Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 8606589, 9 pages
https://doi.org/10.1155/2022/8606589

Research Article

Binary Symmetric Polynomial-Based Protected Fair Secret Sharing and Secure Communication over Satellite Networks

Chao Guo,[1,2] Chenglei Pan,[3] Guangyu Hu,[3] Dingbang Xie,[4] Peiliang Zuo,[1] and Yanyan Han[1]

[1]*Department of Electronics and Communication Engineering, Beijing Electronics Science and Technology Institute, Beijing 100070, China*
[2]*State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710126, China*
[3]*Department of Cyberspace Security, Beijing Electronic Science and Technology Institute, Beijing 100070, China*
[4]*School of Communication Engineering, Xidian University, Xi'an 710126, China*

Correspondence should be addressed to Yanyan Han; hyy@besti.edu.cn

Received 19 February 2022; Revised 30 April 2022; Accepted 24 June 2022; Published 9 August 2022

Academic Editor: A.H. Alamoodi

The rapid establishment of the low Earth orbit (LEO) satellite network in orbit has promoted the development of satellite communication technology. However, with the reduction of access conditions of satellite networks, the problems of data protection and secure communication have attracted extensive attention. A secret sharing scheme is a cryptographic technology that can disperse risks and tolerate intrusion by dividing and storing secrets. Using secret sharing technology in satellite communication can realize information security and data confidentiality. However, if there are cheaters among the participants, existing secret sharing schemes cannot prevent cheaters from sharing secrets exclusively, even if they can detect attacks. For this reason, this paper proposes a satellite based on binary symmetric polynomials protected fair secret sharing and secure communication scheme. In satellite secret refactoring, this scheme can produce a shared session key between two participants, no other key agreement processes, and reduce the scheme in the shared secret and the actual communication satellite application complexity. Users use the session key to encrypt communication to improve security and resist external attacks. The safety and fairness of the scheme are proved against the four attack models. Compared with the existing schemes, the scheme has a lower cost of deception identification on the premise of satisfying security and fairness. This scheme does not require any cryptographic assumptions and is unconditionally secure.

1. Introduction

A satellite network is a unified, organic system composed of various types of satellites in different orbits by maximizing the utilization efficiency of space information resources. It has the characteristics of comprehensive coverage, flexible networking, good transmission effect, and functional diversity, so it is often used in meteorology, scientific research, military, and environmental fields. However, the satellite network has a tremendous negative impact due to satellite node exposure, open channel, complex space environment, highly dynamic network topology, and high link error rate. Limited space-borne resources affect computing power, which will pose a significant threat to the security of satellite networks. Although Vaseghi et al. proposed a chaotic satellite image encryption algorithm in 2021, there are security proof problems [1]. Therefore, it is necessary to design an unconditional security-protected fair secret sharing scheme in the satellite network.

In 1979, Shamir and Blakley proposed a secret sharing scheme based on Lagrange interpolation polynomials and mapping geometry, respectively [2, 3]. The traditional (t, n) secret sharing scheme consists of secret distribution and secret reconstruction: (1) The distributor divides the shared secret into multiple secret shares by calculation and distributes them to participants, respectively. (2) Any participant set greater than or equal to the threshold can present the secret share to reconstruct the shared secret. The proposal of secret sharing provides a new idea for key management, but the traditional secret sharing scheme also has some security problems. In the process of reconstruction, the reconstructors are not completely honest. If the insider attacker shows the false child secret, then the honest participant restores the false secret, and the insider attacker can enjoy the secret to maximize the benefits. If the external attacker collects the subsecrets presented by the honest participants, it can also forge its identity and obtain the same attack effect as the internal attacker. The above spoofing attack raises the fairness issue of secret refactoring: (1) when there are internal or external attackers, all honest reconstructors can recover shared secrets, but attackers cannot reconstruct true shared secrets. (2) When there is no attacker, all refactorers can reconstruct true shared secrets.

One of the most common attacks on the satellite network is the information forgery attack. The attacker forges the illegally stolen data and sends it back to the uplink. The ground cannot distinguish whether the data is from the legitimate node, resulting in the error of the whole data communication. The problem of honest refactorers recovering false secrets arises in secret refactorings. Similarly, satellite communication is broadcast chiefly over a wide range, so if encryption protection technology is not adopted, it can easily lead to data leakage.

Because of the above cheating problems, Rabin and Ben introduced the validation vector to check the correctness of participants' secret shares and detect and identify cheaters [4]. In 1995, Carpentieri proposed a scheme based on the characteristics of reference [4] that reduced the additional verification vectors required by participants [5]. In 2009, Harn and Lin constructed a subsecret consistency deceiver detection and recognition algorithm and proved the scheme's feasibility under three attack models [6]. In 2011, Ghodosi pointed out that the deception detection and recognition algorithm in reference [6] was invalid under its limitations; after he improved the scheme conditions, the scheme with a medium or above the number of participants had high computational complexity for deception recognition [7]. In 2018, Liu et al. constructed two deception detection and recognition algorithms based on binary polynomials and proposed a scheme for nonreconstructors to participate in detection and recognition [8]. The spoofing detection and identification scheme will terminate the protocol immediately when spoofing is detected, which does not apply to the general situation. Secondly, although the cheater is detected and identified, it cannot be prevented from enjoying the shared secret exclusively, which does not meet the fairness of secret reconstruction. Tompa and Woll first proposed the fair secret sharing scheme in 1988 and hid the shared secret in a secret reconstruction sequence, and all participants did not know the location of the real secrets. In the synchronous reconstruction environment, the attacker can successfully attack only when the probability is $1/k$, and the attacker correctly guesses the shared secret reconstruction location [9]. Therefore, this scheme is fair in a synchronous environment. In an asynchronous environment, an attacker can launch an attack and share the secret as long as the child's secret is presented last. In 1995, Lin and Harn used the scheme in reference [4] to verify subsecrets. In addition, the secret reconstruction sequence $\{s_1, \cdots, s_j, s_{j+1}, \cdots, s_k\}$ is constructed, in which $s_j = s, s_{j+1} = s', s'$, participants restore the secret to $s_{j+1} = s'$, the correct secret sharing is the previous s_j, and the scheme meets the fairness in the asynchronous environment [10]. In 2013, Tian et al. used secret consistency and secret reconstruction sequences to construct a fair secret sharing scheme and proved the fairness of the scheme under noncollusive attacks, asynchronous and synchronous collusive attacks [11]. In 2014, Harn pointed out that reference [11] was neither safe nor fair in an asynchronous environment [12]. In 2015, Harn et al. constructed the secret reconstruction sequence and adopted the algorithm in reference [13] to share and reconstruct each secret sequence bit [14]. The scheme was fair and safe in an asynchronous environment. In 2016, Gu et al. proposed a fair secret sharing scheme based on binary symmetric polynomials to provide secure channels between participants. Still, discrete logarithms and hash functions are required to ensure security [15]. In 2017, Zhang et al. constructed a fair secret sharing scheme with absolute security by combining the deception detection and recognition algorithm and secret reconstruction in reference [6] and proved the fairness and security of the scheme under four attack models [16]. In 2019, Yang and Xing constructed a fair secret sharing scheme based on binary asymmetric polynomials and proved the fairness and security of the scheme under four standard attack models [17]. In 2019, Li et al. proposed an unconditional secret sharing scheme [18]. In 2020, Sun improved the recognition algorithm of reference [6] and proposed a fair secret sharing scheme with absolute security [19]. According to the research in reference [7], the security restriction conditions under the three attack models in references [16, 19] are all wrong. Liu et al. proposed a blockchain-based anonymous authentication scheme for air-ground integrated networks, which increased the consumption of satellite resources [20].

