Threshold Signature Scheme without Using Polynomial Interpolation

Lein Harn[1] and Feng Wang[2]
(Corresponding author: Lein Harn)
Department of Computer Science Electrical Engineering, University of Missouri-Kansas City[1]
5110 Rockhill Road, Kansas City, MO 64110, USA
College of Mathematics and Physics, Fujian University of Technology[2]
Fuzhou, Fujian, 350118, China
(Email: harnl@umkc.edu)
(Received Sept. 16, 2014; revised and accepted Jan. 16 & Feb. 8, 2015)

Abstract

In a (t, n) secret sharing scheme (SS), the secret is shared among n shareholders in such a way that (a) with t or more than t shares can recover the secret, and (b) with fewer than t shares cannot obtain the secret. The threshold signature scheme is an application that extends the SS to a digital signature scheme. In a threshold signature scheme, any t or more than t group members can represent the group to generate a group signature; but fewer than t group members cannot generate a group signature. So far, most threshold signature schemes are based on the linear polynomial. In other words, these threshold signature schemes need to overcome the problem of polynomial interpolation. In this paper, we propose a threshold signature scheme based on the Chinese Remainder Theorem (CRT). We describe how to set up the system by a trusted group manager initially and generate pairs of public and private keys for group members. Since our proposed scheme is based on the CRT, there is no polynomial interpolation. The security of our proposed threshold signature scheme is based on the difficulty of solving the discrete logarithm problem.

Keywords: Chinese remainder theorem, polynomial interpolation, multisignature, threshold signature

1 Introduction

Secret Sharing Schemes (SSs) were originally introduced by both Blakley [3] and Shamir [27] independently in 1979 as a solution for safeguarding cryptographic keys and have been studied extensively in the literature. SS has become one of the most basic tools in cryptographic research. In Shamir's SS, a secret s is divided into n shares by a dealer and shares are sent to shareholders secretly. The secret s is shared among n shareholders in such a way that (a) with t or more than t shares can recover the secret, and (b) with fewer than t shares cannot obtain the secret. Shamir's (t, n) SS is based on a linear polynomial and is unconditionally secure. There are other types of threshold SSs, for example, Blakley's scheme [3] is based on the geometry, Mignotte's scheme [22] and Asmuth-Bloom's scheme [1] are based on the Chinese remainder theorem (CRT).

In an SS, the shares can be used for reconstructing the secret for only one time. This is because, in the secret reconstruction, the secret and shares are known to all participated shareholders. Therefore, the efficiency of the SS is very low. However, in a digital signature algorithm, the secret is the private key used for generating a digital signature. Since most digital signature schemes are based on some computation assumptions, the private key can be reused for generating multiple signatures. If the SS is extended to protect the private key of a digital signature scheme, the efficiency of the SS can be improved since the private key of a digital signature is protected based on some computational assumptions.

The threshold cryptography was first introduced by Desmedt in 1987 [5]. Desmedt and Frankel [6] have also proposed the first non-robust threshold signature scheme based on the ElGamal's signature [8]. In a threshold signature scheme, a group manager (GM) is responsible for selecting a pair of private and public keys for the group. The GM divides the group private key into multiple shares (i.e., private keys of members) and gives each share to each member secretly. Later, any t or more than t members can work together to generate a group signature; but, fewer than t members cannot generate a group signature. It is a natural generalization to use the SS in the design of a threshold signature scheme. So far, most threshold signature schemes are based on the linear polynomial. In Shamir's SS based on the linear polynomial, the polynomial interpolation needs to be performed in a field Z_p where p is a prime. Harn [13] proposed a robust threshold signature scheme based on a variation of ElGamal signa-

ture scheme. In Harn's scheme, a special modulus p is selected by the GM where $p-1$ contains a small prime factor q (i.e., $q|p-1$). A generator g with order q is used to compute all modular exponentiations. Under this arrangement, the polynomial interpolation of Shamir's SS can be performed in Z_q. Gennaro et al. [10] have proposed a robust threshold DSS [23] signature.

Desmedt and Frankel [6] have mentioned the difficulty of designing threshold signature schemes based on the RSA signature scheme [25]. The problem is caused by the fact that the polynomial interpolation is over the ring $Z_{\phi(n)}$ where n is the RSA modulus and $\phi(n)$ is the Euler toting function and is not a prime. Desmedt and Frankel [7] have proposed a non-robust threshold RSA signature. Later, De Santis et al. [26] proposed a variation of the Desmedt and Frankel's scheme [7]; but trades interaction for large share size. Both schemes [7, 26] avoid the problem of polynomial interpolation over $Z_{\phi(n)}$ by working instead with over $Z_{\phi(n)}\lfloor x \rfloor/\Phi_q(X)$ where $\Phi_q(X)$ is the qth cyclotomic polynomial and q is a prime. But, this design leads to a much more complicate scheme. Gennaro et al. [11] and Shoup [29] have proposed techniques to make threshold RSA signature scheme robust. There are other types of threshold signature schemes, including Elliptic curve-based [28] and Pairing-based [9] threshold signature schemes in the literature. Readers can refer to [14] for more information on the development of threshold signature schemes.

