On Cheater Identifiable Secret Sharing Schemes Secure Against Rushing Adversary

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

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

Abstract. At EUROCRYPT 2011, Obana proposed a k-out-of-n secret sharing scheme capable of identifying up to t cheaters with probability $1 - \epsilon$ under the condition $t < k/3$. In that scheme, the share size $|V_i|$ satisfies $|V_i| = |S|/\epsilon$, which is almost optimal. However, Obana's scheme is known to be vulnerable to attacks by rushing adversary who can observe the messages sent by the honest participants prior to deciding her own messages. In this paper, we present a new scheme, which is secure against rushing adversary, with $|V_i| = |S|/\epsilon^{n-t+1}$, assuming $t < k/3$. We note that the share size of our proposal is substantially smaller compared to $|V_i| = |S|(t+1)^{3n}/\epsilon^{3n}$ in the scheme by Choudhury at PODC 2012 when the secret is a single field element. A modification of the later scheme is secure against rushing adversary under a weaker $t < k/2$ condition. Therefore, our scheme demonstrates an improvement in share size achieved for the price of strengthening the assumption on t.

Keywords: cheater identifiable secret sharing, Shamir secret sharing, rushing adversary

1 Introduction

Secret sharing, independently introduced by Shamir [1] and Blakley [2], is an important primitive enjoying numerous cryptographic applications such as threshold cryptography [3], secure multiparty computation [4, 5], and (perfectly) secure message transmission [6], to mention a few. A typical example is the *threshold* (or *k-out-of-n*) secret sharing scheme that allows a dealer D to distribute a secret s among a set of n participants (or players) $\{P_1, P_2, \ldots, P_n\}$ in such a way that the following two properties hold: (1) *perfect secrecy*: $k-1$ or less participants can get no information about s from their shares; (2) *correctness*: k or more participants can pool their shares together to reconstruct the secret. In the original setting of secret sharing schemes, it is assumed that all players will provide correct shares when reconstructing the secret. Since this assumption does not model the real life scenario, in which some participants may submit incorrect shares in order to cause the reconstruction of an incorrect secret, a body of work has been done on identifying the cheaters in secret sharing schemes. Next, we will discuss some of the prominent results in this area. If there are more than one cheating participant, we will assume a single malicious adversary who controls their behavior. The adversary is called *rushing*, if she is allowed to observe all the messages sent by honest players (in every round) prior to deciding on cheaters' messages.

1.1 Secret Sharing with Cheaters

In this work, we focus on *secret sharing with cheater identification (SSCI)*. In this setting, the dealer is assumed to be honest. At the reconstruction stage, when a qualified subset of participants pool their shares, they will be able to *identify* cheater(s) among them, who submitted a forged share, as long as the number of cheaters is smaller than a certain bound.

The idea of secret sharing with protection against cheating was pioneered by Tompa and Woll [7]. They modified the (k-out-of-n) Shamir secret sharing scheme [1] to enable the *cheater detection* (not identification). The first secret sharing scheme capable of identifying cheaters is due to Rabin and Ben-Or [4]. Later, McEliece and Sarwate [8] showed that Shamir scheme is cheater identifiable by exhibiting its connection to Reed-Solomon codes. Note that this scheme requires the presence of *more than k* participants in order to carry out cheater identification. In contrast, Kurosawa, Obana and Ogata [9] considered the problem of identifying cheaters when only k players take part in the reconstruction. In particular, they gave a lower bound on the share size in this model:

$$|V_i| \geq \frac{|S|-1}{\epsilon} + 1, \tag{1}$$

where ϵ is the cheaters' success probability.

In this work, we mainly focus on SSCI in the model of Kurosawa *et al.* [9]. They proposed an SSCI scheme identifying $t < k/3$ cheaters in k-out-of-n Shamir secret sharing. Obana [10] improved Kurosawa *et al.*'s scheme by reducing the share size to $|V_i| = |S|/\epsilon$, which is almost optimal, and in addition proposed two (inefficient) SSCI schemes identifying up to $t < k/2$ cheaters.

While the above mentioned schemes can only identify non-rushing cheaters, Choudhury [11] implemented an efficient SSCI scheme which can identify up to $t < k/2$ *rushing* cheaters, achieving the share size $|V_i| = |S|/\epsilon$ when the secret consists of $l = \Omega(n)$ field elements.

Cevallos *et al.* [12] proposed a *robust secret sharing scheme (RSS)* against up to $t < n/2$ rushing cheaters with share size $|V_i| = |S| \left[\log |S| \cdot (t+1)(\frac{e}{\epsilon})^{\frac{2}{t+1}}\right]^{3n}$ which is close-to-optimal. In their scheme, all the n players are required to take part in the reconstruction phase.

We note that in this work, we focus on *public* cheater identification [10,11], where reconstruction is performed such that all the shares are treated equally in terms of their trustworthiness by the reconstruction algorithm. It means that this algorithm can be performed even by an external reconstructor. Such type of schemes are only possible for the case of honest majority.

