Efficient Threshold Secret Sharing Schemes Secure against Rushing Cheaters

Avishek Adhikari[1], Kirill Morozov[2], Satoshi Obana[3], Partha Sarathi Roy[1], Kouichi Sakurai[4,5], and Rui Xu[6]

[1] Department of Pure Mathematics, University of Calcutta
{avishek.adh,royparthasarathi0}@gmail.com
[2] Institute of Mathematics for Industry, Kyushu University
morozov@imi.kyushu-u.ac.jp
[3] Faculty of Computer and Information Sciences, Hosei University
obana@hosei.ac.jp
[4] Faculty of Information Science and Electrical Engineering, Kyushu University
sakurai@csce.kyushu-u.ac.jp
[5] Information Securty Laboratory, Institute of Systems, Information Technologies and Nanotechnologies (ISIT)
[6] Information Security Laboratory, KDDI R&D Laboratories
ru-xu@kddilabs.jp

Abstract. In this paper, we consider three very important issues namely detection, identification and robustness of k-out-of-n secret sharing schemes against rushing cheaters who are allowed to submit (possibly forged) shares *after* observing shares of the honest users in the reconstruction phase. Towards this we present five different schemes. Among these, first we present two k-out-of-n secret sharing schemes, the first one being capable of detecting $(k-1)/3$ cheaters such that $|V_i| = |S|/\epsilon^3$ and the second one being capable of detecting $n-1$ cheaters such that $|V_i| = |S|/\epsilon^{k+1}$, where S denotes the set of all possible secrets, ϵ denotes the successful cheating probability of cheaters and V_i denotes set all possible shares. Next we present two k-out-of-n secret sharing schemes, the first one being capable of identifying $(k-1)/3$ rushing cheaters with share size $|V_i|$ that satisfies $|V_i| = |S|/\epsilon^k$. This is the first scheme whose size of shares does not grow linearly with n but only with k, where n is the number of participants. For the second one, in the setting of public cheater identification, we present an efficient optimal cheater resilient k-out-of-n secret sharing scheme against rushing cheaters having the share size $|V_i| = (n-t)^{n+2t}|S|/\epsilon^{n+2t}$. The proposed scheme achieves *flexibility* in the sense that the security level (i.e. the cheater(s) success probability) is independent of the secret size. Finally, we design an efficient (k, δ) robust secret sharing secure against rushing adversary with optimal cheater resiliency. Each of the five proposed schemes has the smallest share size having the mentioned properties among the existing schemes in the respective fields.

Keywords: Secret Sharing, Cheating Detection, Cheater Identification, Robustness, Rushing Cheaters, Universal Hash, Reed-Solomon Code.

1 Introduction

In the basic form of secret sharing scheme, it is assumed that everyone involved in the protocol is honest or semi honest. But in the real life scenario, this assumption may not hold good. It may happen that some participants behave maliciously during the execution of the protocol. Malicious participants may submit incorrect shares resulting in incorrect secret. This observation leads to some interesting protocols viz. *cheating*

detectable secret sharing scheme (CDSS), *cheater identifiable secret sharing scheme (CISS)*, *robust secret sharing scheme (RSS)*, *verifiable secret sharing scheme (VSS)*.

However, most schemes known so far implicitly assume existence of *fully* simultaneous network, and they do not deal with cheating by rushing cheaters who submit their shares *after* observing shares of honest users. Constructing a secret sharing scheme secure against rushing cheaters is important in many real life applications. For example, consider the following scenario where there is no trusted party to reconstruct a secret and to detect/identify cheaters, and each user independently reconstructs the secret by receiving shares of the other users sent through the network which is not *fully* simultaneous. Since the network is not fully simultaneous, rushing cheaters can determine how to forge their shares after receiving shares of honest users. Rushing cheaters are so powerful that it is difficult to construct a scheme with small share size. In fact, most of efficient schemes whose bit lengths of shares do not grow linearly with n (i.e., the number of participants) (e.g., [20, 7, 29, 26, 2, 1, 25]) are vulnerable to cheating by rushing cheaters.

In this paper we study k-out-of-n secret sharing schemes secure against rushing cheaters with the following properties:

- secret reconstruction algorithm is allowed to perform multiple rounds of interaction with shareholders,
- t or less (computationally unlimited) rushing cheaters who submit their shares *after* observing shares of honest users are detected/identified with probability $1 - \epsilon$ even when k shares are submitted in secret reconstruction,
- secret will be reconstructed with probability $1 - \epsilon$ even when $k - 1$ or less rushing cheaters manipulate their shares at the time of submission when all the honest participants participate in secret reconstruction,
- probability distribution of a secret does not affect the successful cheating probability ϵ of rushing cheaters.

1.1 State of the Art and Our Results

Tompa and Woll [37] first presented a cheating detectable secret sharing scheme (CDSS). That work is followed by several other works (for example, [2, 1, 12, 7, 26, 29]). However, all these schemes can only detect cheating, without revealing the exact identity of the cheaters, who submitted incorrect shares.

McElice and Sarwate [22] pointed out cheater identification in secret sharing schemes by observing the connection between Shamir threshold scheme and Reed-Solomon error correcting codes. However, such approach requires more than k participants in the reconstruction phase of a (k, n) threshold secret sharing scheme to identify cheaters. The question is whether cheater identification is possible or not with the minimum number of shares (namely k), which are required to reconstruct the secret. *Cheater Identifiable Secret Sharing* (CISS) is the answer to this question. There are two types of cheater identification in secret sharing: *private* as e.g. in [31, 8, 28] and *public* as e.g. in [20, 25, 10, 40]. A reconstruction algorithm of CISS with public cheater identification can be run by an external entity. This is an essential advantage of CISS with public cheater identification over those with private one. However, CISS with public cheater identification is only possible for the case of honest majority [20, 25], while for the case of CISS with private cheater identification honest majority is not required [17]. Many

CISS schemes with different features appear in the literature. The schemes differ on the number of tolerable cheaters, type of the adversary (rushing or not), reconstruction efficiency, and flexibility (security level is flexible or not). We call the scheme *flexible*, when the security level (i.e. success probability of the cheater(s)) can be set independently, i.e., independent of the secret size. Flexibility provides the power of partial customization of length of random strings, according to the requirement.

To have a comparison among the schemes present in the literature, let us first fix the notations. Here, we denote the number of malicious or cheating participants by t in a (k, n) CISS, where $k-1$ denotes the number of semi-honest participants and n denotes the total number of participants. It has been proved in [20] and [25] that a CISS scheme, with public cheater identification, capable of identifying up to t cheaters, is possible if and only if $t < k/2$. So any publicly cheater identifiable CISS scheme with $k = 2t+1$ is said to be an *optimal cheater resilient*. The lower bound [20] on the share size $|V_i|$ of such schemes is $|V_i| \geq \frac{|S|-1}{\epsilon} + 1$, where $|S|$ is the size of the secret and ϵ is cheater's success probability. In [25], two publicly cheater identifiable CISS schemes with optimal cheater resilience were proposed. However, both of them were inefficient. Choudhury [10] came up with an efficient solution, but the scheme in [10] deals with multiple secrets. In the case of a single secret, the scheme of [10] is not an optimal one. One improvement came from Xu et al. [40] but they did not achieve the optimal share size. Moreover, their scheme is not an optimal cheater resilient as it tolerates $t < k/3$ cheaters. In [34], Roy et al. provided a CISS scheme with better share size than [10] with optimal cheater resilience. Xu et. al. [41] further proposed an optimal cheater resilient CISS with improved share size.

Though, cheating detectable and identifiable schemes make it apparent that cheating has occurred, they do not necessarily permit honest participants to recover the correct secret. This observation led to another important primitive in the literature of secret sharing known as *robust secret sharing schemes* [31]. Informally, robust secret sharing schemes allow the correct secret to be recovered even when some of the shares presented during an attempted reconstruction are incorrect. Informally, the main goal of RSS is to ensure successful reconstruction of a correct secret (possibly from more than a threshold of k shares), while disregarding identities of the cheaters. RSS provides the guarantee of *correct* secret reconstruction even in presence of malicious participants. In case of up to k cheaters among n ($\geq 3k-2$) participants, it was observed by McEliece and Sarwate [22] that Shamir's secret sharing scheme [35] is by itself robust via its connection to the Reed-Solomon codes. However, for the case when $n = 2k-1$, the above observation does not work. One solution to this problem considered, e.g., by Rabin and Ben-Or [31] is for the dealer to authenticate shares using some message authentication code [38], resulting a large *overhead* which is defined as the total share size minus the size of the secret. In other words, the overhead can be seen as the number of bits added to the share in order to achieve security against malicious adversary. Therefore, the main point in optimization of robust secret sharing is to reduce the *overhead* needed for ensuring robustness while efficiently reconstructing the secret. If efficient reconstruction is not required and $n \geq 2k$ then one may use the ideal (i.e. without any overhead) scheme by Jhanwar and Safavi-Naini [18]. The case $n \geq 2k-1$ can also be handled by the scheme of Cramer et al. [12] which features a constant overhead. Finally, a (quasi-)linear overhead in the number of players and the security parameter with efficient reconstruction was achieved by Cevallos et al. [9].

Roy et al. [33] further reduced the overhead of Cevallos et al. scheme [9] by applying an authentication tag compression technique of Carpentieri [8]. Very recently, Cramer et al. [11] presented a robust secret sharing scheme with constant share size. But unlike our work, this scheme only works for non-rushing cheaters.

Our Contribution: The contributions of the paper are to present five efficient k-out-of-n secret sharing schemes.

1. We present two CDSS which are all the first scheme in each model such that the bit length of shares does not grow linearly with n. We compare the properties of the existing CDSS schemes in Table 1.

Table 1. Comparison of Proposed CDSS with Existing CDSS

Scheme	#Cheaters	Share Size	Rushing
[7]	$t < k$	$\|V_i\| = \|S\|/\epsilon^2$	No
[27]	$t < n$	$\|S\|(\frac{k-1}{\epsilon})^2$	No
Proposed	$t < k/3$	$\|V_i\| = \|S\|/\epsilon^3$	Yes
Proposed	$t < n$	$\|V_i\| = \|S\|/\epsilon^{k+1}$	Yes

2. We also present two public CISS schemes against rushing cheaters. We compare the properties of existing CISS schemes with public cheater identification in Table 2.

Table 2. Comparison of Proposed CISS with Existing CISS

Scheme	#Cheaters	Share Size	Efficiency	Rushing	Flexibility
[20]	$t < k/3$	$\|V_i\| = \|S\|/\epsilon^{t+2}$	Yes	No	Yes
[25]	$t < k/3$	$\|V_i\| = \|S\|/\epsilon$	Yes	No	Yes
[40]	$t < k/3$	$\|V_i\| = \|S\|/\epsilon^{n-t+1}$	Yes	Yes	No
Proposed	$t < k/3$	$\|V_i\| = \|S\|/\epsilon^k$	Yes	Yes	No
[25]	$t < k/2$	$\|V_i\| \approx (n.(t+1).2^{3t-1}\|S\|)/\epsilon$	No	No	No
[25]	$t < k/2$	$\|V_i\| \approx ((n.(t+1).2^{3t})^2\|S\|)/\epsilon$	No	No	No
[10]*	$t < k/2$	$\|V_i\| = (t+1)^{3n}\|S\|/\epsilon^{3n}$	Yes	Yes	No
[34]	$t < k/2$	$\|V_i\| = (t+1)^{2n+k-3}\|S\|/\epsilon^{2n+k-3}$	Yes	Yes	No
[41]**	$t < k/2$	$\|V_i\| = (n-t)^{n+t+2}\|S\|/\epsilon^{n+t+2}$	Yes	Yes	No
Proposed	$t < k/2$	$\|V_i\| = (n-t)^{n+2t}\|S\|/\epsilon^{n+2t}$	Yes	Yes	Yes

* Share size with respect to a single secret.

