Cheater Identifiable Secret Sharing Schemes via Multi-Receiver Authentication

Rui Xu[1], Kirill Morozov[2], and Tsuyoshi Takagi[2]

[1] Graduate School of Mathematics, Kyushu University, Japan
r-xu@math.kyushu-u.ac.jp
[2] Institute of Mathematics for Industry, Kyushu University, Japan
{morozov,takagi}@imi.kyushu-u.ac.jp

Abstract. We introduce two publicly cheater identifiable secret sharing (CISS) schemes with efficient reconstruction, tolerating $t < k/2$ cheaters. Our constructions are based on (k, n) threshold Shamir scheme, and they feature a novel application of multi-receiver authentication codes to ensure integrity of shares.

The first scheme, which tolerates rushing cheaters, has the share size $|S|(n-t)^{n+t+2}/\epsilon^{n+t+2}$ in the general case, that can be ultimately reduced to $|S|(k-t)^{k+t+2}/\epsilon^{k+t+2}$ assuming that all the t cheaters are among the k reconstructing players. The second scheme, which tolerates non-rushing cheaters, has the share size $|S|(n-t)^{2t+2}/\epsilon^{2t+2}$. These two constructions have the smallest share size among the existing CISS schemes of the same category, when the secret is a single field element.

In addition, we point out that an improvement in the share size to $|S|/\epsilon^{n-\lfloor(k-1)/3\rfloor+1}$ can be achieved for a CISS tolerating $t < k/3$ rushing cheaters presented by Xu et al. at IWSEC 2013.

Keywords: Cheater identifiable secret sharing, multi-receiver authentication code, Shamir secret sharing, rushing adversary.

1 Introduction

We consider cheater identifiable secret sharing (CISS) which is an upgrade of (k, n)-threshold secret sharing schemes [1,13] that can tolerate up to t actively corrupt participants. The dealer in CISS is assumed to be honest. The goal in this scenario is to identify cheaters from the threshold k number of players, and to recover a correct secret whenever possible. In this work, we focus on *public* cheater identification, where reconstruction of the secret and cheater identification can be performed by a third party who collects shares from a threshold of players. Note that an honest majority, i.e. $t < k/2$, is necessary in this case, otherwise the dishonest majority of cheaters might simply generate a new consistent set of (authenticated) shares and submit it at the reconstruction. We will consider, in particular, *rushing* cheaters who are allowed to decide their messages (in every round) upon seeing the messages of honest parties.

M. Yoshida and K. Mouri (Eds.): IWSEC 2014, LNCS 8639, pp. 72–87, 2014.

1.1 Related Works

The observation of McEliece and Sarwate [8] on a connection between the Shamir scheme [13] and the Reed-Solomon codes [11] allowed for identification of cheaters, however redundant shares (i.e., more than k of them) were required. The first CISS scheme came from a related area of robust secret sharing (where the secret is always reconstructed from n shares, while no cheater identification is required) when Rabin and Ben-Or [10] proposed to use unconditional authentication codes for enforcing the integrity of shares. Then, a number of proposals for CISS schemes followed where the efforts were directed at achieving efficient reconstruction against maximal number of cheaters, while reducing the share size. We refer the reader to the survey of Martin [7] for the history of this subject.

Recently, Obana [9] proposed a CISS scheme that is secure against $t < k/2$ non-rushing cheaters but has inefficient reconstruction algorithm (the computation complexity is exponential in the number of cheaters). Choudhury [2] presented a CISS scheme secure against $t < k/2$ rushing cheaters with efficient reconstruction. The share size of his scheme is optimal $O(|S|/\epsilon)$ provided that the size of the secret is $\Omega(n)$, where $|S|$ denotes the size of the secret and ϵ is the cheater success probability. In this work, we focus on the scenario where the secret is "short", i.e., it is represented by a single field element – the same scenario as in [9]. In this case, the share size of Choudhury's scheme is far from optimal.

Under assumption of having $t < k/3$ non-rushing cheaters, Obana [9] presented a CISS scheme with nearly optimal share size $|S|/\epsilon$. Xu et al. [15] upgraded the above scheme to security against rushing cheaters for the price of increasing the share size to $|S|/\epsilon^{n-t+1}$.

1.2 Our Contribution

We present two new CISS schemes tolerating up to $t < k/2$ cheaters, which are based on multi-receiver authentication codes [4, 12]. These schemes are introduced below as Proposals 1 and 2.

Proposal 1: Our scheme tolerating rushing cheaters has the share size $|S|(n-t)^{n+t+2}/\epsilon^{n+t+2}$ in the general case. However, if the number of shares presented at the reconstruction is restricted to k, then the share size can be made equal to $|S|(k-t)^{k+2t+1}/\epsilon^{k+2t+1}$. In other words, when restricting the number of reconstructing players, the share size can be reduced. This is an interesting point in the sense that generally, redundant information is used to identify cheaters. However, in this particular case, we observe that some redundant information can be beneficial to the cheaters. In fact, the share size can be reduced even further to $|S|(k-t)^{k+t+2}/\epsilon^{k+t+2}$, under assumption that all the corrupt players always participate in the reconstruction.

Proposal 2: Our scheme tolerating non-rushing cheaters has the share size $|S|(n-t)^{2t+2}/\epsilon^{2t+2}$. Our proposal has smaller share size as compared to $|V_i| = |S|(t+1)^{3n}/\epsilon^{3n}$ in Choudhury scheme [2]. We emphasize that the work [2] presents a scheme tolerating non-rushing adversaries, but it is trivial to extend it to the rushing case, such that the share size is the same in both cases.