On the contrary, in the schemes with *private* identification [4, 13], the share received by a user from the honest dealer is assumed to be trusted. Such types of schemes do not need honest majority for cheater identification, as explained in [10]. The concept of identifying cheaters without an honest majority is further developed by Ishai *et al.* in [14].

1.2 Related Works

To the best of our knowledge, up to date, the constructions of [11] and [12] are the most efficient secret sharing schemes secure against rushing adversary, in terms of their share size. Both of them are based on the paradigm by Rabin and Ben-Or [4]. In these schemes, a pairwise authentication is applied to identify cheaters in the reconstruction phase. More precisely, every player receives $n-1$ tags computed according to some unconditional Message Authentication Code (MAC) for every share, and the corresponding keys are distributed to the other $n-1$ players, respectively. Therefore, every player can check the validity of shares belonging to other players. Hereby, cheaters' success probability is bounded by the successful substitution attack probability of the used MAC. The schemes [11] and [12] also employ some additional (novel) techniques on top of this generic procedure.

In Choudhury's scheme [11], the shared secret is a vector $s = (s_1, s_2, \ldots, s_l)$ from $\mathbb{F}_p^l$, where $\mathbb{F}_p$ is some finite field and $l \geq 1$. Every player P_i obtains Shamir sharing sh_i of s element-wise. Then the sharing algorithm uses a MAC to authenticate sh_i as $t_{i,j} = MAC(sh_i, k_{j,i})$ where $k_{j,i}$ held by player P_j is the authentication key chosen uniformly and randomly from some finite field. At the reconstruction phase, a majority voting is taken based on the result of verifying each player's tags. Each player whose share is not recognized by the majority is identified as cheater (thus this scheme is public cheater identifiable). Choudhury's scheme is asymptotically optimal when $l = \Omega(n)$.

The sharing phase of Cevallos *et al.* [12] is identical to that of Ben-Or [4] except for the MAC used. Here, it is assumed that $n = 2t+1$. The sharing algorithm first chooses the secret $s \in \mathbb{F}$ then calculates tags of P_i's Shamir share as $t_{i,j} = MAC(sh_i, k_{j,i})$. At the reconstruction phase player P_i's share sh_i will be accepted as valid only if it is recognized by $t+1$ players *who hold accepted shares.* After that Reed-Solomon error correction is applied to rule out potential cheaters who are not identified by the majority voting. Due to such an advanced reconstruction phase, shorter keys and tags for the MAC can be used in their scheme, as compared to the straightforward approach. Hereby, a reduction in the share size is achieved.

We point out that in fact, Cevallos *et al.* scheme [12] can identify cheaters. Moreover, it can be modified in a straightforward manner, in order to satisfy the property of SSCI such that only k players can identify up to $t < k/2$ cheaters, since it uses the same message authentication and majority voting strategy as [4] and [11]. However, if there are only k players in the reconstruction phase, then Reed-Solomon error correction will not rule out the potential cheaters who are not identified by the majority voting. Let us explain this point in details: For Reed-Solomon error correction to work, the number of correct symbols must be greater than the degree of the polynomial $f(x)$. In other words, Reed-Solomon error correction can only be used to rule out potential cheaters, if there are at least k honest players in the reconstruction phase. On the other hand, when the number of players taking part in reconstruction is $m < k+t$, Cevallos *et al.*'s scheme [12] will lose the ability to identify potential cheaters using Reed-Solomon error correction. This will imply that short keys and tags for the MAC will be no longer secure (the security will then come exclusively from the employed MAC). Therefore, if we modify Cevallos *et al.* [12] to work as a standard SSCI scheme, it will become equivalent to Choudhury's scheme for a single secret.

Let t be the number of cheaters, $|V_i|$ – the size of a share for every player P_i, m – the number of players involved in the reconstruction phase. We summarize the SSCI schemes of [10, 11] and the RSS scheme of [12], and compare them to our proposal in Table 1, unifying the parameters of all these schemes, for the convenience sake.

From Table 1, we observe that Obana's scheme [10] achieved the nearly optimal share size, since the lower bound on the share size is $|V_i| \geq \frac{|S|-1}{\epsilon} + 1$ according to [9]. However, recall that [10] can only deal with non-rushing adversary. Choudhury's scheme [11] is (almost) asymptotically optimal for large secrets, and it has a de-

Table 1. Comparison of Our Proposal to Existing SSCI schemes.

