Cheating Prevention in Linear Secret Sharing

Josef Pieprzyk and Xian-Mo Zhang

Centre for Advanced Computing – Algorithms and Cryptography
Department of Computing
Macquarie University
Sydney , NSW 2109, AUSTRALIA
E-mail: josef,xianmo@ics.mq.edu.au

Abstract. Cheating detection in linear secret sharing is considered. The model of cheating extends the Tompa-Woll attack and includes cheating during multiple (unsuccessful) recovery of the secret. It is shown that shares in most linear schemes can be split into subshares. Subshares can be used by participants to trade perfectness of the scheme with cheating prevention. Evaluation of cheating prevention is given in the context of different strategies applied by cheaters.

Key Words: Cryptography, Secret Sharing, Linear Secret Sharing Schemes, Cheating Detection, Cheating Identification.

1 Introduction

Secret sharing was introduced by Shamir [14] and Blakley [3]. The main purpose of secret sharing is to allow a group of participants to own a secret. The secret is typically divided into shares and each principal has at least one share of the secret. The ownership is collective, i.e. to recover the secret, a big enough subgroup of participants must get together and pool their shares. The collection of all subsets of participants authorised to recover the secret, is called the access structure. Ito, Saito and Nishizeki [9] showed that there is a perfect secret sharing for any access structure. Benaloh and Leichter [2] and Simmons, Jackson and Martin [15] showed alternative constructions for perfect secret sharing.

Tompa and Woll [16] demonstrated that Shamir threshold secret sharing can be subject to cheating by one or more dishonest participants. At the pooling time, instead of valid shares, dishonest participants submit modified shares to the combiner who recovers the matching polynomial. Knowing an invalid secret returned from the combiner, dishonest participants can correct it and recover the valid secret while honest ones are left with the invalid secret. In order to prevent cheating, Tompa and Woll [16] suggested to make x co-ordinates of the polynomial secret.

2 Results Achieved

The work generalises the TW attack for the broader class of linear secret sharing and defines a new repeat recovery attack (RR attack). In the attack, it is assumed

that the currently active group is engaged in precisely two unsuccessful secret recovery attempts. The RR attack works even if the underlying matrix and shares are secret.

Linear secret sharing is introduced and it is shown that shares can be easily and efficiently converted into independent pieces (sub-shares). The model is applicable to many known secret sharing schemes including the Shamir scheme [14], the modular scheme [1], and the Karnin-Greene-Hellman scheme [10]. We demonstrate how the linear secret sharing with sub-shares can be used to combat cheating.

3 Model of Cheating

Consider a group of n participants $\mathcal{P} = \{P_1, \ldots, P_n\}$. Given any GF($q$). Denote $\mathcal{S} = GF(q)$ as the set of secrets, $\mathcal{S}_i$ – the set of shares assigned to the participant P_i, and Γ – an access structure which consists of all subsets of participants who are able to recover the secret by pooling their shares together. Secret sharing scheme is a collection of two algorithms: distribution algorithm and recovery algorithm. The distribution algorithm also called the dealer is a probabilistic algorithm which takes access structure Γ, the secret $s \in \mathcal{S}$ and a random string $r \in \mathcal{R}$ as an input and assigns shares $s_i \in \mathcal{S}_i$ to participants P_i; $i = 1, \ldots, n$. This can be written as: $D_\Gamma : \mathcal{S} \times \mathcal{R} \rightarrow \mathcal{S}_1 \times \mathcal{S}_2 \times \cdots \times \mathcal{S}_n$.

For a given access structure Γ, the recovery algorithm (also called combiner) is a deterministic algorithm and accepts shares from participants and recovers the secret $s \in \mathcal{S}$ only when the currently active collection $\mathcal{A} \in \Gamma$ of participants belongs to the access structure. Otherwise, the algorithm fails with an overwhelming probability. So knowing the active set $\mathcal{A} \subset \Gamma$ and the corresponding set of shares $\mathcal{S}_\mathcal{A}$, the recovery algorithm is $R_\Gamma : 2^{\mathcal{S}_\mathcal{A}} \rightarrow \mathcal{S}$, where $2^{\mathcal{S}_\mathcal{A}}$ stands for the collection of all subsets of shares that exist for the group $\mathcal{A}$.

Threshold schemes constitute a very interesting class of secret sharing schemes. Their access structure is particularly simple as any t out of n participants are able to run successfully the recovery algorithm. The access structure of a (t, n) threshold scheme is $\Gamma = \{\mathcal{A} : |\mathcal{A}| = t\}$, where $|\mathcal{A}|$ stands for cardinality of the set $\mathcal{A}$ and $\mathcal{A} \subset \mathcal{P}$.

Definition 1. *A linear (t, n) threshold scheme is a scheme for which the distribution algorithm has the form*

$$(s_1, \ldots, s_n) = (r_1, \ldots, r_t)A \tag{1}$$

where A is a $t \times n$ matrix with entries from GF(q), whose any $(t \times t)$ matrix is nonsingular, and each column is assigned to each share, i.e. $s_i \leftrightarrow \eta_i$ where η_i is a column vector of A. The vector $r = (r_1, \ldots, r_t)$ is selected at random so $r_i \in_R GF(q)$ for $i = 1, \ldots, t$. The secret $s = \sum_{i=1}^{t} r_i$. Shares s_i are secret and known to their owners P_i while the matrix A is public. The recovery algorithm collects t shares and a corresponding $(t \times t)$ matrix and recovers the secret by solving an appropriate collection of linear equations.

