A Fair (t, n)-Threshold Secret Sharing Scheme with Efficient Cheater Identifying

Hua Shen, Daijie Sun, Lan Zhao, Mingwu Zhang

A Fair (t, n)-threshold Secret Sharing Scheme with Efficient Cheater Identifying *

Hua Shen[1], Daijie Sun[1], Lan Zhao[1], and Mingwu Zhang[1,2,*]

[1] School of Computer Science, Hubei University of Technology
[2] Hubei Key Laboratory of Intelligent Geo-Information Processing, China University of Geosciences
Corresponding: Mingwu Zhang mzhang@hbut.edu.cn

Abstract. The fairness of secret sharing guarantees that, if either participant obtains the secret, other participants obtain too. The fairness can be threatened by cheaters who was hidden in the participants. To efficiently and accurately identify cheaters with guaranteeing fairness, this paper proposes a fair (t, n)-threshold secret sharing scheme with an efficient cheater identifying ability. The scheme consists of three protocols which correspond to the secret distribution phase, secret reconstruction phase, and cheater identification phase respectively. The scheme's secret distribution strategy enables the secret reconstruction protocol to detect the occurrence of cheating and trigger the execution of the cheater identification protocol to accurately locate cheaters. Moreover, we prove that the scheme is fair and secure, and show that the cheater identification algorithm has higher efficiency by comparing with other schemes.

Keywords: Secret sharing · Cheater identification · Fairness · Attack model.

1 Introduction

In the reconstruction phase of a (t, n)-threshold secret sharing scheme, dishonest participants can reconstruct the real secret because of receiving the valid secret shares. It's unfair for honest participants that they gain the wrong secret because of accepting the invalid secret shares[1]. To address this issue, many researchers have come up with their solutions. Laih and Lee[2] proposed a v-fair (t, n)-threshold secret sharing scheme, in which all participants do not have to show their secret shares simultaneously to recover the secret with the same probability, even if there are $v(< t/2)$ dishonest participants. [3] and [5] further improved Laih scheme[2]. In 2003, Tian[6] utilized the consistency of secret shares to detect attackers, and constructed a fair (t, n)-threshold scheme with the help of the schemes of Tompa and Woll[7]. Harn and Lin[8] also used the

* Supported by the National Natural Science Foundation of China (61702168, 61672010), Hubei Provincial Department of Education Key Project (D20181402), the open research project of Hubei Key Laboratory of Intelligent Geo-Information Processing (KLIGIP-2017A11).

consistency of secret share to design an algorithm to detect cheating behavior and identify cheaters. In 2014, Harn[9] pointed out that the research on asynchronous attack in scheme [6] was incorrect. In 2015, Harn[10] proposed a scheme that can resist asynchronous attacks of external attackers and internal attackers. In 2016, Liu[11] presented a Linear (t, n)-threshold secret sharing scheme in which there is only one honest participant can detect cheaters. Lin[12] constructed a secret sharing scheme which focuses on preventing cheating behavior rather than cheating detection. With the same purpose, in 2018 Liu[13] proposed a (t, n)-threshold secret image sharing scheme. In order to improve the efficiency of the verifiable secret sharing scheme, Mashhadi[14] and Cafaro[15] put forward their schemes respectively, but none of their schemes are unconditionally safe. In 2018, Liu and Yang[17] proposed a cheating identifiable secret sharing scheme by using the symmetric bivariate polynomial, but the scheme does not achieve fairness requirement of secret sharing.

In order to not only identify deception behavior but also efficiently and accurately locate cheaters, this paper propose a fair (t, n)-threshold secret sharing scheme which realizes the fairness through *Distribution protocol* and *Reconstruction protocol*, and achieves the efficiently cheaters identification through *Cheater identification protocol*. Moreover, the presented scheme is unconditional security because of not depending on any security assumptions, and is fair and secure based on four attack models.

The remainder of this paper is organized as follows. We introduce some preliminaries, in Section 2. In Section 3, we present a fair (t, n)-threshold secret sharing scheme with an efficient cheater identifying algorithm. In Section 4, we describe the fairness and security of the proposed scheme, followed by the performance analysis in Section 5. Finally, we conclude this paper.

2 Preliminaries

In this section, we briefly recall some fundamental backgrounds which are used in our scheme and then introduce the attack models of our scheme.

