Secret Sharing Extensions based on the Chinese Remainder Theorem

Kamer Kaya, Ali Aydın Selçuk

Department of Computer Engineering
Bilkent University
Ankara, 06800, Turkey
{kamer,selcuk}@cs.bilkent.edu.tr

Abstract. In this paper, we investigate how to achieve verifiable secret sharing (VSS) schemes by using the Chinese Remainder Theorem (CRT). We first show that two schemes proposed earlier are not secure from an attack where the dealer is able to distribute inconsistent shares to the users. Then we propose a new VSS scheme based on the CRT and prove its security. Using the proposed VSS scheme, we develop joint random secret sharing (JRSS) and proactive SSS protocols, which, to the best of our knowledge, are the first secure protocols of their kind based on the CRT.
Keywords: Verifiability, joint random secret sharing, proactive secret sharing, Chinese Remainder Theorem, Asmuth-Bloom

1 Introduction

Threshold cryptography deals with the problem of sharing a highly sensitive secret among a group of users so that only when a sufficient number of them come together can the secret be reconstructed. Well-known secret sharing schemes (SSSs) in the literature include Shamir [26], based on polynomial interpolation, Blakley [2], based on hyperplane geometry, and Asmuth-Bloom [1], based on the Chinese Remainder Theorem (CRT).

A t-out-of-n secret sharing scheme contains two phases: In the *dealer phase*, the dealer shares a secret among n users. In the *combiner phase*, a coalition of size greater than or equal to t constructs the secret. We call a SSS *verifiable* if each user can verify the correctness of his share in the dealer phase and no user can lie about his share in the combiner phase. Hence, neither the dealer nor the users can cheat in a VSS scheme. Verifiable secret sharing schemes based on Shamir's SSS have been proposed in the literature [9, 23]. These schemes have been extensively studied and used in threshold cryptography and secure multi-party computation [12, 22, 23].

There have been just two CRT-based VSS schemes by Iftene [16] and Qiong et al. [24]. In this paper, we show that these schemes are vulnerable to attacks where a corrupted dealer can distribute *inconsistent* shares without detection such that different coalitions will obtain different values for the secret. To the best of our knowledge, these are the only VSS schemes that have been proposed so far based on the CRT.

A typical application of a VSS scheme is the joint random secret sharing (JRSS) primitive frequently used in threshold cryptography [12, 17, 18, 22, 23]. In a JRSS scheme, all players act as a dealer and jointly generate and share a random secret. So far, there have been no JRSS protocols proposed based on the CRT. Another important extension in threshold cryptography is the *proactivity* feature of secret sharing schemes. With this feature, a SSS has the capability of renewing the shares of the users without changing the long-term secret such that any shares obtained by a corrupted party becomes obsolete. So far, no CRT-based proactive secret sharing (PSS) schemes have been proposed in the literature.

In this paper, we first show why existing attempts of a CRT-based verifiable secret sharing scheme fail when attacked. We then propose a VSS scheme based on the Asmuth-Bloom secret sharing [1] and using this VSS scheme, we propose a JRSS scheme. By combining and extending the ideas used in the VSS and JRSS schemes, we also propose a PSS scheme. To the best of our knowledge the VSS, JRSS, and PSS schemes we propose are the first secure CRT-based schemes of their kind in the literature.

The rest of this paper is organized as follows: In Section 2, we describe the Asmuth-Bloom SSS in detail and introduce the notation we followed in this paper. In Section 3, we describe the VSS schemes proposed in [16, 24] and analyze their flaws. After presenting our VSS scheme in Section 4, we propose the joint random scheme in Section 5. In Section 6, we describe the CRT-based proactive SSS and in Section 7, we analyze the practicability and efficiency of the schemes. Section 8 concludes the paper.

2 Asmuth-Bloom Secret Sharing Scheme

The Asmuth-Bloom SSS [1] shares a secret d among n parties by modular arithmetic such that any t users can reconstruct the secret by the CRT. The scheme presented in Figure 1 is a slightly modified version by Kaya and Selcuk [19] that obtains better security properties.

– *Dealer Phase:* To share a secret d among a group of n users, the dealer does the following:
  - A set of relatively prime integers $m_0 < m_1 < \ldots < m_n$ are chosen, where m_0 is a prime and
$$\prod_{i=1}^{t} m_i > {m_0}^2 \prod_{i=1}^{t-1} m_{n-i+1}. \quad (1)$$
  - Let M denote $\prod_{i=1}^{t} m_i$. The dealer computes $y = d + Am_0$ where A is a positive integer generated randomly subject to the condition that $0 \leq y < M$.
  - The share of the ith user, $1 \leq i \leq n$, is $y_i = y \bmod m_i$.
– *Combiner Phase:* Let S be a coalition of t users gathered to construct the secret. Let M_S denote $\prod_{i \in S} m_i$.
  - Let $M_{S\setminus\{i\}}$ denote $\prod_{j \in S, j \neq i} m_j$ and $M'_{S,i}$ be the multiplicative inverse of $M_{S\setminus\{i\}}$ in $\mathbb{Z}_{m_i}$, i.e., $M_{S\setminus\{i\}} M'_{S,i} \equiv 1 \pmod{m_i}$. First, the ith user computes
$$u_i = y_i M'_{S,i} M_{S\setminus\{i\}} \bmod M_S.$$
  - The users then compute
$$y = \left(\sum_{i \in S} u_i \right) \bmod M_S$$
  and obtain the secret d by computing
$$d = y \bmod m_0.$$

Fig. 1. Asmuth-Bloom secret sharing scheme

According to the Chinese Remainder Theorem, y can be determined uniquely in $\mathbb{Z}_{M_S}$ since $y < M \leq M_S$ for any coalition S of size t.

Kaya and Selcuk [19] showed that the Asmuth-Bloom version presented here is *perfect* in the sense that no coalition of size smaller than t can obtain any information about the secret.

Quisquater et al. [25] showed that when m_is are chosen as consecutive primes, the scheme has better security properties. In this paper, we will also assume that all m_is are prime and we will choose them such that $p_i = 2m_i + 1$ is also a prime for $1 \leq i \leq n$. The notation used in our paper is summarized in Table 1.

Notation	Explanation
n	The number of users.
t	The threshold, the minimum number of users required to construct the secret.
d	The secret to be shared.
m_0	A prime; specifies the domain of $d \in \mathbb{Z}_{m_0}$.
$m_i : 1 \leq i \leq n$	The prime modulus for user i.
$p_i : 1 \leq i \leq n$	A safe prime, $2m_i + 1$.
P	$\prod_{i=1}^{n} p_i$.
y	$d + Am_0$, where A is a random number.
M	The domain of $y \in \mathbb{Z}_M$.
$y_i : 1 \leq i \leq n$	$y \bmod m_i$, the share of user i.
$E(y)$	The commitment value of an integer y.
S	A coalition of users.
M_S	The modulus of coalition S, $\prod_{i \in S} m_i$.

Table 1. Notations

For the protocols in this paper, we assume that private channels exist between the dealer and users. The share of each user is sent via these private channels, hence, no one except the user himself knows the share. Besides, we assume that a broadcast channel exists and if some data is broadcast each user will read the same value. Hence, an adversary cannot send two different values to two different users in a broadcast data.

3 Analysis of the Existing CRT-based VSS Schemes

There have been two different approaches to achieve VSS by a CRT-based secret sharing scheme. The first one, proposed by Iftene [16], obtains a VSS scheme from Mignotte's SSS [20], which is another CRT-based

SSS similar to Asmuth-Bloom. Here, we adapt Iftene's approach to the Asmuth-Bloom SSS. The scheme is given in Figure 2.