Scheme	# Cheaters	Share Size	Adversary	# Players at Reconstruction
Obana [10]	$t < k/3$	$\lvert V_i\rvert = \lvert S\rvert/\epsilon$	Non-rushing	$m \geq k$
Choudhury [11]	$t < k/2$	$\lvert V_i\rvert = \lvert S\rvert(t+1)^{3n}/\epsilon^{3n}$	Rushing	$m \geq k$
Cevallos *et al.* [12]	$t < n/2$	$\lvert V_i\rvert = \lvert S\rvert[\log\lvert S\rvert \cdot (t+1)(\frac{e}{\epsilon})^{\frac{2}{t+1}}]^{3n}$	Rushing	$m = n$
Our Proposal	$t < k/3$	$\lvert V_i\rvert = \lvert S\rvert/\epsilon^{n-t+1}$	Rushing	$m \geq k$

sirable property of identifying rushing cheaters from the minimal number of shares k. However for a single secret, the share size of this scheme (that is $\frac{|S|(t+1)^{3n}}{\epsilon^{3n}}$) is far from optimal. Cevallos *et al.* [12] scheme working with a single secret achieves nearly optimal share size. However, their scheme requires more than $k+t$ players to identify t rushing cheaters.

Now, an interesting open question is to introduce a secret sharing scheme (with share size smaller than those of the above schemes) for a single secret with the property that only k players can identify rushing cheaters. Our proposal fills this gap for $t < k/3$.

1.3 Our Result

We present an SSCI scheme with public cheater identification which is a k-out-of-n secret sharing identifying up to $t < k/3$ rushing cheaters. The share size of our SSCI scheme is $|V_i| = |S|/\epsilon^{n-t+1}$, its parameters are summarized in Table 1.

Note that in Table 1, we provide the share size of Choudhury's scheme [11] for the case of a single secret ($l = 1$). As we mentioned before, if the scheme by Cevallos *et al.* [12] is modified to be an SSCI with the property of identifying cheaters from the minimum number of shares, it will turn into Choudhury's [11] scheme with $l = 1$.

We emphasize that all the schemes mentioned in Table 1 are *not* directly comparable, however we list them together since they provide the same functionality. Hereby, it will help the reader to place our contribution in the context of SSCI and related schemes.

Our contribution is to achieve a tradeoff among the existing secret sharing schemes with cheaters, in terms of tolerable cheaters (t), required players at reconstruction (m), and the share size ($|V_i|$). Hereby we fill the following gap: When the number of rushing cheaters is less than $k/3$ and only k players take part in the reconstruction, our SSCI scheme is superior to the existing schemes in terms of share size.

The closest related work is the one by Choudhury [11] so that we will now provide a detailed comparison with this scheme. The share size of our scheme is $\frac{(t+1)^{3n}}{\epsilon^{2n+t-1}}$ times smaller than that of [11] (in the case of a single secret). This advantage comes for the price of strengthening requirements on the number of cheaters, that our scheme can tolerate, to $t < k/3$.

Let us elaborate more on the savings in the share size that we obtain by providing a specific example. Let us consider the bit length to be added for the sake of cheater identification (we will call it *redundancy*) – it will be computed by taking a logarithm of $|V_i|$ and subtracting $\log |S|$. Then the bit length of redundancy in Choudhury's scheme and ours are respectively:

$$Red_{Cho} = 3n \log(t+1) + 3n \log(\frac{1}{\epsilon}), \tag{2}$$

$$Red_{Our} = (n-t+1) \log(\frac{1}{\epsilon}). \tag{3}$$

From the above equations we can see that asymptotically, our scheme adds at least 3 times less redundancy as compared with Choudhary's scheme if $\frac{1}{\epsilon} >> n$. In Table 2, we compare the redundancy of our scheme to that of [11], fixing the cheater success probability ϵ to be 2^{-80}. For simplicity, we take $t = \lfloor (n-1)/3 \rfloor$, although the maximal tolerable number of cheaters t is $\lfloor (k-1)/3 \rfloor$ and $\lfloor (k-1)/2 \rfloor$ in our scheme and in [11], respectively. We can see from Table 2 that as n grows larger, our scheme needs less and less redundancy as compared with [11]. In particular, even for $n = 4$, our scheme will need 26.7 bytes of redundancy, which is still 4.5 times less than 120.6 bytes needed for Choudhary's scheme. We emphasize again that the reduction of share size comes for the price of sacrifice on the number of tolerable cheaters.

Table 2. Redundancy Needed for Cheater Identification when $t = \lfloor (n-1)/3 \rfloor$, $\epsilon = 2^{-80}$.

n	Red_{Cho}	Red_{Our}	Red_{Cho}/Red_{Our}
4	120.6 B	26.7 B	4.5
1024	33.2 KB	6.7 KB	5.0
2^{18}	9.0 MB	1.7 MB	5.4

Our scheme (as well as [11] and [12]) has two rounds. In fact, it is round-optimal since Cramer *et al.* [15] showed that two rounds of communication is necessary in the rushing adversary model, if the secret sharing scheme requires an agreement among all honest players. Since our scheme is public cheater identifiable, an agreement among all honest players must indeed be achieved.

2 Preliminaries

Let us first fix some notation. 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 $p > n$. All computation is done in the specified Galois fields.

2.1 Security Model and Communication Model

Throughout the paper, we consider an active rushing adversary with unbounded computational power. By being *rushing* we mean that the adversary can observe the information sent by all the honest players at each communication round, prior to deciding on her own messages. The adversary can adaptively corrupt up to t players (which then will be called *cheaters*) during the whole protocol execution provided that $t < k/3$, where k is the threshold of the secret sharing scheme. As usual in SSCI schemes, we assume that adversary cannot corrupt the dealer D.