2.1 Shamir's (t, n)-secret sharing scheme

Shamir's (t, n)-threshold secret sharing scheme [16] is based on Lagrange interpolating polynomial, in which there are n participants $\mathcal{P}$=$\{P_1, \cdots, P_n\}$, and a mutually trusted dealer $\mathcal{D}$. The scheme consist of two algorithms:

- *Distribution algorithm*: The dealer $\mathcal{D}$ first randomly generates a polynomial: $f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{t-1} x^{t-1}$, in which the secret is s=a_0 and all the other coefficients $a_1, \cdots, a_{t-1}$ are chosen from a finite field $\mathbb{F}$, and then $\mathcal{D}$ computes the secret share $s_i = f(i)$ and sends it to the participant P_i, where $i = 1, 2, \cdots, n$.
- *Reconstruction algorithm*: In the reconstruction phase, at least t participants submit their secret shares, the secret s can be reconstructed by calculating the Lagrangian interpolation polynomial through these secret shares.

2.2 Definitions of Consistency and Fairness

Definition 1. *(*consistency*): In a (t, n)-threshold secret sharing scheme, suppose there are m $(m \geq t)$ participants reconstruct the secret. The m shares are consistent if any t shares in them can reconstruct the same secret.*

To check whether m shares are consistent or not, we only need to sequentially execute three steps as follows [6]. (i) Reconstruct a polynomial $g(x)$ using any t shares of the m secret shares. (ii) Check whether the degree of $g(x)$ is $t-1$ or not. (iii) Check whether the remainder $m-t$ secret shares satisfy $g(x)$ or not. If (ii) and (iii) are satisfied, we can conclude that the m shares are consistent.

Definition 2. *(*fairness*): A (t, n)-threshold secret sharing scheme is fair if it can guarantees that either each participant who takes part in reconstructing the secret obtains the same secret, or knows nothing about the mystery.*

Not difficult to find if the m secret shares are consistent, the corresponding scheme is fair.

2.3 Attack models

The aim of our scheme is holding the fairness and secure under the following four attack models. :

- *Non-cooperative attack with synchronisation (NCAS)*: All participants submit the secret shares simultaneously, and that there are no cooperations between dishonest parties.
- *Non-cooperative attack with asynchronisation (NCAAS)*: All participants present secret shares successfully and that there are no cooperations between dishonest parties.
- *Collusion attack with synchronisation (CAS)*: The malicious parties modify their secret shares to deceive the honest parties. We assume that all participants submit their secret shares at the same time. Under this assumption, only when the number of malicious parties is more extensive than or equal to the threshold value t, can the malicious parties successfully deceive the honest parties.
- *Collusion attack with asynchronisation (CAAS)*: The dishonest parties collaboratively modify their secret shares to deceive the honest parties. The participants asynchronously release their secret shares. The best option for dishonest participants is to submit their accordingly modified secret shares after all honest participants have submitted their secret shares.

3 Our schemes

In this section, we introduce our fair (t, n)-threshold secret sharing scheme which consists of three algorithms: distribution algorithm, reconstruction algorithm, and cheater identification algorithm.

3.1 Distribution

The dealer $\mathcal{D}$ wants to share a secret s among n participants $\mathcal{P} = \{P_1, \cdots, P_n\}$. $\mathcal{D}$ first randomly constructs an identifier sequence $\{a_1, a_2, \cdots, a_v\}$ from $\mathbb{Z}_q$, and q is big prime integer. The sequence must satisfy: $a_1 > a_2 > \cdots > a_{l-1} > a_{l+1} > \cdots > a_v > a_l$ where $l \in [1, v]$ is randomly determined by $\mathcal{D}$, and a_l is related to finally recover s. And then, based on the sequence, $\mathcal{D}$ generates v random polynomials through which $\mathcal{D}$ calculates the secret share $s_i = (s_{i_1}, \cdots, s_{i_v})$ for the ith participant. The distribution protocol is shown as:

Distribution protocol
INPUT: the secret s, the parameter v.
OUTPUT: the secret shares $s_1, s_2, \cdots, s_n$.