** With an additional assumption that the adversary can only corrupt k participants who take part in the reconstruction phase, Xu et al. get even smaller share size, however we list their share size in the general case for a fair comparison.

From Table 2 we can see that our share size is smaller than all the other schemes within the same category except for that of Xu et. al. [41]. But, in case of $t = 1$, share size of proposed CISS is smaller than [41] and same in case of $t = 2$. However our scheme provides flexibility in the security level which is not a feature of the scheme

[41]. We achieve flexibility by adapting authentication technique from [39]. Recently, Xu et. al. presented a scheme capable of identifying up to $(k-1)/3$ rushing cheaters [40]. The size of shares $|V_i|$ of the scheme is $|V_i| = |S|/\epsilon^{n-t+1}$ and the bit size of shares still grows linear in n. We proposed a scheme capable of identifying $(k-1)/3$ rushing cheaters which achieves the smallest size of shares when $k < \frac{3}{4}n+1$ and the bit size of shares does not grow linear in n.

3. Our last scheme is a RSS scheme secure against rushing cheaters. We compare the share sizes of the existing efficient RSS schemes in Table 3. In this paper, we further reduce the overhead of construction proposed in [33] by adapting authentication technique from [39]. Since the schemes [9] and [33] are nearly-optimal, we achieve a constant factor (2/3) improvement in the overhead.

Table 3. Comparison of Proposed RSS with Existing Efficient RSS

Scheme	Overhead (bits)
Rabin and Ben-Or [31]	$3(n-1)(2\log(k)+\mu)$
Cevallos et al. [9]	$3(n-1)(\log(k)+\log(m)+\frac{2}{k}(\mu+\log(e)))$
Roy et al. [33]	$(2n+k-3)(\log(k)+\log(N)+\frac{2}{k}(\mu+\log(e)))$
Proposed	$(n+k-1)(\log(k)+\log(N)+\frac{2}{k}(\mu+\log(e)))$

Here, m is the bit length of the secret and m is an integer multiple of N, $k-1$ is the number of cheaters, $n = 2k-1$ is the number of total participants, $e = exp(1)$, and μ is the security parameter *s.t.* the scheme fails to reconstruct the authentic secret with probability at most $2^{-\mu}$.

2 Preliminaries

2.1 Secret Sharing Schemes

In the model of secret sharing schemes, there are n users $\mathcal{P} = \{P_1, \ldots, P_n\}$ and a dealer D. The set of users who are allowed to reconstruct the secret is characterized by an *access structure* $\Gamma \subseteq 2^{\mathcal{P}}$; that is, users $P_{i_1}, \ldots, P_{i_k}$ are allowed to reconstruct the secret if and only if $\{P_{i_1}, \ldots, P_{i_k}\} \in \Gamma$. The model consists of two algorithms: a share generation algorithm $\mathsf{ShareGen}$ and a secret reconstruction algorithm $\mathsf{Reconst}$. The share generation algorithm $\mathsf{ShareGen}$ takes a secret $s \in \mathcal{S}$ as input and outputs a list $(v_1, v_2, \ldots, v_n)$. Each $v_i \in \mathcal{V}_i$ is called a *share* and is given to a user P_i. In a usual setting, $\mathsf{ShareGen}$ is invoked by the dealer. The secret reconstruction algorithm $\mathsf{Reconst}$ takes a list of shares and outputs a secret $s \in \mathcal{S}$.

A secret sharing scheme $\mathbf{SS} = (\mathsf{ShareGen}, \mathsf{Reconst})$ is called *perfect* if the following two conditions are satisfied for the output $(v_1, \ldots, v_n)$ of $\mathsf{ShareGen}(\hat{s})$ where the probabilities are taken over the random tape of $\mathsf{ShareGen}$.

1. if $\{P_{i_1}, \ldots, P_{i_k}\} \in \Gamma$ then $\Pr[\mathsf{Reconst}(v_{i_1}, \ldots, v_{i_k}) = \hat{s}] = 1$,
2. if $\{P_{i_1}, \ldots, P_{i_k}\} \notin \Gamma$ then $\Pr[\mathcal{S} = s \mid \mathcal{V}_{i_1} = v_{i_1}, \ldots, \mathcal{V}_{i_k} = v_{i_k}] = \Pr[\mathcal{S} = s]$ for any $s \in \mathcal{S}$.

We note that only perfect secret sharing schemes are dealt with in this paper.

2.2 Cheating Detectable Secret Sharing against Rushing Cheaters

Tompa and Woll [37] considered the scenario in which cheaters who do not belong to the access structure submit forged shares in the secret reconstruction phase. Such cheaters will succeed if the other users participating in the reconstruction accept an incorrect secret. In this paper, we consider very powerful cheaters called *rushing cheaters* who submit forged shares *after* observing shares of honest users.

As in the ordinary secret sharing schemes, the model of cheating detectable secret sharing scheme against rushing cheaters consists of two algorithms. A share generation algorithm $\mathsf{ShareGen}$ is the same as that in the ordinary secret sharing schemes. A secret reconstruction algorithm $\mathsf{Reconst}$ is slightly changed: the reconstruction algorithm is modeled as an interactive Turing machine which interacts with users multiple times, and users release a part of their shares to $\mathsf{Reconst}$ in each round. Therefore, $\mathsf{Reconst}$ takes round identifier rid, user identifier P_i, and part of share $v_i^{(rid)}$ and state information state_R as input and outputs updated state information. When interactions with users are finished, $\mathsf{Reconst}$ outputs either the secret or the special symbol $\perp$ ($\perp \notin \mathcal{S}$.) $\mathsf{Reconst}$ outputs $\perp$ if and only if cheating has been detected.

Figure 1 below models the interaction between users and the reconstruction algorithm $\mathsf{Reconst}$. Here, a pair of Turing machine $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ representing rushing cheaters $P_{i_1}, \ldots, P_{i_t}$ who try to cheat honest users $P_{i_{t+1}}, \ldots, P_{i_m}$. In the $\mathsf{Game}^{\mathsf{Rushing}}(\mathbf{SS}$,

$\underline{\mathsf{Game}^{\mathsf{Rushing}}(\mathbf{SS}, \mathcal{A})}$
$s \leftarrow \mathcal{S}$ *(according to the probability distribution over $\mathcal{S}$)*
$(v_1, \ldots, v_n) \leftarrow \mathsf{ShareGen}(s)$
$((i_1, \ldots, i_t), (i_{t+1}, \ldots, i_m), \mathsf{state}_C) \leftarrow \mathcal{A}_1()$
$\mathsf{state}_R \leftarrow \emptyset$
for $rid = 1$ **to** `RidMax` **do**
  for $\ell = t+1$ **to** m **do**
    $\mathsf{state}_R \leftarrow \mathsf{Reconst}(rid, P_{i_\ell}, v_{i_\ell}^{(rid)}, \mathsf{state}_R)$
  done
  $((v_{i_1}^{\prime(rid)}, \ldots, v_{i_t}^{\prime(rid)}), \mathsf{state}_C) \leftarrow \mathcal{A}_2(rid, (v_{i_{t+1}}^{(rid)}, \ldots, v_{i_m}^{(rid)}), (v_{i_1}, \ldots, v_{i_t}), \mathsf{state}_C)$
  for $\ell = 1$ **to** t **do**
    $\mathsf{state}_R \leftarrow \mathsf{Reconst}(rid, P_{i_\ell}, v_{i_\ell}^{\prime(rid)}, \mathsf{state}_R)$
  done
done
$\mathsf{output} \leftarrow \mathsf{Reconst}(\cdot, \cdot, \mathsf{state}_R)$

Fig. 1. Game between $\mathsf{Reconst}$ and rushing cheaters for CDSS and CISS

$\mathcal{A}$), $\mathcal{A}_1$ first chooses rushing cheater $P_{i_1}, \ldots, P_{i_t}$ to cheat users $P_{i_{t+1}}, \ldots, P_{i_m}$. Next, in each round, $\mathcal{A}_2$ determines forged share $(v_{i_1}^{\prime(rid)}, \ldots, v_{i_1}^{\prime(rid)})$ to be submitted by rushing cheaters. Note that $\mathcal{A}_2$ takes shares $(v_{i_{t+1}}^{(rid)}, \ldots, v_{i_m}^{(rid)})$ as input in determining forged shares, which captures the rushing capability of cheaters.

The successful cheating probability $\epsilon(\mathbf{SS}, \mathcal{A})$ of the cheaters $\mathcal{A}$ against $\mathbf{SS} = (\mathsf{ShareGen}, \mathsf{Reconst})$ is defined by

$$\epsilon(\mathbf{SS}, \mathcal{A}) = \Pr[s' \leftarrow \mathsf{Reconst}(\cdot, \cdot, \mathsf{state}_R) : s' \in \mathcal{S} \wedge s' \neq s] \,,$$

where the probability is taken over the distribution of $\mathcal{S}$, and the random tapes of ShareGen and $\mathcal{A}$. The security of cheating detectable secret sharing schemes against t rushing cheaters are defined as follows:

Definition 1. *A k-out-of-n secret sharing $\mathbf{SS} = (\mathit{ShareGen}, \mathit{Reconst})$ is called (t, ϵ) cheating detectable against rushing cheaters if $\epsilon(\mathbf{SS}, \mathcal{A}) \leq \epsilon$ for any adversary $\mathcal{A}$.*

2.3 Cheater Identifiable Secret Sharing against Rushing Cheaters

The model of cheater identifiable secret sharing also consists of a share generation algorithm ShareGen and a secret reconstruction algorithm Reconst. As in the model of (t, ϵ) cheating detectable secret sharing, ShareGen takes a secret as input and outputs a list of shares $(v_1, \ldots, v_n)$ and Reconst is also modeled as interactive Turing machine which interacts with users multiple times. The input of Reconst is the same as cheating detectable secret sharing, but final output is slightly different: Reconst in cheater identifiable secret sharing outputs $(\hat{s}, \emptyset)$ if no cheating is detected. On the other hand, if Reconst detects cheating, it outputs $(\perp, L)$ where $\perp$ is a special symbol indicating detection of cheating and L is a list of cheaters.

The security of cheater identifiable secret sharing is formalized through the same game defined in Figure 1. The cheater P_{i_j}, submitting an invalid share, succeeds in cheating if Reconst fails to identify P_{i_j} as a cheater. The successful cheating probability of P_{i_j} against $\mathbf{SS} = (\mathsf{ShareGen}, \mathsf{Reconst})$ is denoted as $\epsilon(\mathbf{SS}, \mathcal{A}, P_{i_j})$ where the probability $\epsilon(\mathbf{SS}, \mathcal{A}, P_{i_j})$ is defined by

$$\epsilon(\mathbf{SS}, \mathcal{A}, P_{i_j}) = \Pr[(s', L) \leftarrow \mathsf{Reconst}(\cdot, \cdot, \mathsf{state}_R) : i_j \notin L].$$

Based on the above definition, we define the security of secret sharing schemes capable of identifying cheaters who submit forged shares as follows:

Definition 2. *A k-out-of-n threshold secret sharing scheme $\mathbf{SS} = (\mathit{ShareGen}, \mathit{Reconst})$ is called a (t, ϵ) cheater identifiable secret sharing scheme if (1) $\epsilon(\mathbf{SS}, \mathcal{A}, P_j) \leq \epsilon$ for any $\mathcal{A}$ representing set of t or less cheaters L, and for any cheater $P_j \in L$ who submits forged share $v'_j \neq v_j$. (2) $P_i \notin L$ for any user P_i who does not forge its share.*

2.4 Robust Secret Sharing against Rushing Cheaters

McEliece and Sarwate [22] considered the scenario in which cheaters submit forged shares in the secret reconstruction phase. Such cheaters will succeed if the other users participating in the reconstruction accept an incorrect secret. In this paper, we consider *rushing cheaters*.