Table 1. Comparison of Our Proposals to Existing CISS schemes

Scheme	Assumption	Share Size	Adversary
Choudhury [2]*	$t < k/2$	$\lvert V_i\rvert = \lvert S\rvert(t+1)^{3n}/\epsilon^{3n}$	Rushing
Our Proposal 1**	$t < k/2$	$\lvert V_i\rvert = \lvert S\rvert(k-t)^{k+t+2}/\epsilon^{k+t+2}$	Rushing
Obana [9]***	$t < k/2$	$\lvert V_i\rvert \approx \lvert S\rvert(nt \cdot 2^{3t})^2/\epsilon^2$	Non-Rushing
Our Proposal 2	$t < k/2$	$\lvert V_i\rvert = \lvert S\rvert(n-t)^{2t+2}/\epsilon^{2t+2}$	Non-Rushing
Obana [9]	$t < k/3$	$\lvert V_i\rvert = \lvert S\rvert/\epsilon$	Non-Rushing
Xu et al. [15]	$t < k/3$	$\lvert V_i\rvert = \lvert S\rvert/\epsilon^{n-t+1}$	Rushing
Our Proposal 3	$t < k/3$	$\lvert V_i\rvert = \lvert S\rvert/\epsilon^{n-\lfloor (k-1)/3\rfloor+1}$	Rushing

* When the secret is a single field element.

** The smallest share size, when restricting to k reconstructing parties such that only these ones can be actively corrupt.

*** The reconstruction needs $\binom{3t}{t+2}$ Lagrange interpolations. For comparison, each of our Proposals 1 and 2 needs one Lagrange interpolation and k polynomial evaluations.

Proposal 3: Under assumption that $t < k/3$, we improve the share size of the scheme [15] from $|S|/\epsilon^{n-t+1}$ to $|S|/\epsilon^{n-\lfloor (k-1)/3\rfloor+1}$ by eliminating some encryption keys in their construction.

Our contributions and the related works are summarized in Table 1.

Remark 1. We emphasize that the main contribution to the share size typically comes from the factor $1/\epsilon$, since one expects the cheating probability ϵ to be made negligible, while the parameters n, k, and t are some constants. Consequently, the major efforts in reducing the share size are made towards reducing the degree of the factor $1/\epsilon$.

2 Preliminaries

Set $[n] = \{1, 2, \ldots, n\}$. The cardinality of the set X is denoted by $|X|$. Let $\mathbb{F}_p$ be a Galois field of a prime order p satisfying $p > n$. Let $\phi(\cdot,\cdot) : \mathbb{F}_p \times [n] \to \mathbb{F}_q$ be a injective function ($q > np$ is a prime power). All computation is done in the specified Galois fields.

2.1 Shamir Secret Sharing

We describe the k-out-of-n threshold secret sharing scheme by Shamir [13]. Such the secret sharing scheme involves a dealer D and n participants $\{R_1, \ldots, R_n\}$, and consists of two algorithms: **ShareGen** and **Reconst**. The **ShareGen** algorithm takes a secret $s \in \mathbb{F}_p$ as input and then outputs a list $(\sigma_1, \ldots, \sigma_n)$. Each σ_i

is respectively distributed to participant R_i and called her share. The algorithm **Reconst** takes a list $(\sigma_1, \ldots, \sigma_m)$ as input and outputs the secret s if $m \geq k$. Otherwise, the **Reconst** outputs $\perp$. Formally, the properties of *correctness* and *perfect secrecy* hold:

1. Correctness: If $m \geq k$, then $\Pr[\mathbf{Reconst}(\sigma_1, \ldots, \sigma_m) = s] = 1$;

2. Perfect secrecy: If $m < k$, then $\Pr[S = s|(V_1 = \sigma_1, \ldots, V_m = \sigma_m)] = \Pr[S = s]$ for any $s \in S$.

In the Shamir scheme, the above mentioned algorithms proceed as follows:

ShareGen

1. For a given secret $s \in \mathbb{F}_p$, the dealer D chooses a random polynomial $f(x) \in \mathbb{F}_p[X]$ with degree at most $k-1$ and $f(0) = s$.

2. For $i \in [n]$, compute $\sigma_i = f(x_i)$ for fixed, public and distinct $x_i \in \mathbb{F}_p$ (where x_i can be seen as a unique identifier for R_i) and send σ_i privately to participant R_i.

Reconst

If $m \geq k$ then output the secret s using the Lagrange interpolation formula, otherwise output $\perp$.

2.2 Cheater Identifiable Secret Sharing

We will focus on cheater identifiable secret sharing that is based on the Shamir scheme. In CISS, we require that the reconstruction algorithm **Reconst** both computes the secret and identifies incorrect shares, which point at cheaters among the involved participants. The output of **Reconst** algorithm is a tuple (s', L), where s' is the reconstructed secret and L is the set of cheaters. If the secret cannot be reconstructed because there are not enough of honest players, it is set to be $\perp$. When $s' \neq \perp$, $s' = s$ except with negligible probability.

The following definitions are developed using those by Choudhury [2] and Xu et al. [15].

Communication Model: We assume that the participants $\mathcal{R} = \{R_1, \ldots, R_n\}$ are connected with the dealer D by private and authenticated channels, and in addition, a broadcast channel is available to every entity. The communication network is assumed to be synchronous and the adversary can be *rushing* or not [5]. In synchronous network the protocols proceed in rounds: the current round is known to all parties, and messages sent in some round are delivered by the beginning of the next round. The term "rushing" refers to allowing the corrupted parties to learn the messages sent by the uncorrupted parties in each round, before sending their own messages for this round.

Adversary Model: There exist two adaptive, computationally unbounded adversaries $\mathcal{A}_{listen}$ and $\mathcal{A}_{cheat}$. The listening adversary $\mathcal{A}_{listen}$ can passively control any $k-1$ parties in $\mathcal{R}$. The cheating adversary $\mathcal{A}_{cheat}$ can adaptively choose to control any t parties in $\mathcal{R}$ in the malicious manner. Additionally, we assume that $\mathcal{A}_{listen}$ and $\mathcal{A}_{cheat}$ do not collude. This implies that $\mathcal{A}_{cheat}$ will not get any information about the computation and communication of the parties, which are under the control of $\mathcal{A}_{listen}$ (but not $\mathcal{A}_{cheat}$) and vice-versa. Intuitively, security against $\mathcal{A}_{listen}$ implies the standard (perfect) secrecy of (k,n)-threshold secret sharing, while security against $\mathcal{A}_{cheat}$ implies protection against active cheaters intending to disrupt the reconstruction of a correct secret. As usual in CISS schemes, we assume that adversaries cannot corrupt the dealer D.