3.1 The TW Attack on Linear Secret Sharing

Tompa and Woll [16] showed that a dishonest participant can modify their share in such a way that the secret can be recovered only by the cheater leaving others with an invalid value. The attack was shown for the Shamir scheme and can be easily modified for any linear scheme. For the sake of clarity, assume that participants use the linear (t, n) threshold scheme whose distribution algorithm is defined by Equation (1). We also assume that the $(n \times t)$ matrix A is publicly known. In Shamir schemes, this translates into the assumption that x co-ordinates of underlying polynomial $f(x)$ assigned to participants are public. Let $\mathcal{A} = \{P_1, \ldots, P_t\}$ be the currently active group of participants who wish to reveal the secret and P_1 be a cheater.

During the pooling time, each participant P_i; $i \neq 1$, submits their correct share s_i. The cheater P_1 gives $\tilde{s}_1 = s_1 - \alpha$ where $\alpha \in_R GF(q)$ is a random integer. The recovery algorithm takes shares and solves

$$(r_1, r_2, \ldots, r_t) = (\tilde{s}_1, s_2, \ldots, s_t)A_{\mathcal{A}}^{-1} \tag{2}$$

where $A_{\mathcal{A}}$ is a $t \times t$ matrix derived from A by removing all columns for participants not in $\mathcal{A}$. By design, the matrix $A_{\mathcal{A}}$ is nonsingular so the vectors $(r_1, \ldots, r_t)$ have a unique solution. Let it be $(\tilde{r}_1, \ldots, \tilde{r}_n)$. The recovery algorithm outputs an invalid secret

$$\tilde{s} = \sum_{i=1}^{t} \tilde{r}_i.$$

We follow Tompa and Woll and claim that P_1 can recover the valid secret while others are left with the invalid one $\tilde{s}$. To do so the cheater must know the currently active group (the matrix $A_{\mathcal{A}}$). Note that

$$\begin{aligned}(\tilde{r}_1, \tilde{r}_2, \ldots, \tilde{r}_t) &= (s_1 - \alpha, s_2, \ldots, s_t)A_{\mathcal{A}}^{-1} \\ &= (s_1, s_2, \ldots, s_t)A_{\mathcal{A}}^{-1} - (\alpha, 0, \ldots, 0)A_{\mathcal{A}}^{-1}\end{aligned}$$

The cheater can easily compute an error vector $\Delta = (\delta_1, \ldots, \delta_t)$ caused by the modification of his share by α. So if

$$(\delta_1, \delta_2, \ldots, \delta_t) = (\alpha, 0, \ldots, 0)A_{\mathcal{A}}^{-1}$$

then the cheater can recover the valid secret

$$s = \tilde{s} + \sum_{i=1}^{t} \delta_i = \sum_{i=1}^{t} (\tilde{r}_i + \delta_i)$$

by adding the correction $\Delta = (\delta_1, \ldots, \delta_t)$. Note that even when the recovery algorithm outputs the vector $\tilde{r} = (\tilde{r}_1, \ldots, \tilde{r}_t)$, none of the active participants is able to detect cheating as $\tilde{r}A_{\mathcal{A}}$ gives the correct share for all active (and honest) participants. The modification done by the cheater is obviously undetectable by honest participants.

A secret sharing may offer a different degree of protection against cheaters applying the TW attack. In general, a cheating participant P_1 can try to achieve the following goals:

– G1 – the cheater wants to recover the valid secret while the honest participants are unable to detect cheating,
– G2 – the cheater wants to recover the valid secret while the honest participants are able to detect cheating.

3.2 Practicality of One-time Recovery

In the unconditionally secure setting, it is customary to assume that secret sharing is designed for a single recovery of the secret. We argue that this assumption does not hold in many practical situations. In particular, consider the following arguments.

Concurrent recovery. Given a group holding a secret using (t, n) where t is smaller than $\frac{n}{2}$. The necessity of recovery of the secret is in many cases triggered by an event that is observed independently by all members of the group (say, a stock exchange crash). It should be no surprise to see more than one subgroup attempting to recover the secret roughly at the same time. There is no facility built in secret sharing to prevent multiple key recovery. Even if it was one, it would mean that recovery would go ahead only after checking whether the secret had not been reconstructed by any subgroup. This is clearly unreasonable and it will not work in cases when there some members of the group are not contactable or simply ceased to exist.

Proxy recovery. Multiple recovery can be put into a good use in the case when the combiner is not trusted. Consider first the case where the TW attack can be actually useful in preventing the combiner from leaking the recovered secret to outsiders. The combiner is normally implemented as a computer program which is run under a watchful eye of a trusted operating system and using underlying secure communication infrastructure to collect shares from participants. Even if the participants are aware about security gaps in the implementation of the combiner, they may still be tempted to use it as it offers communication and computing facilities that otherwise may not be readily available. Assume that the group delegates a member who is entrusted by the group to act on their behalf. Note that this member is the real combiner who is using untrusted one to quickly and efficiently use the (insecure) infrastructure. All participants submit their shares to the untrusted combiner except the delegated member who submits an incorrect share. The combiner recovers an incorrect secret and communicates it to the group (and leaks it to outsiders). Only the delegated member is able to recover the secret. Note that in the unconditionally secure setting, participants may use the untrusted combiner because of the accessible communication infrastructure (they obviously could use authenticated broadcasting). In the conditionally secure setting, participants may use untrusted combiner as a powerful (and insecure) server. A multiple recovery could be useful if the participants are allowed to lie about their shares by "small" modification of their shares.