In this paper, we propose an approach to avoid the problem of polynomial interpolation. We adopt the SS based on the CRT in the design of a threshold signature. Most research papers in the subject of the SS are based on the linear polynomial; but only a few papers are based on the CRT. Polynomial and CRT are two different mathematical tools which can be used to implement a SS scheme. Both tools share many interesting properties. For example, the secret sharing homomorphism proposed by Benaloh [2] implies that the additive sum of shares generated by polynomials/CRTs is a share of additive sum of polynomials/CRTs. On the other hand, both tools are different in many aspects. For example, there is no polynomial interpolation in using CRT. Kaya and Selcuk [16] proposed the first CRT-based threshold decryptions. Later, they proposed a CRT-based threshold DSS signature [17]. But, their scheme needs $2t$ shares to generate a valid threshold signature and the signature generation is very complicate. In 2012, Guo and Chang [12] proposed a weighted threshold signature based on based on the work of Iftene [15], Kaya and Selcuk [16], and generalized Chinese remainder theorem [20]. Their scheme utilizes the cryptographic techniques of extended Asmuth-Bloom sequences [1] based on GCRT and the RSA threshold signature scheme [18]. However, Guo and Chang's scheme is not provable security because that RSA signature is not provable security [21]. In our proposed CRT-based threshold signature, it needs only t or more than t users to jointly generate the signature. Our scheme utilizes the cryptographic techniques of Mignotte's (t, n) threshold SS [22] and Harn's multisignature signature scheme [13]. The signature generation is almost the same as the polynomial-based threshold signatures. We describe how to set up the system by a trusted GM initially and generate pairs of public and private keys for group members. Since our proposed scheme is based on the CRT, there is no polynomial interpolation. The security of our proposed threshold signature scheme is based on the difficulty of solving the discrete logarithm problem.

The rest of this paper is organized as follows. In the next section, we introduce some preliminaries that include CRT, Mignotte's (t, n) threshold SS and a modified signature scheme and multisignature scheme used in our design. In Section 3, we introduce the model of our proposed scheme including entities, informal model and security properties. We propose a novel threshold signature scheme based on the CRT in Section 4. Security analysis and comparisons are included in Section 5. We conclude in Section 6.

2 Preliminaries

In this section, we provide fundamental background used in our design, including the CRT, the Mignotte's threshold SS [22] and Harn's multisignature scheme [13].

2.1 Chinese Remainder Theorem [4]

Given following system of equations as

$$\begin{aligned} x &= s_1 \bmod p_1; \\ x &= s_2 \bmod p_2; \\ &\vdots \\ x &= s_t \bmod p_t, \end{aligned}$$

there is one unique solution as $x = \sum_{i=1}^{t}(N/p_i)\cdot y_i \cdot s_i \bmod N$ where $(N/p_i)\cdot y_i \bmod p_i = 1$, and $N = p_1 \cdot p_2 \cdot \ldots \cdot p_t$, if all moduli are pairwise coprime (i.e., $\gcd(p_i, p_j) = 1$, for every $i \neq j$.)

2.2 Review of Mignotte's Threshold SS

We review Mignotte's threshold secret sharing scheme [22] as follows.

Share generation: A sequence of pairwise coprime positive integers, $p_1 < p_2 < \ldots < p_n$, where p_i is the public information associated with each shareholder, U_i. These public integers need to satisfy that $p_{n-t+2} \cdot p_{n-t+3} \cdot \ldots \cdot p_n < p_1 \cdot p_2 \cdot \ldots \cdot p_t$.

For this given sequence, the dealer chooses the secret s in the range,$R_t = \{s \in Z | p_{n-t+2} \cdot p_{n-t+3} \cdot \ldots \cdot p_n < s < p_1 \cdot p_2 \cdot \ldots \cdot p_t\}$. We call this range, the **t-threshold range**. Share for the shareholder, U_i, is generated as $s_i = s \bmod p_i$, $i = 1, 2, \ldots, n$. s_i is sent to shareholder, U_i secretly.

Remark 1. *The numbers in the t-threshold range, R_t, are integers upper bounded by $p_1 \cdot p_2 \cdot \ldots \cdot p_t$, which is the smallest product of any t moduli, and lower bounded*

by $p_{n-t+2} \cdot p_{n-t+3} \cdot \ldots \cdot p_n$, which is the largest product of any $t-1$ moduli. The secret, s, selected in this range can ensure that (a) the secret can be recovered with any t or more than t shares (i.e., the product of their moduli must be either equal to or larger than $p_1 \cdot p_2 \cdot \ldots \cdot p_t$), and (b) the secret cannot be obtained with fewer than t shares (i.e., the product of their moduli must be either equal to or smaller than $p_{n-t+2} \cdot p_{n-t+3} \cdot \ldots \cdot p_n$). Thus, the t-threshold range determines the threshold of a (t, n) threshold SS.