Therefore, according to the above research progress, in order to adapt to the characteristics of limited satellite resources and narrow bandwidth, combined with the characteristics of low orbit satellite network with wide coverage, low propagation delay, and small transmission loss, this paper proposes an unconditionally secure protected fair secret sharing scheme based on binary symmetric polynomials. Combined with the IoT architecture of low orbit satellites proposed by Ding et al. [21], this scheme can effectively solve the problems of secret distribution and mutual communication in satellite networks [6, 7]. The interplanetary link is formed by multiple low-orbit satellites, and the ground control center or mid-orbit satellites serve as the key distribution center. The users are all kinds of network users who need to provide services in the satellite network. The scheme has a low cost of deception detection and identification. It satisfies fairness and security under four standard attack models, which solves a series of security problems in a satellite network, such as intercepting data transmission by attackers, data leakage, and data tampering.

2. Related Work

2.1. Harn Spoofing Detection Algorithm. Harn and Lin proposed a deception detection algorithm compatible with

Shamir's secret sharing [2, 6]. The algorithm is briefly described as follows.

t represents share, n represents the total number, s stands for secret share, and J represents the number of interpolation points.

Input: $t, n, J = \{i_1, \cdots, i_j\}, s_{i_1}, s_{i_2}, \cdots, s_{i_j}$,where j interpolation points $(i_1, s_{i_1}), \cdots, (i_j, s_{i_j})$ are used to calculate the interpolation polynomial $f(x)$, denoting the order of $f(x)$ as d. If $d = t - 1$, then secret $s = f(0)$.

Output: no cheater, and the secret is s. There are cheaters.

If the participant set is J and the attacker set is $GF(p)$, reference [6] points out that deception detection will always succeed when $(J - C) > (t - 1)$.

2.2. Carpentieri Deception Recognition Algorithm. The scheme in this paper adopts the deceiver recognition algorithm proposed by Carpentieri, which is briefly described as follows [5].

q is a large prime number, $q > n$, and $\mathrm{GF}(q)$ are finite fields, and the secret s is selected on $\mathrm{GF}(q)$.

2.2.1. Secret Distribution. The distributor selects a $k(k \leq n)$ -dimensional vector $d_i \equiv (d_{i,0}, \cdots, d_{i,k-1})$ on $\mathrm{GF}(q)$ for each participant $P_i(i = 1,\cdots,n)$ as its secret share, the distributor randomly selects a nonnull different value $\alpha_1, \cdots, \alpha_n$ on GF (q), a$_i$ is the coefficient of the unknown x, $f(x) = s + a_1x + a_2x^2 + \cdots + a_{k-1}x^{k-1}$, $d_{i,0} = f(\alpha_i)$, for $i = 1, \cdots, n$, the different participant $d_{i,1}, \cdots, d_{i,k-1}$ randomly selects on $\mathrm{GF}(q)$. For any participant P_j, the distributor randomly selects different nonnull values $g_{j,i}, i = 1, \cdots, n$ on $\mathrm{GF}(q)$, calculates $b_{j,i} = g_{j,i} d_{i,0} + \alpha_j d_{i,1} + \cdots + \alpha_j^{k-1} d_{i,k-1}$, and distributes numerical pairs $(g_{j,i}, b_{j,i}), i = 1, \cdots, n, i \neq j$ to each participant P_j.

2.2.2. Deception Identification. After participants P_i show their secret share d_i, any participants P_j can authenticate d_i through an equation $g_{j,i}y_0 + \alpha_j y_1 + \cdots + \alpha_j^{k-1} y_{k-1} = b_{j,i}$. If d_i is the solution vector of the equation, then P_i is the honest participant, otherwise P_i is identified as a cheater.

3. Solution Overview

As shown in Figure 1, this scheme is divided into two scenarios. Solid lines represent communication between users, while dotted lines represent sharing secrets between users. When two users communicate with each other, they cannot communicate with each other directly due to the complex and changeable environment, such as desert, gobi, and sea. First, the ground control center will randomly send the secret share of the unique IN for the participants to the middle Earth orbit (MEO), and then, the MEO will transmit it to the LEO through the intersatellite link. Because the low orbit satellite has the characteristics of wide coverage and good transmission effect, the LEO will send the secret share to two users. Under the condition of ensuring the reliability of each other, the users can generate the session key between each other according to the above scheme. Therefore, when users communicate, they can encrypt and decrypt through the session key; when particular users have a secret to share with ordinary users, a particular user sends a request to the ground control center, the ground control center will randomly to send the secret share of the unique IN for the participants to the MEO, and then, the MEO will transmit it to the LEO through the intersatellite link, and the LEO will send the secret share to ordinary users. After that, secret reconstruction can be started between users. When all cheaters are excluded, the ground control center responds to the special user. The particular user can achieve the purpose of secret sharing, which significantly improves the security of the session and reduces the time of generating the session key.