1. Randomly pick an integer $l \in [1, v]$;
2. Generate $a_1 > a_2 > \cdots > a_{l-1} > a_{l+1} > \cdots > a_v > a_l$;
3. Construct v polynomials of $(t-1)$-degree, like as follows:
   $f_k(x) = a_k + a_{k,1}x + a_{k,2}x^2 + \cdots + a_{k,t-1}x^{t-1} \bmod \mathbb{Z}_q$,
   where $k = 1, \cdots, v$, and $a_{k,1}, \cdots, a_{k,t-1}$ are randomly picked from $\mathbb{Z}_q$;
4. Calculate d to satisfy: $s = a_l \cdot d$;
5. Generate the secret share of ith $(i = 1, \cdots, n)$ participant by computing
   $s_i = (s_{i_1}, s_{i_2}, \cdots, s_{i_v}) = (f_1(i), f_2(i), \cdots, f_v(i))$.

3.2 Reconstruction

Suppose that $m(\geq t)$ participants $\mathcal{R} = \{P_1, \cdots, P_m\}$ cooperate to reconstruct s. Denoted by $\mathcal{P}_{-i} = \mathcal{R}/P_i$. The reconstruction protocol is shown below:

Reconstruction protocol
INPUT: $m(m \geq t)$ secret shares $\{s_1, s_2, \cdots, s_m\}$.
OUTPUT: the set of cheaters $\mathcal{A}$ and the secret s.

1. 1th round: P_i sends s_{i_1} to $\mathcal{P}_{-i}$, and then performs *Receive_share*(k).
2. kth (k from 2 to v) round: If P_i receives all $(k-1)$th items of secret shares sent by $\mathcal{P}_{-i}$, then uses $\{s_{1_{k-1}}, s_{2_{k-1}}, \cdots, s_{m_{k-1}}\}$ to calculate a Lagrange interpolating polynomial $f_{k-1}(x)$. If $f_{k-1}(x)$ is $t-1$ degree, then all participants send the kth items of their secret shares and then perform *Receive_share*(k). Otherwise, all participants utilize the cheater identification protocol and obtain the set $\mathcal{A}$. If $|\mathcal{P}/\mathcal{A}| \geq t$, then all participants $\in \mathcal{P}/\mathcal{A}$ send the kth items of their secret shares and performs *Receive_share*(k); otherwise, protocol is terminated.

Procedure *Receive_share*(k): Receiving the kth item of secret share

1. When P_i has received all kth items of secret shares sent by $\mathcal{P}_{-i}$, he utilizes all these items $\{s_{1_k}, s_{2_k}, \cdots, s_{m_k}\}$ to compute the Lagrange interpolating polynomial $f_k(x)$. If the degree of $f_k(x)$ is $t-1$, then P_i performs step (b). Otherwise, all participants invoke the cheater identification protocol to identify the cheaters, and put them into the cheaters' set $\mathcal{A}$. If $|\mathcal{P}/\mathcal{A}| \geq t$, then the protocol turns to step b; otherwise, it is terminated.
2. Calculate the identifier by using the secret share sent by all participants in $\mathcal{P}/\mathcal{A}$, $a_k = f_k(0)$. If $a_k > a_{k-1}$, then $\mathcal{D}$ sends d to all participants in $\mathcal{P}/\mathcal{A}$, and these participants can calculate $s = a_{k-1} \cdot d$, and then the protocol is terminated; otherwise, all participants in $\mathcal{P}/\mathcal{A}$ send the $(k+1)$-th items of secret shares.

3.3 Cheater identification

To identify the participants who input fake shares, We use a mark vector represents a kind of choice of selecting t participants from m participants, so there are $u = \binom{m}{t}$ mark vectors, denoted by $C_1, \cdots, C_u$. Each mark vector consists of m items, of which the value is 0 or 1, denoted by $C_j = (c_{j_1}, \cdots, c_{j_m}), j = 1, 2, \cdots, u$. Therefore, each mark vector includes t $1's$ and $m-t$ $0's$.