Definition 1 ([15]). A cheater identifiable secret sharing scheme Σ is a tuple $(n, k, S, V, \mathbf{ShareGen}, \mathbf{Reconst})$ consisting of:

- A positive integer n called the number of players;
- A positive integer k denoting the number of honest shares from which the original secret can be reconstructed;
- A finite set S with $|S| \geq 2$, whose elements are called secrets;
- A finite set $V = \{V_1, \ldots, V_n\}$, where V_i is the set of player R_i's shares;
- An algorithm **ShareGen**, that takes as input a secret $s \in S$, and outputs a vector of n shares $(\sigma_1, \ldots, \sigma_n) \in V_1 \times \cdots \times V_n$; and
- An algorithm **Reconst**, that takes as input a vector $(\sigma'_{i_1}, \ldots, \sigma'_{i_m}) \in V_{i_1} \times \cdots \times V_{i_m}$, and outputs a tuple (s', L), where s' is the reconstructed secret and L is the set of identified cheaters.

Recall that the cheating adversary can corrupt at most t players. Denote by $(R_{i_1}, \ldots, R_{i_t})$ the t cheaters under the control of $\mathcal{A}_{cheat}$ and by $\sigma'_{i_1}, \ldots, \sigma'_{i_t}$ their possibly corrupt shares. We define the successful cheating probability to be the probability that *the cheater is not identified when she provided a forged share (thus resulting in a corrupt secret) at the reconstruction.*

Definition 2. For some $s' \neq s$, the successful cheating probability of player R_{i_j} under the control of $\mathcal{A}_{cheat}$ against the cheater identifiable secret sharing scheme $\Sigma = (n, k, S, V, \mathbf{ShareGen}, \mathbf{Reconst})$ is defined as

$$\begin{aligned}&\epsilon(\Sigma, R_{i_j}, \mathcal{A}_{cheat})\\ &= \max_{\sigma'_{i_j} \neq \sigma_{i_j}} \Pr[(s', L) \leftarrow \mathbf{Reconst}(\sigma'_{i_1}, \ldots, \sigma'_{i_t}, \sigma_{i_{t+1}}, \ldots, \sigma_{i_k}) \wedge R_{i_j} \notin L], \end{aligned} \tag{1}$$

where the probability is taken over the distribution of S, and the random coins of **ShareGen** and $\mathcal{A}_{cheat}$.

Henceforth, we will write the above probability as $\epsilon(\Sigma, R_{i_j})$, for short.

Remark 2. For simplicity of our analysis – and similarly to the previous works – we estimate the success probability for a *single* cheater. The overall success probability for the cheating adversary can be estimated using the union bound.

Definition 3. A CISS scheme $\Sigma = (n, k, S, V, \mathbf{ShareGen}, \mathbf{Reconst})$ is called (t, ϵ)-CISS scheme if the following properties hold:

1. Perfect secrecy: At the end of the algorithm **ShareGen**, $\mathcal{A}_{listen}$ has no information about the secret s.
2. $(1-\epsilon)$-correctness: $\epsilon(\Sigma, R_i) \leq \epsilon$ for any cheater R_i under the control of $\mathcal{A}_{cheat}$.

2.3 Unconditional Multi-Receiver Authentication Codes

In the traditional setting of unconditional authentication codes [3], there are three participants: a transmitter, a receiver and an opponent. The task of authentication codes is to prevent the opponent from deceiving the receiver by impersonation attacks and substitution attacks. Desmedt, Frankel and Yung [4] proposed a generalized notion of authentication called unconditional multi-receiver authentication (MRA). An MRA code involves one transmitter, one opponent and n receivers. When authenticating a source, the transmitter broadcasts a message to n receivers and each receiver verifies the authenticity of the message based on their own keys. If an MRA code ensures that neither the outside opponent nor the coalition of t receivers can deceive any other honest player, it is called a (t, n) MRA code.

Desmedt et al. constructed a (t, n) MRA code capable of authenticating a single message. Safavi-Naini and Wang [12] generalized Desmedt et al.'s construction to allow multiple messages to be authenticated with the same key. We will call it a (t, n) MRA code with multiple messages. We briefly describe Safavi-Naini and Wang's construction in Algorithm 1.

Let $Poly_t$ be the set of all polynomials of degree at most t over the finite filed $\mathbb{F}_q$. Define a map $f : \mathbb{F}_q \times Poly_t^{w+1} \to Poly_t$ with $f(s, P_0(x), \ldots, P_w(x)) = P_0(x) + sP_1(x) + \cdots + s^w P_w(x)$, where $P_i(x) \in Poly_t$ for $i = 0, \ldots, w$. For the ease of presentation, set $e = (P_0(x), \ldots, P_w(x))$ and express f as $f_e(s) = A_s(x)$. We also denote by $e_i = (P_0(x_i), \ldots, P_w(x_i))$ the verification key for Player R_i.

Algorithm 1 ((t, n) MRA with w messages)

Assume that $q \geq w$, where w is the number of possible messages, and that $q \geq n$. The system consists of the following steps:

1. **Key distribution:** The key distribution center (KDC) randomly generates $w + 1$ polynomials $e = (P_0(x), P_1(x), \ldots, P_w(x))$, each of degree at most t and chooses n distinct elements $x_1, x_2, \ldots, x_n$ of $\mathbb{F}_q$. KDC makes all x_i public and sends privately $(P_0(x), \ldots, P_w(x))$ to the sender T as her authentication key, and $e_i = (P_0(x_i), \ldots, P_w(x_i))$ to the receiver R_i as her verification key.
2. **Broadcast:** For a message s, T computes $A_s(x) = f_e(s) = P_0(x) + sP_1(x) + \cdots + s^w P_w(x)$ and broadcasts $(s, A_s(x))$.
3. **Verification:** R_i accepts $(s, A_s(x))$ as authentic if $A_s(x_i) = P_0(x_i) + sP_1(x_i) + \cdots + s^w P_w(x_i)$.

It is proven by Safavi-Naini and Wang that Algorithm 1 is a (t, n) MRA code in which each key can be used to authenticate up to w messages with both impersonation and substitution probability $1/q$.