As in the ordinary (k, n) secret sharing schemes, the model of robust secret sharing scheme against rushing cheaters consists of two algorithms. A share generation algorithm ShareGen is the same as that in the ordinary secret sharing schemes. A secret reconstruction algorithm Reconst is changed: the reconstruction algorithm is modeled as an interactive Turing machine which interacts with users multiple times, and users release a part of their shares to Reconst in each round. Therefore, Reconst takes round identifier rid, user identifier P_i, and part of share $v_i^{(rid)}$ and state information state_R as input and outputs updated state information. When interactions with users are finished, Reconst outputs the secret.

Figure 2 below models the interaction between users and the reconstruction algorithm Reconst. Here, a pair of Turing machine $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ representing rushing cheaters $P_{i_1}, \ldots, P_{i_{k-1}}$ who try to cheat honest users $P_{i_k}, \ldots, P_{i_n}$. In the $\mathsf{Game}^{\mathsf{Rushing}}(\mathbf{SS}$,

$\underline{\mathsf{Game}^{\mathsf{Rushing}}(\mathbf{SS}, \mathcal{A})}$
$s \leftarrow \mathcal{S}$ *(according to the probability distribution over $\mathcal{S}$)*
$(v_1, \ldots, v_n) \leftarrow \mathsf{ShareGen}(s)$
$((i_1, \ldots, i_{k-1}), (i_k, \ldots, i_n), \mathsf{state}_C) \leftarrow \mathcal{A}_1()$
$\mathsf{state}_R \leftarrow \emptyset$
for $rid = 1$ **to** `RidMax` **do**
  for $\ell = k$ **to** n **do**
    $\mathsf{state}_R \leftarrow \mathsf{Reconst}(rid, P_{i_\ell}, v_{i_\ell}^{(rid)}, \mathsf{state}_R)$
  done
  $((v_{i_1}^{\prime(rid)}, \ldots, v_{i_{k-1}}^{\prime(rid)}), \mathsf{state}_C) \leftarrow \mathcal{A}_2(rid, (v_{i_k}^{(rid)}, \ldots, v_{i_n}^{(rid)}), (v_{i_1}, \ldots, v_{i_{k-1}}), \mathsf{state}_C)$
  for $\ell = 1$ **to** $k-1$ **do**
    $\mathsf{state}_R \leftarrow \mathsf{Reconst}(rid, P_{i_\ell}, v_{i_\ell}^{\prime(rid)}, \mathsf{state}_R)$
  done
done
$\mathsf{output} \leftarrow \mathsf{Reconst}(\cdot, \cdot, \mathsf{state}_R)$

Fig. 2. Game between Reconst and rushing cheaters for RSS

$\mathcal{A}$), $\mathcal{A}_1$ first chooses rushing cheater $P_{i_1}, \ldots, P_{i_{k-1}}$ and users $P_{i_k}, \ldots, P_{i_n}$ to cheat. Next, in each round, $\mathcal{A}_2$ determines forged share $(v_{i_1}^{\prime(rid)}, \ldots, v_{i_{k-1}}^{\prime(rid)})$ to be submitted by rushing cheaters. Note that $\mathcal{A}_2$ takes shares $(v_{i_k}^{(rid)}, \ldots, v_{i_n}^{(rid)})$ as input in determining forged shares, which captures the rushing capability of cheaters.

The successful cheating probability $\epsilon(\mathbf{SS}, \mathcal{A})$ of the cheaters $\mathcal{A}$ against $\mathbf{SS} = (\mathsf{ShareGen}, \mathsf{Reconst})$ is defined by

$$\epsilon(\mathbf{SS}, \mathcal{A}) = \Pr[s' \leftarrow \mathsf{Reconst}(\cdot, \cdot, \mathsf{state}_R) : s' \in \mathcal{S} \wedge s' \neq s] \,,$$

where the probability is taken over the distribution of $\mathcal{S}$, and the random tapes of ShareGen and $\mathcal{A}$. The security of robust secret sharing schemes against $k-1$ rushing cheaters are defined as follows:

Definition 3. *A k-out-of-n secret sharing $\mathbf{SS} = (\mathsf{ShareGen}, \mathsf{Reconst})$ is called ϵ robust against rushing cheaters if $\epsilon(\mathbf{SS}, \mathcal{A}) \leq \epsilon$ for any adversary $\mathcal{A}$.*

2.5 Building Blocks of Proposed Schemes

In this subsection we briefly review building blocks of proposed schemes: Reed-Solomon code, almost strongly universal family of hash function, and k-wise independent random variables.

Strongly Universal Family of Hash Function: Here, we will review the definitions and constructions of strongly universal families of hash function.

Definition 4. *A family of hash function $H : A \to B$ is called ϵ* almost strongly universal family of hash function (ϵ-ASU_2 for short) *if it satisfies following two conditions:*

1. $|\{h \mid h \in H,\ h(a) = b\}| = |H|/|B|$ *holds for any* $a \in A$ *and for any* $b \in B$.
2. *For any distinct* $a, a' \in A$ *and for any* $b, b' \in B$, *the following equality holds:*

$$\frac{|\{h \mid h \in H,\ h(a) = b,\ h(a') = b'\}|}{|\{h \mid h \in H,\ h(a) = b\}|} \leq \epsilon$$

H is called strongly universal family of hash function (SU_2 for short) *if* $\epsilon = 1/|B|$.

We define a key e of ASU_2 to specify an element of a hash family H and use a notation h_e to denote an element of the H specified by the key e. It is obvious that the size of keys of hash family is identical to the size of hash family $|H|$.

The subscripts 2 of ASU_2 denotes the strength of hash families, and we can define the notion of strongly universal hash family SU_t of strength t for $t \geq 2$ as follows:

Definition 5. *A family of hash function* $H : A \to B$ *is called* strongly universal family of hash function of strength t (SU_t for short) *if* $|\{h \mid h \in H,\ h(a_i) = b_i (i = 1, \ldots, j)\}| = |H|/|B|^j$ *hold for any distinct* $a_1, a_2, \ldots, a_j \in A$, *for any (not necessarily distinct)* $b_1, b_2, \ldots, b_j \in B$ *and for any* $j \leq t$.

Here, we will review some known constructions of (almost) strongly hash families which we will use in the proposed schemes. In [15], the following efficient ASU_2 based on polynomials over a finite field is proposed.

Proposition 1. *Let* $e_0, e_1 \in \mathbb{F}_p$ *and* $s = (s_1, \ldots, s_N) \in \mathbb{F}_{p^N}$, *then the hash family* $H_{1,N} : \mathbb{F}_{p^N} \to \mathbb{F}_p$ *defined by* $H_{1,N} = \{h_{1,N,(e_0,e_1)} \mid h_{e_0,e_1}(s) = e_0 + s_1 e_1 + s_2 e_1^2 + \cdots s_N e_1^N\}$ *is* N/p-ASU_2. *Furthermore, the hash family* $H_{1,1} : \mathbb{F}_p \to \mathbb{F}_p$ *is* SU_2.

It is well known that strongly universal hash families with higher strength are also constructed based on polynomials over a finite field as follows:

Proposition 2. *Let* $e_0, e_1, \ldots, e_{t-1} \in \mathbb{F}_p$ *and* $s \in \mathbb{F}_p$, *then the hash family* $H_{2,t} : \mathbb{F}_p \to \mathbb{F}_p$ *defined by* $H_{2,t} = \{h_{2,t,(e_0,e_1,\ldots,e_{t-1})} \mid h_{2,t,(e_0,e_1\ldots,e_{t-1})}(s) = e_0 + e_1 s + e_2 s^2 + \cdots e_{t-1} s^{t-1}\}$ *is* SU_t.

Proposition 3. *Let* $e_0, e_1 \in \mathbb{F}_p$ *and* $s_i = (s_{i,1}, \ldots, s_{i,N}) \in \mathbb{F}_{p^N}$, *then the hash family* $H_{3,N} : \mathbb{F}_{p^N} \to \mathbb{F}_p$ *defined by* $H_{3,N} = \{h_{3,N,(e_0,e_{1,1},e_{1,2},\ldots,e_{1,n})} \mid h_{e_0,e_{1,i}}(s_i) = \Sigma_{l=1}^{N} e_0^l . s_{i,l} + e_{1,i}\}$ *is* N/p-ASU_2. *Furthermore,* $h_{3,N,(e_0,e_{1,1},e_{1,2},\ldots,e_{1,n})}$ *an be used to authenticate* n *messages.*

This proposition is an adaption of the universal hash family by Wegman and Carter [39].

Proof. Denote by $a_i = h_{e_0,e_{1,i}}(s_i)$, we calculate the probability that the forged message and authentication tag are accepted by the authentication key. We assume the opponent tries her forge after seeing n pairs of message and tag pairs $\{s_1, a_1, \ldots, s_n, a_n\}$ and substitutes all the n messages and tags by $\{s'_1, a'_1, \ldots, s'_n, a'_n\}$. The substitution probability is

$$\Pr[\text{at least one forged message-tag pairs in } \{s'_1, a'_1, \ldots, s'_n, a'_n\} \text{ is accepted} \mid \{s_1, a_1, \ldots, s_n, a_n\} \text{ are accepted }]$$

Denote by E_1 the event that "at least one forged message-tag pairs in $\{s'_1, a'_1, \ldots, s'_n, a'_n\}$ is accepted", and by E_2 the event that "$\{s_1, a_1, \ldots, s_n, a_n\}$ are accepted ".

We first count how many keys satisfy event E_2. This is given by the following system of equations.

$$\Sigma_{l=1}^{N} e_0^l . s_{1,l} + e_{1,i} = a_1 \quad (1)$$
$$\Sigma_{l=1}^{N} e_0^l . s_{2,l} + e_{1,i} = a_2 \quad (2)$$
$$\cdots = \cdots \quad (3)$$
$$\Sigma_{l=1}^{N} e_0^l . s_{n,l} + e_{1,i} = a_n \quad (4)$$

Taking $e_0, e_{1,1}, e_{1,2}, \ldots, e_{1,n}$ as unknowns, for any fixed value of e_0, there exists one and only one solution for this system of equations. Thus, there are in total p solutions to the following equation system. More importantly, each possible key appears with equal probability.