We assume that the entities are connected pairwise by private and authenticated channels, and also that broadcast channel is available.

2.2 Secret Sharing

The n players are denoted by $\{P_1, P_2, \ldots, P_n\}$. Let s be the secret chosen by D from some distribution S, and let σ_i be the share distributed to player P_i. The set of P_i's possible shares is denoted by V_i. By a slight abuse of notation, we also use S to denote the random variable induced by s and V_i as the random variable induced by σ_i.

First, we describe k-out-of-n secret sharing scheme by Shamir [1]. A k-out-of-n secret sharing scheme involves a dealer D and n participants $\{P_1, \ldots, P_n\}$, and consists of two algorithms: **ShareGen** and **Reconst**. The **ShareGen** algorithm takes a secret $s \in \mathbb{F}_p$ as input and outputs a list $(\sigma_1, \ldots, \sigma_n)$. Each σ_i is distributed to participant P_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 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 a fixed $x_i \in \mathbb{F}_p$ (where x_i can be seen as a unique identifier for P_i) and send σ_i privately to participant P_i.

Reconst:

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

Remark 1. For simplicity of our presentation, we will henceforth write the identifier of P_i as i, rather than x_i.

Next, we formalize k-out-of-n SSCI schemes. As compared to ordinary secret sharing schemes, we require that the reconstruction algorithm **Reconst** both computes the secret and identifies incorrect shares that 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, moreover $s' = s$ except with negligible probability. If a secret can not be reconstructed from the given shares, then s' is set to $\perp$, while $L = \emptyset$ denotes the fact that no cheater is identified.

Definition 1. *A k-out-of-n SSCI scheme Σ is a tuple $(n, k, S, V, \textbf{\textit{ShareGen}}, \textbf{\textit{Reconst}})$ 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 $V = \{V_1, V_2, \ldots, V_n\}$, where V_i is the set of player P_i's shares.*

- *An algorithm* ***ShareGen****, that takes as input a secret $s \in S$, and outputs a vector of n shares $(\sigma_1, \sigma_2, \ldots, \sigma_n) \in V_1 \times V_2 \times \cdots \times V_n$; and*
- *An algorithm* ***Reconst****, that takes as input a vector $(\sigma'_{i_1}, \sigma'_{i_2}, \ldots, \sigma'_{i_m}) \in V_{i_1} \times V_{i_2} \times \cdots \times V_{i_m}$, and outputs a tuple (s', L), where s' is the reconstructed secret and L is the set of identified cheaters.*

Remember that t denotes the maximum number of cheaters that a rushing adversary can corrupt. We assume that the players in $A^{(t)} = \{P_{i_1}, P_{i_2}, \ldots, P_{i_t}\}$ are corrupted by the rushing adversary. In the SSCI scheme, a cheater $P_{i_j}(1 \leq j \leq t)$ succeeds if **Reconst** fails to identify P_{i_j} as a cheater when P_{i_j} provides a forged share. Note that if P_{i_j} succeeded in cheating, then the reconstructed secret s' is different from the original secret s. Without loss of generality, we assume that at the reconstruction, the corrupted players P_i's are those with the smallest i's in $[n]$.

Definition 2. *The successful cheating probability of player $P_{i_j} \in A^{(t)}$ against the SSCI scheme $\Sigma = (n, k, S, V,$ **ShareGen**, **Reconst**$)$ is defined as*

$$\begin{aligned}&\epsilon(\Sigma, A^{(t)}, P_{i_j}) \\ &= \Pr[(s', L) \leftarrow \textbf{\textit{Reconst}}(\sigma'_{i_1}, \sigma'_{i_2}, \ldots, \sigma'_{i_t}, \sigma_{i_{t+1}}, \ldots, \sigma_{i_k}) \wedge P_{i_j} \notin L : \sigma'_{i_j} \neq \sigma_{i_j}].\end{aligned} \tag{4}$$

Remark 2. In Definition 2, we set $\epsilon(\Sigma, A^{(t)}, P_{i_j})$ to be the successful cheating probability of an *individual* cheater P_{i_j}. Since at most t players can be corrupted, the overall failure probability for the SSCI scheme (i.e., the probability that at least one cheater in $A^{(t)}$ succeeds) can be upper-bounded using the union bound. We choose the individual successful cheating probability instead of the overall failure probability to be in accordance with the definition of Obana [10].