Secret reconstruction: Given t distinct shares, for example, $\{s_{i_1}, s_{i_2}, \cdots, s_{i_t}\}$, the secret s can be reconstructed by solving the following system of equations as

$$\begin{aligned} x &= s_{i_1} \bmod p_{i_1}; \\ x &= s_{i_2} \bmod p_{i_2}; \\ &\vdots \\ x &= s_{i_t} \bmod p_{i_t}. \end{aligned}$$

Using the standard CRT, a unique solution x is given as $x = \sum_{r=1}^{t} (N/p_{i_r}) \cdot y_{i_r} \cdot s_{i_r} \bmod N$, where $N = p_{i_1} \cdot p_{i_2} \cdot \ldots \cdot p_{i_t}$ and $(N/p_{i_r}) \cdot y_{i_r} \bmod p_{i_r} = 1$.

2.3 Review of Harn's Multisignature Signature Scheme

An efficient digital multisignature signature is proposed in [13]. This multisignature signature allows multiple signers to work together to generate a digital multisignature corresponding to a message. The length of multisignature signature is equivalent to the length of each individual signature.

In this section, we first introduce the modified ElGamal signature scheme used to construct the multisignature signature. We present a formal security proof of this modified scheme. The original ElGamal signature scheme [8] was proposed in 1985; but the security was never proved equivalent to the discrete logarithm problem. In 1996, Pointcheval and Stern [24] used the Forking lemma to prove the security of a slight variation of the original ElGamal signature scheme.

The modified ElGamal signature scheme used to construct a multisignature consists of 3 steps as follows:

- Let p be a large prime and g be a generator of Z_p, then the public key is $y = g^x \bmod p$ and the private key is x;
- The signer picks $k \in Z_{p-1}$ randomly and a cryptographic hash function h, the signature of message m is (r, s) where $r = g^k \bmod p$ and $s = x \cdot h(m, r) - k \cdot r \bmod p - 1$;
- The verification of the signature checks the equation $y^{h(m,r)} = r^r \cdot g^s \bmod p$.

We assume that hash function h behaves like a random oracle, and hence we follow the established cryptographic techniques, i.e., the Oracle Replay Attack and the Forking Lemma as proposed in [24], to prove the security of modified ElGamal signature scheme.

Theorem 1. *The modified ElGamal signature scheme is secure under the random oracle model against known-message attack and against adaptively chosen message attack.*

Proof. For a formal security proof, the hash function $h = h(m, r)$ in modified signature scheme will be treated as a random oracle. We use the method of reductionist proof to prove this Theorem. Suppose that there is an adversary A who can break this scheme, then we can construct an algorithm B that can solve the discrete logarithm problem with non-negligible probability in probabilistic polynomial time. It is to say that given (p, g, y) we can design an algorithm B to output x which satisfies $y = g^x \bmod p$. The algorithm B is described as follows.

Algorithm B sends (p, g, y) to an adversary A, and A requests some queries.

h-query: B maintains a list $L_1 = (m, r, h)$ and initializes it to empty. If A provides a pair (m, r) for h-query, B checks if (m, r) has it in the list L_1. If it is, returns the corresponding h; otherwise, B picks a random $h \in Z_{p-1}$ as a returned value, and adds (m, r, h) into list L_1.

Signature query: B maintains a list $L_2 = (m, r, h, s)$ and initializes it to empty. If A provides a message m for Signature query, B checks if m is in the list L_2. If it is, returns the corresponding (r, s) as m's signature; otherwise, B picks random integers $u, v \in Z_{p-1}$, and computes $r = g^u \cdot y^v \bmod p$, $h = v \cdot g^u \cdot y^v \bmod p - 1$, and $s = -u \cdot g^u \cdot y^v \bmod p - 1$, and checks whether (m, r) is in the list L_1. If it is, returns failure; else, returns the signature (r, s) and adds (m, r, h, s) into the list L_2, adds (m, r, h) into the list L_1. Note that the probability of failure is less than the number of times of requested h-queries and Signature queries divided by the length of hash value times two.

Adversary A outputs a valid signature (r_1, s_1) of the message, m_1, where m_1 is not requested on Signature query.

Using the Oracle Replay Attack and the Forking Lemma as proposed in [24], we can obtain another valid signature (r_1, s_1') of message, m_1. In detail, B resets A two times. In the first time, B records all the transcripts that interacted with A, and in the second time, B does the same thing as the first time except h-query. For instance, B picks a random numbers h_1 as the returned value for the first time and a different random numbers h_1' for the second time when A request h-query on (m_1, r_1). After two rounds of interacting with B, A outputs two valid signatures, (r_1, s_1) and (r_1, s_1') of the message m_1 with different hash values, h_1 and h_1'. Then, A sends (r_1, s_1) and (r_1, s_1')

to B. because both (r_1, s_1) and (r_1, s_1') are m_1's valid signature. So, B obtains $y^{h_1} = {r_1}^{r_1} \cdot g^{s_1} \bmod p$ and $y^{h_1'} = {r_1}^{r_1} \cdot g^{s_1'} \bmod p$. Thus, we have $y^{h_1'-h_1} = g^{s_1'-s_1} \bmod p$. If $\gcd(h_1' - h_1, p-1) = 1$, it is easy to compute the discrete logarithm of y as $x = (h_1' - h_1)^{-1} \bmod p - 1$. This result contradicts to the discrete logarithm assumption. Note that the probability of $\gcd(h_1' - h_1, p-1) = 1$, is big enough and can reach $1/2$ if we let $p = 2q + 1$ for some prime q.

