure channel for delivery of the signed warrant since Schnorr signature scheme is secure. More precisely, the original signer can send the signature v on the warrant m_ω to the proxy signers through a public channel.

5. Performance Evaluation and Numerical Computation Sample

In this section, we compare our scheme with Sun's scheme and Hsu *et al.*'s scheme in terms of computation time, then we provide a simple numerical computation example of our proposed scheme.

5.1. *Performance Evaluation*

We denote the following notations to facilitate the performance evaluation:

m: The time of performing a modular multiplication computation.

in: The time of performing a modular inverse computation.

exp: The time of performing a exponentiation computation.

h: The time of performing a cryptographic hash function mapping strings to a modular group (such as hash functions in Sun's scheme and Hsu *et al.*'s scheme or H_1 in our scheme).

Add: The time of performing a point addition computation.

Sca: The time of performing a scalar multiplication computation.

NA: Not available.

P: The time of performing a pair computation.

H: The time of performing a cryptographic hash function mapping strings to a GDH group (such as H_2 in our scheme).

Table 1 shows us the time cost in computations of Sun's scheme, Hsu *et al.*'s scheme and our scheme which shows our threshold proxy signature is of great efficiency. The Table 1 excludes the computation cost on validating $f_i(v_j)$ in Sun's scheme and Hsu *et al.*'s scheme or corresponding $f_i(j)$ in our scheme. We also assume $Y_G = \prod_{i=1}^{n} y_i,\ (mod\, p)$ in Sun's, Hsu *et al.*'s scheme and corresponding $\sum_{i=1}^{n} Y_i$ in our scheme are precomputed before verification phase.

Since our scheme works on an elliptic curve or a supersinglar elliptic curve, we may have small size keys as a advantage. Moreover the signature size is also small by using

Table 1

Comparisons of efficiency

	Sun's scheme	Hsu *et al.*'s scheme	Our scheme
Secret share generation	$(n^2-1)(t-1)\,\mathbf{m}$ $+\,n(t-1)\,\mathbf{exp}$	$(n+1)\,\mathbf{exp}+n(t-1)\mathbf{m}$	$1\,\mathbf{Sca}+1\,\mathbf{h}+1\,\mathbf{m}$ (*Step* 1 in $\mathcal{TPK}$)
Proxy share generation	The original signer: $t\,\mathbf{exp}+1\,\mathbf{h}$ $+\,(nt-n+1)\,\mathbf{m}$ Each proxy signer: $(t+1)\,\mathbf{exp}+1\,\mathbf{h}$ $+\,(2t-1)\,\mathbf{m}$	The original signer: $t\,\mathbf{exp}+1\,\mathbf{h}$ $+\,(nt-n+1)\,\mathbf{m}$ Each proxy signer: $(t+1)\,\mathbf{exp}+1\,\mathbf{h}$ $+\,(2t-1)\,\mathbf{m}$	The original signer: **NA** Each proxy signer: $(t-1)\,\mathbf{exp}+1\,\mathbf{in}$ $+\,1\,\mathbf{Add}+1\,\mathbf{m}$ $+\,2(t+2)\,\mathbf{m}$ (*Step* 2, 3, 4 in $\mathcal{TPK}$)
Partial proxy signature	$(4t^2-7t+5)\,\mathbf{m}$ $+\,1\mathbf{h}+t^2\mathbf{exp}$	$1\,\mathbf{exp}+(3t-1)\,\mathbf{m}$ $+\,1\mathbf{h}+(t-1)\mathbf{in}$	$1\mathbf{m}{+}1\mathbf{Sca}{+}1\mathbf{H}{+}2\mathbf{P}$
Whole proxy signature	$(3t^2-t-2)\,\mathbf{exp}$ $+\,(6t^2-7t+1)\,\mathbf{m}$ $+\,(t^2-t)\,\mathbf{in}+2\mathbf{h}$	$(t^2+4t)\,\mathbf{exp}+2\,\mathbf{h}$ $+\,(4t^2-t-1)\,\mathbf{m}$ $+\,(t^2-t)\,\mathbf{in}$	$(n+2t-1)\,\mathbf{Add}$ $+\,t\,\mathbf{Sca}+2t\,\mathbf{P}$ (*Step* 1, 2 in $\mathcal{TPS}$)
Proxy signature verification	$4\,\mathbf{exp}+2\,\mathbf{h}$ $+\,(t+3)\,\mathbf{m}$	$4\,\mathbf{exp}+2\,\mathbf{h}+(t+3)\,\mathbf{m}$	$1\mathbf{h}+1\mathbf{Sca}+1\mathbf{H}$ $+\,2\mathbf{P}+(t+3)\mathbf{Add}$ (*Step* 1, 2 in $\mathcal{TPV}$)

our threshold proxy signature scheme, for they are both points on an elliptic curve or a supersinglar elliptic curve.