Multiple recoveries. Multiple recovery can be used to get rid of combiner altogether and the secret reconstruction can then be seen as a probabilistic game with participants broadcasting their shares distorted by a small "noise". It is

expected that if the entropy of noise is appropriately selected then the participants have advantage over outsiders in finding the secret from many "noisy" recoveries. Note that they know precise values of their individual shares while outsiders do not.

3.3 A Repeat-Recovery Attack

To make the presentation clearer, we assume that the currently active subset of participants is $\mathcal{A} = \{P_1, \ldots, P_t\}$ from which P_1 is a cheater and $P_2, \ldots, P_t$ are honest. To prevent cheating, the dealer has distributed pairs (s_i, η_i) secretly to each participant. In the repeat-recovery (RR) attack, any unsuccessful attempt in secret recovery provides the cheater information about the secret. From a point of view of the cheater, the matrix $A_{\mathcal{A}}$ is seen as

$$(\eta_1, \eta_2, \ldots, \eta_t)$$

where vectors $\eta_2, \ldots, \eta_t$ are unknown. The recovery algorithm collects pairs (s_i, η_i), constructs the matrix $A_{\mathcal{A}}$, computes its inverse $A_{\mathcal{A}}^{-1}$ and computes the secret $s = \sum_{i=1}^{t} r_i$ where

$$(r_1, r_2, \ldots, r_t) = (s_1, s_2, \ldots, s_t)A_{\mathcal{A}}^{-1}$$

The secret is sent back to active participants via secure channels.

In the RR attack, the cheater modifies his share and sends the pair $(s_1 + \alpha_1, A_1)$ where $\alpha_1 \in_R GF(q)$. Honest participants provide their pairs. The recovery algorithm computes

$$\begin{aligned}(r_1^{(1)}, r_2^{(1)}, \ldots, r_t^{(1)}) &= (s_1 + \alpha_1, s_2, \ldots, s_t)A_{\mathcal{A}}^{-1} \\ &= (s_1, s_2, \ldots, s_t)A_{\mathcal{A}}^{-1} + (\alpha_1, 0, \ldots, 0)A_{\mathcal{A}}^{-1}\end{aligned}$$

The invalid secret $s' = \sum_{i=1}^{t} r_i^{(1)}$ is returned to all active participants. Next the cheater publicly acknowledges that he has made a mistake while sending the share and asks for another try. In the second attempt, the cheater modifies his share using $\alpha_2 \in_R GF(q)$ and sends $(s_1 + \alpha_2, A_1)$ to the combiner. The recovery algorithm again computes the vector r which is

$$\begin{aligned}(r_1^{(2)}, r_2^{(2)}, \ldots, r_t^{(2)}) &= (s_1 + \alpha_2, s_2, \ldots, s_t)A_{\mathcal{A}}^{-1} \\ &= (s_1, s_2, \ldots, s_t)A_{\mathcal{A}}^{-1} + (\alpha_2, 0, \ldots, 0)A_{\mathcal{A}}^{-1}\end{aligned}$$

The second invalid secret is $s'' = \sum_{i=1}^{t} r_i^{(2)}$. After getting it, the cheater can write a system of two equations with four known integers: s', s'', α_1 and α_2. The system has the form:

$$\begin{aligned}s' &= \sum_{i=1}^{t} r_i^{(1)} = s + \sum_{i=1}^{t} (\alpha_1, 0, \ldots, 0)\eta_i^* \\ s'' &= \sum_{i=1}^{t} r_i^{(2)} = s + \sum_{i=1}^{t} (\alpha_2, 0, \ldots, 0)\eta_i^*\end{aligned}$$

where η_i^* is the i-th column of the matrix $A_{\mathcal{A}}^{-1}$. If the first equation is multiplied by α_2 and the second by α_1, then the cheater obtains:

$$\alpha_2 s' = \alpha_2 s + \sum_{i=1}^{t} (\alpha_1 \alpha_2, 0, \ldots, 0)\eta_i^*$$

$$\alpha_1 s'' = \alpha_1 s + \sum_{i=1}^{t} (\alpha_1 \alpha_2, 0, \ldots, 0)\eta_i^*$$

The secret is

$$s = \frac{\alpha_2 s' - \alpha_1 s''}{\alpha_2 - \alpha_1}$$

The cheater now knows the secret. The above considerations are summarised in the following theorem.