In the following, we assume that there are two signers, A and B, with their private and public keys, (x_A, y_A) and (x_B, y_B) respectively, where $y_A = g^{x_A} \bmod p$ and $y_B = g^{x_B} \bmod p$. To digitally generate a valid multisignature (r, s), by A and B, according to [13], they compute $r_A = g^{k_A} \bmod p$ and $r_B = g^{k_B} \bmod p$, where k_A and k_B are random secrets selected by A and B, respectively from Z_{p-1}. r_A and r_B are exchanged with each other. Then, they compute $r = r_A \cdot r_B \bmod p$. With knowledge of their private keys, they can solve s_A and s_B satisfying $x_A \cdot h(m, r) = s_A + k_A \cdot r \bmod p - 1$ and $x_B \cdot h(m, r) = s_B + k_B \cdot r \bmod p - 1$, respectively. The multisignature of a message m is (r, s), where $s = s_A + s_B \bmod p - 1$. The multisignature can be verified by a verifier by checking whether $y^{h(m,r)} = r^r \cdot g^s \bmod p$. In the next section, we propose a threshold signature scheme based on this multisignature scheme to allow any t or more than t members to represent a group to generate a threshold signature.

There is a threshold signature scheme in [13] which integrates both Shamir's (t, n) SS and Harn's multisignature scheme. In fact, most existing threshold signature schemes are based on the linear polynomial. In the next section, we propose a novel approach to design a threshold signature scheme based on the CRT. We believe that our design opens a new direction to enable CRT-based SS to be integrated into other cryptographic functions. □

3 Models of Proposed Threshold Signature Scheme

3.1 Entities

In our proposed threshold scheme, there is a GM to register n members initially. The GM needs to select a pair of private and public keys of the group and divide the group private key into n shares. Each share will be sent to each member secretly. Later, any t or more than t members can work together to generate a group signature; but, fewer than t members cannot generate a group signature. Any verifier can use the group public key to verify the group signature.

3.2 Informal Model of Our Proposed Scheme

We assume that the GM selects a pair of private and public keys, (x, y), of the group and divides the group private key into n shares, $x_i, i = 1, 2, \ldots, n$, for members in the group, $U = \{U_1, U_2, \ldots, U_n\}$. Each member, U_i, will receive a share (i.e., private key), x_i, from the GM initially as his/her private key. In other words, the GM uses the Mignotte's (t, n) threshold SS to compute private keys, $(x_1, x_2, \ldots, x_n)$ for group members initially. The threshold signature generation, TSS, allows any t or more than t members to generate a group signature. The group signature can be verified according to the signature verification, VS, using the group public key. i.e.,

TSS: $(m, x_{i_1}, x_{i_2}, \ldots, x_{i_t}) \rightarrow$a group signature; where $\forall U_{i_r} \in U$;

VS: (m, a group signature, group public key)$\rightarrow$yes/no.

3.3 Properties

We propose a threshold signature scheme with the following properties:

Protection of private keys. Our scheme protects the secrecy of private keys of the group and members; otherwise, private keys can be used to generate only one group signature.

Unforgibility of group signature. Our scheme ensures that (a) any t or more than t members can work together to generate a valid group signature, and (b) fewer than t members cannot generate a valid group signature.

Fixed length of threshold signature. Our scheme ensures that the length of a threshold signature is fixed (i.e., not depending on the number of signers).

Efficiency of verification. The verification of a group signature is based on the group public key.

4 Proposed Threshold Signature Scheme

4.1 Outline of Our Design

In our proposed scheme, there is a trusted GM who is responsible for setting up the system initially. The GM needs to select public parameters and a pair of private and public keys of the group. The GM needs to register all members initially and follow the Mignotte's SS to divide the group private key into shares (private keys of members) and send a private key, x_i, for each member.

In the threshold signature generation, each group member needs to use his/her private key to generate an individual signature. The individual signature needs to be sent to a signature combiner. The signature combiner can be any participated member who is responsible to collect all individual signatures and produce a group signature. The signature combiner needs to verify each individual signature and then combine all individual signatures into a group signature.