- *Dealer Phase:* To share a secret $d \in \mathbb{Z}_{m_0}$ among a group of n users with verifiable shares, the dealer does the following:
  1. Use the dealing procedure of the Asmuth-Bloom SSS to obtain the shares $y_i = y \bmod m_i$ for each $1 \leq i \leq n$ where $y = d + Am_0 < M$. Choose m_is such that each $p_i = 2m_i + 1$ is also a prime.
  2. Let $g_i \in \mathbb{Z}_{p_i}^*$ be an element of order m_i. The dealer sends y_i to the ith user privately and makes the values p_i, g_i, and $z_i = g_i^y \bmod p_i$ public for $1 \leq i \leq n$. The ith user can find whether his share is valid or not by checking
$$z_i \stackrel{?}{=} {g_i}^{y_i} \bmod p_i. \tag{2}$$
- *Combiner Phase:* Let S be a coalition gathered to construct the secret.
  1. The share y_i of user $i \in S$ can be verified by the other users in S by the verification equation $z_i \stackrel{?}{=} {g_i}^{y_i} \bmod p_i$.
  2. If all shares are valid then the coalition S can obtain the secret d: First, the ith user computes
$$u_i = y_i M'_{S,i} M_{S\setminus\{i\}} \bmod M_S.$$
  3. Then the users compute
$$y = \left(\sum_{i \in S} u_i \right) \bmod M_S$$
and obtain the secret d by computing $d = y \bmod m_0$.

Fig. 2. Iftene's CRT-based VSS extension

If the dealer is honest and the discrete logarithm problem is hard, the scheme in Figure 2 is secure against a dishonest user because the verification data, ${g_i}^y \bmod p_i$, can be used to detect an invalid share from a corrupted user in the first step of the combiner phase.

However, if the dealer is dishonest, he can mount an attack despite the additional verification data above: Let y be an integer and $y_i = y \bmod m_i$ for $1 \leq i \leq n$. In the combiner phase of the Asmuth-Bloom SSS, the minimum number of users required to obtain the secret is t; hence, $y = d + Am_0$ must be smaller than $M = \prod_{i=1}^{t} m_i$. Note that to reconstruct the secret d, each coalition S must first compute $y \bmod M_S$, where $M_S \geq M$. If the dealer distributes the shares for some $y > M$, then y will be greater than M_S for some coalition S of size t. Hence, S may not compute the correct y value and the correct secret d even though $y_i = y \bmod m_i$

for all i. Therefore, the given VSS scheme cannot detect these kinds of inconsistent shares from the dealer where different coalitions end up with different d values. The same problem also arises in Iftene's original VSS scheme [16].

Another VSS scheme based on Asmuth-Bloom secret sharing was proposed by Qiong et al. [24]. Their approach is similar to the Pedersen's VSS [23] based on Shamir's SSS. Their scheme is given in Figure 3.

- *Dealer Phase:* To share a secret $d \in \mathbb{Z}_{m_0}$ among a group of n users with verifiable shares, the dealer does the following:
  1. Use the dealing procedure of the Asmuth-Bloom SSS to obtain the shares $y_i = y \bmod m_i$ for all $1 \leq i \leq n$ where $y = d + Am_0 < M$.
  2. Let p, q be primes such that $q|(p-1)$. Construct the unique polynomial $f(x) \in \mathbb{Z}_q[x]$ where $deg(f(x)) = n-1$ and $f(m_i) = y_i$. Construct a random polynomial $f'(x) \in \mathbb{Z}_q[x]$ where $deg(f'(x)) = n-1$. Let $z_i = f'(m_i)$ for all $1 \leq i \leq n$.
  3. Let $g \in \mathbb{Z}_p$ with order q, h be a random integer in the group generated by g, and $E(a,b) = g^a h^b \bmod p$ for inputs $a, b \in \mathbb{Z}_q^*$. Compute
  $$E_i = E(f_i, f_i') = g^{f_i} h^{f_i'} \bmod p,$$
  where f_i and f_i' are the $(i-1)$th coefficients of $f(x)$ and $f'(x)$, respectively, for all $1 \leq i \leq n$. Broadcast E_is to all users.
  4. Send (y_i, z_i) secretly to the ith user for all $1 \leq i \leq n$.
  5. Each user checks
  $$E(y_i, z_i) \stackrel{?}{\equiv} \prod_{j=1}^{n} {E_j}^{{m_i}^{j-1}} \equiv \prod_{j=1}^{n} g^{f_j {m_i}^{j-1}} \prod_{j=1}^{n} h^{f_j' {m_i}^{j-1}} \equiv g^{y_i} h^{z_i} \pmod p \quad (3)$$
  to verify the validity of his share.
- *Combiner Phase:* Let S be a coalition gathered to construct the secret.
  1. The share (y_i, z_i) of user $i \in S$ can be verified by the other users in S with the verification equality $E(y_i, z_i) \stackrel{?}{\equiv} \prod_{j=1}^{n} {E_j}^{{m_i}^{j-1}} \pmod p$.
  2. If all shares are valid; the coalition S can obtain the secret d by using the reconstruction procedure described in Section 2.

Fig. 3. Qiong et al.'s CRT-based VSS extension

As the scheme shows, Qiong et al. treated the shares of, Asmuth-Bloom SSS as points on a degree-$(n-1)$ polynomial and adopted Pedersen's approach by evaluating the polynomial in the exponent to verify the shares. If the dealer is honest, the scheme in Figure 3 is secure be-

cause the verification data can be used to detect an invalid share from a corrupted user in the first step of the combiner phase.

However, similar to the attack on Iftene's VSS scheme, if the dealer uses some $y > M$ and computes the verification data by using the shares $y_i = y \bmod m_i$, $1 \leq i \leq n$, the verification equation (3) holds for each user. But, for a coalition S where $y > M_S$, the coalition S cannot compute the correct y value and the secret d.

Note that Iftene's VSS scheme uses a separate verification data for each user; hence, even if all verification equations hold, the secret can still be inconsistent for different coalitions. Quiong et al.'s VSS scheme generates a polynomial $f(x)$ from the shares, as in Feldman's and Pedersen's VSS schemes. This polynomial is used to check all verification equations. But the Asmuth-Bloom SSS depends on the CRT and unlike Shamir's SSS, here f is not inherently related to the shares. Hence, even if all equations hold, the shares can still be inconsistent, as we have shown.

4 Verifiable Secret Sharing with the Asmuth-Bloom SSS

As discussed in Section 3, existing CRT-based VSS schemes in the literature cannot prevent a dealer from cheating. To solve this problem, we will use a range-proof technique originally proposed by Boudot [4] and modified by Cao et al. [6].

4.1 Range-Proof Techniques

Boudot [4] proposed an efficient and non-interactive technique to prove that a committed number lies within an interval. He used the Fujisaki-Okamoto commitment scheme [11], where the commitment of a number y with bases (g, h) is computed as

$$E = E(y, r) = g^y h^r \bmod N,$$

where g is an element in $\mathbb{Z}_N^*$, h is an element of the group generated by g, and r is a random integer. As proved in [4, 11], this commitment scheme is statistically secure assuming the factorization of N is not known.