Theorem 1. *Given (t, n) threshold linear secret sharing with the matrix A. Let each participant be assigned her secret pair (s_i, η_i) where s_i is her share and η_i is the column vector of A assigned to P_i (see Equation (1)). Assume that after having collected t pairs (s_i, η_i), the recovery algorithm returns the secret to all active participants via secure channels. Let P_1 be a cheater who modifies his shares by choosing two random modifications α_1 and α_2 ($\alpha_1 \neq \alpha_2$). Then the cheater is able to recover the secret after two unsuccessful attempts. Honest participants do not have any information about the secret except the information provided by the combiner.*

The only point we have not proved is the last statement. From an honest participant point of view, invalid secrets s' and s'' are random variables controlled by random variables α_1 and α_2, respectively. If α_1 and α_2 are two random variables ($\alpha_1 \neq \alpha_2$) selected from all nonzero elements of $GF(q)$, then so are s' and s''. The only information accessible to honest participants is that the valid secret must be different from both s' and s''.

How we can prevent linear secret sharing against the new attack ? The general rule is *Do not give the cheater another chance.* More precisely, the RR attack works only if the same matrix $A_{\mathcal{A}}$ is used twice by the recovery algorithm. To thwart the RR attack, it is enough to replace a single participant by a new one. This changes the matrix and the cheater is unable to recover the valid secret.

Assume that participants have agreed for multiple recovery of the secret. Again the currently active group is $\mathcal{A} = \{P_1, \ldots, P_\ell\}$ where P_1 is cheating and other participants are honest. The cheater may have the following goals:

- Goal $\mathrm{G1}_{RR}$ – recovery of the valid secret while the cheater does not mind to be identified by the honest participants,
- Goal $\mathrm{G2}_{RR}$ – recovery of the valid secret while the cheater wants to remain unidentified.

3.4 Previous Works

The Tompa-Woll attack put the cheating problem in the spot-light. The suggestion of making x co-ordinates secret does not really address the problem but rather removes the main incentive behind cheating. We are interested in cheating detection in the unconditionally secure setting. Rabin and Ben-Or [13] used a system of linear equations to validate shares before they are passed into the combiner. Carpentieri in [5] constructed a similar scheme but with shorter shares. Carpentieri, De Santis and Vaccaro [6] argued that share expansion is unavoidable to detect cheating. They proved that any (t, n) threshold scheme must have the size of share bigger by $\log \frac{1}{\varepsilon}$ than the size of the secret to detect cheating with the probability better than $1 - \varepsilon$.

Note that all solutions for cheating detection presented in the literature suffer from a dramatic share expansion. So dramatic, in fact, that their practicality is questionable. Note that the Rabin and Ben-Or solution requires $(3n - 2)$ additional elements of the length of the secret per participant. Carpentieri managed to reduce this to $t + 2(n - 1)$ elements where t is the threshold parameter and n is the number of participants in the group. The underlying secret sharing is perfect and unconditionally secure. Some other methods of cheating detection are considered in [4, 7, 8, 12].

The solution we propose does not need any share expansion and in fact, can be applied to any linear secret sharing (including Shamir secret sharing). This claim seems to be in odds with the finding of Carpentieri at al [6]. The solution is built on the observation (see [11]) that it is possible to design Shamir secret sharing with divisible shares. The divisibility of shares allowed the authors of [11] to increase the effective threshold of secret sharing. Here we are using it to trade perfectness of the scheme with the cheating detection. In other words, if participants do not care about cheating, they treat their shares as atomic - the scheme is perfect. If however, they choose to detect cheating, they donate a part of their sub-shares to the combiner leaving some sub-shares to verify the correctness of the returned structure. Note that in our solution, the cheating detection is done individually by active participants after the combiner has returned the secret together with additional information to the participants.

4 Linear Secret Sharing with Sub-shares

Consider an ideal and linear (t, n) threshold scheme and P_i holds the share s_i. Assume that $\mathcal{S} = \mathcal{S}_i = GF(q)$ for $i = 1, \ldots, n$. Shares held by participants are normally atomic – either withheld or given out in their totality. Suppose further that $GF(q) = GF(p^v)$, σ be a root of a primitive polynomial $p(y) = a_0 + a_1 y + \cdots + a_{v-1} y^{v-1} + y^v$ of degree v over $GF(p)$. Then any element in $\tau \in GF(p^v)$ can be expressed as $\tau = b_0 + b_1\sigma + \cdots + b_{v-1}\sigma^{v-1}$ where each $b_j \in GF(p)$. We call the vector $(b_0, b_1, \ldots, b_{v-1})$ the *vector representation* of τ.

Now consider a (t, n) threshold scheme described by Equation (1). This equation can be equivalently presented as

$$(\varepsilon_1, \ldots, \varepsilon_{nv}) = (\rho_1, \ldots, \rho_{vt})B \tag{3}$$

where $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_{nv})$ and $\tau = (\rho_1, \ldots, \rho_{tv})$ are created from vectors $(s_1, \ldots, s_n)$ and $(r_1, \ldots, r_t)$ respectively by replacing each entry by its vector representations, and B is an $tv \times nv$ matrix over $GF(p)$ obtained from the matrix A in the following way. We take the unit vector $\rho^{(i)} = (0, \ldots, 0, 1, 0, \ldots, 0)$ which has all zero co-ordinates except the i-th co-ordinate that is "1". Next we find the vector $r^{(i)}$ such that $r^{(i)}$ and $\rho^{(i)}$ present the same integer, in other words, $r^{(i)} = (r_1, \ldots, r_t)$ where each $r_j \in GF(p^v)$, and $\rho^{(i)} = (\rho_1, \ldots, \rho_{tv})$ where each $\rho_j \in GF(p)$, satisfy $\sum_{j=1}^{t} r_j q^{t-j} = \sum_{j=1}^{tv} \rho_j p^{tv-j}$ where $q = p^v$. Then we compute $s^{(i)} = r^{(i)} A$. The vector $s^{(i)}$ is translated into its equivalent $\varepsilon^{(i)}$ – this is the i-th row of B. If we continue this process for all unit vectors $\rho^{(i)}$; $i = 1, \ldots, vt$, we get explicit form of the matrix B with tv rows and nt columns. So we have proved the following.