4.2 Proposed Threshold Signature Scheme

Public and private key generation: The GM selects a sequence of pairwise coprime positive integers, $p_1 < p_2 < \ldots < p_n$, where p_i is the public information associated with member, U_i. These public integers need to satisfy that $p_{n-t+2} \cdot p_{n-t+3} \cdot \ldots \cdot p_n < p_1 \cdot p_2 \cdot \ldots \cdot p_t$. In addition, the GM selects a prime modulus, p, where integers in the set, $\{p_1, p_2, \ldots, p_n\}$, are divisors of $p-1$ (i.e., $p_1, p_2, \ldots, p_n | p-1$), a generator, g, of the subgroup of order $N = p_1 \cdot p_2 \cdot \ldots \cdot p_n$ such that $1 < g < p$.

For this given sequence, the GM chooses the private key x of the group as an integer in the range $R_t = \{x \in Z | p_{n-t+2} \cdot p_{n-t+3} \cdot \ldots \cdot p_n < x < p_1 \cdot p_2 \cdot \ldots \cdot p_t\}$. The public key of the group is $y = g^x \bmod p$. Private key of the member, U_i, is generated as $x_i = x \bmod p_i$. The public key of the member, U_i, is computed as $y_i = g^{x_i} \bmod p$. x_i is sent to each member, U_i, secretly.

Remark 2. *Following steps are used to generate the generator g.*

1) $e = (p-1)/N$;
2) *Set α be any integer satisfying $1 < \alpha < p-1$, such that α differs from any value previously tried;*
3) $g = \alpha^e \bmod p$.;
4) *If $(g = 1)$, then go to step 2; otherwise return g.*

The following lemma proves the order of the generator.

Lemma 1. *For any nonnegative integer b if $g = \alpha^{(p-1)/N} \bmod p$, then $g^b = g^{b \bmod N} \bmod p$.*

Proof. From the Fermat theorem, since $\gcd(\alpha, p) = 1$, we have $\alpha^{p-1} \bmod p = 1$. Hence, for any nonnegative integer c, we have $g^{cN} \bmod p = (\alpha^{(p-1)/N})^{cN} \bmod p == (\alpha^{p-1})^c \bmod p = 1$. Thus, any nonnegative integer b can be represented as $b = dN + z$, where $0 < d, z < N$. Then,$g^b \bmod p = g^{dN+z} \bmod p == g^z \bmod p$. Since $z = b \bmod N$. we have proven this lemma. □

Threshold signature generation:

The proposed scheme allows any t or more than t members to represent the group to generate a group signature. Assume that members in the subset $U = \{U_{i_1}, U_{i_2}, \cdots, U_{i_t}\}$ want to generate a group signature for a message m. There are two parts involved in this phase.

Individual signature generation and verification.

Every member U_{i_v} randomly selects an integer $k_v \in Z_N$ and computes $r_v = g^{k_v} \bmod p$. r_v is made available to all other members in the subset U. After receiving all values, $r_v, v = 1, 2, \ldots, t$, every member U_{i_v} computes $r = (r_1 \cdot r_2 \cdot \ldots \cdot r_t)^{N \backslash N'} \bmod p$, where $N = (p_1 \cdot p_2 \cdot \ldots \cdot p_n)$ and $N' = (p_{i_1} \cdot p_{i_2} \cdot \ldots \cdot p_{i_t})$. Then, every member U_{i_v} uses his/her private key, x_{i_v}, to generate a partial signature of the message m as $s_v = (N'/p_{i_v}) \cdot w_{i_v} \cdot x_{i_v} \cdot h(m, r) - k_v \cdot r \bmod N'$, where $(N'/p_{i_v}) \cdot w_{i_v} \bmod p_{i_v} = 1$. The individual signature, (r_v, s_v) of member U_{i_v} is sent to the signature combiner.

Once receiving the individual signature, (r_v, s_v), from member U_{i_v}, the signature combiner uses the public key, y_{i_v} of member U_{i_v} to verify whether $y_{i_v}^{(N/p_{i_v}) \cdot w_{i_v} \cdot h(m,r)} \stackrel{?}{=} g^{(N/N') \cdot s_v} \cdot r_v^{(N/N') \cdot r} \bmod p$. If it is, the individual signature has been successfully verified.

Theorem 2. *If $y_{i_v}^{(N/p_{i_v}) \cdot w_{i_v} \cdot h(m,r)} = g^{(N/N') \cdot s_v} \cdot r_v^{(N/N') \cdot r} \bmod p$, the individual signature has been verified successfully.*

Proof. With the knowledge of the secrets, x_{i_v} and k_v, member U_{i_v} is able to compute $s_v = (N'/p_{i_v}) \cdot w_{i_v} \cdot x_{i_v} \cdot h(m, r) - k_v \cdot r \bmod N'$, where $N' = p_{i_1} \cdot p_{i_2} \cdot \ldots \cdot p_{i_t}$. Since N' is a factor of N (i.e., $N' | N$), we have $g^{(N/N')}$ is a generator of the subgroup of order N'. Hence, we can get

$$\begin{aligned} y_{i_v}^{(N/p_{i_v}) \cdot w_{i_v} \cdot h(m,r)} &= (g^{(N/N')})^{(N'/p_{i_v}) \cdot x_{i_v} \cdot w_{i_v} \cdot h(m,r)} \\ &= (g^{(N/N')})^{s_v} \cdot (g^{(N/N')})^{k_v \cdot r} \\ &= g^{(N/N') \cdot s_v} \cdot r_v^{(N/N') \cdot r} \bmod p. \end{aligned}$$

□

Threshold signature generation. After all individual signatures, $(r_v, s_v), v = 1, 2, \ldots, t$, having been verified successfully, the threshold signature, (N', r, s) of the message m is computed as $s = (N/N') \cdot (\sum_{v=1}^{t} s_v \bmod N')$.