4. The Project Design

4.1. Scheme Description. The scheme in this section adopts the deceiver recognition algorithm proposed by Carpentieri, which is briefly described as follows [5]. The subground control center D and the set of participants $\{P_1, \cdots, P_n\}$ are defined, and the finite domain of order p is constructed, where $p(p > n)$ is a large prime number, and the secret s is selected on $\mathrm{GF}(p)$.

4.1.1. Secret Distribution. D constructs $k - 1$ degree polynomial $f(x) = s + a_1x + a_2x^2 + \cdots + a_{k-1}x^{k-1} \bmod p$, where the unknown coefficient $a_1, a_2, \cdots, a_k$ is uniformly randomly selected on $\mathrm{GF}(p)$, different values $\alpha_1, \alpha_2, \cdots, \alpha_n$ are randomly selected on $\mathrm{GF}(p) \setminus \{0\}$ and disclosed, and the $k(k \leq n)$-dimensional vector $\mathbf{d_i} \equiv (d_{i,0}, \cdots, d_{i,k-1})$ is generated for each participant $P_i(i = 1, 2, \cdots, n)$ as it is secret share and distributed, where $d_{i,0} = f(\alpha_i) \bmod p$, $d_{i,1}, \cdots, d_{i,k-1}$ is uniformly randomly selected on $\mathrm{GF}(p)$. Participants $P_j(j = 1, 2, \cdots, n)$, D randomly select different values $g_{j,i}(i = 1, 2, \cdots, n, i \neq j)$ on $\mathrm{GF}(p) \setminus \{0\}$, form $n - 1$ pairs of values $(g_{j,i}, b_{j,i})$, and distribute them to P_j, calculating $b_{j,i} = g_{j,i}d_{i,0} + \alpha_j d_{i,1} + \cdots + \alpha_j^{k-1} d_{i,k-1} \bmod p$.

4.1.2. Deception Identification. After the participant P_i receives and presents his secret share $\mathbf{d_i}$ via the satellite network, any participant $P_j(j = 1, 2, \cdots, n, i \neq j)$ can verify $\mathbf{d_i}$ through $b_{j,i} = g_{j,i}y_0 + \alpha_j y_1 + \cdots + \alpha_j^{k-1} y_{k-1} \bmod p$, where y_0, $y_1, \cdots, y_{k-1}$ is unknown. If $\mathbf{d_i}$ is the solution vector of the equation, P_i is identified as an honest participant, otherwise as a cheater. The scheme in this section includes two parts: secret satellite distribution and secret satellite reconstruction. The detailed process is given below.

4.2. Secret Distribution. Assume that the ground control center is D, the threshold value of the scheme is t, and there are n participants $\{P_1, P_2, \cdots, P_n\}$. D constructs the finite domain $\mathrm{GF}(p)$ of order p, and $p(p > n)$ is a large prime number. The select secret s on $\mathrm{GF}(p)$ sets the security parameter v and executes the following algorithm:

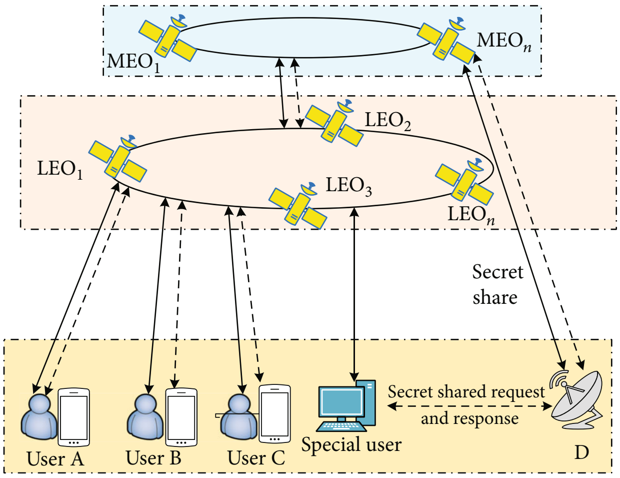

Figure 1: Overview of secret sharing scheme for satellite networks.

(i) Step 1: select random integers $l(1 \le l \le v)$, $a_i(i = 1, 2, \cdots, v, i \ne l)$ is the sequence bit value, and randomly generate a set of sequences:

$$a_1 > a_2 > \cdots > a_{l-1} > a_l < a_{l+1} \cdots < a_v \quad (1)$$

(i) Step2: take $a_i(i = 1, 2, \cdots, v, i \ne l)$ as a constant term to generate a univariate polynomial:

$$f_i(x) = a_i + a_{i,1}x + \cdots + a_{i,t-1}x^{t-1} \bmod p \quad (2)$$

For the sequence l position, a_l is used as a constant term to generate a bivariate symmetric polynomial of degree $t - 1$:

$$F(x, y) = a_l + c_{1,0}x + c_{0,1}y + c_{1,1}xy + \cdots + c_{t-1,t-1}x^{t-1}y^{t-1} \bmod p, \quad (3)$$

where the unknown coefficient $c_{i,j} = c_{j,i}(\forall i, j \in [0, t - 1])$, $F(0, 0) = a_l$

(i) Step 3: calculate what d satisfies $s = a_l \oplus d$.

(ii) Step 4: select $\mathrm{ID}_i((1 \le i \le n)), \mathrm{ID}_i \in GF(p) \setminus \{0\}$ as the identification information of each participant $P_i(1 \le i \le n)$ and make it public to ensure that any two participants meet $\mathrm{ID}_i \ne \mathrm{ID}_j(i \ne j)$. Compute $F_i(y) = F(\mathrm{ID}_i, y) \bmod p$ and distribute it to the actor P_i over a secure channel

(iii) Step 5: generate the secret share of the participant P_i: vector $\mathbf{s_{i,k}} \equiv (s_{i,k,0}, s_{i,k,1}, \cdots, s_{i,k,t-1}), 1 \le k \le v$ is t -dimensional:

(i) When $k = l, \mathbf{s_{i,l}} \equiv (s_{i,l,0} = F(\mathrm{ID}_i, 0) \bmod p, \cdots, s_{i,l,t-1})$

(ii) When $k \ne l, \mathbf{s_{i,k}} \equiv (s_{i,k,0} = f_k(ID_i) \bmod p, \cdots, s_{i,k,\mathrm{t}-1})$

The remaining $nk(t - 1)$ elements $s_{i,k,1}, \cdots, s_{i,k,t-1}(1 \le i \le n, 1 \le k \le v)$ are randomly selected on $\mathrm{GF}(p)$, and v vectors are distributed to participant P_i through the secure channel.

