Simple and Asymptotically Optimal t-Cheater Identifiable Secret Sharing Scheme

Ashish Choudhury

Applied Statistics Unit
Indian Statistical Institute Kolkata India
partho31@gmail.com, partho_31@yahoo.co.in

Abstract. In this paper, we consider the problem of k-out-of-n secret sharing scheme, capable of identifying t cheaters. We design a very simple k-out-of-n secret sharing scheme, which can identify up to t cheaters, with probability at least $1-\epsilon$, where $0 < \epsilon < 1/2$, provided $t < k/2$. This is the maximum number of cheaters, which can be identified by any k-out-of-n secret sharing scheme, capable of identifying t cheaters[1]. In our scheme, the set of all possible i^{th} share $\mathcal{V}_i$ satisfies the condition that $|\mathcal{V}_i| = |\mathcal{S}|/\epsilon^{3n}$, where $\mathcal{S}$ denotes the set of all possible secrets. Moreover, our scheme requires polynomial computation.
In EUROCRYPT 2011, Satoshi Obana presented two SSCI schemes, which can identify up to $t < k/2$ cheaters. However, the schemes require $|\mathcal{V}_i| \approx \frac{(n\cdot(t+1)\cdot 2^{3t-1}\cdot|\mathcal{S}|)}{\epsilon}$ and $|\mathcal{V}_i| \approx \frac{((n\cdot t\cdot 2^{3t})^2\cdot|\mathcal{S}|)}{\epsilon^2}$ respectively. Moreover, both the schemes are computationally *inefficient*, as they require to perform exponential computation in general. So comparing our scheme with the schemes of Obana, we find that not only our scheme is computationally efficient, but in our scheme the share size is significantly smaller than that of Obana. Thus our scheme solves one of the open problems left by Obana, urging to design efficient SSCI scheme with $t < k/2$.
In CRYPT0 1995, Kurosawa, Obana and Ogata have shown that in any SSCI scheme, $|\mathcal{V}_i| \geq \frac{|\mathcal{S}|-1}{\epsilon} + 1$. Though our proposed scheme does not exactly matches this bound, we show that our scheme *asymptotically* satisfies the above bound. To the best of our knowledge, our scheme is the best SSCI scheme, capable of identifying the maximum number of cheaters.

Keywords: Secret Sharing, Cheater Identification, Share Size.

1 Introduction

Consider the following problem: there exists a set of n parties, denoted by $\mathcal{P} = \{P_1, \ldots, P_n\}$ and a special party called *dealer*, denoted by D. The dealer has a secret, which he wants to share among the n parties in such a way, that the following two conditions are satisfied:

[1] In the rest of the paper, we call these schemes as Secret Sharing with Cheater Identification (SSCI).

1. *Correctness*: Any set of k or more parties can reconstruct the secret by pooling their shares.
2. *Perfect Secrecy*: Any set of $k-1$ or less number of parties will have no information about the secret (in information theoretic sense) by pooling their shares.

The above problem is the well known k-out-of-n secret sharing (SS) problem, which was first formulated by Shamir [Sha79] and independently by Blackley [Bla79]. It is one of the fundamental problem in cryptography and has been extensively studied over the past three decades. Any SS scheme consists of the following two phases:

1. *Sharing Phase*: During this phase, D shares the secret among the n parties.
2. *Reconstruction Phase*: In this phase, a set of parties (of size at least k) pool their shares to reconstruct the secret.

In the traditional SS schemes (like the one by Shamir [Sha79]), it is assumed that the parties will submit correct shares during the reconstruction phase. However, this does not model the real life scenario because in practice, some of the parties may produce incorrect shares, in order to ensure that the honest parties reconstruct incorrect secret.

Preventing parties from producing incorrect shares is one of the hot research topics in the area of secret sharing. Tompa and Woll [TW88] first presented an SS scheme, which can *detect* cheaters, when invalid shares are produced during the reconstruction phase. This work is followed by several other works (for example, [Ara07,AO07,CDF+08,CPS02,OA06,OKS06]) where upper bound on the size of the shares are derived and efficient schemes are presented. However, all these schemes can *only detect* cheating, without identifying the exact identity of the cheaters, who submitted incorrect shares.

Secret Sharing with Cheater Identification (SSCI) : Secret sharing schemes, which can not only *detect* the cheaters, but can also *identify* the cheaters (who submitted incorrect shares) is another interesting area of research. McElice and Sarwate [MS81] were the first to point out cheater identification in secret sharing schemes. They observed that the list of shares of a k-out-of-n Shamir secret sharing scheme [Sha79] is nothing, but the components of a Reed-Solomon code [MS78] with dimension k. So if $k+2t+1$ shares are revealed during the reconstruction phase, out of which at most t could be corrupted, then by applying the standard Reed-Solomon decoding algorithm, we can identify the exact identity of the t cheaters, who produced invalid shares. But this process requires the availability of more than k shares during the reconstruction phase. The question is whether we can do the cheater identification with the minimum number of shares (namely k), which are required to reconstruct the secret. SSCI is the answer to this question.