Threshold signature verification: Using the group public key, y, the threshold signature, (N', r, s) of the message m can be verified by first checking whether N is divisible by N' and then checking whether $y^{(N/N') \cdot h(m,r)} \stackrel{?}{=} g^s \cdot r^r \bmod p$. If it is, threshold signature has been successfully verified.

Theorem 3. *If N is divisible by N', and $y^{(N/N') \cdot h(m,r)} = g^s \cdot r^r \bmod p$, the threshold signature has been verified successfully.*

Proof. It is obvious that since $N = p_1 \cdot p_2 \cdot \ldots \cdot p_n$ and $N' = p_{i_1} \cdot p_{i_2} \cdot \ldots \cdot p_{i_t}$. N is divisible by N'. In addition, since every individual signature, (r_v, s_v) satisfies $s_v = (N'/p_{i_v}) \cdot w_{i_v} \cdot x_{i_v} \cdot h(m, r) - k_v \cdot r \bmod N'$, the threshold signature s is $s = (N/N') \cdot (\sum_{v=1}^{t} s_v \bmod N') = (N/N') \cdot (\sum_{v=1}^{t} (N'/p_{i_v}) \cdot w_{i_v} \cdot x_{i_v} \cdot h(m, r) - k_v \cdot r \bmod N')$. According to the secret reconstruction in Mignotte's SS, we have

$$x = \sum_{v=1}^{t} (N'/p_{i_v}) \cdot w_{i_v} \cdot x_{i_v} \bmod N'.$$

Table 1: Comparison with other schemes

Scheme	**Kaya and Selcuk's scheme [17]**	**Guo and Chang's scheme [12]**	**Our scheme**
Players for generating (t, n) *signature*	$2t$	t	t
Whether or not provable security	Yes	No	Yes
Which signature scheme is based on	DSS	RSA	Harn's multisignature signature scheme [13]
Which secret sharing scheme is based on	Asmuth and Bloom's threshold SS [1]	Weighted threshold SS [15]	Mignotte's threshold SS [22]

Thus, we have

$$s = (N/N')\cdot(x\cdot h(m,r) - \sum_{v=1}^{t} k_v\cdot r \bmod N').$$

Hence, we can get

$$\begin{aligned} y^{(N/N')\cdot h(m,r)} &= g^s\cdot(g^{(N/N')})^{(r\cdot\sum_{v=1}^{t} k_v) mod N'} \bmod p \\ &= g^s\cdot(\prod_{v=1}^{t} r_v)^{(N/N')\cdot r} \bmod p \\ &= g^s\cdot r^r \bmod p. \end{aligned}$$

□

5 Security Analysis and Comparisons

In the following discussion, we analyze the properties described in Section 3.3 and compare our scheme with some other threshold schemes [12, 17].

Protection of private keys. Every member U_{i_v} needs to use his/her private key, x_{i_v} to generate an individual signature of the message m as $s_v = (N'/p_{i_v})\cdot w_{i_v}\cdot x_{i_v}\cdot h(m,r) - k_v\cdot r \bmod N'$. The private key, x_{i_v}, cannot be recovered by other members since there is one more secret, k_{i_v}, known only to the member, U_{i_v}.

Unforgibility of group signature. The private key, x, of the group is protected by the SS. It needs t or more than t members to recover the group private key. With fewer than t private keys cannot recover the group private key and therefore cannot generate a valid group signature.

Similar to [19, 30], we suppose that there is an adversary A who can corrupt at most $t-1$ members at the beginning of the signature. The adversary A adaptively chooses messages $m_1, m_2, \ldots, m_k$ for signature query, then the adversary A attempts to forge a valid signature for new message m. If there is no such adversary can successfully forge a valid signature for m with non-negligible probability, we say the threshold signature scheme unforgibility.