(i) Step 6: for sequence bit $k = 1, 2, \cdots, v$, select a nonzero value $g_{j,i,k}(i, j = 1, 2, \cdots, n, i \ne j)$ on a finite field $\mathrm{GF}(p)$ randomly for each participant P_i, $b_{j,i,k} = g_{j,i,k} s_{i,k,0} + \mathrm{ID}_j s_{i,k,1} + \cdots + \mathrm{ID}_j^{t-1} s_{i,k,t-1} \bmod p$ and distribute $(g_{j,i,k}, b_{j,i,k}), i = 1, \cdots, n, i \ne j$ to each participant P_j over a secure channel.

4.3. Secret Refactoring. Assuming the set of reconstructors $\mathbf{R} = \{P_1, \cdots, P_m\}(m \ge t)$, the reconstruction algorithm performs at most v rounds, denoted by $P_{-i} = \mathbf{R} \setminus P_i$. Participants P_i and P_j calculate, respectively $F(\mathrm{ID}_i, \mathrm{ID}_j) \bmod p$ through $F_i(y) \bmod p$ and $F_j(y) \bmod p$, which serves as the session key between ground users. After that, information exchange is carried out in symmetric encryption.

Case 1. Send round k secret quota. All refactorers P_i perform the following algorithms:

Step 1: if the algorithm takes $k = 1$ rounds, P_i send a secret share $s_{i,1}$ to $P_{\text{-}i}$.

Step 2: the algorithm execution cycle is round k. If P_i receives $m - 1$ secret shares of round $k - 1$ sent by $P_{\text{-}i}$, the algorithm perform step 3. Otherwise, the attacker set $\mathbf{C}$ is output, and the algorithm is terminated.

Step3: P_i calculates interpolation polynomial $f'_{k-1}(x)$ through the collected subsecret share $s_{1,k-1,1}, \cdots, s_{m,k-1,1}$. If the polynomial $f'_{k-1}(x)$ is $t - 1$, the secret share of the wheel k is sent; otherwise, a spoofing attack exists. P_i verifies the $m - 1$ subsecret share received by $(g_{i,j,k-1}, b_{i,j,k-1})$, if the verification is passed, P_i will vote for P_j; otherwise, no vote will be given. If P_j gets votes $T < 2/m$ and P_j is marked as a cheater, P_j is removed from the secret reconstruction, and the cheater set $\mathbf{C}$ is entered. Those who voted for P_j also enter the cheater set $\mathbf{C}$. If $|\mathbf{R} \setminus \mathbf{C}| \ge \mathrm{t}$, P_i sends the k wheel secret share; otherwise, the protocol terminates and outputs the deceiver set $\mathbf{C}$.

Case 2. Receive round k secret share. All refactorers P_i perform the following algorithms:

Step 1: if P_i receives all $m - 1$ secret shares of round k sent by $P_{\text{-}i}$, the algorithm calculates interpolation polynomial $f'_k(x)$ through $s_{1,k,1}, s_{2,k,1}, \cdots, s_{m,k,1}$. If the polynomial $f'_k(x)$ is of order $t - 1$, perform step 2. Otherwise, P_i verifies $m - 1$ subsecret shares received by $(g_{i,j,k}, b_{i,j,k})$, and P_i votes for P_j. Otherwise, P_i does not vote. If P_j gets the votes that satisfy $T < 2/m$, P_j is marked as a cheater, P_j is removed from the secret reconstruction, and the cheater set $\mathbf{C}$ is entered. Those who voted for P_j are also entered into the cheater set $\mathbf{C}$. If $|\mathbf{R} \setminus \mathbf{C}| \ge \mathrm{t}$, perform step 2, otherwise, the protocol terminates, and the spoofer set $\mathbf{C}$ is output.

Step 2: all the reconstructors in $\mathbf{R} \setminus \mathbf{C}$ calculating the sequence bits, $a_k = f'_k(0)$, if $a_{k-1} < a_k$ is satisfied, the reconstructors in $\mathbf{R} \setminus \mathbf{C}$ send a request to the ground control center D, and D sends d to the reconstructors in $\mathbf{R} \setminus \mathbf{C}$. After any reconstructor in $\mathbf{R} \setminus \mathbf{C}$ receives d, he reconstructs the secret through the equation $s = a_{k-1} \oplus d$, and the agreement is terminated; otherwise, the secret share of the round $k+1$ is sent.

5. Scheme Analysis

5.1. Security Model. Because the satellite fair secret sharing and secure communication scheme proposed in this section have protected characteristics, it is not necessary to consider any attack from external hostile users or satellites. The scheme is the same as the previous satellite's fair secret sharing and communication scheme. It is assumed that there is a secure channel between the ground control center and participants, so only security in secret reconstruction is considered. To better analyze the safety and fairness of the scheme, the scheme classifies internal hostile user or satellite attacks into the following four types of attacks.

Case 1. Noncooperative attack with synchronization (NCAS). When all refactorers participate in secret reconstruction, the secret share is synchronous. There is no collusion between internal hostile users or satellites, which means that the false secret share presented by the internal enemy can only be a random number from a finite field. And the false secret share is entirely independent of the secret share provided by other real refactorers.

Case 2. Noncooperative attack with as synchronization (NCAAS). When participating in secret reconstruction, all reconstructors show that the secret share is asynchronous, and there is no collusion between internal hostile users or satellites. The best attack idea for internal hostile users or satellites is to finally show the false secret share and collect as many real secret shares as possible.

Case 3. Collusion attack with synchronization (CAS). When all refactorers participate in secret reconstruction, the secret shares they show are synchronous, and there is collusion between internal hostile users or satellites. Internal hostile users or satellites can conspire to generate and produce false secret shares. When the number of false secrets constructed is greater than or equal to the threshold, the other honest reconstructors reconstruct the false secrets constructed by their internal enemies.

Case 4. Collusion attack with asynchronization (CAAS). When all refactorings participate in secret refactoring, the secret share is asynchronous, and there is collusion between internal hostile users or satellites. Same as NCAAS, the best attack idea for internal hostile users or satellites is to choose to show the false secret share finally and collect as many real secret shares as possible before that. The false secret share of conspiracy presented will have a greater chance of attack success.