Formally, we have the following property:

Property 1. The probability that t corrupt receivers and/or the outside opponent succeed in deceiving any receiver R_i is at most

$$\begin{aligned}&\Pr[R_i \text{ accepts } (s_{w+1}, A_{s_{w+1}}(x)) | f_e(s_1) = A_{s_1}(x), \ldots, f_e(s_w) = A_{s_w}(x);\\ &\quad e_{i_1}, \ldots, e_{i_t}] = 1/q.\end{aligned} \tag{2}$$

for any choice of $(s_{w+1}, A_{s_{w+1}}(x))$ with $s_{w+1} \neq s_i$ for $i = 1, \ldots, w$; for any choice of $(P_0(x), \ldots, P_w(x)) \in Poly_t^{w+1}$, and for any $[i_1, \ldots, i_t] \subseteq [n] \setminus \{i\}$.

3 CISS Against Rushing Adversary

In this section, we propose two CISS schemes against a rushing active adversary, $\mathcal{A}_{cheat}$, who can corrupt at most t players provided $t < k/2$. The first one restricts the number of reconstructing players to be exactly k which allows us to achieve a smaller share size compared to the case of allowing more than k players to join the reconstruction. The second one extends to general situation where the number of reconstructing players can be any value m with $k \leq m \leq n$. Since in both schemes $\mathcal{A}_{cheat}$ can corrupt at most t players, the fact that the second scheme requires larger share size implies that the higher ratio of honest players benefits the adversary. This may seem counter-intuitive, but the reason for this is that the adversary is rushing so that more honest players provide more information to her. We will emphasize this point in the proof of security.

3.1 Overview

The basic idea of our proposal is to follow the paradigm of Rabin and Ben-Or [10] that is to use unconditional authentication codes for pairwise authentication and to use the majority voting to identify cheaters. The twist of our scheme is to employ *multi-receiver* authentication codes [4,12], instead of ordinary ones. More specifically, the dealer D generates Shamir shares, denoted $v_{s,i}$, for player R_i and authenticates it using MRA codes. Then the dealer sends $v_{s,i}$, its authentication tag $v_{c,i}(x)$ (note that it is a polynomial), and the verification key to player R_i privately. Reconstruction of the secret is performed in two rounds. In the first round, each player broadcasts her share and authentication tag $(v_{s,i}, v_{c,i}(x))$. In the second round, each player broadcasts her verification key (we emphasize that in MRA each player holds different verification key). After receiving all the above information, the players vote for correctly authenticated shares, and then identify cheaters as the players who did not get enough approvals.

3.2 CISS with Restriction on Reconstructing Players

The following scheme restricts the number of reconstructing players to be exactly the threshold k.

Protocol 1 (ShareGen)

Public parameters: $x_i \in \mathbb{F}_p$ as player R_i's identifier for $i = 1, \ldots, n$.

Input: Secret $s \in \mathbb{F}_p$.

Output: A list of n shares $\sigma_1, \sigma_2, \ldots, \sigma_n$.

A dealer D performs the following:

1. Generate a random degree-$(k-1)$ polynomial $f_s(x)$ over $\mathbb{F}_p$, such that $f_s(0) = s$. Compute $v_{s,i} = f_s(x_i)$, for $i \in [n]$.

2. Uniformly at random, generate $e = (P_0(x), \ldots, P_{k+t_{max}}(x))$ that is an authentication key for a (t, n) MRA code with $k + t_{max}$ messages, where $t_{max} = min\{t-1, n-k\}$, and $P_i(x) \in Poly_t$ is a polynomial of degree at most t over $\mathbb{F}_q$.
3. For $i \in [n]$, compute $v_{c,i}(x) = f_e(\phi(v_{s,i}, i))$ as the authentication tag for $v_{s,i}$. Note that $v_{c,i}(x) \in Poly_t$ is a polynomial of degree at most t over $\mathbb{F}_q$.
4. For $i \in [n]$, set $\sigma_i = \{v_{s,i}, v_{c,i}(x), P_0(x_i), \ldots, P_{k+t_{max}}(x_i)\}$ and distribute it privately to player R_i.

Remark 3. Note that in Step 3, we combine player's share $v_{s,i}$ with her identifier i before authentication. This is because Shamir scheme does not guarantee that each player gets distinct shares. Therefore, a cheater may simply re-use the share and authentication information submitted by any honest player – naturally it would be accepted as authentic. In order to prevent that from happening, we use the injective function $\phi(\cdot, \cdot)$ to make sure that the entities to be authenticated will be distinct for every player even if they received the same share.

Without loss of generality, assume that the first k players want to recover the secret. Moreover, let $\sigma'_i = \{v'_{s,i}, v'_{c,i}(x), P'_0(x_i), \ldots, P'_{k+t_{max}}(x_i)\}$ be the (possibly corrupt) share for player R_i.

Protocol 2 (Reconst)

Input: A list of k shares $(\sigma'_1, \ldots, \sigma'_k)$.
Output: Either $(\perp, L)$ or (s', L), where L is the list of cheaters.

Communication rounds performed by each player $i \in [k]$:
Round 1: Announce $(v'_{s,i}, v'_{c,i}(x))$.
Round 2: Announce $(P'_0(x_i), \ldots, P'_{k+t_{max}}(x_i))$.

Computation by players in $[k]$:

1. For $i \in [k]$, do:
   a) Use the verification key $(P'_0(x_j), \ldots, P'_{k+t_{max}}(x_j))$ to verify the authenticity of $(v'_{s,i}, v'_{c,i}(x))$, for $j \in [k]$.
   b) If less than $t + 1$ keys verify $(v'_{s,i}, v'_{c,i}(x))$ as authentic, then player R_i is put into the cheater list L.
2. If $L = \emptyset$, reconstruct $f'_s(x)$ from k shares $v'_{s,i}$ using Lagrange interpolation and output $(f'_s(0), L)$. Otherwise output $(\perp, L)$.

Remark 4. It is easy to check that in Round 2, the players can broadcast their votes regarding each player's share, instead of their verification keys. Precisely, the player R_i can use her verification key $(P_0(x_i), \ldots, P_{k+t_{max}}(x_i))$ to verify the share $(v'_{s,j}, v'_{c,j}(x))$ announced by player R_j. After verifying all the shares, R_i broadcasts a binary vector of length k indicating her votes against all the k players. Broadcasting every player's votes instead of her verification key can reduce the communication cost of **Reconst** protocol. However, this does not affect the share size.