Theorem 4. *Our proposed threshold signature scheme is secure under the random oracle model against known-message attack and against adaptively chosen message attack.*

Proof. Suppose an adversary A with $t-1$ corrupted members can break the proposed threshold signature scheme. Without loss of generality, we assume that the corrupted members are $U_1, U_2, \ldots, U_{t-1}$. It is to say that the adversary A can forges a signature (N', r, s) of m which satisfies $y^{(N/N')\cdot h(m,r)} = r^r\cdot g^s \bmod p$. In fact, the adversary A cannot obtain the private key x from $t-1$ corrupted members because the private key is protected by the SS, and therefore cannot generate a signature (N', r, s) of m which satisfies $y^{(N/N')\cdot h(m,r)} = r^r\cdot g^s \bmod p$ without knowing private key x according to Theorem 1. Thus, the adversary A must use the $t-1$ private keys of corrupted members to compute the forged signature. In other words, the adversary generates the $t-1$ individual signatures (r_v, s_v) of m satisfying $y_{i_v}^{(N'/p_{i_v})\cdot w_{i_v}\cdot h(m,r)} = g^{(N/N')\cdot s_v}\cdot r_v^{(N/N')\cdot r} \bmod p$, for $v = 1, 2, \ldots, t-1$. Then, the adversary computes $s_t = s - \sum_{v=1}^{t-1} s_v \bmod N'$ and $r_t = r^{(N/N')^{-1} mod N'}\cdot(r_1\cdot r_2\cdot\ldots\cdot r_{t-1})^{-1} \bmod p$ without knowing the private key x_t. Obviously, (r_t, s_t) satisfies $y_t^{(N'/p_t)\cdot w_t\cdot h(m,r)} = g^{(N/N')\cdot s_t}\cdot r_t^{(N/N')\cdot r} \bmod p$; however, in a similar approach as used in proving Theorem 1, it is impossible since this result contradicts to the discrete logarithm assumption. □

Fixed length of threshold signature. The length of the threshold signature is identical to the length of an individual signature.

Efficiency of verification. Any verifier does not need to know the signers of a group signature. The group signature is verified using the public key of the group.

Next, we compare our scheme with some other threshold schemes [12, 17] which are based on the Chinese Remainder Theorem too. The result of comparisons is described in Table 1. Since RSA signature is not provable

security [21], the scheme [12] is not provable security because that the verification of scheme [12] is the same as RSA signature. Furthermore, we can use the existing message-signature pairs (M_1, s_1) and (M_2, s_2) to forge a new message-signature pairs $(M_1 \cdot M_2, s_1 \cdot s_2)$ [21] easily. Therefore, our scheme is more secure than scheme [12]. As for scheme [17], it needs $2t$ players to generate a (t, n) threshold signature. Therefore, our scheme is more efficient than scheme [17] because our scheme needs t players to generate a (t, n) threshold signature.

6 Conclusions

A threshold signature scheme is a useful tool to support the group-oriented application. The threshold signature enables t or more than t members to represent a group to generate a group signature; but, fewer than t group members cannot generate a group signature. Most existing threshold signature schemes are based on the linear polynomial. We propose a threshold signature scheme based on the CRT. By selecting parameters properly, the CRT-based SS can be applied in designing a threshold signature scheme. We believe that our design opens a new direction to enable CRT-based SS to be integrated into other cryptographic functions.

References

[1] C. A. Asmuth, J. Bloom, "A modular approach to key safeguarding," *IEEE Transactions on Information Theory*, vol. IT-29, no. 2, pp. 208–210, 1983.

[2] J. C. Benaloh, "Secret sharing homomorphisms: keeping shares of a secret," in *Proceedings of Advances in Cryptology (Crypto'86)*, LNCS 263, pp. 251–260, Springer, Aug. 1986.

[3] G. R. Blakley, "Safeguarding cryptographic keys," in *Proceedings of American Federation of Information Processing Society (AFIPS'79)*, pp. 313–317, New York, USA, June 1979.

[4] H. Cohen, *A Course in Computational Algebraic Number Theory (Graduate Texts in Mathematics), Fourth ed.*, Springer-Verlag Press, 2000.

[5] Y. Desmedt, "Society and group oriented cryptography: a new concept," in *Proceedings of Advances in Cryptology (Crypto'87)*, LNCS 293, pp. 120–127, Springer, 1978.

[6] Y. Desmedt, Y. Frankel, "Threshold cryptosystems," in *Proceedings of Advances in Cryptology (Crypto'89)*, LNCS 435, pp. 307–315, Springer, Aug. 1989.

[7] Y. Desmedt, Y. Frankel, "Shared generation of authenticators and signatures," in *Proceedings of Advances in Cryptology (Crypto'91)*, LNCS 576, pp. 457–569, Springer, Aug. 1991.

[8] T. ElGamal, "A public key cryptosystem and signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. IT- 31, no. 4, pp. 469–472, 1985.

[9] W. Gao, G. Wang, X. Wang, Z. Yang, "One-round ID-based threshold signature scheme from bilinear pairings," *Informatica*, vol. 20, pp. 461–476, 2009.

[10] R. Gennaro, S. Jarecki, H. Krawczyk, T. Rabin, "Robust threshold DSS signatures," in *Proceedings of Advances in Cryptology (Eurocrypt'96)*, LNCS 1070, pp. 354–371, Springer, May 1996.

[11] R. Gennaro, S. Jarecki, H. Krawczyk, T. Rabin, "Robust and efficient sharing of RSA functions," in *Proceedings of Advances in Cryptology (Crypto'96)*, LNCS 1109, pp. 157–172, Springer, Aug. 1996.