5.2. *Main Numerial Computation of the Scheme*

In our scheme the main idea is to use bilinear pairings to construct new threshold proxy signature schemes. We use Schnorr signature and GDH signature schemes including VSS in the proposed scheme, most of them have standard algorithms which we won't elaborate. The only possible obstacle is how to compute bilinear pairings. Now we will take Tate pairing as an example.

We review definition of Tate pairing first (Galbraith *et al.*, 2002). Let E be an elliptic curve over a finite field F_q. We write $\mathcal{O}_E$ for the point at infinity on E. Let m be a positive integer which is coprime to q. In most applications m is a prime and $m|\sharp E(F_q)$. Let k be a positive integer such that $m|(q^k-1)$. Let $G = E(F_{q^k})$ and write $G[m]$ for the subgroup of points of order m and G/mG for the quotient group (which is also a group of exponent m). Then the Tate pairing is a mapping

$$<\cdot,\cdot>\colon G[m]\times G/mG \to F^*_{q^k}/(F^*_{q^k})^m. \tag{7}$$

The quotient group on the right hand side of (7) can be thought of as the set of equivalence classes of $F^*_{q^k}$ under the equivalence relation $a \equiv b$ if and only if there exists

$c \in F_{q^k}^*$ such that $a = bc^m$. Given the point P compute a function g such that the divisor of g is equal to $l((P) - (\mathcal{O}))$. Then compute a divisor D which is equivalent to $(Q) - (\mathcal{O})$ such that D is disjoint from the support of g. Then the value of the Tate pairing (up to lth powers) is $< P, Q >= g(D)$ where $g(D) = \prod_i g(P_i)^{n_i}$, if $D = \sum_i n_i P_i$.

For each pair of points U, V on the elliptic curve $E(F_{q^k})$, let $g_{U,V}$ be the rational function given by the line $g_{U,V}$: $l_1 y + l_2 x + l_3 = 0$ through U and V. Naturally, if $U = V$, then $g_{U,V}$ is the given by the equation of the tangent line at U, and if either U or V is the point at infinity $\mathcal{O}$, then $g_{U,V}$ represents the vertical line through the other point. Furthermore, for brevity, we write g_U instead of $g_{U,-U}$. We introduce Miller's algorithm for the Tate pairing to compute $< P, Q >$:

- Choose a random point $Q' \in E(F_{q^k})$ and compute $S = Q' + Q \in E(F_{q^k})$.
- Set $t = \lfloor \log_2(m) \rfloor$, and let $(m_t, \cdots, m_0)_2$ be the binary representation of m. Set $f = 1$ and $V = P$.
- For $i = t - 1$ to 0 do
  1. Set $f = f^2(g_{V,V}(S) g_{[2]V}(Q'))/((g_{V,V}(Q') g_{[2]V}(S)))$ and $V = [2]V$.
  2. If $m_i = 1$ then set $f = f(g_{V,P}(S) g_{V+P}(Q'))/(g_{V,P}(Q') g_{V+P}(S))$ and $V = V + P$.
- Return f.

5.2.1. *Example*

We consider elliptic curve E/F_{11}: $y^2 = x^3 + 3x$. If $m = 6$, k can be 2, for $6|(11^2 - 1)$ but 6 can not divide $(11 - 1)$. To compute the Tate pairing $< P, Q >$ for $P = (1, 9)$, $Q = (10, 9i)$ and $m = 6$, we carry out Miller's algorithm.

- We choose $Q' \in E(F_{11^2})$ to be $Q' = (6, 6)$. Then $Q + Q' = (8 + 7i, 10 + 6i)$.
- The binary representation of $m = 6$ is given by $(m_2, m_1, m_0)_2 = (1, 1, 0)_2$, so $t = \lfloor \log_2(6) \rfloor = 2$. Further, we set $f = 1$ and $V = P = (1, 9)$.
- For $i = 1$:
  1. We compute $g_{V,V}$ and $g_{[2]V}$:

     $$g_{V,V} = y + 7x + 6, \quad g_{[2]V} = x + 8.$$

     Then

     $$g_{V,V}(S) = 6, \quad g_{[2]V}(Q') = 3, \quad g_{[2]V}(S) = 5 + 7i, \quad g_{V,V}(Q') = 10,$$

     and thus we set

     $$f = 1^2 \frac{6 \cdot 3}{(5 + 7i) \cdot 10} = 8 + i, \quad V = [2](1, 9) = (3, 5).$$