Lemma 1. *Given a perfect and linear (t, n) threshold scheme whose shares are computed according to Equation (1) where A is the $t \times n$ matrix over $GF(q) = GF(p^v)$, then shares can be equivalently computed by Equation (3) where B is the $tv \times nv$ matrix over $GF(p)$, obtained from the matrix A.*

We further notice that the matrix B in (3) can also be constructed immediately from the matrix A as follows. Define a $v \times v$ matrix D over $GF(p)$ such that:

$$D = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{v-2} & -a_{v-1} \end{bmatrix} \tag{4}$$

Lemma 2. *Let σ be a root of a primitive polynomial $p(y) = a_0 + a_1 y + \cdots + a_{v-1} y^{v-1} + y^v$ of degree v over $GF(p)$. Then $(c_0 + c_1\sigma + \cdots + c_{v-1}\sigma^{v-1})(b_0 + b_1\sigma + \cdots + b_{v-1}\sigma^{v-1}) = (d_0 + d_1\sigma + \cdots + d_{v-1}\sigma^{v-1})$, where each b_j, c_j and d_j are elements in $GF(p)$, if and only if*

$$(c_0, c_1, \ldots, c_{v-1})(b_0 I_v + b_1 D + \cdots + b_{v-1} D^{v-1}) = (d_0, d_1, \ldots, d_{v-1})$$

where D has been defined in (4) and I_v denotes the $v \times v$ identity matrix.

We reconsider a (t, n) threshold scheme described by Equation (1). Due to Lemma 2, Equation (1) can be equivalently presented as

$$(\varepsilon_1, \ldots, \varepsilon_{nv}) = (\rho_1, \ldots, \rho_{tv}) B \tag{5}$$

where $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_{nv})$ and $\tau = (\rho_1, \ldots, \rho_{tv})$ are created from vectors $(s_1, \ldots, s_n)$ and $(r_1, \ldots, r_t)$ respectively by replacing each entry by its vector representations, and B is an $tv \times nv$ matrix over $GF(p)$ obtained from the matrix A by changing σ^i into the matrix D^i ($\sigma^0 = 1$ and $D^0 = I_v$ is the $v \times v$ identity matrix).

A (t, n) Shamir Secret Sharing with v Sub-shares ([11])
Let sub-shares be chosen from $GF(p)$ and create shares with v elements. A single

share consists of v sub-shares. The group $\mathcal{P} = \{P_1, \ldots, P_n\}$ collectively holds the secret.

Dealer

- chooses at random a polynomial $f(x)$ of degree $tv - 1$ with elements from $GF(p)$,
- defines the secret $s = (f(-v), \ldots, f(-1))$,
- communicates shares $s_i = (\varepsilon_{i,1}, \ldots, \varepsilon_{i,v})$ to the participant P_i via a secure channel, where $\varepsilon_{i,j} = f(x_{i,j})$ and $x_{i,j}$ are public co-ordinates assigned to P_i.

Combiner

- is activated by a subgroup $\mathcal{A} \subseteq \mathcal{P}$,
- collects t shares (or alternatively tv sub-shares) from a currently active group $\mathcal{A}$, applies the Lagrange interpolation and recovers the polynomial $f(x)$,
- recovers the secret $s = (f(-v), \ldots, f(-1))$ and distributes it to the group $\mathcal{A}$ via secure channels.

5 Cheating Detection in Linear Secret Sharing with Sub-shares

If a linear (t, n) threshold scheme allows to use sub-shares, there are new possibilities for cheating prevention including the following:

1. non-atomic submission of shares with public matrix B. A participant pools a collection of sub-shares leaving the unused ones for verification purposes,
2. non-atomic submission of shares with partially secret matrix B. A participant submits sub-shares whose corresponding columns of matrix B are public. The columns related to sub-shares used by the participants for verification purposes, are private,
3. submission of sub-shares with secret matrix B.

The linear (t, n) threshold scheme with v sub-shares assigned to each participant according to Formula (3) allows to trade cheating detection ability with the threshold value ℓ. In particular,

- the threshold value is $\ell = t$, then the participants pool all their sub-shares so they collectively provide tv sub-shares (or alternatively t shares) and recover the secret. In this case there is no cheating detection,
- the threshold value $\ell > t$ and each participant submits k sub-shares such that $\ell k = tv$. A participant can use $v - k$ unused sub-shares for verification of the recovered secret.