5.2. Safety Analysis. This section gives a detailed security analysis of the scheme in this section. To clearly represent the security proof process of the scheme, the following assumptions and symbolic definitions are given: suppose the refactorer set is $\mathbf{R} = \{P_1, P_2, \cdots, P_n\} (n \geq t)$, where P_i and $P_j (i \neq j)$ are arbitrary honest refactorers. $\mathscr{A}$ is defined as any internal deceiver. α is the number of internal fraudsters. m is the number of all refactorers. $\mathbf{C}$ is the set of identified internal fraudsters.

Theorem 1. *$\mathscr{A}$ correctly guesses that the probability that round k can reconstruct the shared secret is 1/v.*

Proof. The real shared secret is hidden in the reconstruction sequence by the ground control center. $\mathscr{A}$ does not know the correct location and can only iterate the reconstruction in turn according to the reconstruction sequence. The probability of successfully guessing the real secret location is $1/v$. □

Theorem 2. *In this section's scheme reconstruction process, any cheater will be identified by the honest refactorer, and the fraud identification probability is 1 − 1/(q − 1).*

Proof. Suppose the secret is reconstructed in the k round, and the share of the subsecret shown by the reconstructor P_i to P_j is $s'_{i,k}$, where $s'_{i,k,0} \neq s_{i,k,0}$, $i \neq j$. P_j verifies $s'_{i,k}$ through the $g_{j,i} \in GF(q) - \{0\}$ sent by the distributor, P_j has $q-1$ validation equations, considering two equations:

$$\begin{aligned} g_{j,i,k} y_0 + ID_j y_1 + \cdots + ID_j^{t-1} y_{k-1} = b_{j,i,k} \bmod p, \\ g'_{j,i,k} y_0 + ID_j y_1 + \cdots + ID_j^{t-1} y_{k-1} = b'_{j,i,k} \bmod p, \end{aligned} \tag{4}$$

where $g_{j,i,k} \neq g'_{j,i,k}$, if $s'_{i,k}$ and $s_{i,k}$ are the solutions of these two equations; the two equations are subtracted to obtain $(g_{j,i,k} - g'_{j,i,k}) s_{i,k,0} = b_{j,i,k} - b'_{j,i,k}$, $(g_{j,i,k} - g'_{j,i,k}) s'_{i,k,0} = b_{j,i,k} - b'_{j,i,k}$. Because inequalities $g_{j,i,k} \neq g'_{j,i,k}$ and $s_{j,i,k} \neq s'_{j,i,k}$ are contradictory, there is only one case of the equation satisfying the subsecret share of P_j verifiable P_i. Then, the probability that P_i successfully deceives P_j is at most $1/(q-1)$, and the probability of being recognized by P_j is not less than $1 - 1/(q-1)$. □

Theorem 3. *When $m - \alpha \geq t$, the Harn subsecret consistency detection scheme can always detect deception [6].*

Proof. In 2011, Ghodosi pointed out that the spoofing detection scheme of reference [6] cannot successfully detect spoofing, regardless of whether the secret reconstruction protocol is asynchronous or synchronous [7]. Suppose that there are $q (q \geq 1)$ deceivers $\{P_{i1}, \cdots, P_{iq}\}$ and $t-1$ honest reconstructors in the secret reconstruction process. The deceivers conspire to generate a random $t-1$ degree polynomial $g(x)$. For any honest participant, P_i meets the requirements of $g(i) = 0$. The deceiver calculates the false secret share $g(i1), \cdots, g(iq)$ for himself, and he shows false secret shares to all honest people and the sum of true secrets $h(i1), \cdots, h(iq)$. When honest reconstructors receive false secret shares, their reconstructed polynomial is $h(x) = f(x) + g(x)$. The deceiver can easily calculate the true shared secret $f(0) = h(0) - g(0)$, while the honest reconstructor reconstructs the wrong secret $h(0)$. The highest degree of the false polynomial $h(x)$ is $t-1$, so consistency spoofing detection can be bypassed. If there are at least t honest reconstructors in the scheme, no matter how the deceiver constructs, the highest degree of the polynomial $g(x)$ is at least t, which does not meet the consistency detection. To sum up, when $m - \alpha \geq t$, secret consistency can always successfully detect deception. □

Theorem 4. *When $m - \alpha \geq t$ the scheme in this chapter is safe and fair under NCAS.*

Proof. In the case of NCAS, it is assumed that there is only a single deceiver $\mathscr{A}$ in the secret reconstruction process. According to the attack method in the proof of Theorem 3, the scheme in this section cannot detect deception because the highest degree of the false polynomial $h(x)$ is $t-1$, so the condition $m - \alpha \geq t$ must be satisfied. Due to the lack of cooperation between attackers, arbitrary deceiver $\mathscr{A}$ assumes that the other reconstructors are honest and cannot obtain adequate information through collusion. Suppose that the false subsecret share constructed by $\mathscr{A}$ in round k is $s'_{i,k} \equiv (s_{i,k,0} + s'_{i,k,0}, s_{i,k,1} \cdots, s_{i,k,t-1})$, according to Theorem 1, the probability that the false subsecret share presented by $\mathscr{A}$ is verified by the honest reconstructor is less than that of $1/(q-1)$. So it cannot pass the verification and obtain the votes of other reconstructors, and the k rounds are not necessarily the location of the real secret in the reconstructed sequence. The probability of $\mathscr{A}$ successfully guessing the reconstruction location is $1/v$. When the security parameter is large enough, the probability of $\mathscr{A}$ successfully cheating is negligible. To sum up, when $m - \alpha \geq t$, the scheme in this section is safe and fair under NCAS. □

Theorem 5. *When $m - \alpha \geq t$, the scheme in this section is safe and fair under NCAAS.*