  2. Since $m_1 = 1$, we compute $g_{V,P}$ and g_{V+P}:

     $$g_{V,P} = y + 2x, \quad g_{V+P} = x.$$

Then

$$g_{V,P}(S) = 4 + 9i, \quad g_{V+P}(Q') = 6,$$
$$g_{V,P}(Q') = 7, \quad g_{V+P}(S) = 8 + 7i,$$

and thus we set

$$f = (8 + 2i)\frac{(4 + 9i) \cdot 6}{(8 + 7i) \cdot 7} = 5 + 4i, \quad V = (3,5) + (1,9) = (0,0).$$

- For $i = 0$:
  1. We compute $g_{V,V}$ and $g_{[2]V}$:

     $$g_{V,V} = x, \quad g_{[2]V} = 1.$$

     Then

     $$g_{V,V}(S) = 8 + 7i, \quad g_{[2]V}(Q') = 1,$$
     $$g_{[2]V}(S) = 1, \quad g_{V,V}(Q') = 6,$$

     and thus we set

     $$f = (5 + 4i)^2 \frac{8 + 7i}{6} = 2 + 7i, \quad V = [2](0,0) = (0,0) = \mathcal{O}.$$

  2. Since $m_0 = 0$, end.
- $i = 0$, so the program terminates and returns $f = 2 + 7i$.

All the other implementation of the proposed scheme's application is easy, since they all have standard algorithms.

6. Conclusions

We have used Schnorr signature scheme on an elliptic curve or a supersinglar elliptic curve to create an efficient pairing-based threshold proxy signature Schemes with known signers. The threshold proxy signature scheme is based on secure bilinear pairings which may be the first one using bilinear pairings to create threshold proxy signature with known signers, since bilinear pairings have been found having many good properties in cryptography. In security analysis some theorems have been proved to show the scheme's security, the requirements which the proxy signature satisfies with has been checked also, almost all kinds of attacks are shown to be useless in our scheme. Finally we compare the performance of our scheme with other threshold proxy signature scheme, which shows our scheme is also of great efficiency with small size keys and signature.

Acknowledgements

The author would like to thank anonymous referees for their suggestions to improve the paper. Besides, this article is supported by the National Natural Science Foundation of China under Grant No. 60072018, the National Science Fund 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

Boldyreva, A. (2003). Threshold signature, multisignature and blind signature schemes based on the Gap–Diffe–Hellman-group signature scheme. In *Public Key Cryptograpy – PKC 2003*, LNCS 2567. Springer-Verlag, Berlin. pp. 31–46.

Boneh, D., B. Lynn, H. Shacham (2001). Short signatures from the weil pairing. In *Advances in Cryptology-Asiacrypt'01*, LNCS 2248. Springer-Verlag, Berlin. pp. 514–532.

Boneh, D., M. Franklin (2001). ID-based encryption from the weil-pairing. In *Advance in Cryptology-CRYPTO'2001*, LNCS 2139. Springer-Verlag, Berlin. pp. 213–229.

Boneh, D., I. Mironov, V. Shoup (2003). A secure signature scheme from bilinear maps. In *Proceedings of RSA-CT'03*, LNCS 2612. Springer-Verlag, Berlin. pp. 98–110.

Galbraith, S.D., K. Harrison, D. Soldera (2002). Implementing the tate pairing. In *Algorithmic Number Theory 5th International Symposium*, ANTS-V, LNCS 2369. Springer-Verlag, Berlin. pp. 324–337.

Hsu, C.L., T.S. Wu, T.C. Wu (2001). New nonrepudiable threshold proxy signature scheme with known signers. *Journal of Systems and Software*, **58**, 119–124.

Hwang, M.S., I.C. Lin, J.L. Lu Eric (2000). A secure nonrepudiable threshold proxy signature scheme with known signers. *Informatica*, **11**(2), 137–144.

Hwang, S.J., C.C. Chen (2003). Cryptanalysis of nonrepudiable threshold proxy signature scheme with known signers. *Informatica*, **14**(2), 205–212.

Joux, A., K. Nguyen (2001). *Separating Decision Diffe–Hellman from Diffe–Hellman in Cryptographic Groups*. Cryptology ePrint Archive – 2001/03.

Li, J.G., Z.F. Cao (2002). Improvement of a threshold proxy signature scheme. *Journal of Computer Research and Development*, **39**(11), 515–518 (in Chinese).

Li, J.G., Z.F. Cao, Y.C. Zhang (2002). Improvement of M-U-O and K-P-W proxy signature schemes. *Journal of Harbin Institute of Technology* (New Series), **9**(2), 145–148.

Li, J.G., Z.F. Cao, Y.C. Zhang (2003). Nonrepudiable proxy multi-signature scheme. *Journal of Computer Science and Technology*, **18**(3), 399–402.