Theorem 1. *If $t < k/2$ then the scheme described above is a (t,ϵ)-CISS against rushing adversary such that*

$$|S| = p,\ \epsilon = \frac{k-t}{q},\ q \geq n \cdot p,\ |V_i| = p \cdot q^{k+2t+1} = \frac{|S|(k-t)^{k+2t+1}}{\epsilon^{k+2t+1}}. \quad (3)$$

For proving Theorem 1, we will use the following two lemmas.

Lemma 1. *The above (k,n)-CISS has perfect secrecy, i.e. $\mathcal{A}_{listen}$ has no information about the secret s at the end of* ***ShareGen****.*

Proof. We can assume *w.l.o.g.* that the passive adversary $\mathcal{A}_{listen}$ corrupts the first $k-1$ players after **ShareGen**. $\mathcal{A}_{listen}$ will know $k-1$ Shamir shares $(v_{s,1}, \ldots, v_{s,k-1})$ from which she can get no information about the secret s due to perfect secrecy of Shamir scheme. Besides the Shamir shares, $\mathcal{A}_{listen}$ also knows the verification keys for $k-1$ players and the $k-1$ authentication tags. But the authentication key $e = (P_0(x), \ldots, P_{k+t_{max}}(x))$ is randomly generated independently of the secret s, and it decides the verification key for each player. So the verification keys leak no information about s. Moreover, the authentication tags are decided by the Shamir shares and the authentication key, they also do not give any information on the secret s. Thus we have proven that after **ShareGen**, $\mathcal{A}_{listen}$ gets no information about the secret s. □

Lemma 2. *In the above CISS, $\epsilon(\Sigma, R_i) \leq \frac{k-t}{q}$ for any player R_i under control of $\mathcal{A}_{cheat}$.*

Proof. We divide all the n players into the following two groups: The *active* group $(R_1, \ldots, R_k)$ who take part in the reconstruction phase; and the *inactive* group $(R_{k+1}, \ldots, R_n)$ who just hold their shares. Recall that $\mathcal{A}_{cheat}$ can corrupt at most t players. Assume $\mathcal{A}_{cheat}$ corrupts t' players in the active group and t'' players in the inactive group such that $t' + t'' = t$. Note that $t' \geq 1$ (which implies $t'' \leq t-1$), since $\mathcal{A}_{cheat}$ has to corrupt at least one player in the active group in order to cheat the honest players. Combining this observation with $t'' \leq n-k$, we get $t'' \leq t_{max} = min\{t-1, n-k\}$. Suppose *w.l.o.g.* that $\mathcal{A}_{cheat}$ corrupts $R_1, \ldots, R_{t'}$ in the active group and $R_{k+1}, \ldots, R_{k+t''}$ in the inactive group. Remember that since the adversary $\mathcal{A}_{cheat}$ is rushing, she can see all the communication of honest players during each round, prior to deciding her own messages. Denote the verification key for R_i by $e_i = (P_0(x_i), \ldots, P_{k+t_{max}}(x_i))$. We summarize the view of the adversary in Table 2.

Suppose *w.l.o.g.* that player R_1 under control of $\mathcal{A}_{cheat}$ submits a forged share $\sigma'_1 = \{v'_{s,1}, v'_{c,1}(x), e'_1\}$. If R_1 is not identified as a cheater, then at least one honest player will accept $(v'_{s,1}, v'_{c,1}(x))$ as authentic. At the end of the first round R_1 has to submit $(v'_{s,1}, v'_{c,1}(x))$ with $v'_{s,1} \neq v_{s,1}$. At that time, she can see $(v_{s,1}, \ldots, v_{s,k+t''})$, $(v_{c,1}(x), \ldots, v_{c,k+t''}(x))$, and $(e_1, \ldots, e_{t'})$, $(e_{k+1}, \ldots, e_{k+t''})$. From the $t'+t''=t$ verification keys and the $k+t''$ authentication tags R_1 cannot generate a new authentication tag for $\phi(v'_{s,1}, x_1)$. This is because $t'' \leq t_{max}$, so $k+t'' \leq k+t_{max}$. Recall that we use (t,n) MRA with $k+t_{max}$ messages in the **ShareGen** Protocol. At the end of round 1, $\mathcal{A}_{cheat}$ has seen at most $k+t_{max}$

Table 2. View of $\mathcal{A}_{cheat}$ in **Reconst**

First round	Second round
$(v_{s,1}, v_{c,1}(x), e_1)$	$(v_{s,1}, v_{c,1}(x), e_1)$
$\cdots$	$\cdots$
$(v_{s,t'}, v_{c,t'}(x), e_{t'})$	$(v_{s,t'}, v_{c,t'}(x), e_{t'})$
$(v_{s,t'+1}, v_{c,t'+1}(x))$	$(v_{s,t'+1}, v_{c,t'+1}(x), e_{t'+1})$
$\cdots$	$\cdots$
$(v_{s,k}, v_{c,k}(x))$	$(v_{s,k}, v_{c,k}(x), e_k)$
$(v_{s,k+1}, v_{c,k+1}(x), e_{k+1})$	$(v_{s,k+1}, v_{c,k+1}(x), e_{k+1})$
$\cdots$	$\cdots$
$(v_{s,k+t''}, v_{c,k+t''}(x), e_{k+t''})$	$(v_{s,k+t''}, v_{c,k+t''}(x), e_{k+t''})$

authentication tags and knows t verification keys. By Equation (2) in Property 1 we have for any honest player R_j where $j \in [k] \setminus [t']$,

$$\Pr[R_j \text{ accepts } (v'_{s,1}, v'_{c,1}(x)) | \text{ the view of } \mathcal{A}_{cheat}] = 1/q.$$

The probability that one honest player accepts R_1's fake share is $1/q$. Now we consider the optimal strategy for the adversary $\mathcal{A}_{cheat}$. Given the construction of the CISS scheme, especially the use of (t, n) MRA code with $k + t_{max}$ messages, no matter how the adversary distributes his corruption between the active group and inactive group, he can not get advantage over the MRA code. Thus the optimal choice for $\mathcal{A}_{cheat}$ is to corrupt t players in the active group so that any cheater under her control only needs to get one vote of support from the honest players (since the cheaters will surely support each other). Then, there are $k - t$ honest players whom R_1 can cheat. By the union bound, the probability that R_1 will not be identified as a cheater is at most $(k - t)/q$, which concludes the proof. □