In the model of SSCI, there exists a set of n parties, denoted by $\mathcal{P} = \{P_1, \ldots, P_n\}$ and a special party called the *dealer*, denoted by D. There exists two different centralized adversaries, denoted by $\mathcal{A}_{Lis}$ and $\mathcal{A}_{Cheat}$ respectively.

The adversary $\mathcal{A}_{Lis}$ is a *static, computationally unbounded passive* adversary, who can control any $k-1$ out of the n parties. The parties under the control of $\mathcal{A}_{Lis}$ will honestly follow the protocol, but at the same time will leak complete information about their internal state and computation to $\mathcal{A}_{Lis}$. On the other hand, the adversary $\mathcal{A}_{Cheat}$ is a *static, computationally unbounded active* adversary, who can control any t out of the n parties in *Byzantine* fashion. Thus $\mathcal{A}_{Cheat}$ will not only have full information about the computation and communication of the parties under its control, but $\mathcal{A}_{Cheat}$ can also dictate these parties to behave in any arbitrary manner during the protocol. *Moreover, it is assumed that $\mathcal{A}_{Lis}$ does not co-operate with $\mathcal{A}_{Cheat}$. This implies that $\mathcal{A}_{Cheat}$ will not get any information about the computation and communication of the parties, which may be under the control of $\mathcal{A}_{Lis}$, but not under the control of $\mathcal{A}_{Cheat}$. Similarly, $\mathcal{A}_{Lis}$ will not get any information about the computation and communication of the parties, which may be under the control of $\mathcal{A}_{Cheat}$, but not under the control of $\mathcal{A}_{Lis}$.* To illustrate this, consider $\mathcal{P} = \{P_1, \ldots, P_5\}$ and let $t = 3$ and $k = 4$. Then it may happen that $\mathcal{A}_{Lis}$ controls P_1, P_2 and P_3, while $\mathcal{A}_{Cheat}$ controls P_3, P_4 and P_5. So $\mathcal{A}_{Lis}$ will only know the computation and communication of the parties P_1, P_2 and P_3, with no access to the computation and communication of the parties P_4 and P_5. Similarly, $\mathcal{A}_{Cheat}$ will have access to the computation and communication of the parties P_3, P_4 and P_5, with no access to the computation and communication of the parties P_1 and P_2.

Any SSCI scheme consists of the following two phases:

1. *Sharing Phase*: During this phase, D takes the secret S and generates n shares for the secret, denoted by $Sh_1, \ldots, Sh_n$ and assign Sh_i to party P_i.
2. *Reconstruction Phase*: During this phase, a set of m parties, where $m \geq k$, *publicly* produce their shares to reconstruct the secret. These m parties can be any m parties out of the n parties.
   (a) Then a cheating identification algorithm is publicly applied on the m shares produced by the m parties to identify the invalid shares.
   (b) Let L be the set of parties, who are identified to be the cheaters by the cheater identification algorithm.
   (c) If $(m-|L|) \geq k$, then a publicly known reconstruction function, say Rec, is applied on the shares produced by the parties not in L, to reconstruct a secret $\hat{S}$. Finally, $\hat{S}$ and L is the output of the reconstruction phase.
   (d) If $(m-|L|) < k$, then $\perp$ and L is the output of the reconstruction phase.

Any SSCI scheme should satisfy the following properties:

1. *Perfect Secrecy*: At the end of the sharing phase, the adversary $\mathcal{A}_{Lis}$ should not get any information about the secret S (in information theoretic sense) from the shares of the parties (at most $k-1$) under its control.
2. *Correctness*: The following two conditions must be satisfied:
   (a) During the reconstruction phase, if any party P_i is under the control of $\mathcal{A}_{Cheat}$ and produces incorrect share $Sh_i' \neq Sh_i$, then except with error probability ϵ, P_i will be identified as a cheater and will be included in the set L. Here $0 < \epsilon < 1/2$.

(b) During the reconstruction phase, if any $\hat{S} \neq \perp$ is reconstructed, then $S = \hat{S}$, except with error probability ϵ, where $0 < \epsilon < 1/2$.

We next note down few important notes.

Note 1. (**A Note on the Correctness Conditions**): Note that the two conditions under Correctness are equivalent, in the sense that one implies the other. This is because an incorrect secret $\hat{S} \neq S$ is reconstructed if and only if a party under the control of $\mathcal{A}_{Cheat}$, is successfully able to produce an invalid share, without being identified by the cheater identification algorithm. □

Note 2. (**A Note on the Rushing Adversary**): Notice that we assume that both $\mathcal{A}_{Lis}$ and $\mathcal{A}_{Cheat}$ are *static* and does not follow *rushing* strategy. Specifically, during the reconstruction phase, if some party P_i is under the control of $\mathcal{A}_{Cheat}$, then P_i *does not wait* for the honest parties to reveal their shares, and accordingly modify his share, before producing his share (possibly corrupted). This is in contrast to the well known *rushing* strategy (see, for example [GIKR01]), where the corrupted P_i would have first listened to the shares revealed by the honest parties and accordingly, would have modified his share and then would have produced his share (possibly corrupted). This is in accordance with all the previous SSCI schemes, where rushing is not allowed. Thus, the share produced by any cheater during the reconstruction phase will be completely independent of the shares produced by the honest parties.