Li, J.G., Z.F. Cao, Y.C. Zhang, J.Z. Li (2003). Cryptographic analysis and modification of proxy multi-signature scheme. *High Technology Letters*, **13**(4), 1–5 (in Chinese).

Mambo, M., K. Usuda, E. Okamoto (1996a). Proxy signature: delegation of the power to sign messages. In *IEICE Trans Fundam*, E79-A(9). pp. 1338–1354.

Mambo, M., K. Usuda, E. Okamoto (1996b). Proxy signature for delegating signing operation. In: *Proc 3rd ACM Conference on Computer and Communications Security*, New Dehli, India. ACM press, New York. pp. 48–57.

Neuman, B.C. (1993). Proxy-based authorization and accounting for distrubuted computing systems. In *Proc 13th International Conference on Distrubuted Systems*. IEEE Computer Society Press, Pennsylvania. pp. 283–291.

Pedersen, T.P. (1991). Non-interactive and information theoretic secure verifiable secret sharing. *Advance in Cryptology-ASIACRYPTO'91*, LNCS 576. Springer-Verlag, Berlin. pp. 129–140.

Pointcheval, D., J. Stern (1996). Security proofs for signature schemes. In *In Eurocrypt'96*, LNCS 1070. Springer-Verlag, Berlin. pp. 387–398.

Sun, H.M. (1999). An efficient nonrepudiable threshold proxy signature scheme with known signers. *Computer Communications*, **22**(8), 717–722.

Sun, H.M., N.Y. Lee, T. Hwang (1999). Threshold proxy signatures. *IEE Proc-Computers & Digital Techniques*, **146**(5), 259–263.

Zhang, F., K. Kim (1997). Threshold proxy signature schemes. In *Information Security Workshop*, Japan, LNCS 1396. Springer-Verlag, Berlin. pp. 191–197.

Zhang, F., K. Kim (2003). Efficient id-based blind signature and proxy signature from bilinear parings. In *Proc of ACISP'03*, July 9–11, Wollongong, Australia, LNCS 2727. Springer-Verlag, Berlin. pp. 312–323.

H.F. Qian was awarded BS degree and a master degree (on algebraic geometry) in Mathematic Department from East China Normal University, China, in 2000 and 2003, respectively, and now is a doctoral candidate in the Department of Computer Science and Engineering, Shanghai Jiao Tong University. His main research interests include network security, cryptography and algebraic geometry.

Z.F. Cao received a BS degree in computer science and PhD degree in mathematics from Harbin Institute of Technology, China, in 1983 and 1999, respectively. He became the youngest associate professor and professor in China, in 1987 and 1991, respectively. Since 2002, he has been the professor and the doctoral supervisor of Shanghai Jiao Tong University. Dr. Cao is the member of many academic organizations such as Expert Group of the National Information Security Technology and director of National Association for Cryptologic Research (China). And he is a reviewer of Mathematical Reviews (USA) and Zentrallbatt MATH (Germany). His main research areas are number theory, modern cryptography, theory and technology of information security etc. He is the gainer of the first prize of Award for Science and Technology in Chinese University and the National Outstanding Youth Fund of China etc.

Q.S. Xue received a BS degree in computer science and technology in Shandong Normal University and a Master degree in computer application from Shandong University, China in 1995 and 2000, respectively, and is now a doctoral candidate in the Department of Computer Science and Engineering, Shanghai Jiao Tong University. His research interests include network security and cryptography.

Efektyvi poravimu grįsta slenkstinio įgaliotojo parašo schema su žinomais pasirašiusiaisiais

Haifeng QIAN, Zhenfu CAO, Qingshui XUE

Nuo tada, kai slenkstinio įgaliotojo parašas buvo pasiūlytas, visos slenkstinio įgaliotojo parašo schemos yra grindžiamos diskretaus logaritmo uždaviniais modulinėje multiplikacinėje grupėje, kuri turi g elementą su dideliu pirminiu skaičiumi. Dabar šio tipo parašo schemos tampa vis sudėtingesnės ir, kad užtikrinus saugumą ir specifinius reikalavimus, sunaudojama daug laiko.

Šiame straipsnyje siūlomas poravimu grįstą slenkstinio įgaliotojo parašo schemą su žinomais pasirašiusiaisiais, analizuojamos ir tikrinamos šitokios jos savybės: neišsižadėjimo, padirbinėjimo, tapatumo nustatymo, atskiriamumo, patikrinamumo, apsisaugojimo nuo klaidingo parašo teisių naudojimo ir kt. Be to, mes parodome, kad pasiūlytoji schema yra efektyvesnė nei ta, kurią pasiūlė Sun, Hsu ir kiti.