Next we count how many keys satisfy both events E_1 and E_2. The keys which satisfy both events must *in addition* satisfy *at least one* of the following n equations. In the following equations, we assume that $s'_{i,j} \neq s_{i,j}$ for all $j \in [N]$. This is quite a natural assumption since the opponent wants to forge the messages she would choose a different message other than what is sent by the transmitter. The reason that we assume all the forged messages are different from the authentic ones is simply because this case maximizes the successful probability of a substitution attack by the opponent.

$$\Sigma_{l=1}^{N} e_0^l . s'_{1,l} + e_{1,i} = a'_1 \quad (5)$$
$$\Sigma_{l=1}^{N} e_0^l . s'_{2,l} + e_{1,i} = a'_2 \quad (6)$$
$$\cdots = \cdots \quad (7)$$
$$\Sigma_{l=1}^{N} e_0^l . s'_{n,l} + e_{1,i} = a'_n \quad (8)$$

If we subtract Eq.(9) by Eq.(13), we get $\Sigma_{l=1}^{N} e_0^k . (s_{1,l} - s'_{1,l}) = a_1 - a'_1$. We rephrase it as $f(e_0) = \Sigma_{l=1}^{N} e_0^k . (s_{1,l} - s'_{1,l}) - (a_1 - a'_1) = 0$. The function $f(e_0)$ represents a polynomial of degree at most N in the variable e_0. Since $s_1 \neq s'_1$, for any (s'_1, a'_1) of the opponent's choice, $f(e_0) \neq 0$. Thus there exist at most N values of e_0 satisfying Eqs.(9) and Eq.(13). We record these values $e^*_{0,1,1}, \ldots, e^*_{0,1,N}$. This means, the forged message-tag pair s'_1, a'_1 would not be accepted as authentic if and only $e_0 \in \mathbb{F} \setminus \{e^*_{0,1,1}, \ldots, e^*_{0,1,N}\}$. The same arguments hold for message-tag pairs s'_i, a'_i, that is s'_i, a'_i will not be accepted as authentic if and only if $e_0 \in \mathbb{F} \setminus \{e^*_{0,i,1}, \ldots, e^*_{0,i,N}\}$, for any $i \in [n]$. So for any value of $\{s'_1, a'_1, \ldots, s'_n, a'_n\}$, there are at most $p - (p - N \cdot n) = N \cdot n$ keys satisfying the first equation system and at least one equation from the second equation system. In a nutshell, there are in total at most $N \cdot n$ keys satisfying both event E_2 and event E_1. $\Pr[E_1|E_2] \leq N \cdot n/p$. It is easy to show that for any particular (s'_i, a'_i) with $s'_i \neq s_i$, the probability that it is accepted as authentic is $N/p = N/|\mathbb{F}|$. Hence, the proposition. □

***k*-wise Independent Random Variables:** k-wise independent random variables is used to construct schemes presented in Sections 4 and 6.

Definition 6. *Random variables $X_1, \ldots, X_n$ over a finite set S is called* k-wise independent *if $\Pr[X_{i_1} = \alpha_{i_1}, \ldots, X_{i_k} = \alpha_{i_k}] = 1/p^k$ holds for any k indices $i_1, \ldots, i_k \in \{1, \ldots, n\}$ and for any $\alpha_{i_1}, \ldots, \alpha_{i_k} \in S$ where p is a cardinality of finite set S.*

It is well known that a polynomials of degree $k-1$ over a finite field form a k-wise independent random variables.

Proposition 4. *Let a polynomial* $r(x) = r_0 + r_1 x + \cdots + r_{k-1}x^{k-1}$ *be a randomly chosen polynomial over* $\mathbb{F}_p$. *Then* $X_1 = r(1), X_2 = r(2), \ldots, X_n = r(n)$ *is* k*-wise independent random variables over* $\mathbb{F}_p$.

We note that the size of memory to store the above k-wise independent variables $X_1, \ldots, X_n$ is p^k since it suffice to store all the coefficients of $r(x)$.

Reed-Solomon Error Correcting Code: Let $C(x) \in \mathbb{F}_p$ be a polynomial of degree at most t, and let $i_1, \ldots, i_k$ are distinct elements of $\mathbb{F}_p$ then it is well known that $(C(i_1), C(i_2), \ldots, C(i_k))$ constitutes a codeword of Reed-Solomon code with minimum hamming distance $k-t$. Therefore, when $k \leq 3t+1$ (i.e., $t < k/3$) holds, Reed-Solomon code corrects up to t errors with probability 1. Since the capability of Reed-Solomon code to identify cheaters in secret sharing schemes was first pointed out by McEliece and Sarwate [22], Reed-Solomon code has played the central role in secret sharing scheme capable of preventing cheating by $(k-1)/3$ cheaters. In fact, it has been used to identify up to $(k-1)/3$ cheaters in various schemes [20, 25, 40].

3 A Scheme Capable of Detecting $(k-1)/3$ Rushing Cheaters

In this section, we present a scheme capable of detecting cheating by $(k-1)/3$ rushing cheaters. The scheme is constructed based on the schemes presented in [7, 26] which are capable of identifying $k-1$ *non-rushing* cheater.

The basic idea of the proposed scheme is as follows. The share generation algorithm ShareGen generates shares $(v_{s,1}, \ldots, v_{s,n})$ of a secret s using Shamir's (k, n) threshold scheme. The share generation algorithm also generate shares $(v_{e,1}, \ldots, v_{e,n})$ for a key e of almost strongly universal hash family $H : \mathcal{S} \to \mathcal{H}$ using Shamir's $(t+1, n)$ threshold scheme. Furthermore, ShareGen generates shares $(v_{a,1}, \ldots, v_{a,n})$ for hash value $a = h_e(s)$ using Shamir's (k, n) threshold scheme. The share v_i of user P_i is of the form $v_i = (v_{s,i}, v_{e,i}, v_{a,i})$.

Since $v_{s,i}$ and $v_{a,i}$ are shared using (k, n) threshold scheme, $k-1$ users do not obtain any information about the secret even if they know the value of the key e. To guarantee security against t rushing cheaters, The secret reconstruction algorithm Reconst receives part of share $(v_{s,i}, v_{a,i})$ from P_i $(i = 1, 2, \ldots, k)$ in the first round, and then receives v_e from P_i in the second round. Since the key e of hash family is shared using $(t+1, n)$ threshold scheme, t rushing cheaters do not obtain any information about e even at the end of the first round. Therefore, the cheater cannot guess correct hash value a' for a forged secret s' in the first round. Further, from the error correcting capability of $(t+1, n)$ threshold scheme, rushing cheaters cannot alter the value of e no matter what shares they submit in the second round. The above discussion, together with the properties of ASU_2, directly derives the security of the scheme against rushing cheaters. The complete description of the scheme is given as follows.

Share Generation: On input a secret $s \in \mathbb{F}_{p^N}$, the share generation algorithm outputs a list of shares $(v_1, \ldots, v_n)$ as follows:

1. Generate a random polynomial $f_s(x) \in \mathbb{F}_{p^N}[X]$ of degree $k-1$ such that $f_s(0) = s$.

2. Generate random polynomials $C_{e_0}(x), C_{e_1}(x) \in \mathbb{F}_p[X]$ of degree t. We will use $(e_0, e_1) = (C_{e_0}(0), C_{e_1}(0))$ as a key of $H_{1,N}$.
3. Generate a random polynomial $a(x) \in \mathbb{F}_p$ of degree $k-1$ such that $a(0) = h_{1,N,(e_0,e_1)}(s)$.
4. Output a list of share $(v_1, \ldots, v_n)$ where $v_i = (f_s(i), C_{e_0}(i), C_{e_1}(i), a(i))$.

Secret Reconstruction: On input m shares $(v_{i_1}, \ldots, v_{i_m})$ (where $m \geq k$ and $v_i = (v_{s,i}, v_{e_0,i}, v_{e_1,i}, v_{a,i})$ for $1 \leq i \leq n$), the secret reconstruction algorithm outputs s or $\perp$ according to the following procedure:

1. [Round 1] Receive $(v_{s,i_1}, v_{a,i_1}), \ldots, (v_{s,i_m}, v_{a,i_m})$ from $P_{i_1}, \ldots, P_{i_m}$, respectively.
2. [Round 2] Receive $(v_{e_0,i_1}, v_{e_1,i_1}), \ldots, (v_{e_0,i_m}, v_{e_1,i_m})$ from $P_{i_1}, \ldots, P_{i_m}$, respectively.
3. Reconstruct $C_{e_0}(x)$ and C_{e_1} from $v_{e_0,i_1}, \ldots, v_{e_0,i_m}$ and $v_{e_1,i_1}, \ldots, v_{e_1,i_m}$, respectively, using a decoding algorithm of generalized Reed-Solomon Code (e.g., Berlekamp-Welch algorithm), and compute $e_0 = C_{e_0}(0)$ and $e_1 = C_{e_1}(0)$.
4. Output $\perp$ if error is detected.
5. Reconstruct $\hat{s} = f_{\hat{s}}(0)$ and $\hat{a} = f_{\hat{a}}(0)$ from $v_{s,i_1}, \ldots, v_{s,i_m}$ and $v_{a,i_1}, \ldots, v_{a,i_m}$, respectively.
6. Output $\hat{s}$ if $h_{1,N,(e_0,e_1)}(\hat{s}) = \hat{a}$ holds. Otherwise, output $\perp$.

Theorem 1. *If $t < k/3$ holds then the above scheme is (t, ϵ) cheating detectable k-out-of-n secret sharing scheme against rushing cheaters such that $|\mathcal{S}| = p^N$, $\epsilon = N/p$, and $|\mathcal{V}_i| = p^{N+3} \approx |\mathcal{S}|(\frac{\log|\mathcal{S}|}{\epsilon \log(1/\epsilon)})^3$. In particular, $|\mathcal{V}_i| = |\mathcal{S}|/\epsilon^3$ holds when $N = 1$.*

Proof. First, we will prove the scheme is *perfect*. Suppose that users $P_{i_1}, \ldots, P_{i_{k-1}}$ try to compute any partial information about the secret s. Since $v_{s,i_1}, \ldots, v_{s,i_{k-1}}$ is generated using Shamir's (k, n) threshold scheme, they do not obtain any information about the secret from $v_{s,i_1}, \ldots, v_{s,i_{k-1}}$. Therefore, the scheme is proven to be perfect if the equations $h_{1,N,(e_0,e_1)}(s) = a$ does not reveal any information about the secret. Since shares $v_{e_0,i}$ and $v_{e_1,i}$ are generated using Shamir's (t, n) threshold scheme, $P_{i_1}, \ldots, P_{i_{k-1}}$ obtain values of e_0 and e_1 from their shares. However, since a share $v_{a,i}$ is generated using Shamir's (k, n) threshold scheme, $P_{i_1}, \ldots, P_{i_{k-1}}$ obtain no information about the value $h_{1,N,(e_0,e_1)}(s)$. Therefore, they do not obtain any information about the secret s even if they know e_0 and e_1, which shows that the scheme is perfect.