Interestingly, our scheme can be easily modified to deal with the rushing strategy, provided one extra round of communication is allowed during the reconstruction phase. We will discuss more about this during the formal discussion of our scheme. □

Note 3. (**Difference Between SSCI and Verifiable Secret Sharing (VSS)**): In any VSS scheme [CGMA85], it is assumed that D may also be corrupted and he may distribute inconsistent shares. Moreover, during the reconstruction phase, the corrupted parties may produce invalid shares. So during the sharing phase, the parties have to interact with each other to ensure that D has distributed consistent shares. And during the reconstruction phase, cheater identification algorithm has to be applied to identify the cheaters.

On the other hand, in SSCI schemes, D is assumed to be honest and he will distribute consistent shares to the parties. It is only during the reconstruction phase that the parties may try to cheat by producing invalid shares. So SSCI has *weaker* requirements than VSS. Nevertheless, SSCI is an important problem in its own right and has been studied extensively in the literature (see the **Existing Literature** in the sequel). □

Parameters of any SSCI Scheme : Any SSCI scheme has the following important parameters:

1. *Secret Space* $\mathcal{S}$: It is the set of all possible secrets, from which D will select an element to share, according to the underlying probability distribution of $\mathcal{S}$. Without loss of generality, we assume that D can select any element from

$\mathcal{S}$ as the secret, uniformly and randomly. This is in accordance with all the previous SSCI schemes.

2. i^{th} *Share Space* $\mathcal{V}_i$: It is the set of all possible i^{th} share, which can occur during any execution of the SSCI scheme. During the execution of the protocol, any value from $\mathcal{V}_i$ can be assigned as the i^{th} share to P_i. The choice of the value depends upon the secret to be shared and the random coin tosses of D.
3. *Eavesdropping Threshold*: It is the maximum number of parties $k-1$ which can be under the control of $\mathcal{A}_{Lis}$ during the sharing phase.
4. *Cheating Threshold*: It is the maximum number of parties t which can be under the control of $\mathcal{A}_{Cheat}$ during the reconstruction phase.
5. *Computational Complexity*: It is the amount of computation, done throughout the protocol. An SSCI scheme will be called *efficient*, if it performs computation, which is polynomial in n, k and t.

Types of SSCI Schemes : In the literature, there are two types of SSCI schemes:

1. *Secret Sharing with Private Cheater Identification*: In these schemes, during the reconstruction phase, the cheater identification algorithm will take a *base share* and a list of other shares and try to identify the cheaters. So the base share becomes a *basis* for deciding whether a participant submitting a share during the reconstruction phase is a cheater or not. Such schemes were studied in [RBO89,Car95,OK00]
2. *Secret Sharing with Public Cheater Identification*: In these schemes, during the reconstruction phase, the cheater identification algorithm is applied publicly to a list of shares, which are revealed by a set of parties publicly. So there is no concept of base share and each revealed share has equivalent weightage in the cheater identification algorithm.

In this paper, our focus is on SSCI with public cheater identification. Extensive research has been done in the past to establish bounds on $|\mathcal{V}_i|$ and study the relationship between t and k. We now give a brief overview of the existing literature on SSCI schemes.

Existing Literature on SSCI Schemes with Public Cheater Identification : It is well known that SSCI scheme, capable of identifying up to t cheaters is possible if and only if $t < k/2$ [KOO95,Oba11]. So any SSCI scheme where $k = 2t + 1$ is said to have *optimal cheating threshold*. In [KOO95], it is shown that in any SSCI scheme, the following lower bound must be satisfied:

$$|\mathcal{V}_i| \geq \frac{|\mathcal{S}| - 1}{\epsilon} + 1. \tag{1}$$

So any SSCI scheme, where $|\mathcal{V}_i|$ exactly matches the above bound is said to be *optimal*. We now summarize the properties of the best known existing SSCI schemes in Table 1.

Table 1. Properties of the best known existing SSCI schemes with public cheater identification

Reference	t	$\lvert\mathcal{V}_i\rvert$	Efficient/Inefficient
[KOO95]	$t < k/3$	$\lvert\mathcal{V}_i\rvert = \lvert\mathcal{S}\rvert/\epsilon^{t+2}$	Efficient
[Oba11]	$t < k/3$	$\lvert\mathcal{V}_i\rvert = \lvert\mathcal{S}\rvert/\epsilon$	Efficient
[Oba11]	$t < k/2$	$\lvert\mathcal{V}_i\rvert \approx (n \cdot (t+1) \cdot 2^{3t-1} \cdot \lvert\mathcal{S}\rvert)/\epsilon$	*Inefficient*
[Oba11]	$t < k/2$	$\lvert\mathcal{V}_i\rvert \approx ((n \cdot t \cdot 2^{3t})^2 \cdot \lvert\mathcal{S}\rvert)/\epsilon^2$	*Inefficient* [a]

[a] In [Oba11], two different (inefficient) schemes were presented with $t < k/2$.