After Boudot, Cao et al. [6] applied the same proof technique with a different commitment scheme,

$$E = E(y) = g^y \bmod N,$$

to obtain shorter range proofs. Here, we will use Cao et al.'s non-interactive range-proof scheme as a black box. For further details, we refer the user

to [4, 6]. For our needs, we modified the commitment scheme to

$$E = E(y) = g^y \bmod PN,$$

where $P = \prod_{i=1}^{n} p_i$ and N is an RSA composite whose factorization is secret. Note that even if $\phi(P)$ is known, $\phi(PN)$ cannot be computed since $\phi(N)$ is secret. Throughout the section, we will use $\text{RngPrf}(E(y), M)$ to denote the range proof that a secret integer y committed with $E(y)$ is in the interval $[0, M)$.

4.2 A CRT-Based VSS Scheme

In our VSS scheme, the RSA composite N is an integer generated jointly by the users and the dealer, where its prime factorization is not known. Such an integer satisfying these constraints can be generated by using the protocols proposed for shared RSA key generation [3, 10] at the beginning of the protocol. Note that we do not need the private and the public RSA exponents in our VSS scheme as in the original protocols [3, 10]; hence, those parts of the protocols can be omitted.

Let $g_i \in \mathbb{Z}_{p_i}^*$ be an element of order m_i. Let $P = \prod_{i=1}^{n} p_i$ and

$$g = \left(\sum_{i=1}^{n} g_i P_i' \frac{P}{p_i} \right) \bmod P, \tag{4}$$

where $P_i' = \left(\frac{P}{p_i}\right)^{-1} \bmod p_i$ for all $1 \leq i \leq n$, i.e., g is the unique integer in $\mathbb{Z}_P$ satisfying $g_i = g \bmod p_i$ for all i. Our VSS scheme is described in Figure 4.

4.3 Analysis of the Proposed VSS Scheme

We analyze the correctness of the scheme and its security against passive and active attackers below:

Correctness Aside from the verification equation, the scheme uses the original Asmuth-Bloom scheme. Hence, for correctness, we only need to show that when the dealer and the users are honest, the verification equations in the dealer and combiner phases hold. Note that the condition $y < M$ is checked in Step 3 of the dealer phase by using $\text{RngPrf}(E(y), M))$. Furthermore, for a valid share y_i,

– *Dealer Phase:* To share a secret $d \in \mathbb{Z}_{m_0}$ among a group of n users with verifiable shares, the dealer does the following:
  1. Use the dealing procedure of the Asmuth-Bloom secret sharing scheme described in Section 2 to obtain the shares
  $$y_i = y \bmod m_i$$
  for each $1 \leq i \leq n$ where $y = d + Am_0 < M = \prod_{i=1}^{t} m_i$. Note that the m_is are large primes, where $p_i = 2m_i + 1$ is also a prime for $1 \leq i \leq n$.
  2. Let N be an integer whose prime factorization is not known by the users and the dealer. Compute $E(y) = g^y \bmod PN$. Send y_i to the ith user secretly for all $1 \leq i \leq n$ and broadcast $(E(y), \mathrm{RngPrf}(E(y), M))$.
  3. The ith user checks
  $$g_i{}^{y_i} \stackrel{?}{\equiv} E(y) \pmod{p_i} \tag{5}$$
  to verify $y_i = y \bmod m_i$. Then he checks the validity of the range proof to verify $y < M$.
– *Combiner Phase:* Let S be a coalition gathered to construct the secret.
  1. The share y_i of user $i \in S$ can be verified by the other users in S with the verification equality $g_i{}^{y_i} \stackrel{?}{\equiv} E(y) \pmod{p_i}$.
  2. If all shares are valid, the participants can obtain the secret d by using the reconstruction procedure described in Section 2. Otherwise, the corrupted users are disqualified.

Fig. 4. CRT-based verifiable secret sharing scheme

$$\begin{aligned} E(y) \bmod p_i &= g^y \bmod PN \bmod p_i \\ &= g^y \bmod p_i \\ &= g_i{}^y \bmod p_i \\ &= g_i{}^{y_i} \bmod p_i. \end{aligned}$$

Hence, if the dealer and the users behave honestly, the verification equation holds and the ith user verifies that his share is a residue modulo m_i of the integer $y < M$ committed with $E(y)$.

Security For the security analysis, we will first show that the underlying SSS is perfect, as proved by Kaya et al. [19], i.e., no coalition of size smaller than t can obtain any information about the secret.

Theorem 1 (Kaya and Selcuk [19]). *For a passive adversary with $t-1$ shares in the VSS scheme, every candidate for the secret is equally*

likely, i.e., the probabilities $\Pr(d = d')$ *and* $\Pr(d = d'')$ *are approximately equal for all* $d', d'' \in \mathbb{Z}_{m_0}$.

Proof. Suppose the adversary corrupts $t-1$ users and just observes the inputs and outputs of the corrupted users without controlling their actions, i.e., the adversary is honest in user actions but curious about the secret. Let S' be the adversarial coalition of size $t-1$, and let y' be the unique solution for y in $Z_{M_{S'}}$. According to (1), $M/M_{S'} > m_0$, hence, $y' + jM_{S'}$ is smaller than M for $j < m_0$. Since $\gcd(m_0, M_{S'}) = 1$, all $(y' + jM_{S'}) \bmod m_0$ are distinct for $0 \leq j < m_0$, and there are m_0 of them. That is, d can be any integer from $\mathbb{Z}_{m_0}$. For each value of d, there are either $\lfloor M/(M_{S'}m_0) \rfloor$ or $\lfloor M/(M_{S'}m_0) \rfloor + 1$ possible values of y consistent with d, depending on the value of d. Hence, for two different integers in $\mathbb{Z}_{m_0}$, the probabilities of d equals these integers are almost equal. Note that $M/(M_{S'}m_0) > m_0$ and given that $m_0 \gg 1$, all d values are approximately equally likely.

Aside from the shares, the only additional information a corrupted user can obtain are $E(y)$ and $\mathrm{RngPrf}(E(y), M)$. Given that the discrete logarithm problem is hard and Cao et al.'s range-proof technique is computationally secure, the proposed VSS scheme is also computationally secure. □

The shares distributed by a dealer are said to be inconsistent if different coalitions of size at least t obtain different values for the secret. The following theorem proves that the dealer cannot distribute shares inconsistent with the secret.

Theorem 2. *A corrupted dealer cannot cheat in the VSS scheme without being detected. I.e., if the shares are inconsistent with the secret* d *then at least one verification equation does not hold.*

Proof. Let $U = \{1, \ldots, n\}$ be the set of all users. If the shares are inconsistent, for two coalitions S and S' with $|S|, |S'| \geq t$,

$$\left(\sum_{i \in S} y_i M'_{S,i} M_{S \setminus \{i\}} \right) \bmod M_S \neq \left(\sum_{i \in S'} y_i M'_{S',i} M_{S' \setminus \{i\}} \right) \bmod M_{S'}.$$

Hence,

$$y = \left(\sum_{i=1}^{n} y_i M'_{U,i} M_{U \setminus \{i\}} \right) \bmod M_U > M.$$

If this is true then the dealer cannot provide a valid range proof $\mathrm{RngPrf}(E(y), M)$. So, when a user tries to verify that $y < M$, the range proof will not be verified.

If the dealer tries to use a different $y' \neq y$ value in the commitment $E(y')$ and generates a valid proof $\mathrm{RngPrf}(E(y'), M)$, the verification equation (5) will not hold for some user i.