Next, we prove that if $t < k/3$ the scheme is (t, ϵ) cheating detectable. Here, we consider the worst case where just k users $P_1, \ldots, P_k$ take part in secret reconstruction. Without loss of generality, we can assume $P_1, \ldots, P_t$ are cheater who try to fool $P_{t+1}, \ldots, P_k$ by submitting $(v'_1, \ldots, v'_t)$ to the secret reconstruction algorithm. Since cheaters are rushing, cheaters know all values of s (a value secret reconstructed from $(v_{s,1}, v_{s,2}, \ldots, v_{s,k})$), s' (a value reconstructed from $(v'_{s,1}, \ldots, v'_{s,t}, v_{s,t+1} \ldots, v_{s,k})$), a (a value reconstructed from $(v_{a,1}, v_{a,2}, \ldots, v_{a,k})$) and a' (a value reconstructed from $(v'_{a,1}, \ldots, v'_{a,t}, v_{a,t+1} \ldots, v_{a,k})$) after observing part of shares $(v_{s,i}, v_{a,i})$ submitted by honest users $P_{t+1}, \ldots, P_k$ in the first round. However, at this stage, cheaters do not obtain any information about the values of e_0 and e_1 since they are shared among users using Shamir's (t, n) threshold scheme. Moreover, even rushing cheaters cannot forge part of their shares $(v_{e_0,i}, v_{e_1,i})$ in order to alter the values of e_0 and e_1 reconstructed. In fact, if cheater P_i forge $(v_{e_0,i}, v_{e_1,i})$ into $(v'_{e_0,i}, v'_{e_1,i})$ in the second round, then P_i is identified as a cheater with probability 1 by t-error correction capability of Reed-Solomon codes. Therefore, the best strategy for cheater P_i is to submit $(v_{e_0,i}, v_{e_1,i})$ as

is, which ensure that e_0 and e_1 are reconstructed without being forged. Now we compute the successful cheating probability ϵ of rushing cheaters. The cheaters succeed in cheating if $h_{1,N,(e_0,e_1)}(s') = a'$ holds. Since rushing cheater P_i must submit $(v'_{s,i}, v'_{a,i})$ in the first round without knowing the values of e_0 and e_1, The successful cheating probability of cheaters are computed by $\Pr[h_{1,N,(e_0,e_1)}(s') = a' \mid h_{1,N,(e_0,e_1)}(s) = a]$ where the probability is taken only over e_0 and e_1 since s, s', a and a' are known to cheaters when they submit $(v'_{s,i}, v'_{a,i})$. Since $H_{1,N}$ is N/p-ASU_2 and (e_0, e_1) are chosen uniformly and randomly from the set of keys satisfying $h_{1,N,(e_0,e_1)}(s) = a$, it is easy to see that $\Pr[h_{1,N,(e_0,e_1)}(s') = a' \mid h_{1,N,(e_0,e_1)}(s) = a] = N/p$ holds, which directly implies that the successful cheating probability of cheaters $P_1, \ldots, P_t$ is upper bounded by N/p. □

4 A Scheme Capable of Detecting $n-1$ Rushing Cheaters

In this section, we present a scheme capable of detecting cheating by $n-1$ rushing cheaters. The idea of the scheme is similar to the scheme presented in the previous section in the sense that the share generation algorithm generates a key e of ASU_2 and $a = h_e(s)$ to check the correctness of a secret reconstructed. However, since $t = n-1$ holds, we cannot use error correcting code to ensure correct reconstruction of e. In the proposed scheme, instead of sharing a single key e, the share generation algorithm generates a key e_i and a hash value a_i for each user P_i who verifies $h_{e_i}(s) = a_i$ to check the correctness of the secret s. However, unfortunately, the above naive scheme cannot be *perfect* since user P_i can compute (possibly partial) information about the secret from information held by P_i: $h_{e_i}(s) = a_i, e_i$ and a_i. For example, consider the case where we use $H_{1,1}$ defined in Proposition 1 for underlying hash family. In this case, any single user P_i can compute s from $e_{0,i}, e_{1,i}$ and $a_i = e_{0,i} + e_{1,i} \cdot s$ by $s = (a_i - e_{0,i}) \cdot e_{1,i}^{-1}$.

We introduce an additional trick to overcome this problem. Namely, we make the hash values a_i shared among users so that unauthorized set of users cannot obtain any information about the hash values and therefore any information about the secret. However, sharing completely independent and random hash values $a_1, \ldots, a_n$ causes the size of shares grow linearly with n. To reduce the size of share, we make hash values $a_1, \ldots, a_n$ derived from $(k-1)$-wise independent random variables, and share the seed a of the random values $a_1, \ldots, a_n$ instead of sharing $a_1, \ldots, a_n$ themselves. By replacing completely random $a_1, \ldots, a_n$ with $(k-1)$-wise independent random variables does not affect the perfectness of the resulting scheme since $k-1$ users do not obtain any relation among hash values due to $(k-1)$-wise randomness of hash values. The complete description of the scheme is given as follows.

Share Generation: The share generation algorithm $\mathsf{ShareGen}$ takes a secret $s = (s_0, s_1, ..., s_{N-1}) \in \mathbb{F}_{p^N}$ as input and outputs a list of shares $(v_1, \ldots, v_n)$ according to the following procedure:

1. Generate a random polynomial $f_s(x) \in \mathbb{F}_{p^N}[X]$ of degree $k-1$ such that $s = f_s(0)$.
2. Generate a random polynomial $f_a(x) \in \mathbb{F}_{p^{k-1}}[X]$ of degree $k-1$. We will use $f_a(0) = (a_0, a_1, \ldots, a_{k-2})$ as coefficients of a polynomial $a(x)$ (i.e., $a(x) = a_0 + a_1 x + \cdots a_{k-2} x^{k-2}$) used to derive hash values in $(k-1)$-wise independent manner.
3. Compute keys $(e_{i,0}, e_{i,1})$ $(1 \le i \le n)$ of almost strongly universal family $H_{1,N}$ independently and randomly satisfying $h_{1,N,(e_{i,0},e_{i,1})}(s) = a(i)$.

4. Output $(v_1, \ldots, v_n)$ where the share v_i of the user P_i is defined by $v_i = (f_s(i), f_a(i), e_{i,0}, e_{i,1})$.

Secret Reconstruction:

1. [Round 1] Receive $(v_{s,i_1}, v_{a,i_1}), \ldots, (v_{s,i_m}, v_{a,i_m})$ from $P_{i_1}, \ldots, P_{i_m}$, respectively.
2. [Round 2] Receive $(e_{i_1,0}, e_{i_1,1})$ from $(e_{i_m,0}, e_{i_m,1})$ from $P_{i_1}, \ldots, P_{i_m}$, respectively.
3. Reconstruct s and $a(x)$ from $(v_{s,i_1}, \ldots, v_{s,i_m})$ and $(v_{a,i_1}, \ldots, v_{a,i_m})$, respectively.
4. Check if $h_{1,N,(e_{i_j,0},e_{i_j,1})}(s) = a(i_j)$ holds for all i_j $(1 \leq j \leq m)$.
5. Output s if the above equation holds for all i_j, otherwise output $\perp$.

The following theorem gives the security properties of the proposed scheme.

Theorem 2. *If $t \leq n-1$ holds then the above scheme is (t, ϵ) cheating detectable k-out-of-n secret sharing scheme against rushing cheaters such that $|\mathcal{S}| = p^N$, $\epsilon = \frac{N}{p}$, and $|\mathcal{V}_i| = p^{N+k+1} \approx |\mathcal{S}| \cdot (\frac{\log |\mathcal{S}|}{\epsilon \log(1/\epsilon)})^{k+1}$. In particular, $|\mathcal{V}_i| = |\mathcal{S}|/\epsilon^{k+1}$ holds when $N = 1$.*

Proof. First, we will prove the scheme is *perfect*. Suppose that users $P_{i_1}, \ldots, P_{i_{k-1}}$ try to compute any partial information about the secret s. Since $v_{s,i_1}, \ldots, v_{s,i_{k-1}}$ are generated using Shamir's (k, n) threshold scheme, they do not obtain any information about the secret from $v_{s,i_1}, \ldots, v_{s,i_{k-1}}$. Therefore, the scheme is proven to be perfect if the equations $h_{1,N,(e_{i_j,0},e_{i_j,1})}(s) = a(i_j)$ $(1 \leq j \leq k-1)$ does not reveal any information about the secret. Since shares $v_{a,i_1}, \ldots, v_{a,i_{k-1}}$ (i.e., shares for $a = (a_0, \ldots, a_{k-2})$) is also generated by Shamir's (k, n) threshold scheme $P_{i_1}, \ldots, P_{i_{k-1}}$ obtain no information about the hash value $a(i_j)$. Therefore, the participants cannot obtain any information from the knowledge $h_{1,N,(e_{i_j,0},e_{i_j,1})}(s) = a(i_j)$, which shows that the scheme is perfect.

Next, we prove that the scheme is $(n-1, \epsilon)$ cheating detectable. Without loss of generality, we can assume $P_2, \ldots, P_n$ are cheaters who try to fool P_1 by submitting $(v'_2, \ldots, v'_k)$ to the secret reconstruction algorithm. Since cheaters are rushing, cheaters know all values of s (a value secret reconstructed from $(v_{s,1}, v_{s,2}, \ldots, v_{s,k})$), s' (a value reconstructed from $(v_{s,1}, v'_{s,2}, \ldots, v'_{s,k})$), $a(x)$ (a polynomial reconstructed from $(v_{a,1}, v_{a,2}, \ldots, v_{a,k})$) and $a'(x)$ (a polynomial reconstructed from $(v_{a,1}, v'_{a,2}, \ldots, v'_{a,k})$) after observing $(v_{s,1}, v_{a,1})$ submitted by P_1 even when $k = n$. The cheaters succeed in cheating P_1 if $h_{1,N,(e_{1,0},e_{1,1})}(s') = a'(1)$ holds. We will show the success cheating probability is upper bounded by N/p. Since cheaters know $h_{e_1}(s) = a(1)$ the successful cheating probability can be computed by $\Pr[h_{1,N,(e_{1,0},e_{1,1})}(s') = a'(1) \mid h_{1,N,(e_{1,0},e_{1,1})}(s) = a(1)]$. From the second property of the almost strongly universal hash family and the fact e_1 is chosen uniformly and randomly from the set of keys such that $h_{1,N,(e_{1,0},e_{1,1})}(s) = a(1)$, the following equation holds:

$$\begin{aligned}
&\Pr[h_{1,N,(e_{1,0},e_{1,1})}(s') = a'(1) \mid h_{1,N,(e_{1,0},e_{1,1})}(s) = a(1)] \\
&= \frac{|\{(e_{1,0}, e_{1,1}) \mid h_{1,N,(e_{1,0},e_{1,1})}(s) = a(1), h_{1,N,(e_{1,0},e_{1,1})}(s') = a'(1)\}|}{|\{(e_{1,0}, e_{1,1}) \mid h_{1,N,(e_{1,0},e_{1,1})}(s) = a(1)|} \leq \epsilon\,,
\end{aligned}$$

which directly implies that the successful cheating probability of cheaters $P_2, \ldots, P_n$ is upper bounded by N/p. □

5 A Scheme Capable of Identifying $(k-1)/3$ Rushing Cheaters

In this section, we present a scheme capable of identifying $(k-1)/3$ rushing cheaters. The scheme is constructed based on the scheme presented in [25] which is capable of identifying $(k-1)/3$ *non-rushing* cheater.

Roughly speaking, the share v_i of the scheme in [25] consists of (1) a share $v_{s,i}$ of Shamir's (k,n) threshold scheme for a secret s, and (2) a hash value $v_{C_i} = h_{2,t+1}(v_{s,i})$ where $h_{2,t+1} \in H_{2,t+1}$ is a strongly universal hash function of strength $t+1$ (see Proposition 2 for the complete description). Unfortunately, the scheme in [25] is vulnerable to cheating by a single rushing cheater no matter what order partial shares are sent to the reconstruction algorithm. This is because rushing cheaters obtain complete information about the hash function $h_{2,t+1}$ before they send v'_{C_i} to the reconstruction algorithm.