Cheater identification protocol
INPUT: m, t, k, $\{s_{1_k}, s_{2_k}, \cdots, s_{m_k}\}$.
OUTPUT: the set of cheaters $\mathcal{A}$.
All the m reconstruction participants do:

1. Generate u mark vectors $C_1, C_2, \cdots, C_u$.
2. Based on the mark vector C_j $(j = 1, 2, \cdots, u)$ (that is, based on $S'_k = \{s_{i'_k} | c_{j_{i'}} = 1\}$ $(i' = 1, 2, \cdots, m)$), each participant yields the Lagrange interpolating polynomial $f_k^j(x)$. Therefor, each participant can obtain $f_k^1(x), f_k^2(x), \cdots, f_k^u(x)$.
3. According to $f_k^1(x), f_k^2(x), \cdots, f_k^u(x)$, each participant can obtain u values of the identifier a_k, that is $a_k^1 = f_k^1(0), a_k^2 = f_k^2(0), \cdots, a_k^u = f_k^u(0)$. These values might different or the same. Find the most frequently occurring value in them, the value is the value of a_k.
4. And then extract the corresponding mark vectors from $\{C_1, \cdots, C_u\}$. Use $\mathcal{C}^{succ}$ denote the set of these corresponding mark vectors.
5. Perform Logic Or operation on $\mathcal{C}^{succ}$, the participants corresponding to the items whose values are 0 in the result mark vector are cheaters, and then add these participants to $\mathcal{A}$, finally return $\mathcal{A}$.

4 Security and correctness analysis

Theorem 1. *In our proposed scheme, the probability that each participant successfully guesses the secret s is $1/v$.*

Proof. The dealer $\mathcal{D}$ hides the secret s into the polynomial $f_l(x)$, where $l \in [1, v]$ is randomly chosen by $\mathcal{D}$, therefore, the participants successfully guess the value of l with the probability $1/v$.

$\mathcal{P} = \{P_1, \cdots, P_m\}$ $(t \leq m \leq n)$ denotes all participants who take part in the secret reconstruction phase, $\mathcal{P}_I = \{P_{i_1}, \cdots, P_{i_\alpha}\} \subseteq \mathcal{P}$ denotes the set of cheaters in $\mathcal{P}$, $\mathcal{P}_{-I} = \mathcal{P}/\mathcal{P}_I$ denotes the set of honest participants in $\mathcal{P}$.

Theorem 2. *Under non-cooperative attack with synchronisation (NCAS), when $m > t$, our scheme is secure and fair.*

Proof. NCAS assumes that all participants present shares at the same time and that there is no cooperation between cheaters. Suppose that in the k-round reconstruction stage, the cheaters in $\mathcal{P}_I$ send invalid secret shares. Since there is no cooperation between the cheaters, their invalid secret shares can only be random numbers in $\mathbb{Z}_q$. When $m > t$, these secret shares could not pass the consistency test, and the attack is immediately detected. In order to restore s, the attackers in $\mathcal{P}_I$ need to guess in which polynomial s is hidden and which honest participants are involved. According to **Theorem 1**, the maximum successful probability is $1/v$. If v is large enough, the probability can be ignored. Therefore, under non-cooperative attack, when $m > t$, our scheme is secure and fair.

Theorem 3. *Under non-cooperative attack with asynchronisation (NCAAS), when $\{(m-\alpha < t-1) \cap (m > t)\} \cup \{m-\alpha \geq t+1\}$, our scheme is secure and fair.*

Proof. NCAAS assumes that all participants present shared shares successively without cooperation between attackers. A cheater' ideal attack is to show the secret share at the end, because he can obtain all the shares before others. When $m - \alpha \geq t + 1$, that is, there are no less than $t + 1$ honest participants, who show the secret shares firstly. Therefore, the attackers can reconstruct the correct polynomial $f_k(x)$ (suppose in k-round) based on t real secret shares, and then obtain the a_k. The attackers can show the real secret shares in the first l rounds and show a fake secret share in $(l + 1)$th round. However, the fake secret share cannot pass the consistency test, and the attack behavior can be detected, which trigger the execution of cheater identification algorithm. The right identifier a_{l+1} can be reconstructed based on the $m - \alpha$ real secret shares, because $\binom{m-\alpha}{t} > 1$, the a_{l+1} is correct identifier which can be used to identify the attackers, therefore, the attackers could not gain d from the dealer to obtain s. When $m - \alpha < t + 1$, for an attacker, even if he finally shows his secret share, he can only obtain at most $t - 1$ real secret shares, so he can not reconstruct any $t - 1$-degree polynomial, as a result he can not recover s. In order to detect attacks, m should greater than t. In conclusion, when $\{(m - \alpha < t - 1) \cap (m > t)\} \cup \{m - \alpha \geq t + 1\}$, the proposed scheme is secure and fair.