[12] C. Guo, C. C. Chang, "Proactive weighted threshold signature based on Generalized Chinese Remainder Theorem," *Journal of Electronic Science and Technology*, vol. 10, pp. 250–255, 2012.

[13] L. Harn, "Group-oriented (t,n) threshold digital signature scheme and digital multisignature," *IEE Proceedings of Computer Digital Technique*. vol. 141, pp. 307–313, 1994.

[14] M. S. Hwang, T. Y. Chang, "Threshold signatures: current status and key issues," *International Journal of Network Security*, vol. 1, pp. 123–137, 2005.

[15] S. Iftene, "General secret sharing based on the Chinese remainder theorem with applications in e-voting," *Electronic Notes in Theoretical Computer Science*, vol. 186, pp. 67–84, 2007.

[16] K. Kaya, A. A. Selcuk, "Threshold cryptography based on Asmuth-Bloom Secret sharing," *Information Sciences*, vol. 177, pp. 4148–4160, 2007.

[17] K. Kaya, A. A. Selcuk, "Sharing DSS by the Chinese Remainder Theorem," *Journal of Computational and Applied Mathematics*, vol. 259, pp. 495–502, 2014.

[18] K. Kaya, A. A. Selcuk, "Robust threshold schemes based on the Chinese remainder theorem," in *Proceedings of Advances in Cryptology (AFRICACRYPT'08)*, Casablanca, Morocco, 2008.

[19] S. Kim, J. Kim, J. H. Cheon, S. Ju, "Threshold signature schemes for ElGamal variants," *Computer Standards & Interfaces*, Vol. 33, pp. 432–437, 2011.

[20] Y. P. Lai, C. C. Chang, "Parallel Computation Alalgorithms for Generalized Chinese Remainder Theorem," *Computers and Electrical Engineering*, vol. 29, pp. 801–811, 2003.

[21] W. Mao, *Modern Cryptography: Theory and Practice*, Publishing House of Electronic Industry, Beijing, China, 2004.

[22] M. Mignotte, "How to share a secret," in *Proceedings of Cryptography-Proceedings of the Workshop on Cryptography*, LNCS 149, pp. 371–375, Springer, 1983.

[23] National Institute of Satndards and Technology (NIST), The digital signature standard proposed by NIST, 1992.

[24] D. Pointcheval, J. Stern, "Security proofs for signature schemes," in *Proceedings of Advances in Cryptology (Eurocrypt'96)*, LNCS 1070, pp. 387–98, Springer, May 1996.

[25] R. Rivest, A. Shamir, L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of ACM*, vol. 21, pp. 120–126, 1978.
[26] A. De Santis, Y. Desmedt, Y. Frankel, M. Yung, "How to share a function securely," in *Proceedings of 26th Annual ACM Symposium on Theory of Computing (STOC'94)*, pp. 522–533, Canada, May 1994.
[27] A. Shamir, "How to share a secret," *Communications of ACM*, vol. 22, no. 11, pp. 612–613, 1979.
[28] Y. Shang, X. Wang, Y. Li, Y. Zhang, "A general threshold signature scheme based on Elliptic Curve," in *Proceedings of the 2nd International Conference on Computer and Information Application (ICCIA'12)*, pp. 89–92, Taiyuan, Shanxi, China, Dec. 2012.
[29] V. Shoup, "Practical threshold signatures," in *Proceedings of Advances in Cryptology (Eurocrypt'00)*, LNCS 1807, pp. 207–220, Springer, May 2000.
[30] H. Xiong , Z. G. Qin, F.G. Li, "Identity-based Threshold Signature Secure in the Standard Model," *International Journal of Network Security*, Vol. 10, pp. 75–80, 2010.

Lein Harn received the B.S. degree in electrical engineering from the National Taiwan University in 1977, the M.S. degree in electrical engineering from the State University of New York-Stony Brook in 1980, and the Ph.D. degree in electrical engineering from the University of Minnesota in 1984. In 1984, he joined the Department of Electrical and Computer Engineering, University of Missouri- Columbia as an assistant professor, and in 1986, he moved to Computer Science and Telecommunication Program (CSTP), University of Missouri, Kansas City (UMKC). While at UMKC, he went on development leave to work in Racal Data Group, Florida for a year. Currently, he is a Professor at the Department of Computer Science Electrical Engineering, UMKC. His research interests include cryptography, network security, and wireless communication security. He has published a number of papers on digital signature design and applications and wireless and network Security. He has written two books on security. He is currently investigating new ways of using secret sharing in various applications.

Feng Wang was born in Shandong province, China, in 1978. He received his B.S. degree in Mathematics from Yantai Normal University (now named Ludong University), Yantai, in 2000 and the M.S. degree in Applied Mathematics from the Guangzhou University, Guangzhou, in 2006. Currently, he is a Lecturer in the College of Mathematics and Physics at Fujian University of Technology and a visiting scholar in Department of Information Engineering and Computer Science at Feng Chia University. His research interests include computer cryptography and information security.