Proof. In the case of NCAAS, it is assumed that there is only a single deceiver $\mathscr{A}$ in the secret reconstruction process. When $m - \alpha \geq t$, the number of honest reconstructors in H is not less than t. Because it is an asynchronous environment, the best attack strategy of $\mathscr{A}$ is to let the honest reconstructor show the real subsecret share first and then $\mathscr{A}$ reconstruct the secret polynomial $f_i(x)$ through $t-1$ real subsecret shares. Therefore, in the first l rounds of secret reconstruction, $\mathscr{A}$ chooses to show the real subsecret share. In the round $l+1$, $\mathscr{A}$ found that the real secret reconstruction position was in the previous round, condition $m - \alpha \geq t$ limits that $\mathscr{A}$ cannot attack in the way shown in the proof of Theorem 3. $\mathscr{A}$ can only randomly select random numbers on $\mathrm{GF}(q)$ to construct false subsecret share $s'_{i,k} \equiv (s'_{i,k,0}, s_{i,k,1} \cdots, s_{i,k,t-1})$. At this time, the false subsecret share constructed by $\mathscr{A}$ cannot pass the consistency detection. The secret reconstruction enters the identification algorithm. $\mathscr{A}$ obtains the number of votes $T < m/2$ and is identified as a deceiver. It is removed from the reconstruction process and added to the attacker set C. Honest refactorers in $|\mathrm{R} \setminus \mathrm{C}|$ continue to execute the reconstruction protocol, requests d from D, and then reconstructs the real secret $s = a_l \oplus d$. To sum up, when $m - \alpha \geq t$, the scheme is safe and fair under NCAAS. □

Theorem 6. *When $(m - \alpha \geq t) \cup (m > 2(\alpha - 1))$, the scheme in this section is safe under CAS.*

Proof. In the case of CAS, when $\alpha \geq t$, the deceiver set α can calculate the secret polynomial $f_i(x)$ in advance. As described in Theorem 3-(1) of reference [15], the attack mode passes consistency detection (for example, when $\alpha = t$, $m - \alpha = t - 1$, $m = 2t - 1$. α attackers can precalculate sequence bits to show legal secret shares in the first l rounds. In the round l, the subsecret shares of other real reconstructors are calculated by using the reconstructed correct sequence bit polynomial $f_{l+1}(x)$. Assuming $s_{1,l+1,1}, s_{2,l+1,1}, \cdots, s_{t-1,l+1,1}$, the false polynomial $f'(x)$ is constructed by using $t-1$ real subsecret shares and a random number s'_{l+1}. Use $f'(x)$ to generate the subsecret share $s'_{t,l+1,1}, s'_{t+1,l+1,1}, \cdots, s'_{2t-2,l+1,1}$ of the remaining $t-1$ deceivers. At this time, the subsecret share shown by all the reconstructors is $s_{1,l+1,1}, s_{2,l+1,1}, \cdots, s_{\mathrm{t}-1,l+1,1}, s'_{t,l+1,1}, s'_{t+1,l+1,1}, \cdots, s'_{2t-2,l+1,1}, s'_{l+1}$. The polynomial obtained by all honest reconstructors is $f'(x)$, so it can pass the consistency detection). Perhaps as shown in Theorem 3, conspiring to calculate the false subsecret share passes the consistency detection, so $m - \alpha \geq t$ is required. The false subsecret share constructed by $\mathscr{A}$ cannot pass the consistency detection, and the secret reconstruction enters the identification algorithm. The honest reconstructor will not vote for any $\mathscr{A}$ after identification, and the internal cheater set can vote for each other. Therefore, the scheme needs to meet $m > 2(\alpha - 1)$. At this time, the attacker is eliminated, and the honest reconstructor reconstructs the real shared secret according to the protocol. When $\alpha < t$, the deceiver set can only pass the consistency detection through the attack shown in Theorem 3 when the condition $(m - \alpha \geq t) \cup (m > 2(\alpha - 1))$ is satisfied, the protocol is executed normally, or $t-1$ conspirators guess the share of the tth subsecret, and the guessing probability is negligible. The attacker can only show α random numbers and cannot pass the consistency detection. To sum up, when $(m - \alpha \geq t) \cup (m > 2(\alpha - 1))$, the scheme in this section is safe and fair under CAS. □

Theorem 7. *When $(m - \alpha \geq t) \cup (m > 2(\alpha - 1))$, the scheme in this section is safe under CAAS.*

Proof. In the case of CAAS, no matter whether the number of fraudsters α is greater than or equal to the threshold value t, due to the asynchronous environment, any $\mathscr{A}$ can always collect $t-1$ real subsecret shares and calculate whether the previous round is a real satellite secret reconstruction location. Therefore, in the first round l, $\mathscr{A}$ shows the real subsecret share. Until round $l+1$, $\mathscr{A}$ reconstructs the secret polynomial $f_{l+1}(x)$ through the collected real subsecret share and finds that the real secret reconstruction position is in the previous round. It selects the two attack methods described in Theorem 6 to pass the consistency detection; when the condition $(m-\alpha \geq t) \cup (m > 2(\alpha - 1))$ is satisfied, the honest refactor can execute the protocol normally. To sum up, when $(m-\alpha \geq t) \cup (m > 2(\alpha - 1))$, the scheme is safe and fair under CAAS. □

6. Scheme Comparison and Performance Analysis

From the perspective of security fairness, reference [7] points out that Harn deception detection and identification has security problems [5]. But references [16, 19] are not perfect based on Harn deception detection [6]. Under NCAS $(m > t)$, NCAAS $(m-\alpha < t-1) \cap (m > t)$, and CAS $(\alpha < t) \cap (m > t)$, the deceiver can successfully bypass the subsecret consistency detection algorithm through the attack method shown in Theorem 3. And the deceiver cannot be recognized by the honest reconstructor. Therefore, the restrictions listed in the above different scenarios should be changed $m-\alpha \geq t$. Only in this way can the scheme be safe and fair. Under CAS and CAAS, when the number of honest reconstructors is close to that of deceivers, the scheme in this paper needs fewer participants than references [16, 19]. The scheme in reference [11] cannot completely resist asynchronous attacks and synchronous collusion attacks. The schemes in references [14, 15] only consider the fairness of secret reconstruction in an asynchronous environment but do not consider CAS and NCAS. And the schemes do not meet complete fairness, and both need a hash function to ensure security. When deception is detected, the scheme stops immediately, which is not applicable in the actual environment. Compared with the above scheme, the protocol will not terminate immediately when deception is detected, to ensure that honest participants can reconstruct satellite secrets. Secondly, the scheme does not need the protection of a similar hash function, meets unconditional security, and ensures secure communication.