Theorem 4. *Under collusion attack with synchronisation (CAS), when* $\{(\alpha < t) \cap (m > t)\} \cup \{(\alpha \geq t) \cap (m - \alpha > \alpha + t - 1)\}$*, our scheme is secure and fair.*

Proof. CAS assumes that all participants present secret shares simultaneously and that multiple attackers conspire to attack the scheme. Suppose there are α cheaters in k-round. (i) When $\alpha \geq t$, if the number of honest participants is less than t, that is, $m - \alpha < t$, then cheaters can cooperate to forge a set of invalid secret shares which can pass consistency detection. The specific process is as follows: Cheaters first use their secret shares to recover an interpolation polynomial, then utilize the polynomial to calculate the secret shares held by other honest participants, and then generate their false secret shares based on the secret shares of other honest participants. For example, $\alpha = t$, $m - \alpha = t - 1$, $m = 2t - 1$, use $\{P_1, \cdots, P_{t-1}\}$ denote honest participants, use $\{P_t, P_{t+1}, \cdots, P_{2t-1}\}$ denote cheaters. Cheaters can use their true secret share $\{s_{t_k}, s_{t+1_k}, \cdots, s_{2t-1_k}\}$ to calculate the interpolation polynomial $f_k(x)$, so they can show the true secret shares in the first l rounds, and in $(l+1)$th round, they can use $f_{l+1}(x)$ to obtain other honest participants' secret shares $\{s_{1_{l+1}}, \cdots, s_{t-1_{l+1}}\}$, and calculate another $(t-1)$-degree polynomial $f'_{l+1}(x)$ by using secret shares $\{s_{1_{l+1}}, s_{2_{l+1}}, \cdots, s_{t-1_{l+1}}\}$ and a random value $s'_{t_{l+1}}$. And then, cheaters use $f'_{l+1}(x)$ to calculate $t-1$ invalid secret shares $\{s'_{t_{l+1}}, s'_{t+1_{l+1}}, \cdots, s'_{2t-1_{l+1}}\}$. Finally, the secret shares shown by all participants as follows: $\{s_{1_{l+1}}, s_{2_{l+1}}, \cdots, s_{t-1_{l+1}}, s'_{t_{l+1}}, s'_{t+1_{l+1}}, \cdots, s'_{2t-1_{l+1}}\}$. These m secret shares can pass consistency detection when $m-\alpha \geq t$. The secret shares forged by the above method in $(l+1)$th round cannot pass consistency detection. By executing the identification algorithm, m real secret shares can used to reconstruct the correct identifier a_{l+1} at $\binom{m-\alpha}{t}$ times, while $t-1$ real secret shares and an invalid secret share can be utilized to reconstruct a wrong identifier a'_{l+1} at $\binom{\alpha+t-1}{t}$ times. Therefore, we have $\binom{m-\alpha}{t} > \binom{\alpha+t-1}{t}$. That is, $m-\alpha > \alpha+t-1$, under this condition, the invalid secret shares can be detected, and cheaters cannot obtain d from the dealer and recover s. But the honest participants can gain d and reconstruct s. (ii) If $\alpha < t$, these α cheaters can not use their real secret shares to forge the invalid secret shares that can pass the consistency detection. When $m > t$, this attack can not pass the consistency detection. If cheaters want to reconstruct s, they can only guess the value of l, the probability of successfully guessing is only $1/v$. From what has been discussed above, when $\{(\alpha < t) \cap (m > t)\} \cup \{(\alpha \geq t) \cap (m - \alpha > \alpha + t - 1)\}$, our scheme is secure and fair.