Hence, the VSS scheme guarantees that the n distributed shares are consistent and that they are residues of some number $y < M$. □

Theorem 3. *A user cannot cheat in the VSS scheme without being detected; i.e., if a share given in the combiner phase is inconsistent with the secret, then the verification equation does not hold.*

Proof. When a user i sends an incorrect share $y_i' \neq y_i = y \bmod m_i$ in the combiner phase, the verification equation

$$E(y) \stackrel{?}{\equiv} {g_i}^{y_i} \pmod{p_i}$$

will not hold because $E(y) = g^y \bmod PN$, $p_i | P$, and since the order of $g_i \in \mathbb{Z}_{p_i}$ is m_i, the only value that satisfies the verification equation is y_i. □

Therefore, we can say that the scheme is secure for up to $t-1$ corrupted users and no participant can cheat in any phase of the scheme.

5 Joint Random Secret Sharing

Joint random secret sharing protocols enable a group of users to jointly generate and share a secret where a trusted dealer is not available. Although there have been JRSS schemes based on Shamir's SSS, so far no JRSS scheme has been proposed based on the CRT. Here we describe a JRSS scheme based on the VSS scheme in Section 4. We first modify (1), used in the Asmuth-Bloom secret sharing scheme in Section 2 to

$$\prod_{i=1}^{t} m_i > n {m_0}^2 \prod_{i=1}^{t-1} m_{n-i+1}. \tag{6}$$

We also change the definition of M as $M = \left\lfloor (\prod_{i=1}^{t} m_i)/n \right\rfloor$. The proposed JRSS scheme is given in Figure 5.

– *Dealing Phase:* To jointly share a secret $d \in \mathbb{Z}_{m_0}$ the users do the following:
  1. Each user chooses a secret $d_i \in \mathbb{Z}_{m_0}$ and shares it by using the VSS scheme as follows: He first computes
  $$y^{(i)} = d_i + A_i m_0,$$
  where $y^{(i)} < M = \lfloor (\prod_{i=1}^{t} m_i)/n \rfloor$. Then the secret for the jth user is computed as
  $$y_j^{(i)} = y^{(i)} \bmod m_j.$$
  He sends $y_j^{(i)}$ to user j secretly for all $1 \leq i \leq n$ and broadcasts $(E(y^{(i)}), \mathrm{RngPrf}(E(y^{(i)}), M))$.
  2. After receiving shares the jth user verifies them by using the verification procedure in (5). Let $\mathcal{B}$ be the set of users whose shares are verified correctly. The jth user computes his overall share
  $$y_j = \left(\sum_{i \in \mathcal{B}} y_j^{(i)} \right) \bmod m_j$$
  by using the verified shares.
– *Combiner Phase:* Let S be a coalition of t users gathered to construct the secret.
  1. The share y_i of user $i \in S$ can be verified by the other users in S with the verification equation
  $$g^{y_i} \stackrel{?}{\equiv} \left(\prod_{j \in \mathcal{B}} E(y^{(j)}) \right) \pmod{p_i}. \tag{7}$$
  2. If all shares are valid, the participants obtain the secret
  $$d = \left(\sum_{i \in \mathcal{B}} d_i \right) \bmod m_0$$
  by using the reconstruction procedure described in Section 2.

Fig. 5. CRT-based joint random secret sharing scheme.

5.1 Analysis of the Proposed JRSS Scheme

Correctness Observe that when all users behave honestly, the JRSS scheme works correctly. Let $y = \sum_{i\in\mathcal{B}} y^{(i)}$. It is easy to see that $y < \prod_{i=1}^{t} m_i$, since $y^{(i)} < M$ for all $i \in \mathcal{B}$, where $|\mathcal{B}| \leq n$ and $M = \left\lfloor (\prod_{i=1}^{t} m_i)/n \right\rfloor$. One can see that $y_j = y \bmod m_j$ for all $j \in \mathcal{B}$ by checking

$$\begin{aligned} y \bmod m_j &= \left(\sum_{i\in\mathcal{B}} y^{(i)} \right) \bmod m_j \\ &= \left(\sum_{i\in\mathcal{B}} y_j^{(i)} \right) \bmod m_j \\ &= y_j \bmod m_j = y_j. \end{aligned}$$

Hence, each y_i satisfies $y_i = y \bmod m_i$ and $y < \prod_{i=1}^{t} m_i$; y can be constructed with t shares.

For correctness of the verification procedure in (7), one can observe that

$$\begin{aligned} \left(\prod_{i\in\mathcal{B}} E(y^{(i)}) \right) \bmod p_i &= g^{\sum_{i\in\mathcal{B}} y^{(i)}} \bmod p_i \\ &= g^{y} \bmod p_i = {g_i}^{y} \bmod p_i \\ &= {g_i}^{y_i} \bmod p_i. \end{aligned}$$

Hence, when every user behaves honestly, the proposed JRSS scheme works correctly.

Security We will show that no coalition of size smaller than t can obtain any information about the secret.

Theorem 4. *For a passive adversary with $t-1$ shares in the JRSS scheme, every candidate for the secret is equally likely. I.e., the probabilities $\Pr(d = d')$ and $\Pr(d = d'')$ are approximately equal for all $d', d'' \in \mathbb{Z}_{m_0}$.*

Proof. Suppose the adversary corrupts $t-1$ users and just observes the inputs and outputs of the corrupted users without controlling their actions, i.e., the adversary is honest in user actions but curious about the secret. Let S' be the coalition of the users corrupted by the adversary. Shares are obtained when each user shares his partial secret d_i, i.e., the

adversary will obtain $t-1$ share for each d_i. We will prove that the probabilities that $d_i = d_i'$ and $d = d_i''$ are almost equal for two secret candidates $d_i', d_i'' \in \mathbb{Z}_{m_0}$.

We already proved that the Asmuth-Bloom SSS described in Section 2 is perfect with equation (1). By using the shares of S', the adversary can compute $y'^{(i)} = y^{(i)} \bmod M_{S'}$. But even with these shares, there are $\frac{M}{M_{S'}}$ consistent $y^{(i)}$s that are smaller than M and congruent to $y'^{(i)}$ modulo $M_{S'}$. By replacing (1) with (6) and changing the definition of M to $\lfloor (\prod_{i=1}^{t} m_i)/n \rfloor$, the value of the ratio

$$\frac{M}{M_{S'}} > \frac{M}{\prod_{i=1}^{t-1} m_{n-i+1}} \approx \frac{\prod_{i=1}^{t} m_i}{n \prod_{i=1}^{t-1} m_{n-i+1}}$$

is greater than ${m_0}^2$. Hence, even with $t-1$ shares, there are still ${m_0}^2$ candidates for each $y^{(i)}$ which is used to share the secret d_i. Since $\gcd(m_0, M_{S'}) = 1$, there are approximately m_0 $y^{(i)}$s, consistent with a secret candidate d_i'. Hence, for a secret candidate d_i' the probability that $d_i = d_i'$ is approximately equal to $\frac{1}{m_0}$ and the perfectness of the scheme is preserved.

Aside from the shares, the only other information the adversary can observe are the commitments and range-proofs. Given that the discrete logarithm problem is hard and Cao et al.'s range proof scheme is secure, the proposed JRSS scheme is also computationally secure. □

A corrupted user cannot cheat in the JRSS scheme without being detected. Since we are using a VSS scheme, while user i is sharing his partial secret d_i, the conditions of the Asmuth-Bloom SSS must be satisfied as proved in Theorem 2. Furthermore, if user i sends an incorrect share in the combiner phase, the verification equation (7) will not hold. As a result, we can say that the JRSS scheme is secure for up to $t-1$ corrupted users and no user can cheat in any phase of the scheme.