From the perspective of scheme complexity, the scheme reconfiguration protocol in this section requires $\theta(v)$ a round of secret reconfiguration protocols to achieve fairness, which is the same as the fair secret sharing scheme proposed in references [11, 14–16, 19]. From the perspective of each round of reconfiguration protocol, each participant in this scheme receives k elements on $\mathrm{GF}(p)$ from D, and additional $2(n-1)$ elements, $F(\mathrm{ID}_i, y) \bmod p$ containing t elements for generating the session key, there are $k+t+2(n-1)$ in total. In the fair secret sharing scheme constructed for binary polynomials in reference [17], the additional verification elements $a_{j,i,l}$ and $b_{j,i,l}$ distributed to participants are nk. Each round D has to construct x binary asymmetric polynomials of order nk and distribute many additional elements. Compared with this, this scheme has a key free negotiation process between participants, fewer additional elements, and better communication and computational efficiency. Sun proposed an efficient deception recognition algorithm. The method of logic or operation between correctly labeled vectors is used to replace the $m-t$ sub-Lagrange interpolation in reference [6], reducing the fraud identification overhead [19]. The scheme in this secret uses subsecret consistency for deception detection, which is the same as references [16, 19], only $O(1)$. The computational complexity of the deception identification algorithm of Harn and Lin is $O(m!)$ [6]. Similarly, the computational complexity of the deception identification algorithm in the scheme of Zhang et al. is also $O(m!)$ [16]. Although the deception identification algorithm in the Sun scheme reduces the overhead, the computational complexity is also $O(m!)$ [19]. The scheme deception identification algorithm in this paper only needs $m-1$ times of solution verification operation of secret share polynomial, and the computational complexity is $O(m)$. According to the discussion in reference [8], in the deception identification algorithm in reference [7], assuming threshold $t=6$, the number of participants is required to be $m \geq 16$, and the identification algorithm requires 2^{64} times of the Shamir secret reconstruction operation. Therefore, the scheme of references [16, 19] is not practical. To more intuitively represent the fraud detection and identification overhead between different schemes, suppose T_p is the modular exponentiation operation time, $T_L(m)$ is the interpolation operation time of m points, T_H is the hash operation time, and T_v is the polynomial solution verification operation time. As shown in Table 1, the scheme in this section is compared with other fair secret sharing schemes in detail.

7. Parameter Analysis

In Sections 4.1 and 4.2 of this paper, it can be seen that the threshold value is an important parameter affecting satellite secret distribution and satellite secret reconstruction. Furthermore, it has a crucial impact on the generation of binary symmetric polynomials and the order of interpolation polynomials. It can be seen from reference [23] that the security and reliability of the (n, k) tthreshold secret sharing scheme are closely related to the key update cycle and the threshold value. Therefore, choosing the appropriate key update cycle and threshold is of great significance in improving the security of this scheme.

7.1. Key Update Cycle and Key Share Leakage Rate. When an attacker intercepts the shared secret share between satellite nodes, it is called key share leakage and $P(t)$ is used to represent the distribution function of the key share leakage rate with time t:

$$P(t) = 1 - e^{-\lambda t}. \quad (5)$$

Figure 2 shows the probability distribution of key share leakage with the key update cycle T, x is λ, as can be seen from the figure, λ at the same time, the larger the key update cycle, the higher the key share leakage rate. In Figure 2, λ takes 0.02, 0.04, and 0.06, respectively, which are the corresponding values of $P(t)$.

Table 1: Comparison of fair secret sharing schemes.

Scheme	Safe passage between participants	Completely fair	Security assumptions	Spoofing detection overhead	Cheater identification overhead
Reference [11]	No	No	No	$T_L(t)+(m-t)T_v$	mT_v
Reference [16]	No	No	No	$T_L(m)$	$C_m^t T_L(t)+(m-t)T_L(m)$
Reference [19]	No	No	No	$T_L(m)$	$C_m^t T_L(t)$
Reference [15]	Yes	No	DLP	mT_p+T_H	No
Reference [22]	Yes	No	Yes	No	No
Our scheme	Yes	Yes	No	$T_L(m)$	$T_v(m-1)$

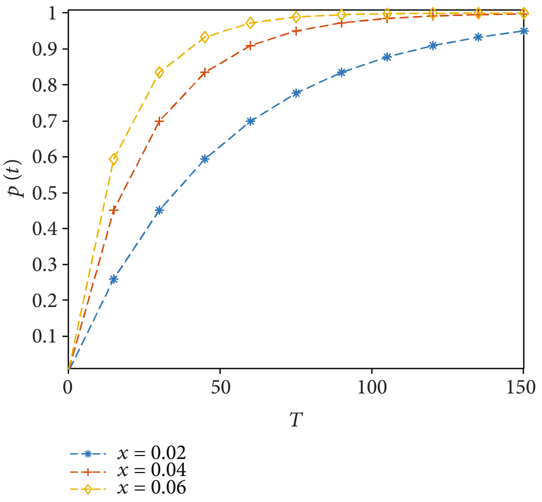

Figure 2: Key share leakage rate graph.

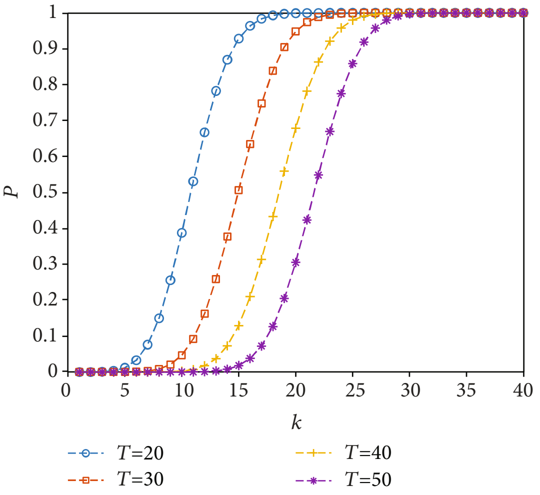

Figure 3: P versus k under different T.

7.2. Influence of Threshold on Network Security. Each node of the satellite network has different key shares. In a key update cycle, the probability of the key share being intercepted by the attacker is as follows:

$$P=\sum_{i=0}^{k-1} C_n^i p(T)^i (1-p(T))^{n-i}=\sum_{i=0}^{k-1} C_n^i \left(1-e^{-\lambda T}\right)^i \left(e^{-\lambda T}\right)^{n-i}, \tag{6}$$

where P stands for network security, $n=40, \lambda=0.015$, the variation curve of P concerning k is given in Figure 3, and the values of $T=20, 30, 40, 50$. As can be seen from the figure, when the key update period T remains unchanged, P will gradually increase with the increase of the threshold value k. when k increases to a certain extent, P approaches 1. When t is different, P corresponding to the same k value is also different. Therefore, it is necessary to increase the threshold value while increasing the key update cycle to improve network security. To improve the security and reliability of the (n, k) threshold secret sharing scheme, it is necessary to set the key update cycle and threshold reasonably.