Theorem 5. *Under collusion attack with asynchronisation (CAAS), when* $m - \alpha > \alpha + t - 1$*, our scheme is secure and fair.*

Proof. CAAS assumes that all participants present secret shares successively and that multiple cheaters conspire to attack the scheme. For cheaters, the ideal mode of attack is to present the secret shares at the end, so that they can obtain the real secret shares presented by previous honest participants. When $m - \alpha \geq t$, there are not less than t honest participants, who first show the secret shares. Attackers

use $t-1$ real secret shares (according to the method of **Theorem 4**) to forge α invalid secret shares. Because $m-\alpha \geq t$, these invalid secret shares cannot pass consistency detection. By executing the identification algorithm, $m-\alpha$ real secret shares can be used to recover the correct identifier a_{l+1} $\binom{m-\alpha}{t}$ times, while $t-1$ real secret shares and an invalid secret share can be utilized to reconstruct a wrong identifier a'_{l+1} $\binom{\alpha+t-1}{t}$ times. Therefore, we have $\binom{m-\alpha}{t} > \binom{\alpha+t-1}{t}$. Concretely, under $m-\alpha > \alpha+t-1$, these invalid secret shares can be detected, and cheaters cannot gain d from the dealer and reconstruct s. But the honest participants can obtain d and recover s. Therefore, when $m-\alpha > \alpha+t-1$, the proposed scheme is secure and fair.

Theorem 6. *Under the conditions mentioned above, our cheater identification algorithm is correct.*

Proof. The key to prove the correctness of the cheater identification protocol is to prove the most frequently occurring value in $\{a_k^1 = f_k^1(0), \cdots, a_k^u = f_k^u(0)\}$ is the correct value of a_k. In the cheater identification protocol, interpolating polynomials are reconstructed only based on t secret shares, therefore, only when the t secret shares are real can the correct value of a_k be recovered. To guarantee the most frequently occurring value in $\{a_k^1 = f_k^1(0), \cdots, a_k^u = f_k^u(0)\}$ is the correct value of a_k, the following condition must be satisfied:

$$\binom{m-\alpha}{t} > \frac{1}{2}\binom{m}{t}.$$

We have,

$$\frac{(m-\alpha)!}{(m-\alpha-t)!t!} > \frac{1}{2}\cdot\frac{m!}{(m-t)!t!} = \frac{1}{2}\cdot\frac{(m-\alpha)!\alpha!}{(m-t)!t!}$$
$$\Rightarrow \frac{(m-\alpha)!}{(m-\alpha-t)!} > \frac{1}{2}\cdot\frac{(m-\alpha)!\alpha!}{(m-t)!} = \frac{1}{2}\cdot\frac{(m-\alpha)!}{(m-\alpha-t)!}$$

Since the inequality is always true, our cheater identification algorithm is correct.

5 Performance

The following two examples are given to respectively calculate the maximum number of attackers α_{max} under the four types of attack models. Taking (7, n) threshold scheme as an example, assuming $m=9$ and $m=11$, where m is the number of participants who take part in the secret reconstruction phase. Under NCAS, according to **Theorem 2**, when $m > t$ our scheme is secure and fair, so $\alpha_{max} = 9$. Similarly, under NCAAS, according to **Theorem 3**, when $\{(m-\alpha < t-1) \cap (m > t)\} \cup \{m-\alpha \geq t+1\}$ our scheme is secure and fair, which means $\alpha_{max} = 9$. From the analysis of **Theorem 4**, Under CAS, when

$\{(\alpha < t) \cap (m > t)\} \cup \{(\alpha \geq t) \cap (m - \alpha > \alpha + t - 1)\}$ the proposed scheme is safe and fair, so $\alpha_{max} = 6$. According to the analysis of **Theorem 5**, Under CAAS, our scheme can defend at most 1 cheaters, as shown Table 1. Based on a similar analysis process, when $m = 11$, the values of α_{max} are shown as in Table 1.

Table 1. (7,n)-threshold scheme, $m = 9$ or $m = 11$

Attack model	Conditions	α_{max} $(m = 9)$	α_{max} $(m = 11)$
NCAS	$m > t$	9	11
NCAAS	$\{(m - \alpha < t - 1) \cap (m > t)\} \cup \{m - \alpha \geq t + 1\}$	9	11
CAS	$\{(\alpha < t) \cap (m > t)\} \cup \{(\alpha \geq t) \cap (m - \alpha > \alpha + t - 1)\}$	6	6
CAAS	$m - \alpha > \alpha + t - 1$	1	2

Different from Tian and Peng's[18] scheme, our scheme does not depend on any security assumptions, it is a unconditional security scheme. Compared to Tian's[6], Harn's[9], Harn-Lin's[8] and Liu-Yang's[17] secret sharing schemes, our scheme achieves fairness but they do not have, as shown in Table 2.