Our Results : From Table 1, we find that best known SSCI schemes with *optimal cheating threshold* are due to [Oba11]. But both these schemes are inefficient. Moreover, the share size $|\mathcal{V}_i|$ of these schemes is no where near to the lower bound, given in Eqn. 1. In [Oba11], it is left as an open problem to design an efficient SSCI scheme with $k = 2t + 1$, with reduced share size $|\mathcal{V}_i|$. In this paper, we make a positive step towards this direction. Specifically, we design a new SSCI scheme with $k = 2t+1$, whose properties are summarized in Table 2.

Table 2. Properties of our new SSCI scheme

$t < k/2$	$\lvert\mathcal{V}_i\rvert = \lvert\mathcal{S}\rvert/\epsilon^{3n}$	Efficient

Comparing Table 2 with the last two rows of Table 1, we find that not only our scheme is computationally efficient, but in our scheme the share size is significantly smaller than that of Obana [Oba11]. Moreover, though our proposed scheme does not exactly match the lower bound given in Eqn. 1, we show that our scheme *asymptotically* satisfies the bound. To the best of our knowledge, our scheme is the best SSCI scheme, capable of identifying the maximum number of allowed cheaters.

Roadmap : In the next section, we discuss few notations and preliminaries. Our new SSCI scheme is presented in Sec. 3. In Sec. 4, we show that our proposed scheme *asymptotically* satisfies the lower bound given in Eqn. 1. In Sec. 5 we show that how we can modify our scheme to deal with a rushing adversary (cheater). We end the paper with a conclusion and directions for further research.

2 Preliminaries and Notations

We assume that the underlying network is a synchronous network and every party knows the identity of every other party. Let $GF(p)$ be a Galois field, where p is a prime power, satisfying the conditions that $p \geq n$ and $p = \frac{1}{\epsilon}$. We

assume that all computation and communication in our scheme is performed over $GF(p)$. In our scheme, the error probability of ϵ comes from the fact that in our scheme, a cheater will be successful, if he can correctly guess a random value, selected uniformly and randomly by D from $GF(p)$. And the cheater will be able to do so with probability at most $\frac{1}{\epsilon}$ (the formal details will appear in our scheme). We are now ready to discuss our scheme, which we do in the next section.

3 Our New SSCI Scheme

Let $k = 2t + 1$. Then we present a very simple and efficient SSCI scheme. Our scheme allows D to shares a secret $S = (s^1, \ldots, s^\ell)$, consisting of ℓ elements, selected uniformly and randomly[2] from $GF(p)$, where $\ell > 1$. So $|\mathcal{S}| = p^\ell$. In our scheme, the share Sh_i of each party P_i will consist of $\ell + 3n$ elements from $GF(p)$. Thus $|\mathcal{V}_i| = p^{\ell+3n} = \frac{|\mathcal{S}|}{\epsilon^{3n}}$, as $p = 1/\epsilon$. We now discuss the high level idea of the protocol.

High Level Idea of the Protocol : The first step that D would do is to generate n Shamir shares [Sha79] for each $s^l \in S$, for a k-out-of-n Shamir secret sharing. Let $Sha_{l,i}$ denote the i^{th} Shamir share of the l^{th} secret s^l, for $l = 1, \ldots, \ell$ and $i = 1, \ldots, n$. Then D will give ℓ i^{th} Shamir shares, namely $Sha_{1,i}, \ldots, Sha_{\ell,i}$ to party P_i. Till now, D has done k-out-of-n Shamir secret sharing for each $s^l \in S$. However, this is not sufficient to get an SSCI scheme, as there is no way by which we can identify a corrupted party, who may submit incorrect Shamir share during the reconstruction phase. So D has to also give some *authentication information* about the Shamir shares to the parties, which will enable the parties to identify the cheaters (submitting incorrect Shamir shares) with very high probability.

The authentication information about the shares is distributed by D, as explained in Table 3. We explain the distribution, as done by D, with respect to a pair of parties $P_i, P_j \in \mathcal{P}$. A similar distribution will be done by D, for every $P_i, P_j \in \mathcal{P}$.

Now the distribution as done by D in Table 3 achieves the following properties:

1. If P_i is *corrupted* and under the control of $\mathcal{A}_{Cheat}$ and if P_j is *honest*, then later, except with probability $\frac{1}{p} = \epsilon$, party P_i cannot produce an incorrect $p_i'(x) \neq p_i(x)$, without being caught by the honest P_j. This is because P_i will have no information about the authentication key $key_{i,j}$ and the authentication value $Auth_{i,j}$ held by the honest P_j. So the only way, a corrupted P_i can cheat an honest P_j is by *guessing* the value of $key_{i,j}$, which he can do with probability at most $1/p = \epsilon$. So the distribution will help to achieve the *Correctness* property.

[2] Our scheme will work even if there exists a probability distribution, associated with $\mathcal{S}$.

Table 3. Information distributed by D to a pair of parties $P_i, P_j \in \mathcal{P}$.

Communication by D to Party P_i	Communication by D to Party P_j
1. Polynomial $p_i(x)$ of degree-$(\ell - 1)$ where $p_i(x) = Sha_{1,i} + Sha_{2,i} \cdot x + \ldots + Sha_{\ell,i} \cdot x^{\ell-1}$. This is same as giving the i^{th} Shamir share of each s^l to P_i.	1. A random *authentication key*, denoted by $key_{i,j}$, selected uniformly and randomly from $GF(p)$. 2. *Authentication value* of $p_i(x)$, namely $Auth_{i,j} = p_i(key_{i,j})$.