6 Proactive Secret Sharing

Proactive SSS protocols enable shareholders to jointly renew their shares without changing and revealing the long-term secret. By this feature, the shares compromised by an adversary can be made obsolete with the update process. Proactive secret sharing schemes have been investigated by several researchers and various schemes are proposed in the literature [7, 8, 14, 15, 21].

In a proactive SSS, at the end of a certain time period τ, first the corrupted users are identified in the detection procedure and then all such

users are rebooted, i.e., the adversary is removed from the computers of the users and all of the past information is erased. Subsequently, the new shares of the rebooted users are recovered in the recovery procedure. Then, the shares of the remaining users are refreshed by a renewal procedure. At the end of this protocol, the long-term secret remains the same although the shares of the users for the next period are renewed. This update phase is repeated at the end of each time period.

Adversary model: We assume the *mobile adversary model* of Herzberg et al. [15]. In this model, the adversary is allowed to move among players and can corrupt users at any time. The only restriction on the adversary is that he cannot corrupt more than $t-1$ distinct users in a time period where $t < n/2$ is the threshold of the secret sharing scheme and n is the number of users. If a user is corrupted during the course of the update phase executed at the end of time period τ, he is considered corrupted for both time periods τ and $\tau+1$. With this model, Herzberg at al. proposed an efficient and secure Shamir-based proactive SSS.

We use $\mathcal{A}_\tau$ to denote the set of users where adversarial behavior is detected in their actions in time period τ, and $\mathcal{B}_\tau$ to denote the set of remaining users. A user is disqualified and becomes a member of $\mathcal{A}_\tau$ if his share is inconsistent with the secret or if he tries to cheat in the share-update phase. Each disqualified user is rebooted at the beginning of the update phase, i.e., all the information including the secret share is erased, hence, the new share of a corrupted user must be recovered by the users in $\mathcal{B}_\tau$.

6.1 CRT-based Proactive Secret Sharing Scheme

To obtain a proactive SSS, we first modify equation (1), used in the Asmuth-Bloom secret sharing scheme in Section 2 to

$$\prod_{i=1}^{t} m_i > n{m_0}^3 \prod_{i=1}^{t-1} m_{n-i+1}. \tag{8}$$

We also change the definition of M to

$$M = \left\lfloor \frac{\prod_{i=1}^{t} m_i}{n m_0} \right\rfloor.$$

In the proposed proactive sharing scheme, first a secret is shared by a dealer as described in Figure 6 by using the VSS scheme proposed in Section 4.

Dealer Phase: The dealer shares a secret $d \in \mathbb{Z}_{m_0}$ by equation (8) and M, using the VSS scheme proposed in Section 4. Similar to the VSS, let $p_i = 2m_i + 1$ be a prime for $0 \leq i \leq n$. As in the VSS, $y = d + Am_0$ is an integer smaller than M, $y_i = y \bmod m_i$ is the share of user i, and the commitment $E(y) = g^y \bmod PN$ is broadcast with the range proof for y.

Fig. 6. CRT-based proactive SSS: The dealer phase.

The share update phase executed at the end of a time period τ has three phases:

1. *Detection*: If a user j is detected as corrupted he is rebooted and becomes a member of $\mathcal{A}_\tau$.
2. *Share Recovery*: The share of each rebooted user $j \in \mathcal{A}_\tau$ is reconstructed by the remaining users in $\mathcal{B}_\tau$.
3. *Share Renewal*: The users jointly share 0 by setting $d_i = 0$ for $1 \leq i \leq n$ in the JRSS protocol of Figure 5. Then they add these renewal shares to their previous ones and obtain their new shares.

Detection If a user does not participate in a protocol where he is an active member, or if the information he sends does not verify correctly, we say that an adversarial action is detected. However, when an adversary *silently* corrupts some users by only modifying their local data, we cannot detect such inconsistencies after the adversary detaches himself from the user. Hence, to protect proactiveness, we need to periodically test the correctness of users' local data. To do this, we use the $E(y)$ values each player holds, as in Figure 7.

Note that $t < n/2$, hence, there are at least t honest users who have not been silently or actively corrupted by an adversary. Since the views of all honest users are the same, an inconsistent value will be detected by at least t users.

Share Recovery At the beginning of the update phase, the shares of the rebooted users will still be missing. To recover the share of a rebooted user $j \in \mathcal{A}_{\tau-1}$, each user $\mathcal{B}_{\tau-1}$ shares a random multiple of $m_j \in [0, M)$; hence, the sum of these shared values, which will be denoted by z, will be a multiple of m_j. This ensures that when the users in $\mathcal{B}_{\tau-1}$ add their new shares to their old ones for y, they obtain a share for an integer $y' = y + z$, where $y_j \equiv y \equiv y' \pmod{m_j}$. After obtaining a share for y', each user in $\mathcal{B}_{\tau-1}$ sends it to the jth user via a private channel so the jth user can

1. Let $E^{(i)}(y)$ denotes the local copy of the commitment $E(y)$ for user i. For detection, the ith user first checks the validity of his share by using the equation

$$y_i \stackrel{?}{=} E^{(i)}(y) \bmod p_i.$$

   If the equation is not satisfied he broadcasts an inconsistency warning, reboots himself and becomes a member of $\mathcal{A}_\tau$.
2. If the equation is satisfied, after receiving each $E^{(i)}(y)$ for $i \in \mathcal{B}_\tau$, the jth user tries to find an inconsistency

$$E^{(i_1)}(y) \neq E^{(i_2)}(y)$$

   for $i_1, i_2 \in \mathcal{B}_\tau$ and he broadcasts an accusation either for user i_1, i_2 or both, depending which is inconsistent with at least t $E^{(i)}(y)$s.
3. If at least t accusations are broadcasted for a user i, he is rebooted with the users who did not accuse him.

Fig. 7. CRT-based proactive SSS: The detection procedure.

compute y' and hence, y_j. This share recovery procedure is described in detail in Figure 8.

Share Renewal After the recovery procedure, each user $i \in \mathcal{B}_\tau$ has a share $y_i = y \bmod m_i$, where $y = d + Am_0 < M$. The idea used in this phase is similar to the one in the JRSS scheme described in Section 5. Instead of a random secret, each user shares 0 by some $y^{(i)} \equiv 0 \pmod{m_0}$, and $y^{(i)} \in [0, M)$, hence, the overall shared value will be a multiple of m_0. So, when a user adds his renewal shares with his old share y_i, he obtains a new share y_i'', which is a residue of an integer $y'' \in \left[0, \prod_{i=1}^{t} m_i\right)$ such that $d = y'' \bmod m_0$. In the next time period, y'' will be the new y. The share renewal procedure is described in Figure 9.