8. Conclusion

This paper proposes a protected secret sharing scheme for satellite networks based on binary symmetric polynomials, points out the conditional errors in references [15, 17], and proves the complete security fairness under four attack models. Compared with the existing fair secret sharing schemes, this scheme has two characteristics: The first is verifiable multisecret sharing. This scheme can effectively ensure participants' effectiveness with secret shares before secret transmission. Secondly, suppose participants want to communicate with each other, after ensuring participants' effectiveness. In that case, participants can communicate through the key distributed by the distribution center to form a session key to resist the external attack of satellite communication node attackers. There is no need for additional key negotiation processes between participants to reduce the number of interactions to improve the performance of the satellite network. Thus, it can reduce the bit error rate of the link and ensure safe communication between users. At the same time, the scheme does not rely on any security assumptions, is unconditionally secure, and has low fraud detection and identification overhead, which reduces the cost of remote maintenance and management of satellite networks and improves reliability and security.

Data Availability

All data, models, and code generated or used during the study appear in the submitted article.

Conflicts of Interest

The authors declare that they have no conflicts of interest to report regarding the present study.

Acknowledgments

This work is supported by The State Key Laboratory of Integrated Services Networks, Xidian University (ISN22-13).

References

[1] B. Vaseghi, S. S. Hashemi, S. Mobayen, and A. Fekih, "Finite time chaos synchronization in time-delay channel and its application to satellite image encryption in OFDM communication systems," *IEEE Access*, vol. 9, pp. 21332–21344, 2021.

[2] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612-613, 1979.

[3] G. R. Blakley, "Safeguarding cryptographic keys," in *MARK 1979: International Workshop on Managing Requirements Knowledge*, pp. 313–318, IEEE, Piscataway, NJ, 1979.

[4] T. Rabin and M. Ben-Or, "Verifiable secret sharing and multiparty protocols with honest majority," in *STOC 1989: Proceedings of the twenty-first annual ACM symposium on theory of computing*, pp. 73–85, ACM, New York, NY, 1989.

[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] L. Harn and C. Lin, "Detection and identification of cheaters in (t, n) secret sharing scheme," *Designs, Codes and Cryptography*, vol. 52, no. 1, pp. 15–24, 2009.

[7] H. Ghodosi, "Comments on Harn–Lin's cheating detection scheme," *Designs, Codes and Cryptography*, vol. 60, no. 1, pp. 63–66, 2011.

[8] Y. Liu, C. Yang, Y. Wang, L. Zhu, and W. Ji, "Cheating identifiable secret sharing scheme using symmetric bivariate polynomial," *Information Sciences*, vol. 453, pp. 21–29, 2018.

[9] M. Tompa and H. Woll, "How to share a secret with cheaters," *Journal of Cryptology*, vol. 1, no. 3, pp. 133–138, 1989.

[10] H.-Y. Lin and L. Harn, "Fair reconstruction of a secret," *Information Processing Letters*, vol. 55, no. 1, pp. 45–47, 1995.

[11] Y. Tian, J. Ma, C. Peng, and Q. Jiang, "Fair (t, n) threshold secret sharing scheme," *IET Information Security*, vol. 7, no. 2, pp. 106–112, 2013.

[12] L. Harn, "Comments on 'Fair (t, n) threshold secret sharing scheme'," *IET Information Security*, vol. 8, no. 6, pp. 303-304, 2014.

[13] L. Harn, "Secure secret reconstruction and multi-secret sharing schemes with unconditional security," *Security and Communication Networks*, vol. 7, no. 3, 573 pages, 2014.

[14] L. Harn, C. Lin, and Y. Li, "Fair secret reconstruction in (t, n) secret sharing," *Journal of Information Security and Applications*, vol. 23, pp. 1–7, 2015.

[15] W. Y. Gu, F. Y. Miao, and X. T. He, "Fair secret sharing scheme based on bivariate symmetric polynomials," *Computer Engineering and Applications*, vol. 52, no. 13, pp. 38–42, 2016.

[16] B. H. Zhang, X. J. Xie, and Y. S. Tang, "Unconditionally secure fair secret sharing scheme," *Journal of Cryptography*, vol. 4, no. 6, pp. 537–544, 2017.

[17] W. W. Yang and Y. Q. Xing, "Fair secret sharing scheme based on binary asymmetric polynomial," *Journal of Network and Information Security*, vol. 5, no. 1, pp. 22–29, 2019.

[18] J. Li, X. Wang, Z. Huang, L. Wang, and Y. Xiang, "Multi-level multi-secret sharing scheme for decentralized e-voting in cloud computing," *Journal of Parallel and Distributed Computing*, vol. 130, pp. 91–97, 2019.

[19] D. J. Sun, *Efficient Deception Detection Secret Sharing Scheme and Its Application Research*, Hubei University of Technology, 2020.

[20] X. Liu, A. Yang, C. Huang, Y. Li, T. Li, and M. Li, "Decentralized anonymous authentication with fair billing for space-ground integrated networks," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 8, pp. 7764–7777, 2021.

[21] X. J. Ding, T. Hong, R. Liu, W. T. Peng, and G. X. Zhang, "Research on architecture of LEO satellite internet of things and key technologies," *Space-Integrated-Ground Information Networks*, vol. 2, no. 4, pp. 10–18, 2021.

[22] C. Y. Luo, W. Li, H. L. Li, and B. Jian, "Measurement method for space networks authenticated key security under distributed CA," *Dianzi Yu Xinxi Xuebao/Journal of Electronics and Information Technology*, vol. 31, no. 10, pp. 2316–2320, 2009.

[23] K. Xue, W. Meng, H. Zhou, D. S. L. Wei, and M. Guizani, "A lightweight and secure group key based handover authentication protocol for the software-defined space information network," *IEEE Transactions on Wireless Communications*, vol. 19, no. 6, pp. 3673–3684, 2020.