2. If P_i is *honest* and if P_j is under the control of $\mathcal{A}_{Lis}$, then $\mathcal{A}_{Lis}$ will learn $key_{i,j}$ and the value $Auth_{i,j} = p_i(key_{i,j})$. This will leak "some information" about the polynomial $p_i(x)$ (and hence about the i^{th} Shamir share of each s^l) to $\mathcal{A}_{Lis}$. And this will *violate* the perfect secrecy condition.

So from the above discussion, we find that by simply doing the distribution of information as done in Table 3, we cannot get an SSCI scheme, as it will fail to preserve the perfect secrecy property. To preserve the perfect secrecy property, we add an *extra* step to the distribution, as done in Table 3. The final distribution as done by D is shown in Table 4.

Table 4. Information finally distributed by D to a pair of parties $P_i, P_j \in \mathcal{P}$ in our SSCI scheme.

Communication by D to Party P_i	Communication by D to Party P_j
1. Polynomial $p_i(x) = Sha_{1,i} + Sha_{2,i} \cdot x + \ldots + Sha_{\ell,i} \cdot x^{\ell-1}$. 2. A random *masking key*, denoted by $mask_{i,j}$, selected uniformly and randomly from $GF(p)$.	1. A random *authentication key*, denoted by $key_{i,j}$, selected uniformly and randomly from $GF(p)$. 2. *Masked authentication value* of $p_i(x)$, namely $Auth_{i,j} = p_i(key_{i,j}) + mask_{i,j}$.

Now it is easy to see that the distribution of information, as done in Table 4, will help to achieve perfect secrecy, as well as the correctness property. For correctness, if P_i is *corrupted* and P_j is honest, then the probability that P_i will be able to cheat P_j by producing incorrect Shamir shares (i.e., incorrect polynomial $p_i(x)$) is still $1/p = \epsilon$. This is because P_i will have to correctly guess $key_{i,j}$. On the other hand, if P_i is *honest*, then even if P_j is under the control of $\mathcal{A}_{Lis}$, the adversary $\mathcal{A}_{Lis}$ will get no information about $p_i(x)$ after knowing $Auth_{i,j}$. This is because $\mathcal{A}_{Lis}$ will have no information about the masking key $mask_{i,j}$, which is held with honest P_i. Our SSCI scheme is formally given in Fig. 1. The protocol **ShareGen** is the protocol for the sharing phase, while protocol **ReConst** is the protocol for the reconstruction phase.

We now prove the properties of our SSCI scheme.

Fig. 1. SSCI Scheme with $k = 2t + 1$

Sharing Phase: Protocol ShareGen

Input: Secret $S = (s^1, \ldots, s^\ell)$, containing ℓ elements from $GF(p)$.
Output: A list of n shares $Sh_1, \ldots, Sh_n$.
Computation by D:

1. For $l = 1, \ldots, \ell$, select a random degree-$(k-1)$ polynomial $f_l(x)$ over $GF(p)$, such that $f_l(0) = s^l$. Compute $Sha_{l,i} = f_l(i)$, for $i = 1, \ldots, n$.
2. For $i = 1, \ldots, n$, let $p_i(x) = Sha_{1,i} + Sha_{2,i} \cdot x + \ldots + Sha_{\ell,i} \cdot x^{\ell-1}$ be the polynomial of degree-$(\ell - 1)$. Corresponding to the polynomial $p_i(x)$, select n random masking keys $mask_{i,1}, \ldots, mask_{i,n}$ and n random authentication keys $key_{i,1}, \ldots, key_{i,n}$ from $GF(p)$. Compute $Auth_{i,j} = p_i(key_{i,j}) + mask_{i,j}$, for $j = 1, \ldots, n$.
3. For $i = 1, \ldots, n$, assign party P_i the share Sh_i, where Sh_i consists of the following:
   (a) Polynomial $p_i(x)$;
   (b) The n masking keys $mask_{i,1}, \ldots, mask_{i,n}$;
   (c) The n authentication keys $key_{1,i}, \ldots, key_{n,i}$ and the corresponding n authentication values $Auth_{1,i}, \ldots, Auth_{n,i}$.

Thus $Sh_i = \{p_i(x), (mask_{i,1}, \ldots, mask_{i,n}), (key_{1,i}, \ldots, key_{n,i}), (Auth_{1,i}, \ldots, Auth_{n,i})\}$.

Secret Reconstruction and Cheater Identification: Protocol ReConst

Input: A list of m shares, where $m \geq k$. Let $CORE$ be the set of m parties who have produced their shares and let Sh'_i denote the share produced by $P_i \in CORE$. Moreover, let $Sh'_i = \{p'_i(x), (mask'_{i,1}, \ldots, mask'_{i,n}), (key'_{1,i}, \ldots, key'_{n,i}), (Auth'_{1,i}, \ldots, Auth'_{n,i})\}$ for each $P_i \in CORE$. Furthermore, at most t out of these m shares can be corrupted.