Proof of Theorem 1: Combining Lemmas 1 and 2, it is easy to see that the above scheme is a (t,ϵ)-CISS with $t < k/2$ and $\epsilon = \frac{k-t}{q}$. Let us now calculate the share size. Each player gets her share $\sigma_i = (v_{s_i}, v_{c_i}(x), e_i)$, where $v_{s_i} \in \mathbb{F}_p$, $v_{c_i}(x) \in Poly_t$ and $e_i \in \mathbb{F}_q^{k+t_{max}+1}$. So the share size is $|V_i| = p \cdot q^{t+1+k+t_{max}+1} = pq^{k+t+t_{max}+2}$. Taking $p = |S|$, $q = \frac{k-t}{\epsilon}$ and $t_{max} = min\{t - 1, n - k\}$, one gets the desired results in Theorem 1. Note that for the ease of presentation, we take $t_{max} = t - 1$. □

Remark 5. The restriction on the number of shares present at the reconstruction can be achieved even if more than k players are present. Trivially, the players can decide that only some (e.g., k randomly chosen) shares should be input into the reconstruction algorithm, while the rest of the players never disclose their shares. A problem of this solution is that even a single cheater will be able to disrupt the reconstruction.

If we assume that the active adversary $\mathcal{A}_{cheat}$ can only corrupt the players in the active group (i.e., the players who participate in the reconstruction phase), then we can use a (t, n) MRA code with k messages and get a scheme with even smaller share size. We summarize this observation in Theorem 2.

Theorem 2. *Under the assumption that $\mathcal{A}_{cheat}$ can only corrupt the players in the active group and $t < k/2$, we get a (t,ϵ)-CISS against rushing adversary such that*

$$|S| = p,\ \epsilon = \frac{k-t}{q},\ q \geq n \cdot p,\ |V_i| = p \cdot q^{k+t+2} = \frac{|S|(k-t)^{k+t+2}}{\epsilon^{k+t+2}}. \tag{4}$$

We note that the later CISS scheme has the smallest known share size among existing CISS schemes in the same category.

3.3 (k, n)-CISS without Restriction on Reconstructing Players

In the general case, we may not be able to restrict the number of shares appearing at the reconstruction. Moreover, we may encounter a problem mentioned in Remark 5. Therefore, we extend the construction of the previous subsection to fit a general setting where the number of reconstructing players m can be any value between (and including) k and n.

The general scheme is almost identical to the restricted version in the last subsection except that we use the MRA code capable of authenticating n messages, thus increases the share size slightly. For completeness, we provide the scheme below.

As before, all the players are divided into active group $(R_1, \ldots, R_m)$ and inactive group $(R_{m+1}, \ldots, R_n)$.

CISS scheme with $k \leq m \leq n$ reconstructing players.

Protocol 3 (ShareGen-General)

Public parameter: $x_i \in \mathbb{F}_p$ as player R_i's identifier for $i = 1, \ldots, n$.

Input: Secret $s \in \mathbb{F}_p$.
Output: A list of n shares $\sigma_1, \sigma_2, \ldots, \sigma_n$.

A dealer D performs the following:

1. Generate a random degree-$(k-1)$ polynomial $f_s(x)$ over $\mathbb{F}_p$, such that $f_s(0) = s$. Compute $v_{s,i} = f_s(x_i)$, for $i \in [n]$.
2. Uniformly at random, generate authentication key $e = (P_0(x), \ldots, P_n(x))$ for a (t, n) MRA code with n messages, where $P_i(x) \in Poly_t$ is a polynomial of degree at most t over $\mathbb{F}_q$.
3. For $i \in [n]$, compute $v_{c,i}(x) = f_e(\phi(v_{s,i}, i))$ as the authentication tag for $v_{s,i}$, where $v_{c,i}(x) \in Poly_t$ is a polynomial of degree at most t over $\mathbb{F}_q$.
4. For $i \in [n]$, set $\sigma_i = (v_{s,i}, v_{c,i}(x), e_i)$ and distribute it privately to player R_i, where $e_i = (P_0(x_i), \ldots, P_n(x_i))$ is the verification key of R_i.

Protocol 4 (Reconst-General)

Input: A list of m shares $(\sigma'_1, \ldots, \sigma'_m)$.
Output: Either $(\perp, L)$ or (s', L), where L is the list of cheaters.

Communication rounds performed by each player $i \in [m]$:
Round 1: Announce $(v'_{s,i}, v'_{c,i}(x))$.
Round 2: Announce e'_i.

Computation by players in $[m]$:

1. For $i \in [m]$, do:
   a) Use the verification key e_j to verify the authenticity of $(v'_{s,i}, v'_{c,i}(x))$, for $j \in [m]$.
   b) If less than $t+1$ verification keys accept $(v'_{s,i}, v'_{c,i}(x))$ as authentic, then player R_i is put into the cheater list L.
2. If $m - |L| \geq k$, reconstruct $f'_s(x)$ from $m - |L|$ shares $v'_{s,i}$ using the Lagrange interpolation
   a) If degree of $f'_s(x)$ is at most k, output $(f'_s(0), L)$.
   b) Otherwise output $(\perp, L)$.
3. If $m - |L| < k$, output $(\perp, L)$.

Theorem 3. *If $t < k/2$ then the scheme described above is a (t,ϵ)-CISS against rushing adversary (with no restriction on the number of reconstructing players) such that*

$$|S| = p,\ \epsilon = \frac{n-t}{q},\ q \geq n \cdot p,\ |V_i| = p \cdot q^{n+t+2} = \frac{|S|(n-t)^{n+t+2}}{\epsilon^{n+t+2}}. \tag{5}$$

Proof (sketch). Perfect secrecy is shown by the same argument as in the proof of Lemma 1.

For $(1-\epsilon)$-correctness, note that the rushing adversary $\mathcal{A}_{cheat}$ can observe at most n authentication tags after Round 1 of **Reconst-General**. Since $\mathcal{A}_{cheat}$ can corrupt at most t players, clearly she can get the verification keys of at most t players. Since **ShareGen-General** uses (t, n) MRA code with n messages to authenticate the shares, the argument for $(1-\epsilon)$-correctness follows from that of Lemma 2. An important difference is that this time, there are at most $m - t$ honest players whom $\mathcal{A}_{cheat}$ can cheat. So that the upper bound of cheating probability is $\epsilon = \frac{n-t}{q}$ which is computed for the case when all the players appear at the reconstruction phase.