Before we describe the new scheme, we introduce the following notation. Given a linear secret sharing scheme (t, n) with v sub-shares assigned to participants according to Formula (3). Then

(a) B denotes a $tv \times nv$ matrix such that any $tv \times tv$ sub-matrix (obtained by removing the suitable number of columns) is nonsingular,
(b) B_{P_i} denotes a $tv \times v$ sub-matrix of B that contains columns corresponding to all sub-shares assigned to the participant P_i,
(c) $B_{\mathcal{A}}$ denotes a $tv \times av$ sub-matrix of B that contains all columns corresponding to the group $\mathcal{A}$ of a participants,
(d) $B_{\mathcal{A}(k)}$ denotes a $tv \times ak$ sub-matrix of $B_{\mathcal{A}}$ that is obtained by removing $(v-k)$ columns corresponding to participant $P_i \in \mathcal{A}$. This matrix defines columns related to sub-shares provides by the participants of the group $\mathcal{A}$.

A (ℓ, n) Secret Sharing with Cheating Detection

Dealer

- constructs a (t, n) Shamir secret sharing with v sub-shares such that there is an integer k $(k < v)$ such that $k\ell = tv$. The scheme is described by Formula (3) where P_i is assigned a share $s_i = (\varepsilon_{(i-1)v+1}, \ldots, \varepsilon_{iv})$ which consists of v sub-shares. All sub-shares create the vector $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_{nv})$. The vector $\rho = (\rho_1, \ldots, \rho_{tv})$ is typically selected at random while the secret $s = (\rho_1, \ldots, \rho_v)$,
- communicates the shares to participants via confidential channel (the matrix B is public).

Combiner

- is activated by a collection $\mathcal{A}$ of ℓ active participants,
- collects k sub-shares from each participant $P_i \in \mathcal{A}$ and determines the matrix $B_{\mathcal{A}(k)}$,
- computes the vector $\rho = (\rho_1, \ldots, \rho_{tv})$,
- communicates the vector ρ to active participants via secure channel.

Verification - Each participant $P_i \in \mathcal{A}$

- takes the vector ρ and checks

$$(\varepsilon_{(i-1)v+1}, \ldots, \varepsilon_{iv}) \stackrel{?}{=} (\rho_1, \ldots, \rho_{tv}) B_{P_i}$$

  If the check holds, the participant computes the secret from the vector ρ. Otherwise, P_i aborts the scheme and announces a cheating attempt to the other participants.

Lemma 3. *Given a system of n equations in t unknowns $(n \geq t)$ generated from a linear (t, n) threshold scheme described by*

$$(\alpha_1, 0, \ldots, 0, \alpha_{t+1}, \ldots, \alpha_n) = (r_1, r_2, \ldots, r_t) A$$

where co-ordinates $\alpha_2 = 0, \ldots, \alpha_t = 0$. Assume also that the solution for the vector $r = (r_1, \ldots r_t)$ is different from zero. Then co-ordinates $\alpha_1, \alpha_{t+1}, \ldots, \alpha_n$ are different from zero.

A cheating participant clearly controls his own sub-shares but has no influence on the sub-shares owned by the other active participants. Unlike in the original TW attack, the cheater this time has a limited knowledge about which sub-shares are actually submitted to the combiner. As a result, he does not know which columns of the matrix $B_{\mathcal{A}}$ are used by the combiner. In other words, the cheater knows $B_{\mathcal{A}}$ but does not know $B_{\mathcal{A}(k)}$. In general, a cheating participant P_1 can try to achieve the following goals:

- G1 – the cheater wants to recover the valid vector ρ (or the valid secret) while leaving the honest participants with the invalid vector ρ that passes the verification for all honest participants $P_2, \ldots, P_\ell$,
- G2 – the cheater wants to recover the valid vector ρ (or the valid secret) while leaving the honest participants with the invalid vector ρ that fails verification for some honest participants,

Theorem 2. *Given (ℓ, n) secret sharing with cheating detection based on (t, n) scheme with v sub-shares. Assume that the active group $\mathcal{A} = \{P_1, \ldots, P_\ell\}$ includes a single cheater P_1. Then P_1 attains the goal*

- *G1 only if $\lambda = v$, where $\lambda = \frac{k\ell-1}{\ell-1}$,*
- *G2 if $k \leq \lambda < v$ and the probability of success is smaller than or equal to*

$$\binom{\lambda}{k}^{\ell-1} / \binom{v}{k}^{\ell-1}$$

Assume that there are multiple cheaters who collectively create a group $\mathcal{B} = \{P_1, \ldots, P_m\}$ within the group $\mathcal{A} = \{P_1, \ldots, P_\ell\}$. The cheaters now control mk positions. They also know the a part of the matrix B which corresponds to the currently active group $\mathcal{A}$. By a manipulation of their own mk sub-shares they can easily target any $tv - 1$ sub-shares of honest participants so they will not change their values after cheating (see Lemma (3)). Note that the parameter

$$\lambda = \frac{k\ell - 1}{\ell - m}$$

denotes the average numbers of zeroes available for the group to be added to honest participant contributions. The conclusions obtained in Theorem 2 are still valid after rather trivial adjustments.