Output: Either $(\perp, L)$ or $(\hat{S}, L)$, where L is the list of cheaters.

1. For each $P_i \in CORE$, compute $Support_i = \{P_j \in CORE : Auth'_{i,j} = p'_i(key'_{i,j}) + mask'_{i,j}\}$. If $Support_i \geq t + 1$, then consider $p'_i(x)$ as a valid polynomial.
2. If $Support_i \leq t$ for some $P_i \in CORE$ then P_i is identified as a cheater and added to a list of cheaters L, which is initially $\emptyset$.
3. If $|L| > m - k$ then output $(\perp, L)$.
4. If $|L| \leq m - k$, then consider all the valid polynomials $p'_i(x)$, such that $P_i \notin L$. Let each such $p'_i(x)$ be of the form $p'_i(x) = Sha'_{1,i} + Sha'_{2,i} \cdot x + \ldots + Sha'_{\ell,i} \cdot x^{\ell-1}$.
5. For $l = 1, \ldots, \ell$, check using Lagrange interpolation, whether all the values $Sha'_{l,i}$'s, where $P_i \notin L$, lie on a unique degree-$(k-1)$ polynomial, say $f'_l(x)$.
   (a) If $\exists l \in \{1, \ldots, \ell\}$, such that all the values $Sha'_{l,i}$'s, where $P_i \notin L$, does not lie on a unique degree-$(k-1)$ polynomial, then output $(\perp, L)$.
   (b) If for every $l \in \{1, \ldots, \ell\}$, all the values $Sha'_{l,i}$'s, where $P_i \notin L$, lie on a unique degree-$(k-1)$ polynomial, say $f'_l(x)$, then let $\hat{S} = (f'_1(0), \ldots, f'_\ell(0))$. Output $(\hat{S}, L)$.

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

PROOF: Without loss of generality, let the listening adversary $\mathcal{A}_{Lis}$ controls the first $k-1$ parties, namely $P_1, \ldots, P_{k-1}$ during the sharing phase. So the adversary will know the polynomials $p_1(x), \ldots, p_{k-1}(x)$ and hence the Shamir shares $Sha_{l,i}$, for $l = 1, \ldots, \ell$ and $i = 1, \ldots, k-1$. However, from these shares, the adversary will not get any information about $s^1, \ldots, s^\ell$, due to the properties of k-out-of-n Shamir secret sharing scheme [Sha79]. The listening adversary will also know the values $Auth_{k,i}, \ldots, Auth_{n,i}$, for $i = 1, \ldots, k-1$. But this will not reveal any information about the polynomials $p_k(x), \ldots, p_n(x)$, as the adversary will not know the corresponding masking keys $mask_{k,i}, \ldots, mask_{n,i}$. So during the sharing phase, the listening adversary will only know $k-1$ points on each $f^l(x)$, for $l = 1, \ldots, \ell$. The secrecy of $S = (s^1, \ldots, s^\ell)$ now follows from the properties of k-out-of-n Shamir secret sharing scheme [Sha79]. □

Lemma 2 (Correctness). *The scheme in Fig. 1 satisfies correctness condition. That is, during the reconstruction phase, if any $P_i \in CORE$ is under the control of A_{Cheat} and produces $p'_i(x) \neq p_i(x)$, then except with error probability ϵ, P_i will be identified as a cheater and will be included in the list L.*

PROOF: Without loss of generality, let $CORE$ consists of the first m parties, namely $P_1, \ldots, P_m$, where $m \geq k$. Moreover, let $P_1, \ldots, P_t$ be the under the control of $\mathcal{A}_{Cheat}$. Now suppose that P_1 submits $p'_1(x) \neq p_1(x)$ 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 $\mathcal{A}_{Cheat}$. 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 $p'_1(key'_{1,j}) + mask'_{1,j} = Auth'_{1,j}$. Now notice that $\mathcal{A}_{Cheat}$ will have no information about the authentication key $key'_{1,j} = key_{1,j}$ and the corresponding authentication value $Auth'_{1,j} = Auth_{1,j}$, as they are with the honest party P_j. Moreover, $Auth'_{1,j} = Auth_{1,j} = p_1(key_{1,j}) + mask_{1,j}$. So the probability that P_1 can ensure that $p_1(key_{1,j}) + mask_{1,j} = p'_1(key_{1,j}) + mask'_{1,j}$, even if $p'_1(x) \neq p_1(x)$ is same as the probability that P_1 correctly guesses $key_{1,j}$. But the probability that P_1 correctly guesses $key_{1,j}$ is $\frac{1}{p} = \epsilon$, as $key_{1,j}$ is uniformly and randomly selected from $GF(p)$. □

Lemma 3 (Share Size). *In our SSCI scheme, $|\mathcal{V}_i| = \frac{|\mathcal{S}|}{\epsilon^{3n}}$.*

PROOF: During the sharing phase, each party gets a polynomial of degree-$(\ell-1)$, consisting of ℓ coefficients, n masking keys, n authentication keys and n authentication values from $GF(p)$. So $|\mathcal{V}_i| = p^{\ell+3n} = p^\ell \cdot p^{3n} = |\mathcal{S}| \cdot p^{3n} = \frac{|\mathcal{S}|}{\epsilon^{3n}}$. This is because the secret S consists of ℓ elements from $GF(p)$ and hence the secret space $\mathcal{S}$ has the cardinality $|\mathcal{S}| = p^\ell$. Moreover, $p = \frac{1}{\epsilon}$. □

We now finally state the following theorem.