The remaining task is just to evaluate the share size. Again, we have got $\sigma_i = (v_{s,i}, v_{c,i}(x), e_i)$, where $v_{s,i} \in \mathbb{F}_p$, $v_{c,i}(x) \in Poly_t$ and $e_i \in \mathbb{F}_q^{n+1}$. Therefore, the share size is $|V_i| = p \cdot q^{t+1+n+1} = pq^{n+t+2}$. Taking $p = |S|$ and $q = (n-t)/\epsilon$, we get the results claimed in Theorem 3. □

Remark 6. From the above proof, we can see that the higher ratio of honest players leaks more information to the rushing adversary and provides more targets for the adversary to attack. Thus, it is not surprising that our general scheme requires larger share size than its restricted version. We also note that the proof of Choundury [2] did not pay attention to this phenomenon and the failure probability and share size in their proposal is written incorrectly. The correct share size should be $|V_i| = |S|(n-t)^{3n}/\epsilon^{3n}$ for a single secret rather than $|V_i| = |S|(t+1)^{3n}/\epsilon^{3n}$. However, this does not affect the performance of his scheme in the asymptotic case, since one usually takes $n << 1/\epsilon$.

4 CISS against Non-Rushing Adversary

Our proposal follows the same pattern as the two previous schemes, but now we can perform reconstruction in a single round. Also, we only need to take care of t shares available to the adversary A_{cheat}. This allows us to reduce the share size, as compared to the previous schemes. For completeness, we provide a description of our protocol below.

Protocol 5 (ShareGen-NR)

Public parameter: $x_i \in \mathbb{F}_p$ as player R_i's identifier for $i = 1, \ldots, n$.

Input: Secret $s \in \mathbb{F}_p$.
Output: A list of n shares $\sigma_1, \sigma_2, \ldots, \sigma_n$.

A dealer D performs the following:

1. Generate a random degree-$(k-1)$ polynomial $f_s(x)$ over $\mathbb{F}_p$, such that $f_s(0) = s$. Compute $v_{s,i} = f_s(x_i)$, for $i \in [n]$.
2. Randomly and uniformly generate authentication key $e = (P_0(x), \ldots, P_t(x))$ for a (t, n) MRA code with t messages, where $P_i(x) \in Poly_t$ is a polynomial of degree at most t over $\mathbb{F}_q$.
3. For $i \in [n]$, compute $v_{c,i}(x) = f_e(\phi(v_{s,i}, i))$ as the authentication tag for $v_{s,i}$, where $v_{c,i}(x) \in Poly_t$ is a polynomial of degree at most t over $\mathbb{F}_q$.
4. For $i \in [n]$, set $\sigma_i = \{v_{s,i}, v_{c,i}(x), P_0(x_i), \ldots, P_t(x_i)\}$ and distribute it privately to player R_i.

Without loss of generality, assume that the first $m \geq k$ players want to recover the secret.

Protocol 6 (Reconst-NR)

Input: A list of m shares $(\sigma'_1, \ldots, \sigma'_m)$.
Output: Either $(\perp, L)$ or (s', L), where L is the list of cheaters.

Communication rounds performed by each player $i \in [m]$:
Round 1: Announce $(v'_{s,i}, v'_{c,i}(x), P'_0(x_i), \ldots, P'_t(x_i))$.

Computation by players in $[m]$:

1. For $i \in [m]$, do:
a) Use the verification key $(P'_0(x_j), \ldots, P'_t(x_j))$ to verify the authenticity of $(v'_{s,i}, v'_{c,i}(x))$, for $j \in [m]$.
b) If less than $t+1$ verification keys accept $(v'_{s,i}, v'_{c,i}(x))$ as authentic, then player R_i is put into the cheater list L.

2. If $m - |L| \geq k$, reconstruct $f'_s(x)$ from $m - |L|$ shares $v'_{s,i}$ using Lagrange interpolation
   a) If degree of $f'_s(x)$ is at most k, output $(f'_s(0), L)$.
   b) Otherwise output $(\perp, L)$.
3. If $m - |L| < k$, output $(\perp, L)$.

Theorem 4. *If $t < k/2$ then the scheme described above is a (t,ϵ)-CISS against non-rushing adversary such that*

$$|S| = p,\ \epsilon = \frac{n-t}{q},\ q \geq n \cdot p,\ |V_i| = p \cdot q^{2t+2} = \frac{|S|(n-t)^{2t+2}}{\epsilon^{2t+2}}. \tag{6}$$

Proof (sketch). Perfect secrecy is easy to show similarly to the proof of Lemma 1.

Note that now the active adversary $\mathcal{A}_{cheat}$ is non-rushing. So that she can get the view of at most t players. Since the above scheme uses (t, n) MRA codes with t messages, the adversary can successfully generate a fake share and its authentication tag with probability $1/q$. When all players get involved in the reconstruction phase, there are at most $n - t$ honest players for $\mathcal{A}_{cheat}$ to cheat. Thus, the cheating probability for a cheater is $\epsilon = (n - t)/q$, and the share size follows easily. □

Remark 7. Assume that a trusted third party (usually called a reconstructor) collects the shares from the players, and then runs the reconstruction algorithm on them. If we assume in addition that the parties submit their shares to the reconstructor over point-to-point private channels, then the rushing adversary has exactly the same power as the non-rushing one.

5 Improvement of IWSEC 2013 Scheme

Xu et al. [15] presented a (t, ϵ)-CISS scheme capable of identifying $t < k/3$ rushing cheaters with share size $|V_i| = \frac{|S|}{\epsilon^{n-t+1}}$. We make a proposal to improve the share size of their scheme to $|V_i| = \frac{|S|}{\epsilon^{n-\lfloor (k-1)/3 \rfloor+1}}$. The intuition for the improvement comes from a somewhat counter-intuitive property that the larger number of cheaters t require the smaller share size. Therefore, replacing it with a maximum possible value will lead to improving the share size for any t.