6 Prevention against the RR Attack

Consider a (ℓ, n) secret sharing with cheating detection. Assume that participants have agreed for multiple recovery of the secret. Again the currently active group is $\mathcal{A} = \{P_1, \ldots, P_\ell\}$ where P_1 is cheating and other participants are honest. The cheater may have the following goals:

- Goal $\mathrm{G1}_{RR}$ – recovery of the valid vector ρ (and the secret) while the cheater does not mind to be identified by the honest participants,

– Goal $G2_{RR}$ – recovery of the valid vector ρ (and the secret) while the cheater wants to remain unidentified.

Honest participants are likely to have a single goal in mind, namely, they wish to detect cheating and identify the cheater so they can create a new active group without the cheater and recover the secret.

To achieve their goals both parties may choose different strategies. The cheater may use

– Strategy $S1_c$ – the cheater modifies the sub-share vector ε by designing a vector α of the length ℓv with $k\ell - 1$ zeros distributed evenly among $\ell - 1$ honest participants. The vector α is fixed for the duration of all recoveries. At the pooling stage, the cheater selects her k sub-shares at random.
– Strategy $S2_c$ – as the strategy $S1_c$ except the cheater chooses α independently for each recovery.

Honest participants may apply

– Strategy $S1_h$ – for each recovery, an honest participant chooses at random k sub-shares.
– Strategy $S2_h$ – for the first recovery, an honest participant chooses at random k sub-shares and then keeps re-sending them for other recoveries.

Note that if the cheater applies the strategy $S2_c$ while honest ones behave according to the strategy $S2_h$, then the cheater will always succeed (recovers the correct vector ρ and the secret) while the honest ones have no knowledge about the secret. In other words, the cheater applies the RR attack and achieves $G1_{RR}$. Clearly, the honest participants are able to identify the cheater.

It is most likely that both parties (the honest participants and the cheater) will use strategies $S1_h$ and $S1_c$, respectively. As the two strategies are identical we call them the strategy S1. To achieve the goal $G2_{RR}$, the cheater must use this strategy so her behaviour is identical to behaviour of honest participants. On the other hand, honest participants are discouraged to use the strategy $S2_h$ as this strategy makes them vulnerable to the RR attack.

So all participants use the strategy S1 and obviously, they agreed for multiple recovery. In practice, however, it is reasonable to assume that after two unsuccessful recoveries, some participants may not wish to take part in further attempts or more likely, the active group may wish to replace some participants by the new ones. In any case, every unsuccessful recovery reveals an incorrect vector ρ' which permits active participants to recover the vector ε' (by multiplying ρ' by the public $B_{\mathcal{A}}$). A part of ε' corresponding to P_i contains at last k sub-shares that have been indeed submitted by them the other $v - k$ vary depending on how particular participants selected their sub-shares.

A probabilistic model of multiple recovery can be as follows. Given a field $GF(q)$ from which all sub-shares are chosen and a fixed and unknown vector $\beta = (\beta_1, \ldots, \beta_v)$ (it contains all sub-shares of a given participant whose sub-shares are to be identified); $\beta_i \in GF(q)$. To identify the vector, one queries a probabilistic oracle (the combiner). For each enquire, the oracle returns a vector $\gamma = (\gamma_1, \ldots, \gamma_v)$ which

– contains at least k correct co-ordinates. The oracle selects randomly one pattern of k correct co-ordinates out of $\binom{v}{k}$ possible ones.
– contains at most $v-k$ co-ordinates which are chosen randomly, uniformly and independently from the set $GF(q)$.

Note that each co-ordinate in the vector β is in fact, a random variable X with the probability distribution of the following form:

$$P(X=a) = \begin{cases} \frac{k}{v} + (1-\frac{k}{v})\frac{1}{q} \text{ if a is the correct sub-share;} \\ (1-\frac{k}{v})\frac{1}{q} \quad \text{otherwise.} \end{cases}$$

To justify this, it is enough to observe that the given co-ordinate with the correct sub-share occurs $\binom{v-1}{k-1} / \binom{v}{k} = \frac{k}{v}$ times assuming the the oracle chooses k out of v position at random. The correct sub-share may also happen if the co-ordinate has not been chosen and the correct sub-share has been selected from $GF(q)$. The wrong value of sub-share occurs when the co-ordinate has not been selected to the subset of k out of v correct sub-shares and the random element tossed from all $GF(q)$ is different from the correct sub-share.

Identification of correct sub-shares can be accomplished using standard statistical tools such as hypothesis testing. We can put forward two hypotheses related to the binary random variable $X \in \{0,1\}$. The first one

$$H_0 \mapsto P(X=a) = \begin{cases} \frac{k}{v} + (1-\frac{k}{v})\frac{1}{q} \text{ if } a=0; \\ (1-\frac{k}{v})\frac{q-1}{q} \quad \text{otherwise.} \end{cases}$$

and its alternative

$$H_1 \mapsto P(X=a) = \begin{cases} (1-\frac{k}{v})\frac{1}{q} \quad \text{if } a=0; \\ \frac{k}{v} + (1-\frac{k}{v})\frac{q-1}{q} \text{ otherwise.} \end{cases}$$

7 Conclusions

Tompa and Woll were first to demonstrate how a dishonest participant can cheat others during the recovery of secret. The TW attack is applicable if the recovery algorithm is run once only. We have argued that in many applications the assumption about single recovery is not reasonable and there is no mechanism incorporated into secret sharing which would prevent it from multiple recoveries. If we allow multiple recoveries than a new threat called the repeat recovery attack emerges. Unlike the TW attack, the RR attack can be successful even if the x co-ordinates are secret. Note that this is a surprising result as the cheater does not have any information about x co-ordinates while being able to determine the valid secret ! The RR attack can be put to a good use so an active group can recover secret after the combiner has been compromised. A trusted participant plays the role of combiner using the compromised combiner.