Theorem 1. *Let $k = 2t+1$, $p = 1/\epsilon$ and $|\mathcal{S}| = p^\ell$, where $\ell > 1$. Then there exists an efficient SSCI scheme, which can identify up to t cheaters, such that $|\mathcal{V}_i| = \frac{|\mathcal{S}|}{\epsilon^{3n}}$.*

PROOF: The proof follows from the previous three lemmas. □

4 Asymptotic Optimality of Our Scheme

We now show that our scheme (presented in the last section), asymptotically satisfies the lower bound of Kurosawa et al. [KOO95], as given in Eqn. 1. According to the lower bound,

$$|\mathcal{V}_i| \geq \frac{|\mathcal{S}| - 1}{\epsilon} + 1,$$

which implies that

$$\log |\mathcal{V}_i| \geq \log (|\mathcal{S}| - 1) - \log \epsilon,$$

and which further implies that $\log |\mathcal{V}_i| = \Omega(\log |\mathcal{S}|)$.

Now in our scheme, $|\mathcal{V}_i| = \frac{|\mathcal{S}|}{\epsilon^{3n}}$. Thus, $\log |\mathcal{V}_i| = \log |\mathcal{S}| - 3n \log \epsilon$. In our scheme, $|\mathcal{S}| = p^{\ell}$ and so $\log |\mathcal{S}| = \ell \cdot \log p$. So for sufficiently large value of ℓ (for example, $\ell = n$), in our scheme $\log |\mathcal{V}_i| = \mathcal{O}(\log |\mathcal{S}|)$. Thus, our scheme asymptotically satisfies the lower bound. So we can state the following theorem.

Theorem 2 (Asymptotic Optimality). *Our SSCI scheme asymptotically satisfies the lower bound*

$$|\mathcal{V}_i| \geq \frac{|\mathcal{S}| - 1}{\epsilon} + 1.$$

PROOF: Follows from the above discussion. □

5 SSCI Scheme against Rushing Cheater

The SSCI scheme presented in Sec. 3 will fail to achieve its properties against a *rushing* $\mathcal{A}_{Cheat}$. Recall that a rushing adversary [GIKR01] is an adversary, who first wait to listen all the messages sent by the honest parties, before sending his own messages in any protocol. If $\mathcal{A}_{Cheat}$ is rushing, then she can foil our scheme as follows: During the reconstruction phase, $\mathcal{A}_{Cheat}$ will first wait for all the honest parties (at least $t+1$) in $CORE$ to submit their complete shares (which includes the authentication keys and authentication values corresponding to the $p_i(x)$ polynomials of all the n parties). Now $\mathcal{A}_{Cheat}$ will know the authentication keys and the corresponding authentication values, held by the honest parties in $CORE$, corresponding to the $p_i(x)$ polynomials of the parties under the control of $\mathcal{A}_{Cheat}$. So now, the parties under the control of $\mathcal{A}_{Cheat}$ can produce any $p'_i(x) \neq p_i(x)$, which matches the authentication values, held by the honest parties in $CORE$. This will ensure that even if a corrupted party $P_i \in CORE$ produces incorrect $p'_i(x) \neq p_i(x)$, still all the honest parties in $CORE$ are present in $Support_i$ and hence P_i is not identified as a cheater. More specifically, Lemma 2 will not hold if $\mathcal{A}_{Cheat}$ is rushing.

To deal with the above problem, we modify the reconstruction phase as follows: the reconstruction phase will now consists of two rounds, where instead of producing the shares in a single round (as done in our scheme), the parties will submit the shares in parts in two consecutive rounds. During the first round, the parties will only submit their $p_i(x)$ polynomial. And during the second round, the parties will submit the remaining portion of their share, namely the masking

keys, authentication keys and the authentication values. It is now easy to see that by doing so, we can tolerate even a rushing $\mathcal{A}_{Cheat}$. This is because the parties under the control of $\mathcal{A}_{Cheat}$ will have to produce their $p_i(x)$ polynomial during the first round itself. And while producing these polynomials (possibly changed), $\mathcal{A}_{Cheat}$ will have no information about the authentication keys and the authentication values, as held by the honest parties in $CORE$, corresponding to the polynomials of the parties under the control of $\mathcal{A}_{Cheat}$. This is because now the authentication keys and authentication values will be revealed only during the second round, once the polynomials are revealed by *all* the parties in $CORE$. Now by incorporating this modification, our SSCI scheme will work even against a rushing $\mathcal{A}_{Cheat}$.

6 Conclusion and Directions for Further Research

In this paper, we have solved one of the open problems, raised in [Oba11]. Specifically, we have designed a very simple and computationally efficient SSCI scheme with public cheater identification, which can identify the maximum number of allowed cheaters. Moreover, the share size in our scheme is significantly smaller than that of [Oba11]. Furthermore, we have shown that our scheme *asymptotically* matches the lower bound on the share size of SSCI schemes, as given in [KOO95]. It is an interesting open question to see whether we can design efficient SSCI scheme, which can identify the maximum number of allowed cheaters, such that the share size of the scheme *exactly* matches the lower bound of [KOO95].