Next, we briefly describe the **ShareGen** protocol by Xu et al.

1. For a secret $s \in \mathbb{F}_p$, the dealer D generates Shamir share $v_{s,i}$ for each player R_i.

2. The dealer D generates a random polynomial $g(x) \in Poly_t$ over $\mathbb{F}_q$ as the authentication key.

3. The dealer authenticates each share $v_{s,i}$ using $g(x)$ and the corresponding tag is $v_{c,i} = g(\phi(v_{s,i}, i))$, where x_i is the public identifier for player R_i and $\phi(\cdot, \cdot)$ is an injective function.

4. For $i \in [t]$, set $\overline{v}_{c,i} = v_{c,i}$; for $i \in [n] \setminus [t]$, set $\overline{v}_{c,i} = v_{c,i} + k_i$ where k_i is the one-time pad key.

5. For $i \in [n] \setminus [t]$ share the key k_i among the n players using a $(t+1, n)$ Shamir secret sharing scheme. Each player R_j's share for k_i is denoted $k_{j,i}$.

6. The share for player R_i is $\sigma = (v_{s,i}, \overline{v}_{c,i}, k_{i,t+1}, \ldots, k_{i,n})$.

Player R_i's share consists of $v_{s,i} \in \mathbb{F}_p$, $\overline{v}_{c,i} \in \mathbb{F}_q$ and $n-t$ shares for the one-time pad keys. So the share size is $|V_i| = p \cdot q^{n-t+1}$. As we mentioned above, the share size increases while the number of cheaters decreases. For example, when there is only one cheater, the share size will get to its maximum $|V_i| = p \cdot q^n$.

Therefore, instead of generating a polynomial of degree at most t, the dealer must always generate a polynomial $g(x)$ of degree at most $\lfloor k-1/3 \rfloor$ to authenticate the Shamir shares in the above step 2. Then, the $\lfloor k-1/3 \rfloor$ of the authentication tags do not need to be encrypted since the polynomial $g(x)$ serves as strongly universal$_{\lfloor k-1/3 \rfloor+1}$ hash function (see the detailed explanation in [15]). Therefore, the number of encryption keys will be reduced to $n - \lfloor k-1/3 \rfloor$. Correspondingly, the share size in Xu et al.'s scheme can be reduced to $|V_i| = |S|/\epsilon^{n-\lfloor (k-1)/3 \rfloor+1}$ that does not depend on the number of cheaters.

6 Conclusion

We presented CISS schemes tolerating $t < k/2$ cheaters, which utilize the properties of multi-receiver authentication codes to reduce the share size, as compared to the existing constructions based on traditional message authentication codes. From our CISS against rushing adversary, we get a somewhat counter-intuitive observation that higher ratio of honest players benefits the rushing adversary. On the one hand, this is true because the rushing adversary gets more information and more targets to attack. On the other hand, this problem might be circumvented by more sophisticated constructions. For example, when more than k players participate in the reconstruction phase, we can incorporate Reed-Solomon error correction into our CISS scheme in order to reduce the success probability of cheaters. This will be a direction for our future work.

Acknowledgments. R.X. is supported by The China Scholarship Council, No. 201206340057. K.M. is supported by a *kakenhi* Grant-in-Aid for Young Scientists (B) 24700013 from Japan Society for the Promotion of Science.

The authors would like to thank the anonymous reviewers of IWSEC 2014 for their helpful comments.

References

1. Blarkley, G.R.: Safeguarding cryptographic keys. In: Proceedings of AFIPS 1979 National Computer Conference, vol. 48, pp. 313–317 (1979)
2. Choudhury, A.: Brief announcement: optimal amortized secret sharing with cheater identification. In: Kowalski, D., Panconesi, A. (eds.) Proceedings of the 2012 ACM Symposium on Principles of Distributed Computing (PODC 2012), pp. 101–102. ACM, New York (2012)

3. Simmons, G.J.: A survey of information authentication. Proceedings of the IEEE 76(5), 603–620 (1988)
4. Desmedt, Y., Frankel, Y., Yung, M.: Multi-receiver/multi-sender network security: efficient authenticated multicast/feedback. In: Eleventh Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 1992, pp. 2045–2054. IEEE (1992)
5. Canetti, R.: Security and composition of multiparty cryptographic protocols. Journal of Cryptology 13(1), 143–202 (2000)
6. Kurosawa, K., Obana, S., Ogata, W.: t-cheater identifiable (k, n) threshold secret sharing schemes. In: Coppersmith, D. (ed.) CRYPTO 1995. LNCS, vol. 963, pp. 410–423. Springer, Heidelberg (1995)
7. Martin, K.M.: Challenging the adversary model in secret sharing schemes. In: Coding and Cryptography II. Proceedings of the Royal Flemish Academy of Belgium for Science and the Arts, pp. 45–63 (2008)
8. McEliece, R.J., Sarwate, D.V.: On sharing secrets and Reed-Solomon codes. Commun. ACM 24(9), 583–584 (1981)
9. Obana, S.: Almost optimum t-Cheater Identifiable secret sharing schemes. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 284–302. Springer, Heidelberg (2011)
10. Rabin, T., Ben-Or, M.: Verifiable secret sharing and multiparty protocols with honest majority. In: Johnson, D.S. (ed.) Proceedings of the Twenty-first Annual ACM Symposium on Theory of Computing (STOC 1989), pp. 73–85. ACM, New York (1989)
11. Reed, I.S., Solomon, G.: Polynomial codes over certain finite fields. J. Soc. Ind. Appl. Math. 8(2), 300–304 (1960)
12. Safavi-Naini, R., Wang, H.: New results on multi-receiver authentication codes. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 527–541. Springer, Heidelberg (1998)
13. Shamir, A.: How to Share a Secret. Commun. ACM 22(11), 612–613 (1979)
14. Tompa, M., Woll, H.: How to share a secret with cheaters. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 261–265. Springer, Heidelberg (1987), Journal version in: J. Cryptol. 1(2), 133–138 (1988)
15. Xu, R., Morozov, K., Takagi, T.: On cheater identifiable secret sharing schemes secure against rushing adversary. In: Sakiyama, K., Terada, M. (eds.) IWSEC 2013. LNCS, vol. 8231, pp. 258–271. Springer, Heidelberg (2013)