To make rushing cheaters impossible to obtain complete information about the hash function, we modify the scheme in a way that hash function h is chosen from $H_{2,k+t}$ instead of $H_{2,t+1}$. This modification makes rushing cheater difficult to cheat the scheme since at least $k+t$ shares are required to obtain complete information about the hash function h. Furthermore, to prevent rushing cheaters from modifying the hash function h, we introduce an additional share $v_{E,i}$ in the proposed scheme. Here, $v_{E,i}$ is a share of $(n, t+1)$ threshold scheme for a secret $(e_{t+1}, \ldots, e_{t+k-1}) \in \mathbb{F}_{p^{k-1}}$ where $e_{t+1}, \ldots, e_{t+k-1}$ represent higher-degree coefficients of $h \in H_{2,k+t}$. With the help of $v_{E,i}$, we can convert hash values $h(\psi_{i_1}), \ldots, h(\psi_{i_k})$ into hash values $\hat{h}(\psi_{i_1}), \ldots, \hat{h}(\psi_{i_k})$ in such a way that $\hat{h} \in H_{2,t+1}$ and that $\hat{h}(\psi)$ is a correct hash value of ψ if and only if $h(\psi)$ is a correct hash value of ψ. Since converted hash function $\hat{h}$ is a element of $H_{2,t+1}$, we can identify even rushing cheaters, as in the cheater identification procedure presented in [25]. The complete description of the proposed scheme is as follows:

Share Generation: On input a secret the share generation algorithm outputs a list of shares $(v_1, \ldots, v_n)$ as follows:

1. Generate a random polynomial $f_s(x) \in \mathbb{F}_p[X]$ of degree $k-1$ such that $f_s(0) = s$.
2. Generate a random polynomial $C(x) = \sum_{i=0}^{k+t-1} e_i x^i \in \mathbb{F}_q[X]$ of degree $k+t-1$ where q is a prime power satisfying $q \geq n \cdot p$.
3. Generate a random polynomial $C_E(x) \in \mathbb{F}_{q^{k-1}}[X]$ of degree t such that $C_E(0) = (e_{t+1}, e_{t+2}, \ldots, e_{t+k-2}, e_{t+k-1}) \in \mathbb{F}_{q^{k-1}}$ (i.e., $C_E(0)$ represents higher degree coefficients of $C(x)$.)
4. Output a list of share $(v_1, \ldots, v_n)$ where $v_i = (f_s(i), C(\psi(i, f_s(i)), C_E(i))$ and $\psi : [1,n] \times \mathbb{F}_p \to \mathbb{F}_q$ is an arbitrary 1-to-1 function.

Secret Reconstruction: On input m shares $(v_{j_1}, \ldots, v_{j_m})$ (where $m \geq k$ and $v_i = (v_{s,i}, v_{C,i}, v_{E,i})$ for $1 \leq i \leq n$), the secret reconstruction algorithm outputs $(s, \emptyset)$ or $(\perp, L)$ according to the following procedure:

1. Choose k users $i_1, \ldots, i_k$ arbitrarily.
2. [Round 1] Force $P_{i_1}, \ldots, P_{i_k}$ submit $(v_{s,i_1}, v_{C,i_1}), \ldots, (v_{s,i_k}, v_{C,i_k})$, respectively.
3. [Round 2] Force $P_{i_1}, \ldots, P_{i_k}$ submit $v_{E,i_1}, \ldots, v_{E,i_k}$, respectively.
4. Reconstruct $C_E(x)$ using a decoding algorithm of generalized Reed-Solomon Code.
5. Compute a list L' by $L' = \{i_j \mid v_{E,i_j} \neq C_E(i_j)\}$.

6. Compute $(e_{t+1}, \ldots, e_{t+k-1}) = C_E(0)$.
7. Compute $\hat{v}_{C,i_j} = v_{C,i_j} - \sum_{\ell=t+1}^{t+k-1} e_\ell \cdot \psi(i_j, v_{s,i_j})^\ell$.
8. Reconstruct $\hat{C}(x) = \sum_{\ell=0}^{t} e_\ell x^\ell$ from $\hat{v}_{C,i_1}, \ldots, \hat{v}_{C,i_k}$ using a decoding algorithm of generalized Reed-Solomon Code again.
9. Compute a list L by $L = L' \cup \{i_j \mid \hat{v}_{C,i_j} \neq \hat{C}(\psi(i_j, v_{s,i_j}))\}$.
10. Reconstruct $f_s(x)$ from $v_{s,i_1}, \ldots, v_{s,i_k}$ and output $(\perp, L)$ if $L \neq \emptyset$. Otherwise, output $(f_s(0), \emptyset)$.

Theorem 3. *If $t < k/3$ holds then the above scheme is a (t, ϵ) cheater identifiable k-out-of-n secret sharing scheme against rushing adversaries such that $|\mathcal{S}| = p$, $\epsilon = 1/q$, and $|\mathcal{V}_i| = p \cdot q^k = |\mathcal{S}|/\epsilon^k$.*

Proof. First, we show that the scheme is perfect. It is well known that $v_{s,i_1}, \ldots, v_{s,i_{k-1}}$ do not reveal any information about the secret since each $v_{s,i}$ is a share of Shamir's k-out-of-n secret sharing scheme. Further, it is easy to see that the knowledge about $v_{C,i}$ and $v_{E,i}$ do not reveal any information about the secret since the polynomials $C(x)$ and $C_E(x)$ are completely independent of the secret s.

Next we show that the scheme is (t, ϵ) cheater identifiable against rushing cheaters. The following two facts are important to prove (t, ϵ) cheater identifiability of the scheme:

1. A family of functions $\{C(x) \mid C(x) \in \mathbb{F}_q[X], \deg(C(x)) \leq t + k - 1\}$ is a strong family of universal hash functions $\mathbb{F}_q \to \mathbb{F}_q$ with strength $t + k$. Therefore, even rushing cheaters who observed t shares of cheaters as well as $k - 1$ honest users cannot send a correct value of $C(\psi')$ for unknown ψ' with probability better than $1/q$ in the first round.
2. $(C_E(x_1), C_E(x_2), \ldots, C_E(x_k))$ and $(\hat{C}(x_1), \hat{C}(x_2), \ldots, \hat{C}(x_k))$ are codewords of the Reed-Solomon Code with minimum distance $k - t$. Therefore, if $t < k/3$ holds, then $C_E(x)$ and $\hat{C}(x)$ can be reconstructed correctly even when t points are forged.

Without loss of generality, we can assume $P_k, \ldots, P_{t+k-1}$ are cheaters who cooperatively cheat users $P_1, \ldots, P_{k-1}$ by forging (part of) their shares. We consider the worst case where honest users $P_1, \ldots, P_{k-1}$ and the rushing cheater P_k are chosen to submit their shares to Reconst (this is the worst case since rushing cheater can observe the most number of shares in cheating).

Since only P_k is a cheater, P_k submits forged $v'_{s,k}$ in the first round. In this case, P_k is not identified as a cheater only if he submits correct $v'_{C,k}$ such that $v'_{C,k} = C(\psi(v'_{s,k}, k))$ since Reconst can recover correct $\hat{C}(x)$ whatever $v'_{E,k}$ he submits, and $\hat{v}_{C,k} = \hat{C}(\psi(v'_{s,k}, k))$ holds if and only if $v'_{C,k} = C(\psi(v'_{s,k}, k))$. It is easy to see that P_k cannot guess correct $v'_{C,k}$ with probability better than $1/q$ since $C(x)$ belongs to a strongly universal family of hash functions with strength $t + k$. where the probability is taken over the random choice of $C(x)$. □

Note: Successful cheating probability ϵ can be made chosen flexibly in the above scheme by using techniques introduced in [25].

6 A Scheme Capable of Identifying $(k-1)/2$ Rushing Cheaters

In this section, we present a scheme capable of identifying $(k-1)/2$ rushing cheaters. The scheme is based on a standard construction first presented in [31] such that the

share v_i consists of (1) share $v_{s,i}$ of Shamir's (k, n) threshold scheme for a secret, (2) keys of ASU_2 (unconditionally secure MAC) to check the correctness of $v_{s,j}$ $(j \neq i)$, and (3) hash values to prove the correctness of $v_{s,i}$. Unfortunately, the bit length of the resulting scheme still grows linearly with n. Though, with the help of tag compression technique by Carpentieri [8], the proposed scheme reduces the number of keys of ASU_2, which results in smaller size of shares compared to the schemes by Roy et.al. [34] and by Choudhury [10]. The complete description of the proposed scheme is as follows:

Share Generation: On input a secret $s \in \mathbb{F}_{p^N}$, the share generation algorithm ShareGen outputs a list of shares $(v_1, \ldots, v_n)$ as follows:

1. Generate a random polynomial $f_s(x)$ of degree at most $(k-1)$ in x from $\mathbb{F}_{p^N}[X]$ such that $f_s(0) = s$ and compute $f_s(i) = v_{s,i}$ in $\mathbb{F}_{p^N}$, where $i = 1, \ldots, n$.
2. Generate a random $e_{0,i} \in_R \mathbb{F}_p$ and a random polynomial of degree at most $k-1$ with free coefficient 0, $a_i(x) = a_{i,1}x + a_{i,2}x^2 + \cdots + a_{i,k-1}x^{k-1}$, from $\mathbb{F}_p[X]$.
3. Compute $a_{i,j} = a_i(j)$ and $e_{1,i,j} = a_j(i) - \Sigma_{l=1}^{N} e_{0,i}^{l} \cdot v_{s,j,l}$ for $i \in [n] \setminus j$.
4. Compute $v_i = (v_{s,i}, a_i(x), e_{0,i}, e_{1,i,1}, \ldots, e_{1,i,i-1}, e_{1,i,i+1}, \ldots, e_{1,i,n})$.

Secret Reconstruction: Denote the set of m $(\geq k)$ participants taking part in the reconstruction as *core*. On input a list of m shares, the secret reconstruction algorithm Reconst output a secret and a list of identities of cheaters or $\perp$ and a list of identities of cheaters as follows.

1. [Round 1] Receive $v'_{s,i}, a'_{i,1}, \ldots, a'_{i,k-1}$ from each $P_i \in core$.
2. [Round 1] Receive $e'_{0,i}, e'_{1,i,1}, \ldots, e'_{1,i,n}$ from each $P_i \in core$.
3. **Computation:** For each $P_i \in core$, computes $support_i = \{P_j : \Sigma_{l=1}^{N} e'^{l}_{0,j} \cdot v'_{s,i,l} + e'_{1,j,i} = a'_{i,1}j + a'_{i,2}j^2 + \cdots + a'_{i,k-1}j^{k-1}\} \cup \{P_i\}$.
   If $|support_i| < t+1$, then put P_i in L, where L is the list of the cheaters.
4. – If $m - |L| \geq k$ **:** Using $v'_{s,i}$ for all $P_i \in core \setminus L$, interpolate a poly $f'_s(x)$. If degree of $f'_s(x)$ is less or equal to k, output $(f'_s(0), L)$ otherwise output $(\perp, L)$.
   – If $m - |L| < k$ **:** Output $(\perp, L)$.

Lemma 1. *The above scheme provides perfect secrecy. That is, any adversary $\mathcal{A}$ controlling any $(k-1)$ parties during the sharing phase, will get no information about the secret s.*

Proof. Without loss of generality, we may assume that the first $(k-1)$ participants, i.e., $P_1, \ldots, P_{k-1}$, are under the control of the adversary $\mathcal{A}$. The listening adversary has the following information.

$$\begin{pmatrix} v_{s,1} & a_{1,1} & a_{1,2} & \cdots & a_{1,k-1} & e_{0,1} & \perp & e_{1,1,2} & \cdots & e_{1,1,n} \\ v_{s,2} & a_{2,1} & a_{2,2} & \cdots & a_{2,k-1} & e_{0,2} & e_{1,2,1} & \perp & \cdots & e_{1,2,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ v_{s,k-1} & a_{k-1,1} & a_{k-1,2} & \cdots & a_{k-1,k-1} & e_{0,k-1} & e_{1,k-1,1} & e_{1,k-1,2} & \cdots & e_{1,k-1,n} \end{pmatrix}$$

Now, according to Lagrange's interpolation, k Shamir shares $v_{s,i}$ fully define a degree-$(k-1)$ polynomial. On the other hand, $k-1$ such values provide no information on s, according to the perfect privacy property of Shamir scheme. Thus, the adversary needs to choose one more $v_{s,i}$, where $i \in \{1, 2, \ldots, n\} \setminus I$ and $I = \{1, 2, \ldots, k-1\}$.