Note that y will remain less than $\prod_{i=1}^{t} m_i$ provided that m_0, which is a very large integer, is greater than the number of times the update procedure is applied, since $y^{(i)} < M$, $|\mathcal{B}_\tau| < n$, and

$$m_0 \sum_{i \in \mathcal{B}_\tau} y^{(i)} < nm_0 M = nm_0 \left\lfloor \frac{\prod_{i=1}^{t} m_i}{nm_0} \right\rfloor \leq \prod_{i=1}^{t} m_i.$$

6.2 Security

Assume that the update phase is started at the end of the τth time period. In the share recovery procedure, the participants share z, which is equivalent to 0 modulo m_j for a rebooted user j. Also, in the share

1. To recover the share of a compromised user $j \in \mathcal{A}_\tau$ each user $i \in \mathcal{B}_\tau$ chooses an integer
$$y^{(i,j)} = A_i m_j,$$
where A_i is a random integer such that $y^{(i,j)} < M$ and then shares it among $\mathcal{B}_\tau$ by computing the secrets
$$y_k^{(i,j)} = y^{(i,j)} \bmod m_k$$
for each user $k \in \mathcal{B}_\tau$. He sends $y_k^{(i,j)}$ to user k secretly and broadcasts $(E(y^{(i,j)}), \mathrm{RngPrf}(E(y^{(i,j)}), M))$.
2. After receiving the shares $y_k^{(i,j)}$ from each $i \in \mathcal{B}_\tau$, the kth player verifies them by using the verification procedure in equation (5). In addition, each commitment is checked by $E(y^{(i,j)}) \bmod p_j \stackrel{?}{=} 1$. If a verification equation does not hold for a user, he is disqualified.
3. The kth user computes his ephemeral secret
$$y'_k = \left(y_k + \sum_{i \in \mathcal{B}_\tau} y_k^{(i,j)} \right) \bmod m_k$$
and sends it to user j secretly.
4. After receiving the shares, y'_ks, from each user $k \in \mathcal{B}_\tau$, the jth player verifies them by using the verification procedure in equation (5) for y'. The verification data for $y' = y + \sum_{i \in \mathcal{B}_\tau} y^{(i,j)}$ can be computed as
$$E(y') = E(y) \prod_{i \in \mathcal{B}_\tau} E(y^{(i,j)}).$$
If a verification equation does not hold for a user, he is disqualified for time period τ and $\tau + 1$.
5. The compromised user jth computes
$$y' = \sum_{k \in \mathcal{B}_\tau} y'_k M'_{\mathcal{B}_\tau, k} M_{\mathcal{B}_\tau \setminus \{k\}} \bmod M_{\mathcal{B}_\tau}$$
where $M_{\mathcal{B}_\tau \setminus \{k\}} M'_{\mathcal{B}_\tau, k} \equiv 1 \pmod{m_k}$. He computes his share as $y_j = y' \bmod m_j$.

Fig. 8. CRT-based proactive SSS: The share recovery procedure.

1. Each user i in $\mathcal{B}_\tau$ shares 0 by first computing

$$y^{(i)} = A_i m_0$$

where A_i is a random integer such that $y^{(i)} < M$. Then he computes the share for user $j \in \mathcal{B}_\tau$ as

$$y_j^{(i)} = y^{(i)} \bmod m_j.$$

He sends $y_j^{(i)}$ to each user j secretly and broadcasts $\Big(E(y^{(i)}), \text{RngPrf}(E(y^{(i)}), M)\Big)$.

2. After receiving the shares $y_j^{(i)}$ for $i \in \mathcal{B}_\tau$, the jth user verifies them by using the verification procedure in equation (5). Besides, each commitment is checked for $E(y^{(i)}) \stackrel{?}{\equiv} 1 \pmod{p_0}$. If the verification equation of user i does not hold, he is disqualified, i.e., he is moved from $\mathcal{B}_\tau$ to $\mathcal{A}_\tau$ and $\mathcal{A}_{\tau+1}$. The jth user updates his overall share as

$$y_j'' = \left(y_j + \sum_{i \in \mathcal{B}_\tau} y_j^{(i)} \right) \bmod m_j.$$

3. The new verification data for $y'' = y + \sum_{i \in \mathcal{B}_\tau} y^{(i)}$ is computed as

$$E(y'') = E(y) \prod_{i \in \mathcal{B}_\tau} E(y^{(i)}).$$

Fig. 9. CRT-based proactive SSS: The share renewal procedure.

renewal procedure, the participants jointly share $y'' - y$, where y'' is the new shared integer and y is the previous one. With the following theorems, we will prove that the perfectness condition is preserved in the dealing phase and the shared integers in the next two phases are not computable by a passive adversary.

Theorem 5. *The modified secret sharing scheme with the new*

$$M = \left\lfloor \frac{\prod_{i=1}^{t} m_i}{nm_0} \right\rfloor$$

and equation (8) is perfect in the sense that the probabilities $\Pr(d = d')$ *and* $\Pr(d = d'')$ *are approximately equal for all* $d', d'' \in \mathbb{Z}_{m_0}$.

Proof. Let S' be a corrupted coalition of $t-1$ users. For perfectness, we need to check the value of $M/M_{S'}$ which is

$$\frac{M}{M_{S'}} > \frac{\prod_{i=1}^{t} m_i}{nm_0 \prod_{i=n-t+1}^{n} m_i} > {m_0}^2$$

due to equation (8). Similar to the proof of Theorem 1, we can say that the perfectness condition is preserved. □

Lemma 1. *For a passive adversary that has corrupted* $t-2$ *users in the recovery procedure, there are at least* ${m_0}^2$ *possible candidates for each* $y^{(i,j)}$ *used.*

Proof. Let S' be the set of $t-2$ corrupted users. In the recovery procedure, first each user i shares a $y^{(i,j)}$ where adversary has $t-2$ shares for each of them, i.e., $y_k^{(i,j)}$ for $k \in S' \setminus \{j\}$. So, for a shared value $y^{(i,j)}$, the adversary can only have the shares of S' and additional information that $y^{(i,j)} \equiv 0 \pmod{m_j}$. Hence, although the adversary can obtain $y^{(i,j)} \bmod M_{S'}$, there are still

$$\frac{M}{M_{S'}} = \frac{\prod_{i=1}^{t} m_i}{nm_0 M_{S'}} > {m_0}^2$$

candidates for $y^{(i,j)}$ since $y^{(i,j)} < M$. □

Lemma 2. *Let* j *be the rebooted user whose share is being recovered in the recovery procedure. For a passive adversary that has corrupted* $t-1$ *users including* j*, there are at least* ${m_0}^2$ *possible values for each uncompromised* y_i*, the secret share of user* i.

Proof. In Step 3 of the recovery procedure described in Figure 8, an honest user i computes his ephemeral secret

$$y_i' = \left(y_i + \sum_{k \in \mathcal{B}_\tau} y_i^{(k,j)} \right) \bmod m_i$$

and sends it to user j, who has been corrupted by the adversary. Note that y_i is masked with $y_i^{(k,j)}$s, where

$$y_i^{(k,j)} = y^{(k,j)} \bmod m_i,$$

and due to Lemma 1, from the adversary's point of view there are at least ${m_0}^2$ candidates, with the same remainder in modulo $M_{S'}$, for each $y^{(k,j)}$. Hence, there are at least ${m_0}^2$ candidates for each $y_i^{(k,j)}$ since $(m_i, M_{S'}) = 1$. This also proves that there are at least ${m_0}^2$ candidates for $y_i = d + Am_0 \bmod m_i$. □

Theorem 6. *For a passive adversary in the recovery procedure, two secrets $d', d'' \in \mathbb{Z}_{m_0}$ are equally likely.*

Proof. Let j be the rebooted user whose share is being recovered. Since user j was corrupted in time period τ, the adversary can have at most $t-2$ additional users corrupted in the recovery procedure. Beside these $t-2$ users, the adversary is allowed to corrupt only the jth user again. Due to the mobile adversary model this is the best the adversary can do. Let S' be the set of $t-1$ corrupted users including user j.