Definition 3. *A k-out-of-n SSCI scheme $\Sigma = (n, k, S, V,$ **ShareGen**, **Reconst**$)$ is called (t, ϵ) SSCI scheme if the following properties hold:*

1. *Perfect secrecy: At the end of the algorithm* ***ShareGen****, any set of players of size at most $k - 1$ have no information about the secret s.*
2. *ϵ-Correctness: $\epsilon(\Sigma, A^{(t)}, P_i) \leq \epsilon$ for any $A^{(t)}$ denoting the set of t or less rushing cheaters, for any cheater $P_i \in A^{(t)}$. If at least k honest players join the reconstruction protocol, the secret will be correctly recovered unless the cheaters remained detected.*

Remark 3. Note that if at least k honest players take part in the reconstruction protocol, successful identification of cheaters is equivalent to recovering the original secret. The secret is not correctly recovered if and only if one or more cheaters are undetected. However, if less than k honest players are available, our scheme can only identify the cheaters with overwhelming probability without recovering the original secret. This is an intrinsic limitation of SSCI schemes since we only require k players to identify the cheaters.

Remark 4. Our protocol, as well as the works of [4, 10–12], prevents false positive error, i.e., honest participants will never be identified as cheaters.

2.3 Reed-Solomon Error Correction

Let $f(x) \in \mathbb{F}_p[X]$ be a polynomial of degree at most k. Let $x_1, x_2, \ldots, x_n \in \mathbb{F}_p$, for $n > k$, be pairwise distinct interpolation points. Then $C = (f(x_1), f(x_2), \ldots, f(x_n))$ is a codeword of Reed-Solomon error correction code [16]. Reed-Solomon code can correct up to $\frac{n-k}{2}$ erroneous symbols, i.e. when t out of n evaluation points $f(x_i)$ $(1 \leq i \leq n)$ are corrupted, the polynomial can be uniquely determined if and only if $n - k > 2t$. Note that there exist efficient algorithms implementing Reed-Solomon decoding, such as Berlekamp-Welch algorithm [17]. We refer the reader to [18] for details on Reed-Solomon codes.

3 Our Proposal

In this section, we describe our k-out-of-n SSCI scheme secure against $t < k/3$ rushing adversary. We suppose that $m \geq k$ participants take part in the reconstruction phase.

3.1 Overview

Our proposal departs from Obana's scheme [10] and improves it in the following manner. Consider k-out-of-n Shamir secret sharing. Since the maximum number of cheaters is $\lfloor (k-1)/3 \rfloor$ and at least k players will take part in the reconstruction phase, Obana [10] uses a polynomial of degree t to compute authentication tags for each player's share. The degree-t polynomial can be recovered given at least $k \geq 3t+1$ players' tags, t of which might be corrupted, using Reed-Solomon decoding (with probability 1). In this scheme, protection against rushing adversary is not provided, since the latter can see all the tags of the k players and recover the polynomial (since $k \geq t+1$). In other words, the adversary can recover the authentication key, so that she will be able to forge authentication tag for an arbitrary value submitted as her share.

In order to deal with this problem, we split the reconstruction phase into two rounds. In the first round, only the Shamir shares and masked authentication tags are revealed. Then in the second round, the masking key will be submitted by each player. We share the masking key between all the n players using a $(t+1)$-out-of-n Shamir secret sharing, such that any t corrupted players can neither get any information about the key nor alter it in the reconstruction phase.

Unfortunately, the necessity to share the masking keys takes the share size of our scheme away from the optimal bound. However, we observe that there is no need to mask *all* of the authentication tags: since the knowledge of any t of them gives no advantage to the adversary, it suffices to mask only $n - t$ of them.

3.2 Our Scheme

Let q be a prime power such that $q \geq n \cdot p$ and let $\phi : \mathbb{F}_p \times [n] \rightarrow \mathbb{F}_q$ be an injective function.

Our proposed scheme is described below.

Protocol 1 (ShareGen)
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 $v_{s,i} = f_s(i)$, for $i = 1, 2, \ldots, n$.
2. Select a random degree-t polynomial $g(x)$ over $\mathbb{F}_q$. Compute $v_{c,i} = g(\phi(v_{s,i}, i))$.
3. (a) For $i = 1, 2, \ldots, t$: Set $\overline{v_{c,i}} = v_{c,i}$;
   (b) For $i = t+1, t+2, \ldots, n$: Randomly and uniformly generate a key $k_i \in \mathbb{F}_q$, and compute $\overline{v_{c,i}} = v_{c,i} + k_i$.
4. For $i = t+1, t+2, \ldots, n$: Generate a random degree-t polynomial $h_i(x)$ over $\mathbb{F}_q$, such that $h_i(0) = k_i$. Compute $k_{i,j} = h_i(j)$, for $j = 1, 2, \ldots, n$.
5. For $i \in [n]$, set $\sigma_i = \{v_{s,i}, \overline{v_{c,i}}, k_{t+1,i}, \ldots, k_{n,i}\}$ and distribute it privately to player P_i.

Remark 5. Note that in Step 2, we must combine player's share $v_{s,i}$ with her identifier i before authentication, since otherwise a cheater can "steal" a share and its authentication tag from some other player pretending that she has received the same share without being detected. However, when we authenticate the combination of the share and the identifier of a player, which is $\phi(v_{s,i}, i)$, the entities to be authenticated will be distinct for every player even if they received the same share, since $\phi(\cdot, \cdot)$ is an injective function.