Without loss of generality, we may assume that the adversary tries to learn $v_{s,k}$ with the information at hand. Note that each player P_i $(i \in I)$ has the information $(e_{0,i}, e_{1,i,k})$ regarding $v_{s,i}$. Now,

$$\begin{aligned}
\Sigma_{l=1}^{N} e_{0,1}^{l} v_{s,k,l} + e_{1,1,k} &= a_{k,1}1 + a_{k,2}1^2 + \cdots + a_{k,k-1}1^{k-1} \\
\Sigma_{l=1}^{N} e_{0,2}^{l} v_{s,k,l} + e_{1,2,k} &= a_{k,1}2 + a_{k,2}2^2 + \cdots + a_{k,k-1}2^{k-1} \\
\cdots &= \cdots \\
\Sigma_{l=1}^{N} e_{0,k-1}^{l} s_{s,k,l} + e_{1,k-1,k} &= a_{k,1}(k-1) + a_{k,2}(k-1)^2 + \cdots + a_{k,k-1}(k-1)^{k-1}
\end{aligned}$$

Suppose, the adversary $\mathcal{A}$ tries to find out $v_{s,k,1}$. Now, as the matrix

$$\begin{bmatrix}
1 & 1^2 & \ldots & 1^{k-1} \\
2 & 2^2 & \ldots & 2^{k-1} \\
\ldots & \ldots & \ldots & \ldots \\
k-1 & (k-1)^2 & \ldots & (k-1)^{k-1}
\end{bmatrix}$$

is non-singular, the above system of linear equations is consistent for all possible values of $v_{s,k,1}$. Similarly, for other $v_{s,k,l}$. So, the best probability for $\mathcal{A}$ to guess $v_{s,k}$ is $(1/p)^N = 1/p^N$.

Note also that the adversary can construct such system of linear equations for every P_j for $j \in \{k, \ldots, n\}$. However, all these systems of equations are consistent. In other words, for any fixed value of $v_{s,k}$, there exists one and only one solution satisfying all equations available to the adversary. This essentially means that all possible values of $v_{s,k}$ are consistent with the view of the adversary. So that the adversary has no information regarding the secret s. Hence, the theorem. □

Lemma 2. *The proposed scheme satisfies correctness condition. That is, during the reconstruction phase, if any $P_i \in$ core is under the control of rushing $\mathcal{A}$ and produces $v'_{s,i} \neq v_{s,i}$, then except with error probability $\epsilon = \frac{m-t}{|\mathbb{F}_p|}$, P_i will be identified as a cheater and will be included in the list L.*

Proof. Without loss of generality, let *core* be formed by the first m parties, namely $P_1, \ldots, P_m$, where $m \geq k$. Moreover, let $P_1, \ldots, P_t$ be under the control of $\mathcal{A}$. Now suppose that P_1 submits $v'_{s,1} \neq v_{s,1}$ and P_1 is not identified as a cheater. This implies that $|support_1| \geq t+1$. In the worst case, $P_1, \ldots, P_t$ may be present in $support_1$, as all of them are under the control of $\mathcal{A}$. But $|support_1| \geq t+1$ implies that there exists at least one honest party in *core*, say P_j, such that $P_j \in support_1$. This is possible only if $\Sigma_{l=1}^{N} e_{0,j}^{l} v'_{s,1,l} + e_{1,j,1} = ja'_{1,1} + j^2 a'_{1,2} + \ldots + j^{k-1} a'_{1,k-1}$. Now in *Round* 1 of reconstruction phase each player P_i broadcasts $v_{s,i}, a_{i,1}, \ldots, a_{i,k-1}$ and in *Round* 2 of reconstruction phase P_i broadcasts $e_{0,i}, e_{1,i,1}, \ldots, e_{1,i,i-1}, e_{1,i,i+1}, \ldots, e_{1,i,n}$.

After round 1 of the reconstruction phase, the cheating adversary can see the Shamir share and authentication tags of each player. And $\mathcal{A}$ also knows the authentication keys of player $P_1, P_2, \ldots, P_t$. But he does not know the authentication keys of players $P_{t+1}, \ldots, P_m$.

Now we evaluate the probability that P_1 succeeds in deceiving at least one honest player to accept her fake share and fake tag. This probability is described by the

following formula.

$$\Pr[\text{at least one player in } [P_{t+1}, \dots P_m] \text{ accepts } (v'_{s,1}, a'_1(x)) \\ | \, [P_{t+1}, \dots P_m] \text{ accept } (v_{s,1}, a_1(x), \dots, v_{s,n}, a_n(x))]$$

Denote by E_1 the event that
"at least one player in $[P_{t+1}, \dots, P_n]$ accepts $(v'_{s,1}, a'_1(x))$", and by E_2 the event that "$[P_{t+1}, \dots, P_n]$ accept $v_{s,1}, a_1(x), \dots, v_{s,n}, a_n(x))$".

Now, using the same argument as in Proposition 3, we can conclude that $\Pr[E_1|E_2] < (m-t)/p$.

So we get ϵ-correctness for $\epsilon = (m-t)/p$. Hence, the theorem. □

Theorem 4. *If $t < k/2$ holds then the above scheme is a (t, ϵ) cheater identifiable k-out-of-n secret sharing scheme against rushing adversaries such that $|S| = p^N$, $\epsilon = \frac{m-t}{p}$, and $|V_i| = |S|\frac{(m-t)^{n+2t}}{\epsilon^{n+2t}}$.*

Remark 1. During the sharing phase, each party gets 1 element from the field $\mathbb{F}_{p^N}$ and $n+k-1$ elements from the field $\mathbb{F}_p$. So, $|V_i| = p^N \cdot p^{n+k-1} = (m-t)^{n+2t}|S|/\epsilon^{n+2t}$. So, share size will be at most $(n-t)^{n+2t}|S|/\epsilon^{n+2t}$, when all the participants participate in the reconstruction phase and share size will be at least $(k-t)^{n+2t}|S|/\epsilon^{n+2t}$, when only k number of participants participate in the reconstruction phase. Moreover, if $t = 1, 2$, the proposed CISS scheme is the best one, with respect to the share size, among all the existing efficient CISS schemes secure against rushing adversary when we consider the worst case scenario.

Remark 2. In the proposed CISS, error probability does not depend on the size of the secret space. We can independently choose the error probability according to the security parameter. Hence, our proposed scheme has *flexibility* property. So, within the natural restrictions, the parameters can be set flexibly.

7 Robust Secret Sharing Capable of Tolerating $(n-1)/2$ Rushing Cheaters

In our proposal, we use the new share authentication method and adapt it to the reconstruction technique of [9]. The complete description of the proposed scheme is as follows:

Share Generation: On input a secret $s \in \mathbb{F}_{2^m}$, the share generation algorithm ShareGen outputs a list of shares $(v_1, \dots, v_n)$ as follows:

1. Generate a random polynomial $f_s(x)$ of degree at most $(k-1)$ in x from $\mathbb{F}_{2^m}[X]$ such that $f_s(0) = s$ and compute $f_s(i) = v_{s,i}$ in $\mathbb{F}_{2^m}$, where $i = 1, \dots, n$.
2. Generate a random $e_{0,i} \in_R \mathbb{F}_{2^q}$ and a random polynomial of degree at most $k-1$ with free coefficient 0, $a_i(x) = a_{i,1}x + a_{i,2}x^2 + \cdots + a_{i,k-1}x^{k-1}$ from $\mathbb{F}_{2^q}[X]$, where $m = Nq$.
3. Compute $a_{i,j} = a_i(j)$ and $e_{1,i,j} = a_j(i) - \Sigma_{l=1}^{N} e_{0,i}^l \cdot v_{s,j,l}$ for $i \in [n] \setminus j$.
4. Compute $v_i = (v_{s,i}, a_i(x), e_{0,i}, e_{1,i,1}, \dots, e_{1,i,i-1}, e_{1,i,i+1}, \dots, e_{1,i,n})$.

Secret Reconstruction: On input a list of n shares, the secret reconstruction algorithm Reconst output a secret or $\perp$ as follows.

1. [Round 1] Receive $v'_{s,i}, a'_{i,1}, \ldots, a'_{i,k-1}$ from each P_i.
2. [Round 1] Receive $e'_{0,i}, e'_{1,i,1}, \ldots, e'_{1,i,n}$ from each P_i.
3. **Local Computation:**
   - Set z_{ij}, $i, j \in \{1, 2, \ldots, n\}$, to be 1 if P_i's authentication tag is accepted by P_j, i.e., if $\Sigma_{l=1}^{N} e'^{l}_{0,j} \cdot v'_{s,i,l} + e'_{1,j,i} = a'_{i,1}j + a'_{i,2}j^2 + \cdots + a'_{i,k-1}j^{k-1}$ otherwise set z_{ij} to 0.
   - computes the largest set $\mathcal{I} \subseteq \{1, 2, \ldots, n\}$ with the property that

     $$\forall i \in \mathcal{I} : |\{j \in \mathcal{I} | z_{ij} = 1\}| = \Sigma_{j \in \mathcal{I}} z_{ij} \geq k.$$

     Clearly, $\mathcal{I}$ contains all honest participants. Let $c = |\mathcal{I}| - k$ be the maximum number of corrupted participants in $\mathcal{I}$.
4. Using the error correction algorithm for Reed-Solomon code, each participant computes a polynomial $f_s(x) \in \mathbb{F}_{2^m}[X]$ of degree at most $k-1$ such that $f_s(i) = v'_{s,i}$ for at least $k + \frac{c}{2}$ participants i in $\mathcal{I}$.
   If no such polynomial exists then output $\perp$,
   otherwise, output $s = f_s(0)$.

Remark 3. In the proposed scheme, a tradeoff between cheating probability and share size can be arranged. So, within the natural restrictions, the parameters can be set flexibly. Hence, q can be smaller or larger than m.

Lemma 3. *Any corrupted participant P_i who submits $v'_{s,i} \neq v_{s,i}$ in Round 1 of the reconstruction phase will be accepted by an honest participant with probability at most $\epsilon = \frac{N}{2^q}$.*

Proof. Without loss of generality, we assume that the corrupted participant is P_1 who submits $v'_{s,1} \neq v_{s,1}$ in Round 1 of the reconstruction phase. P_1 will be accepted by honest P_j if $\Sigma_{l=1}^{N} e'^{l}_{0,j} \cdot v'_{s,1,l} + e'_{1,j,1} = a'_{1,1}j + a'_{1,2}j^2 + \cdots + a'_{1,k-1}j^{k-1}$. Now using the argument of proof of Proposition 3, we can conclude that $\epsilon = \frac{N}{2^q}$. □

Theorem 5. *For any positive integer k such that $n = 2k-1$, the proposed construction forms (k, δ)-robust secret sharing scheme for n participants with the space of secrets $\mathbb{F}_{2^m}$ and*

$$\delta \leq e.(k\epsilon)^{k/2}$$

where $e = exp(1)$ and $\epsilon = \frac{N}{2^q}$.