From Lemma 2, we know that there are at least ${m_0}^2$ candidates for $y_i = d + Am_0 \bmod m_i$. Since $\gcd(m_0, m_i) = 1$ these ${m_0}^2$ candidates cover all m_0 secret candidates at least m_0 times. Hence, all secret candidates are equally likely. □

Lemma 3. *For a passive adversary that has corrupted $t-1$ users in the share renewal procedure, there are at least m_0 possible candidates for each $y^{(i)}$.*

Proof. Assume that the adversary corrupted $t-1$ users in time period τ without being detected. Let S' denote this set of corrupted users. Considering $M \approx (\prod_{i=1}^{t} m_i)/(nm_0)$, we know that

$$M = \frac{\prod_{i=1}^{t} m_i}{nm_0} > {m_0}^2 \prod_{i=1}^{t-1} m_{n-i+1} > {m_0}^2 M_{S'}$$

due to equation (8).

For a shared value $y^{(i)} = A_i m_0$, the adversary will know that $y^{(i)} \equiv 0 \pmod{m_0}$. Since $y^{(i)} < M$, there are $\frac{M}{m_0} > m_0 M_{S'}$ candidates for $y^{(i)} \equiv 0 \pmod{m_0}$. In addition, the adversary can compute $y^{(i)} \bmod M_{S'}$ by using the $t-1$ shares he obtained for $y^{(i)}$. But, there are still $\frac{M}{m_0 M_{S'}} > m_0$ candidates for $y^{(i)}$. □

Theorem 7. *An adversary with $t-1$ corrupted shares in the share renewal procedure cannot compute a new share in time period $\tau+1$ from an old share he has from time period τ.*

Proof. In Step 2 of the renewal procedure described in Figure 9, the jth user updates his overall share as

$$y_j'' = \left(y_j + \sum_{i \in \mathcal{B}_\tau} y_j^{(i)} \right) \bmod m_j.$$

From Lemma 3, there are at least m_0 possible candidates for $y^{(i)}$ and $\gcd(m_j, m_0 M_{S'}) = 1$. Hence, there are at least m_0 possible candidates for each

$$y_j^{(i)} = y^{(i)} \bmod m_j.$$

Hence, the adversary cannot compute y_j'' even if he knows the old share y_j. □

By Theorems 6 and 7, the proposed PSS scheme is secure against passive adversaries in Herzberg et al.'s mobile adversary model.

Security Analysis for an Active Adversary As proved in Sections 4.3 and 5.1, in the proposed VSS and JRSS schemes, if a user tries to cheat by sending inconsistent information he will be detected easily since some verification equations will not hold.

In the share renewal and share recovery phases, we use modified versions of the JRSS scheme, where the shared values are congruent to 0 with respect to moduli m_0 and m_j, respectively, and where user j has been rebooted before the execution of the update phase. To verify these restrictions, in the second step of Figure 9, a user j checks his share for $y^{(i)}$ by using the verification procedure in equation (5) and checks

$$E(y^{(i)}) \stackrel{?}{\equiv} 1 \pmod{p_0}.$$

In the second step of Figure 8, a user k also verifies his share for $y^{(i,j)}$ by using the verification procedure in equation 5 and checks

$$E(y^{(i,j)}) \stackrel{?}{\equiv} 1 \pmod{p_j}.$$

Note that the other restrictions are also verified since they are automatically checked by the proposed VSS scheme. Therefore, an active adversary cannot send inconsistent data without being detected.

7 Practicality and Efficiency of the Schemes

If both p and $2p+1$ are prime numbers then p is called a Sophie Germain prime. It is believed that the number of Sophie Germain primes is infinite and due to the conjecture of Hardy and Littlewood [13], for sufficiently large N the number of Sophie Germain primes less than N is

$$2C \int_2^N \frac{dx}{\log x \log 2x} \approx \frac{2CN}{(\ln N)^2}, \tag{9}$$

where $C \approx 0.66$ is the twin prime constant. The accuracy of the conjecture and the ratio can be seen in Table 2.

N	Actual	Integral	Ratio
1,000,000	7746	7811	6917
10,000,000	56032	56128	50822
100,000,000	423140	423295	389107
1,000,000,000	3308859	3307888	3074425
10,000,000,000	26569515	26568824	24902848
100,000,000,000	218116524	218116102	205808662

Table 2. Number of Sophie Germain primes less than N [5]. Second column is the actual number of Sophie Germain primes less than N. Third and fourth columns are the integral and ratio approximations in the left and right side of (9), respectively.

For the proposed VSS, JRSS, and PSS schemes, a sequence $m_1 < m_2 < \cdots < m_n$ consisting of n Sophie Germain primes is needed. And for the security issues, this sequence must also satisfy (1) for the VSS scheme. Let us assume that m_0, the number of secret candidates, is a k-bit prime. From (1), first, each m_i must be at least a $2k$-bit Sophie Germain prime. We know that such primes exist since the number of Sophie Germain primes is infinite. Second, we need to argue that we can find a Sophie

Germain sequence for every t and k such that the product of t smallest number in the sequence is larger than the product of $t-1$ largest ones and ${m_0}^2$. Note that the Hardy-Littlewood conjecture says that the density of the Sophie Germain primes less than N is proportional with $1/(\ln N)^2$, where the prime number theorem says that the density of primes less than N is proportional with $1/(\ln N)$. Hence, considering $N \gg \ln N$, finding an Asmuth-Bloom sequence with Sophie Germain primes satisfying (1) should not be much harder than finding such a sequence with ordinary primes.

An informal analysis of the existence of a desired sequence and the information rate of the proposed schemes can be given as follows: Let m_0 be a k-bit prime. Considering n (the number of participants) is in the order of hundreds or thousands, from (9), the number of $2k$-bit Sophie Germain primes is

$$\frac{2C2^{2k+1}}{(\ln 2^{2k+1})^2} - \frac{2C2^{2k}}{(\ln 2^{2k})^2} = \frac{C2^{2k+1}}{(\ln 2)^2}\left(\frac{2}{(2k+1)^2} - \frac{1}{(2k)^2}\right) \gg n.$$

Let m_1 be a $2k$-bit Sophie Germain prime and $\ell = \ln m_1$. Let m_i be the $i-1$st Sophie Germain prime after m_1. Due to (9), we can assume that $m_i \approx m_1 + (i-1)\ell^2$. Note that the ratio m_i/m_j for $i < j$ is bounded above by $\left(1 + \frac{n\ell^2}{m_1}\right)$. Hence, the inequality

$$m_1 > \frac{{m_0}^2 \prod_{i=1}^{t-1} m_{n-i+1}}{\prod_{i=1}^{t-1} m_{i+1}}$$

is satisfied when

$$m_1 > {m_0}^2 \left(1 + \frac{n\ell^2}{m_1}\right)^{t-1}.$$

Since $m_1 \gg n\ell^2$ and $m_1 \gg t$, we can choose $m_1 \approx {m_0}^2$, and the information rate of the VSS scheme becomes $|m_0|/|m_n| \approx |m_0|/|{m_0}^2 + 4n(\ln m_0)^2| \approx 1/2$. A similar analysis can be devised for the JRSS and PSS schemes. For the JRSS scheme, (1) is replaced with (6). So,

$$m_1 > n{m_0}^2 \left(1 + \frac{n\ell^2}{m_1}\right)^{t-1}.$$

And for the PSS scheme, (1) is replaced with (8), hence,

$$m_1 > n{m_0}^3 \left(1 + \frac{n\ell^2}{m_1}\right)^{t-1}.$$

So the information rate for the JRSS and PSS schemes are,

$$\frac{|m_0|}{|nm_0{}^2 + 4n(\ln m_0)^2|} \approx \frac{1}{2} \text{ and } \frac{|m_0|}{|nm_0{}^3 + 4n(\ln m_0)^2|} \approx \frac{1}{3},$$

respectively.