Let $CORE = \{i_1, i_2, \ldots, i_m\}$ be the set of identifiers of the m participants who want to recover the secret. Moreover, let $\sigma'_{i_j} = \{v'_{s,i_j}, \overline{v'_{c,i_j}}, k'_{t+1,i_j}, \ldots, k'_{n,i_j}\}$ for each $i_j \in CORE$. Furthermore, at most t out of m shares can be corrupted in a rushing fashion.

Protocol 2 (Reconst)

Input: A list of m shares $(\sigma'_{i_1}, \sigma'_{i_2}, \ldots, \sigma'_{i_m})$, where $m \geq k$.
Output: Either $(\perp, L)$ or (s', L), where L is the list of cheaters.

Communication rounds performed by each player $i_j \in CORE$:

1. Announce $\{v'_{s,i_j}, \overline{v'_{c,i_j}}\}$.
2. Announce $\{k'_{t+1,i_j}, k'_{t+2,i_j}, \ldots, k'_{n,i_j}\}$.

Computation by players in $CORE$:

1. For each $i_j \in CORE \bigcap \{t+1, t+2, \ldots, n\}$, reconstruct k'_{i_j} from $\{k'_{i_j,i_1}, \ldots, k'_{i_j,i_m}\}$ using Reed-Solomon decoding.
2. For $i_j \in CORE \bigcap \{1, 2, \ldots, t\}$, set $v'_{c,i_j} = \overline{v'_{c,i_j}}$;
   For $i_j \in CORE \bigcap \{t+1, t+2, \ldots, n\}$, compute $v'_{c,i_j} = \overline{v'_{c,i_j}} - k'_{i_j}$.
3. Reconstruct $g'(x)$ from $v'_{c,i_1}, v'_{c,i_2}, \ldots, v'_{c,i_m}$ using Reed-Solomon decoding.
4. Check if $v'_{c,i_j} = g'(\phi(v'_{s,i_j}, i_j))$ holds for $1 \leq j \leq m$.
   If $v'_{c,i_j} \neq g'(\phi(v'_{s,i_j}, i_j))$ then i_j is added to the list of cheaters L.
5. If $|L| > m - k$ then output $(\perp, L)$, otherwise:
   Reconstruct $f'_s(x)$ from (k or more) shares v'_{s,i_j} such that $i_j \in CORE \setminus L$ using Lagrange interpolation.
   If $\deg(f'_s) \leq k - 1$, output $(f'_s(0), L)$, otherwise output $(\perp, L)$.

Note that the condition $|L| > m - k$ in Step 5 means that the number of honest players is less than k.

Remark 6. For simplicity of our presentation, in the above protocol, we omitted the check similar to that in Step 4, which must be performed in Step 1. In details: When reconstructing in Step 1 the polynomial $h_{i_j}(x)$ (the one used to share the key k_{i_j}) with Reed-Solomon decoding, we must check whether P_i provided a forged share $k'_{i_j,i} \neq k_{i_j,i}$ and put her into the list L, if this is the case. However, we note that in this step, under assumption that $t < k/3$, the cheaters who submitted a forged share will be identified with probability 1.

4 Security Proof

The security of our SSCI scheme is argued in the following theorem.

Theorem 1. *If $t < k/3$ then the scheme described above is a (t,ϵ)* SSCI against rushing adversary *such that*

$$|S| = p,\ \epsilon = 1/q,\ q \geq n \cdot p,\ |V_i| = p \cdot q^{n-t+1} = |S|/\epsilon^{n-t+1}. \tag{5}$$

Proof. First, we show that the scheme satisfy perfect secrecy. Suppose that $k-1$ players $\{P_{i_1}, P_{i_1}, \dots, P_{i_{k-1}}\}$ want to get the secret from their shares. Denote by $\sigma_{i_j} = \{v_{s,i_j}, \overline{v_{c,i_j}}, k_{t+1,i_j}, \dots, k_{n,i_j}\}$ the share of player P_{i_j}. Due to the secrecy of Shamir scheme, the values $(v_{s,i_1}, v_{s,i_2}, \dots, v_{s,i_{k-1}})$ do not reveal any information about the secret. Moreover, it is easy to see that the knowledge about $\overline{v_{c,i_j}}$ and $(k_{t+1,i_j}, k_{t+2,i_j}, \dots, k_{n,i_j})$ does not leak any information about the secret since the polynomial $g(x)$ and the masking keys $(k_{t+1}, k_{t+2}, \dots, k_n)$ are chosen independently of the secret s.

Next we show that our scheme is ϵ-correct. Our proof follows the lines of [10].