Table 2. Security comparison

Scheme	Tian[6]	Harn-Lin[8]	Liu-Yang[17]	Tian-Peng[18]	ours
Security assumption	no	no	no	ECDLP	no
Fairness	no	no	no	no	yes

In [8], Harn and Lin proposed a secret sharing scheme that can identify cheaters. In their scheme, the correct secret needs to be confirmed and the secret share of each participant needs to be verified. In our scheme, we removed the process of validating each participant's secret share but achieves the same function of [8]. Therefore, our scheme has higher operating efficiency than [8].

6 Conclusion

In this paper, we study the cheater identification issue and the fairness problem in the reconstruction phase of secret sharing, and propose a fair (t, n) secret sharing scheme including a efficient cheater identification algorithm. By comparing with the existing verifiable secret sharing schemes, it can be found that our scheme achieves fairness. Compared with the fair secret sharing scheme, our cheater identification algorithm has a lower computational complexity. Moreover, we analyzed the security of our proposed scheme under four different attack models.

References

1. Zhang M., Zhang Y., Jiang Y., Shen J.: Obfuscating EVES algorithm and its application in fair electronic transactions in public cloud systems. IEEE System Journal **13**(2), 1478–1486 (2019)
2. Laih C. S., Lee Y. C.: V-fairness (t, n) secret sharing scheme. IEE Proceedings-Computers and Digital Techniques **144**(4), 245–248 (1997)
3. Lee Y. C.: A simple (v, t, n)-fairness secret sharing scheme with one shadow for each participant. In: International Conference on Web Information Systems and Mining, pp. 384–389. Springer, Berlin, Heidelberg (2011)
4. Li X., Zhu Y., Wang J., Liu Z., Liu Y., Zhang M.: On the Soundness and Security of Privacy-Preserving SVM for Outsourcing Data Classification. IEEE Transactions on Dependable and Secure Computing **15**(5), 906–912 (2018)
5. Yang J. H., Chang C. C., Wang C. H.: An efficient v-fairness (t, n) threshold secret sharing scheme. In: 2011 Fifth International Conference on Genetic and Evolutionary Computing, pp. 180–183. IEEE (2011)
6. Tian Y., Ma J., Peng C., Jiang Q.: Fair (t, n) threshold secret sharing scheme. IET Information Security **7**(2), 106–112 (2013)
7. Tompa M., Woll H.: How to share a secret with cheaters. Journal of Cryptology **1**(3), 133–138 (1989)
8. Harn L., Lin C.: Detection and identification of cheaters in (t, n) secret sharing scheme. Designs, Codes and Cryptography **52**(1), 15–24 (2009)
9. Harn L.: Comments on'fair (t, n) threshold secret sharing scheme. IET Information Security **8**(6), 303–304 (2014)
10. Harn L., Lin C., Li Y.: Fair secret reconstruction in (t, n) secret sharing. Journal of Information Security and Applications **23**, 1–7 (2015)
11. Liu Y.: Linear (k, n) secret sharing scheme with cheating detection. Security and Communication Networks **9**(13), 2115–2121 (2016)
12. Lin P.: Distributed secret sharing approach with cheater prevention based on qr code. IEEE Transactions on Industrial Informatics **12**(1), 384–392 (2016)
13. Liu Y., Sun Q, Yang C.: (k, n) secret image sharing scheme capable of cheating detection. EURASIP Journal on Wireless Communications and Networking **1**, 72 (2018)
14. Mashhadi S., Dehkordi M. H., Kiamari N.: Provably secure verifiable multi-stage secret sharing scheme based on monotone span program. IET Information Security **11**(6), 326–331 (2017)
15. Cafaro M., Pelle P.: Space-efficient verifiable secret sharing using polynomial interpolation. IEEE Transactions on Cloud Computing **6**(2), 453–463 (2018)
16. Shamir A.: How to share a secret. Communications of the ACM **22**(11), 612–613 (1979)
17. Liu Y., Yang C., Wang Y., Zhu L., Ji W.: Cheating identifiable secret sharing scheme using symmetric bivariate polynomial. Information Sciences **453**, 21–29 (2018)
18. Tian Y., Peng C., Zhang R., Chen Y.: A practical publicly verifiable secret sharing scheme based on bilinear pairing. In: International Conference on Anti-counterfeiting, pp. 71–75. IEEE (2008)