A (ℓ, n) secret sharing scheme with cheater detection is built on the basis of (t, n) threshold scheme with v sub-shares. Unlike in standard secret sharing, participants are expected to split their sub-shares into two subsets. One subset is submitted to the combiner while the other is left for verification purposes. The combiner returns the vector ρ (instead of the recovered secret) which further is used by participants to check its validity. The sharing scheme with cheater detection has the following properties:

- The scheme can be used as a standard (t, n) secret sharing with atomic shares.
- Cheating according to the TW attack (the goal G1) is only successful if the parameters $v = \lambda$ (λ specifies the number of sub-shares per participants which could be manipulated by cheater by putting zeroes in her α vector). For any $v > \lambda$, the goal G1 cannot be achieved.
- The recovery of the valid secret by the cheater while the honest participants detect cheating (the goal G2) can be achieved with the probability of guessing the subset of sub-shares submitted to the combiner.

The secret sharing scheme with cheater detection is also investigated in the context of the RR attack. We argued that if the cheater wishes not to be identified then she must fix her (modified) sub-shares for the duration of multiple recoveries. The cheater and honest participants are now trying to identify their sub-shares. This scenario resembles the TW attack if the number of recoveries is big enough so everybody can correctly identify sub-shares. Note, however, that honest participants may wish to refuse to participate after some number of recoveries when the probability of guessing of correct secret by the cheater becomes too high.

Acknowledgement

The work was partially supported by Australian Research Council grant A00103078.

References

1. C. Asmuth and J. Bloom. A modular approach to key safeguarding. *IEEE Transactions on Information Theory*, IT-29 No. 2:208–211, 1983.
2. J. Benaloh and J. Leichter. Generalised secret sharing and monotone functions. In S. Goldwasser, editor, *Advances in Cryptology - CRYPTO'88*, LNCS No. 403, pages 27–36. Springer-Verlag, 1988.
3. G. R. Blakley. Safeguarding cryptographic keys. In *Proc. AFIPS 1979 National Computer Conference*, pages 313–317. AFIPS, 1979.
4. E.F. Brickell and D.R. Stinson. The detection of cheaters in threshold schemes. In S. Goldwasser, editor, *Advances in Cryptology - CRYPTO'88*, LNCS No. 403, pages 564–577. Springer-Verlag, 1988.
5. M. Carpentieri. A perfect threshold secret sharing scheme to identify cheaters. *Designs, Codes and Cryptography*, 5(3):183–187, 1995.

6. M. Carpentieri, A. De Santis, and U. Vaccaro. Size of shares and probability of cheating in threshold schemes. In T. Helleseth, editor, *Advances in Cryptology - EUROCRYPT'93*, LNCS No. 765, pages 118–125. Springer, 1993.
7. H. Ghodosi, J. Pieprzyk, R. Safavi-Naini, and H. Wang. On construction of cumulative secret sharing. In C. Boyd and E. Dawson, editor, In *Proceedings of the Third Australasian Conference on Information Security and Privacy (ACISP'98)*, LNCS No. 1438, pages 379–390. Springer-Verlag, 1998.
8. H. Lin, and L. Haen. A generalised secret sharing scheme with cheater detection. In H. Imai, R. Rivest, and T. Matsumoto, editor, In *Proceedings of ASIACRYPT'91*, LNCS No. 739, pages 149-158. Springer-Verlag, 1993.
9. M. Ito, A. Saito, and T. Nishizeki. Secret sharing scheme realizing general access structure. In *Proceedings IEEE Globecom '87*, pages 99–102. IEEE, 1987.
10. E.D. Karnin, J.W. Greene, and M.E. Hellman. On secret sharing systems. *IEEE Transactions on Information Theory*, IT-29:35–41, 1983.
11. K. Martin, J. Pieprzyk, R. Safavi-Naini, and H. Wang. Changing thresholds in the absence of secure channels. In *Proceedings of the Fourth Australasian Conference on Information Security and Privacy (ACISP'99)*, LNCS No. 1587, pages 177–191. Springer-Verlag, 1999.
12. . W. Ogata and K. Krurosawa. Optimum secret shares scheme secure against cheating. In U. Maurer, editor, *Advances in Cryptology - EUROCRYPT'96*, LNCS No. 1070, pages 200–211.
13. T. Rabin and M. Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. In *Proceedings of 21st ACM Symposium on Theory of Computing*, pages 73–85, 1989.
14. A. Shamir. How to share a secret. *Communications of the ACM*, 22:612–613, November 1979.
15. G.J. Simmons, W. Jackson, and K. Martin. The geometry of shared secret schemes. *Bulletin of the ICA*, 1:71–88, 1991.
16. M. Tompa and H. Woll. How to share a secret with cheaters. *Journal of Cryptology*, 1(2):133–138, 1988.