Let us observe the following two facts:

1. For $x_1, \dots, x_k \in \mathbb{F}_q$, $(g(x_1), g(x_2), \dots, g(x_k))$ is a codeword of the Reed-Solomon code with minimum distance $k-t$ (since $\deg(g(x)) \leq t$). According to the Reed-Solomon error correction, if $k-t > 2t$ (i.e., $t < k/3$) the degree-t polynomial $g(x)$ can be correctly reconstructed from the k points even if t of them are forged. For the same reason, the masking keys $(k_{t+1}, k_{t+2}, \dots, k_n)$ can be correctly recovered by k players.

2. The set of functions $\{g(x) | g(x) \in \mathbb{F}_q[X], \deg(g(x)) \leq t\}$ is a class of strongly universal$_{t+1}$ hash functions $\mathbb{F}_q \to \mathbb{F}_q$ [19]; that is, the following equality holds for any distinct $x_1, \dots, x_t, x_{t+1} \in \mathbb{F}_q$ and the following $y_1, y_2, \dots, y_t, y_{t+1} \in \mathbb{F}_q$:

$$\Pr[g(x_{t+1}) = y_{t+1} | g(x_1) = y_1, g(x_1) = y_2, \dots, g(x_t) = y_t] = 1/q. \tag{6}$$

Let us suppose without loss of generality that the rushing adversary corrupts $P_{i_1}, \dots, P_{i_t}$ and $CORE \bigcap \{1, 2, \dots, t\} = \{i_1, i_2, \dots, i_l\}$ $(l \leq t)$. Remember that since the adversary is rushing, she can see all the communication of honest players during each round, prior to deciding her own messages. We summarize the view of the adversary in Table 3.

Suppose $P_{i*} \in \{P_{i_1}, P_{i_2}, \overline{\dots}, P_{i_t}\}$, who knows the values $\sigma_{i_1}, \sigma_{i_2}, \dots, \sigma_{i_t}$, submits a forged share $\sigma'_{i*} = (v'_{s,i*}, \overline{v'_{c,i*}}, k'_{i_l+1,i*}, \dots, k'_{i_m,i*})$. P_{i*} is not identified as a cheater only if he submits $\overline{v'_{c,i*}}$ such that $\overline{v'_{c,i*}} = g(\phi(v'_{s,i*}, i*)) + k_{i*}$. At the end of the first round, P_{i*} has to hand in the values $(v'_{s,i*}, \overline{v'_{c,i*}})$. At that time, she can see

Table 3. Adversary's View in **Reconst**.

First Round:	Second Round:
$(v_{s,i_1}, \overline{v_{c,i_1}}, k_{i_{t+1},i_1}, \ldots, k_{i_n,i_1})$	$(v_{s,i_1}, \overline{v_{c,i_1}}, k_{i_{t+1},i_1}, \ldots, k_{i_n,i_1})$
$\cdots$	$\cdots$
$(v_{s,i_t}, \overline{v_{c,i_t}}, k_{i_{t+1},i_t}, \ldots, k_{i_n,i_t})$	$(v_{s,i_t}, \overline{v_{c,i_t}}, k_{i_{t+1},i_t}, \ldots, k_{i_n,i_t})$
$(v_{s,i_{t+1}}, \overline{v_{c,i_{t+1}}})$	$(v_{s,i_{t+1}}, \overline{v_{c,i_{t+1}}}, k_{i_{l+1},i_{t+1}}, \ldots, k_{i_m,i_{t+1}})$
$\cdots$	$\cdots$
$(v_{s,i_m}, \overline{v_{c,i_m}})$	$(v_{s,i_m}, \overline{v_{c,i_m}}, k_{i_{l+1},i_m}, \ldots, k_{i_m,i_m})$

$(\overline{v_{c,i_1}}, \overline{v_{c,i_2}}, \ldots, \overline{v_{c,i_m}})$, $(k_{i_{t+1},i_1}, k_{i_{t+1},i_2}, \ldots, k_{i_{t+1},i_t})$, $\ldots$, $(k_{i_n,i_1}, k_{i_n,i_2}, \ldots, k_{i_n,i_t})$. From $(k_{j,i_1}, k_{j,i_2}, \ldots, k_{j,i_t})$, $(t+1 \le j \le n)$ the cheater P_{i*} can have no information about the masking key k_j since it is shared by the $(t+1)$-out-of-n Shamir scheme. For any $i_j \in CORE \bigcap \{t+1, t+2, \ldots, n\}$, $\overline{v_{c,i_j}} = g(\phi(v_{s,i_j}, i_j)) + k_{i_j}$ looks like a random value to P_{i*}, since k_{i_j} will not be revealed until the second round, and before it serves as a one-time pad. For $i_j \in \{i_1, i_2, \ldots, i_l\}$, P_{i*} will see the values of the function $g(x)$, namely $g(\phi(v_{s,i_j}, i_j)) = v_{c,i_j}$ for $1 \le j \le l$, and $l \le t$. After the second round, all the keys $k_{i_{l+1}}, k_{i_{l+2}}, \ldots, k_{i_m}$ can be correctly reconstructed – since the polynomial $h_j(x)$ hiding k_{i_j} is of degree t and $t < k/3$, we can use Reed-Solomon error correction algorithm to recover the key k_{i_j} despite possibly t corrupted shares of k_{i_j}. By the similar reason, the polynomial $g(x)$ can be correctly reconstructed as well. Since P_{i*} submits the forged share $\sigma'_{i*} = (v'_{s,i*}, \overline{v'_{c,i*}}, k'_{i_l+1,i*}, \ldots, k'_{i_m,i*})$ before he knows the corresponding masking keys, the following holds:

$$\Pr[g(\phi(v'_{s,i*}, i*)) = \overline{v'_{c,i*}} - k_{i*} \,|\, g(\phi(v_{s,i_j}, i_j)) = v_{c,i_j} \ : 1 \le j \le l \, (l \le t)] \le 1/q, \tag{7}$$

where the probability is taken over the random choice of $g(x)$, and $k_{t+1}, k_{t+2}, \ldots, k_n$, and $h_{t+1}(x), h_{t+2}(x), \ldots, h_n(x)$. From the above discussion we can see that any cheater will be identified except with probability at most $1/q$. Therefore, our SSCI scheme satisfies the ϵ-correctness property with $\epsilon = 1/q$.

It is easy to compute the share size as $|V_i| = p \cdot q^{n-t+1} = |S|/\epsilon^{n-t+1}$.

5 Conclusion

We proposed an SSCI scheme capable of identifying $t < k/3$ rushing cheaters. Our scheme is superior to that of Choudhury [11] (for the single secret) and Cevallos *et al.* [12] (if no more than k players can take part in the reconstruction phase), when the number of cheaters is less than $k/3$.

According to the lower bound (1) from Kurosawa *et al.* [9], our scheme is not optimal in the sense of share size $|V_i|$. It is an interesting open problem to design an SSCI scheme against $t < k/2$ (or at least $t < k/3$) *rushing* cheaters with optimal (or at least constant in n, k and t) size of $|V_i|$, even for sharing of a single field element.

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.

K.M. and T.T. would like to thank Satoshi Obana for introducing them to this research topic and for some motivating discussions.

The authors would also like to thank Rui Zhang and anonymous reviewers of IWSEC 2013 for some valuable comments that helped to improve this presentation.

References

1. Shamir, A.: How to share a secret. Commun. ACM **22**(11) (1979) 612–613
2. Blakley, G.: Safeguarding cryptographic keys. In: AFIPS:79 National Computer Conference, IEEE Computer Society (1979) 313–317
3. Desmedt, Y.: Threshold cryptography. European Transactions on Telecommunications **5**(4) (1994) 449–458
4. Rabin, T., Ben-Or, M.: Verifiable secret sharing and multiparty protocols with honest majority (extended abstract). In: STOC 1989. (1989) 73–85
5. Cramer, R., Damgård, I., Maurer, U.: General secure multi-party computation from any linear secret-sharing scheme. In: Advances in CryptologyEUROCRYPT 2000, Springer (2000) 316–334
6. Dolev, D., Dwork, C., Waarts, O., Yung, M.: Perfectly secure message transmission. J. ACM **40**(1) (1993) 17–47
7. Tompa, M., Woll, H.: How to share a secret with cheaters. J. Cryptology **1**(2) (1988) 133–138
8. McEliece, R., Sarwate, D.: On sharing secrets and reed-solomon codes. Commun. ACM **24**(9) (1981) 583–584
9. Kurosawa, K., Obana, S., Ogata, W.: t-cheater identifiable (k, n) threshold secret sharing schemes. In: CRYPTO 1995. (1995) 410–423
10. Obana, S.: Almost optimum t-cheater identifiable secret sharing schemes. In: EUROCRYPT 2011. (2011) 284–302
11. Choudhury, A.: Brief announcement: optimal amortized secret sharing with cheater identification. In: PODC 2012. (2012) 101–102
12. Cevallos, A., Fehr, S., Ostrovsky, R., Rabani, Y.: Unconditionally-secure robust secret sharing with compact shares. In: EUROCRYPT 2012. (2012) 195–208
13. Carpentieri, M.: A perfect threshold secret sharing scheme to identify cheaters. Des. Codes Cryptography **5**(3) (1995) 183–187
14. Ishai, Y., Ostrovsky, R., Seyalioglu, H.: Identifying cheaters without an honest majority. In: TCC. (2012) 21–38
15. Cramer, R., Damgård, I., Fehr, S.: On the cost of reconstructing a secret, or vss with optimal reconstruction phase. In: CRYPTO. (2001) 503–523
16. Reed, I., Solomon, G.: Polynomial codes over certain finite fields. Journal of the Society for Industrial & Applied Mathematics **8**(2) (1960) 300–304
17. Welch, L., Berlekamp, E.: Error correction for algebraic block codes (December 30 1986) US Patent 4,633,470.
18. Roth, R.: Introduction to coding theory. Cambridge University Press (2006)
19. Wegman, M., Carter, L.: New hash functions and their use in authentication and set equality. J. Comput. Syst. Sci. **22**(3) (1981) 265–279