Proof. **Privacy:** Follows from Lemma 1.
Reconstructability: From Lemma 3, we have found that $Pr(z_{ij} = 1) \leq \epsilon$. The rest of the proof is the same as in [9, Theorem 3.1]. □

Discussion:

Let us compute the share size. During the sharing phase, each party gets one element from $\mathbb{F}_{2^m}$ and $n+k-1$ elements from $\mathbb{F}_{2^q}$. Therefore, the share size of each participant is $m + (n + k - 1)q$ bits.

Consider the following instantiation. By Theorem 5, the resulting secret sharing scheme is a δ-robust for $\delta \leq e.(k\epsilon)^{k/2}$. Therefore, for a given security parameter μ, setting $q = \lceil \log k + \log N + \frac{2}{k}(\mu + \log(e)) \rceil$, we obtain $\delta \leq 2^{-\mu}$.

Every perfectly secure secret sharing scheme must have the share size at least that of the secret. The first term in the sum is responsible for this, while the second term characterizes an overhead required for the share authentication. In Table 3, we compare the overhead of our scheme with those of the schemes by Rabin and Ben-Or [31], and Cevallos et al [9]. We see that, our scheme reduces the overhead by the constant factor as compared to that of Cevallos et al.

8 Concluding Remarks

In this paper, we have presented five k-out-of-n secret sharing schemes secure against rushing adversaries with the following properties:

- capable of *detecting* up to $(k-1)/3$ rushing cheaters such that $|V_i| = |S|/\epsilon^3$,
- capable of *detecting* up to $n-1$ rushing cheaters such that $|V_i| = |S|/\epsilon^{k+1}$,
- capable of *identifying* up to $(k-1)/3$ rushing cheaters such that $|V_i| = |S|/\epsilon^k$,
- capable of *identifying* up to $(k-1)/2$ rushing cheaters such that $|V_i| = |S|(\frac{(n-t)^{n+2t}}{\epsilon^{n+2t}})$,
- *robust* secret sharing tolerate up to $(n-1)/2$ rushing cheaters such that $|V_i| = m + (n+k-1)q$ bit.

The first three schemes are all the first scheme in each model such that the bit length of shares does not grow linearly with n. The last two schemes are all the first scheme in each model with lowest share size compared to existing schemes.

To derive a lower bound of sizes of share for various models of secret sharing schemes secure against rushing cheaters will be our future work.

9 Acknowledgments

The research of A.A. and P.S.R. is supported in part by NBHM, DAE, Government of India (No 2/48(10)/2013/ NBHM(R.P.)/R&D II/695). K.S., A.A., and P.S.R. are thankful to DST, Govt. of India and JSPS, Govt. of Japan for providing partial support for this collaborative research work under India Japan Cooperative Science Programme (vide Memo no. DST/INT/JSPS/P-191/2014 dated May 27, 2014). K.M. is supported by a *kakenhi* Grant-in-Aid for Scientific Research (C) No. 15K00186 from Japan Society for the Promotion of Science. R.X. was supported by The China Scholarship Council, No. 201206340057 and National Natural Science Foundation of China No. 61472470.

References

1. T. Araki, "Efficient (k, n) Threshold Secret Sharing Scheme Secure against Cheating from $n-1$ Cheaters," Proc. ACISP'07, Lecture Notes in Computer Science, vol. 4586, Springer Verlag, pp. 133–142, 2007.
2. T. Araki and S. Obana, "Flaws in Some Secret Sharing Schemes against Cheating," Proc. ACISP'07, Lecture Notes in Computer Science, vol. 4586, Springer Verlag, pp. 122–132, 2007.
3. G. R. Blakley, "Safeguarding cryptographic keys," Proc. AFIPS 1979, National Computer Conference, vol. 48, pp. 313–137, 1979.

4. E. F. Brickell and D. R. Stinson, "The Detection of Cheaters in Threshold Schemes," SIAM Journal on Discrete Mathematics, vol. 4, no. 4, pp. 502–510, 1991.
5. M. Carpentieri, "A Perfect Threshold Secret Sharing Scheme to Identify Cheaters," Designs, Codes and Cryptography, vol. 5, no. 3, pp. 183–187, 1995.
6. M. Carpentieri, A. De Santis and U. Vaccaro, "Size of Shares and Probability of Cheating in Threshold Schemes," Proc. Eurocrypt'93, Lecture Notes in Computer Science, vol. 765, Springer Verlag, pp. 118–125, 1993.
7. S. Cabello, C. Padró and G. Sáez, "Secret Sharing Schemes with Detection of Cheaters for a General Access Structure," Designs, Codes and Cryptography, vol. 25, no. 2, pp. 175–188, 2002.
8. Carpentieri, M.: *A perfect threshold secret sharing scheme to identify cheaters.* Design Codes Cryptography 5(3), 183-187 (1995)
9. A. Cevellos, S. Fehr, R. Ostrovsky and Y. Rabani, "Unconditionally-secure Robust Secret Sharing with Compact Shares," Proc. Eurocrypt 2012, Lecture Notes in Computer Science, vol. 7237, Springer Verlag, pp. 195–208, 2012.
10. A. Choudhury, "Brief announcement: Optimal Amortized Secret Sharing with Cheater Identification," Proc. PODC 2012, ACM, pp. 101–101, 2012.
11. R. Cramer, I. Damgård, N. Döttling, S. Fehr and G. Spini, "Linear Secret Sharing Schemes from Error Correcting Codes and Universal Hash Functions," Proc. Enrocrypt 2015, Lecture Notes in Computer Science Volume 9057, 2015, pp. 313-336, 2015.
12. R. Cramer, I. Damgård and S. Fehr, "On the Cost of Reconstruction a Secret, or VSS with Optimal Reconstruction Phase," Proc. Crypto'01, Lecture Notes in Computer Science, vol. 2139, Springer Verlag, pp. 503–523, 2001.
13. R. Cramer, Y. Dodis, S. Fehr, C. Padró, D. Wichs, "Detection of Algebraic Manipulation with Applications to Robust Secret Sharing and Fuzzy Extractors," Proc. Eurocrypt 2008, Lecture Notes in Computer Science, vol. 4965, Springer Verlag, pp. 471–488, 2008.
14. Chor, B., Goldwasser, S., Micali, S., and Awerbuch, B.: *Verifiable Secret Sharing and Achieving Simultaneity in the Presence of Faults (Extended Abstract).* FOCS 1985, 383-395 (1985)
15. B. den Boer, "A Simple and Key-Economical Unconditional Authentication Scheme," Journal of Computer Security, vol. 2, pp. 65–71, 1993.
16. D. Dolev, C. Dwork, O. Waarts, M. Yung, "Perfectly Secure Message Transmission," Journal of the ACM, vol. 40, no. 1, pp. 17–47, 1993.
17. Ishai, Y., Ostrovsky, R., Seyalioglu, H.: *Identifying cheaters without an honest majority.* TCC 2012, 21-38 (2012)
18. Jhanwar M. P., Safavi-Naini R.: *Unconditionally-secure ideal robust secret sharing schemes for threshold and multilevel access structure.* J. Mathematical Cryptology 7(4), 279-296 (2013)
19. Johansson T., Kabatianskii G., Smeets B.: *On the relation between A-codes and codes correcting independent errors.* EUROCRYPT 93, 1-11 (1994)
20. K. Kurosawa, S. Obana and W. Ogata, "t-Cheater Identifiable (k, n) Secret Sharing Schemes," Proc. Crypto'95, Lecture Notes in Computer Science, vol. 963, Springer Verlag, pp. 410–423, 1995.
21. F. MacWilliams and N. Sloane, "The Theory of Error Correcting Codes," North Holland, Amsterdam, 1977.
22. R. J. McEliece and D. V. Sarwate, "On Sharing Secrets and Reed-Solomon Codes," Communications of the ACM, vol 24, issue 9, pp. 583-584, 1981.
23. Martin K. M.: *Challenging the adversary model in secret sharing schemes.*
24. Martin, K.M., Paterson, M.B., Stinson, D.R.: *Error decodable secret sharing and one-round perfectly secure message transmission for general adversary structures.* Cryptography and Communications 3(2), 65-86 (2011)
25. S. Obana, "Almost Optimum t-Cheater Identifiable Secret Sharing Schemes," Proc. Eurocrypt 2011, Lecture Notes in Computer Science, vol. 6631, Springer Verlag, pp. 284–302, 2011.
26. S. Obana and T. Araki, "Almost Optimum Secret Sharing Schemes Secure Against Cheating for Arbitrary Secret Distribution," Proc. Asiacrypt 2006, Lecture Notes in Computer Science, vol. 4284, Springer Verlag, pp. 364–379, 2006.
27. W. Ogata and H. Eguchi, "Cheating Detectable Threshold Scheme against Most Powerful Cheaters for Long Secrets," Designs, Codes and Cryptography, published online, October 2012, 2012.
28. Ogata W., Kurosawa K.: *Provably secure metering scheme.* ASIACRYPT 2000, 388-398 (2000).
29. W. Ogata, K. Kurosawa and D. R. Stinson, "Optimum Secret Sharing Scheme Secure against Cheating," SIAM Journal on Discrete Mathematics, vol. 20, no. 1, pp. 79–95, 2006.
30. T. Pedersen, "Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing," Proc. Crypto'91, Lecture Notes in Computer Science, vol 576, Springer Verlag, pp. 129–149, 1991.

31. T. Rabin and M. Ben-Or, "Verifiable Secret Sharing and Multiparty Protocols with Honest Majority," Proc. STOC'89, pp. 73–85.
32. T. Rabin, "Robust Sharing of Secrets When the Dealer is Honest or Cheating," Journal of the ACM, vol. 41, no. 6, pp. 1089–1109, 1994.
33. Roy P. S., Adhikari A., Xu R., Kirill M., Sakurai K.: *An Efficient Robust Secret Sharing Scheme with Optimal Cheater Resiliency.* SPACE 2014, 47-58 (2014)
34. Roy P. S., Adhikari A., Xu R., Kirill M., Sakurai K.: *An Efficient t-Cheater Identifiable Secret Sharing Scheme with Optimal Cheater Resiliency.* eprint.iacr.org/2014/628.pdf
35. A. Shamir, "How to Share a Secret," Communications of the ACM, vol. 22, issue 11, pp. 612–613, 1979.
36. D. R. Stinson, "On the Connections between Universal Hashing, Combinatorial Designs and Error-Correcting Codes," Congressus Numerantium 114, pp. 7–27, 1996.
37. M. Tompa and H. Woll, "How to Share a Secret with Cheaters," Journal of Cryptology, vol. 1, no. 3, pp. 133-138, 1989.
38. Wegman M.N., Lawrence Carter J.: *New classes and applications of hash functions.* FOCS 1979, 175-182 (1979)
39. Wegman M.N., Lawrence Carter J. "New hash functions and their use in authentication and set equality," Journal of Computer and System Sciences 22 (1981), 265-279.
40. R. Xu, K. Morozov and T. Takagi, "On Cheater Identifiable Secret Sharing Schemes Secure against Rushing Adversary," Proc. IWSEC 2013, Lecture Notes in Computer Science, vol 8231, Springer Verlag, pp. 258–271, 2013.
41. Xu R., Morozov K., Takagi T.: *Cheater Identifiable Secret Sharing Schemes via Multi-Receiver Authentication.* IWSEC 2014, 72-87 (2014)