Although the proposed schemes are not ideal, they are practical since, the information rates for the VSS, JRSS, and PSS schemes are approximately 1/2, 1/2 and 1/3, respectively. Note that the denominators of these rates are actually the exponents of m_0 in (1), (6), and (8). In [1], the perfectness definition used by Asmuth and Bloom guarantees that an adversary with $t-1$ shares cannot eliminate any secret candidate, i.e., the cardinality of the secret candidate set is not reduced. Note that the perfectness definition used in Theorem 1 and Section 2 is stronger than the one used by Asmuth and Bloom. As Theorem 1 shows, with (1) not only no secret candidate can be eliminated but also the probabilities of each candidate being the secret are almost equal. Instead of (1), Asmuth and Bloom used the following equation:

$$\prod_{i=1}^{t} m_i > m_0 \prod_{i=1}^{t-1} m_{n-i+1}. \tag{10}$$

If (10) is used the VSS and JRSS schemes (the right hand side of (10) is multiplied by n for the JRSS scheme) will still be perfect in the sense that no secret candidate is eliminated in case of an adversary obtains $t-1$ shares. On the other hand, the information rates of these schemes will be approximately 1 and the schemes would be almost ideal. For the PSS scheme, if we use $m_0{}^2$ instead of $m_0{}^3$ in the left of (8), i.e.,

$$\prod_{i=1}^{t} m_i > nm_0{}^2 \prod_{i=1}^{t-1} m_{n-i+1},$$

although Theorems 5 and 8 can be rewritten with the weaker perfectness definition, in Lemma 9, we cannot guarantee that there are m_0 possible candidates for each $y^{(i)}$ since the adversary also knows that $y^{(i)} \equiv 0 \mod m$. Hence, we cannot use the same trick to obtain a smaller information rate for the PSS scheme.

8 Conclusion

In this paper, we proposed a practical CRT-based verifiable secret sharing scheme. We showed that previous solutions for this problem did not

guarantee the consistency of the shares. We also put forth a secure JRSS scheme based on Asmuth-Bloom secret sharing as a naive application of the VSS scheme. Additionally, by extending the ideas used in the VSS and JRSS schemes, we proposed a PSS scheme where the shares of the users are periodically renewed without changing the long-term secret. To the best of our knowledge, the proposed schemes are the first CRT-based secure VSS, JRSS, and PSS schemes in the literature.

References

1. C. Asmuth and J. Bloom. A modular approach to key safeguarding. *IEEE Trans. Information Theory*, 29(2):208–210, 1983.
2. G. Blakley. Safeguarding cryptographic keys. In *Proc. of AFIPS National Computer Conference*, 1979.
3. D. Boneh and M. Franklin. Efficient generation of shared RSA keys. *Journal of the ACM*, 48(4):702–722, 2001.
4. F. Boudot. Efficient proofs that a committed number lies in an interval. In *Proc. of EUROCRYPT'2000*, volume 1807, pages 431–444. Springer-Berlin, 2000.
5. C. K. Caldwell. An amazing prime heuristic, November, 2000. http://www.utm.edu/~caldwell/preprints/Heuristics.pdf.
6. Z. Cao and L. Liu. Boudot's range-bounded commitment scheme revisited. In *Proc. of the ICICS'07*, volume 4861 of *LNCS*, pages 230–238, 2007.
7. P. D'Arco and D. Stinson. On unconditionally secure robust distributed key distribution centers. In *Proc. of ASIACRYPT'02*, volume 2501, pages 346–363. Springer-Verlag, 2002.
8. P. D'Arco and D. Stinson. On unconditionally secure proactive secret sharing scheme and distributed key distribution centers, May 2002.
9. P. Feldman. A practical scheme for non-interactive verifiable secret sharing. In *Proc. of FOCS'87*, pages 427–437. IEEE, 1987.
10. Y. Frankel, P. D. MacKenzie, and M. Yung. Robust efficient distributed RSA-Key generation. In *Proc. of STOC'98*, pages 663–672. ACM Press, New York, 1998.
11. E. Fujisaki and T. Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations. In *Proc. of CRYPTO'97*, volume 1294 of *LNCS*, pages 16–30, London, UK, 1997. Springer-Verlag.
12. R. Gennaro, S.Jarecki, H. Krawczyk, and T. Rabin. Robust threshold DSS signatures. *Information and Computation*, 164(1):54–84, 2001.
13. G. H. Hardy and J. E. Littlewood. 'Some problems of partition numerorum'; III: On the expression of a number as a sum of primes. *Acta Mathematica*, 44:1–70, 1922.
14. A. Herzberg, M. Jakobsson, S. Jarecki, H. Krawczyk, and M. Yung. Proactive public key and signature systems. In *ACM'97 - Computer and Communication Security*, pages 100–110. ACM, 1997.
15. A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung. Proactive secret sharing or: How to cope with perpetual leakage. In *Proc. of CRYPTO'95*, volume 963 of *LNCS*, pages 339–352, London, UK, 1995. Springer-Verlag.
16. S. Iftene. Secret sharing schemes with applications in security protocols. Technical Report TR 07-01, http://thor.info.uaic.ro/~tr/tr07-01.pdf, University Alexandru Ioan Cuza of Iaşi, Faculty of Computer Science, January, 2007.

17. I. Ingemarsson and G. J. Simmons. A protocol to set up shared secret schemes without the assistance of a mutually trusted party. In *Proc. of EUROCRYPT'91*, volume 547 of *LNCS*, pages 266–282. Springer-Verlag, 1990.
18. W. A. Jackson, K. M. Martin, and C. M. OKeefe. Mutually trusted authority-free secret sharing schemes. *Journal of Cryptology*, 10(4):261–289, 1997.
19. K. Kaya and A. A. Selçuk. Threshold cryptography based on Asmuth-Bloom secret sharing. *Information Sciences*, 177(19):4148–4160, 2007.
20. M. Mignotte. How to share a secret ? In *Proc. of the Workshop on Cryptography*, volume 149 of *LNCS*, pages 371–375. Springer-Verlag, 1983.
21. V. Nikov, S. Nikova, B. Preneel, and J. Vandewalle. Applying general access structure to proactive secret sharing schemes. In *Proc. of 23rd Symposium on Information Theory in the Benelux*, pages 197–206. Springer-Verlag, 2002.
22. T. P. Pedersen. Distributed provers with applications to undeniable signatures. In *Proc. of EUROCRYPT'91*, volume 547 of *LNCS*, pages 221–242. Springer-Verlag, 1991.
23. T. P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *Proc. of CRYPTO'91*, volume 576 of *LNCS*, pages 129–140, London, UK, 1992. Springer-Verlag.
24. L. Qiong, W. Zhifang, N. Xiamu, and S. Shenghe. A non-interactive modular verifiable secret sharing scheme. In *Proc. of ICCCAS'05*, volume 1, pages 84–87. IEEE, 2005.
25. M. Quisquater, B. Preneel, and J. Vandewalle. On the security of the threshold scheme based on the Chinese Remainder Theorem. In *Proc. of PKC'02*, volume 2274 of *LNCS*, pages 199–210, London, UK, 2002. Springer-Verlag.
26. A. Shamir. How to share a secret? *Communications of the ACM*, 22(11):612–613, 1979.