Acknowledgement: The author would like to thank Prof. Kaoru Kurosawa for answering several questions related to secret sharing with cheater identification schemes.

References

[AO07] T. Araki and S. Obana. Flaws in some secret sharing schemes against cheating. In J. Pieprzyk, H. Ghodosi, and E. Dawson, editors, *Information Security and Privacy, 12th Australasian Conference, ACISP 2007, Townsville, Australia, July 2-4, 2007, Proceedings*, volume 4586 of *Lecture Notes in Computer Science*, pages 122–132. Springer Verlag, 2007.

[Ara07] T. Araki. Efficient (k, n) threshold secret sharing schemes secure against cheating from $n-1$ cheaters. In J. Pieprzyk, H. Ghodosi, and E. Dawson, editors, *Information Security and Privacy, 12th Australasian Conference, ACISP 2007, Townsville, Australia, July 2-4, 2007, Proceedings*, volume 4586 of *Lecture Notes in Computer Science*, pages 133–142. Springer Verlag, 2007.

[Bla79] G. Blakley. Safeguarding cryptographic keys. In *Proc. AFIPS'79 National Computer Conference*, volume 48, pages 313–317, New York, June 1979.

[Car95] M. Carpentieri. A perfect threshold secret sharing scheme to identify cheaters. *Des. Codes Cryptography*, 5(3):183–187, 1995.

[CDF+08] R. Cramer, Y. Dodis, S. Fehr, C. Padró, and D. Wichs. Detection of algebraic manipulation with applications to robust secret sharing and fuzzy extractors. In N. P. Smart, editor, *Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings*, volume 4965 of *Lecture Notes in Computer Science*, pages 471–488. Springer Verlag, 2008.

[CGMA85] B. Chor, S. Goldwasser, S. Micali, and B. Awerbuch. Verifiable Secret Sharing and Achieving Simultaneity in the Presence of Faults (Extended Abstract). In *26th Annual Symposium on Foundations of Computer Science, 21-23 October 1985, Portland, Oregon, USA*, pages 383–395. IEEE, 1985.

[CPS02] S. Cabello, C. Padró, and G. Sáez. Secret sharing schemes with detection of cheaters for a general access structure. *Des. Codes Cryptography*, 25(2):175–188, 2002.

[GIKR01] R. Gennaro, Y. Ishai, E. Kushilevitz, and T. Rabin. The Round complexity of verifiable secret sharing and secure multicast. In *Proceedings on 33rd Annual ACM Symposium on Theory of Computing, July 6-8, 2001, Heraklion, Crete, Greece*, pages 580–589. ACM, 2001.

[KOO95] K. Kurosawa, S. Obana, and W. Ogata. t-cheater identifiable (k, n) threshold secret sharing schemes. In D. Coppersmith, editor, *Advances in Cryptology - CRYPTO '95, 15th Annual International Cryptology Conference, Santa Barbara, California, USA, August 27-31, 1995, Proceedings*, volume 963 of *Lecture Notes in Computer Science*, pages 410–423. Springer Verlag, 1995.

[MS78] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error Correcting Codes*. North-Holland Publishing Company, 1978.

[MS81] R. J. McEliece and D. V. Sarwate. On sharing secrets and reed solomon codes. *Communications of the ACM*, 24(9):583–584, 1981.

[OA06] S. Obana and T. Araki. Almost optimum secret sharing schemes secure against cheating for arbitrary secret distribution. In X. Lai and K. Chen, editors, *Advances in Cryptology - ASIACRYPT 2006, 12th International Conference on the Theory and Application of Cryptology and Information Security, Shanghai, China, December 3-7, 2006, Proceedings*, volume 4284 of *Lecture Notes in Computer Science*, pages 364–379. Springer Verlag, 2006.

[Oba11] S. Obana. Almost optimum t-cheater identifiable secret sharing schemes. In K. G. Paterson, editor, *Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings*, volume 6632 of *Lecture Notes in Computer Science*, pages 284–302. Springer Verlag, 2011.

[OK00] W. Ogata and K. Kurosawa. Provably secure metering scheme. In T. Okamoto, editor, *Advances in Cryptology - ASIACRYPT 2000, 6th International Conference on the Theory and Application of Cryptology and Information Security, Kyoto, Japan, December 3-7, 2000, Proceedings*, volume 1976 of *Lecture Notes in Computer Science*, pages 388–398. Springer Verlag, 2000.

[OKS06] W. Ogata, K. Kurosawa, and D. R. Stinson. Optimum secret sharing scheme secure against cheating. *SIAM J. Discrete Math.*, 20(1):79–95, 2006.

[RBO89] T. Rabin and M. Ben-Or. Verifiable secret sharing and multiparty protocols with honest majority. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washigton, USA*, pages 73–85. ACM, 1989.

[Sha79] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[TW88] M. Tompa and H. Woll. How to share a secret with cheaters. *J. Cryptology*, 1(2):133–138, 1988.