Wireless Communications and Mobile Computing

# Access Control and Privacy-Enhanced Technology in Next Generation Communication Networks

Lead Guest Editor: Yunchuan Guo
Guest Editors: Haitao Xu and Ming Li

# Access Control and Privacy-Enhanced Technology in Next Generation Communication Networks

Wireless Communications and Mobile Computing

# Access Control and Privacy-Enhanced Technology in Next Generation Communication Networks

Lead Guest Editor: Yunchuan Guo
Guest Editors: Haitao Xu and Ming Li

# Contents

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 4466942, 16 pages
https://doi.org/10.1155/2022/4466942

*Research Article*

# A Secure and Cached-Enabled NDN Forwarding Plane Based on Programmable Switches

**Ningchun Liu**,[1] **Shuai Gao**,[1] **Lei Yu**,[1] **and Guobiao He**[2]

[1]*School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, China*
[2]*National Computer Network Emergency Response Technical Team/Coordination Center of China (CNCERT/CC), Beijing, China*

Correspondence should be addressed to Shuai Gao; shgao@bjtu.edu.cn

Received 3 June 2022; Accepted 28 September 2022; Published 3 November 2022

Academic Editor: Haitao Xu



Recently, the rapid development of software-defined networking (SDN) and programming protocol-independent packet processors (P4) provides a potential possibility for the deployment of Named Data Networking (NDN), which has aroused tremendous attention in academia. Existing P4-based NDN solutions mainly focus on how to describe the stateful forwarding characteristics of NDN in a programmable switch environment. However, the existing solutions still face many challenges such as cache availability and data confidentiality and do not support retransmission of interest packets and multicast forwarding of data packets. In this paper, we propose a new NDN forwarding plane based on programmable switches to address the above challenges. We design a decoupled cache module to avoid a large impact on the data plane forwarding performance when the cache function is enabled. Also, we enhance the design of the existing P4-based NDN forwarding plane to support interest retransmission and multicast forwarding of data packets. In addition, with the advantage of network programmability of P4 technology, we extend the content permutation algorithm and integrate it into the NDN forwarding plane, which makes our scheme support lightweight secure forwarding. Finally, we evaluate our scheme in the prototype system and conduct comparative experiments with representative schemes. Experiment results show that our scheme outperforms it in terms of content retrieval latency and received throughput and can support lightweight secure forwarding with low cost.

## 1. Introduction

With the rapid development of network technology, content services represented by online videos have gradually become the main services of today's Internet. According to data provided by the Cisco Visual Networking Index, video traffic will account for 82 percent of IP traffic by 2022 [1]. In this context, the traditional TCP/IP network adopts a host-centric communication mode and faces many challenges such as redundant transmission. Furthermore, users are more concerned about the content itself, rather than the location of the content provider. Since 1999, a series of ICN architectures have been proposed. ICN adopts a content-centric communication model and decouples the location of content and content providers. In the ICN, content copies can be cached at intermediate routing nodes in response to subsequent requests for the same. Because it has the characteristics of ubiquitous caching in the network, it is considered to help reduce redundant transmission in the network [2].

As the most promising project in the ICN architecture, NDN follows a content-centric design philosophy and addresses and routes based on named data at the network layer. The forwarding based on named data in NDN is fundamentally different from the packet processing logic of the traditional TCP/IP network, and the existing software-based NDN routers greatly limit its forwarding performance, which makes NDN encounter challenges in large-scale deployment [3]. The emergence of software-defined networking (SDN) and programming protocol-independent packet processors (P4) provides a potential possibility for constructing high-performance NDN routers and deploying NDN on a large scale, which has attracted widespread attention in the academic community.

Recently, some researchers have carried out some works on P4-based NDN [3–6], which mainly focus on using P4 to

describe the stateful forwarding characteristics of NDN. Specifically, these works explore how to implement the parsing of hierarchical content names and three data structures in forwarding model of NDN, including forwarding information base (FIB), content store (CS), and pending interest table (PIT). However, the initial goal of the P4 design is protocol-independent forwarding. Since there is no special design for in-network caching, how to use P4 to describe the characteristics of NDN in-network caching has become the challenge faced by researchers. Existing works either use the register [7] and queue [8] of the programmable switch or implement the conceptual logic of content storage [4], which has the problem of small cache space and poor availability. In addition, although NDN adopts a content-centric security model and can protect the security of data by encrypting data at the application layer, it is necessary to carry out a corresponding security design for each application with security requirements. Compared with traditional NDN data encryption at the application layer, P4 provides the potential possibility for NDN network layer data encryption.

In this paper, we propose a new NDN architecture with a secure and cached-enabled NDN forwarding plane based on programmable switches. In this architecture, a decoupled cache module is presented to avoid a large impact on the data plane forwarding performance, where the CS module is decoupled into the CS-list and the CS-server. The bloom filter-based CS-list enables that only interest packets being satisfied by the local cache can be forwarded to the CS-server. By extending the PIT in the existing P4-based NDN solutions, our scheme support the retransmission of interest packets. Besides, to support multicast forwarding feature of data packets, we add a data clone operation to the forwarding pipeline, which significantly reduces the congestion of data packets in the programmable switch and improves the network throughput. In addition, our scheme also supports secure forwarding to mitigate the threat of eavesdropping attacks. The secure forwarding can be divided into two phases. First, the control plane sends a security configuration to a group of programmable switches. The security configuration mainly consists of symmetric keys written into registers on programmable switches. Second, according to the requirements of service, programmable switches in data plane can encrypt and decrypt security-sensitive traffic against eavesdropping attacks combined with content permutation.

Finally, we build the prototype system based on the software programmable switches, which realizes the basic functions of NDN, especially the high-availability in-network caching function and lightweight secure forwarding function proposed in this paper. Next, we conduct comparative experiments to evaluate the performance of our mechanism with representative AES-based scheme, S-BOX matrix-based scheme, and NDN.p4 solution. Experiment results show that our scheme outperforms in terms of content retrieval latency and received throughput. Besides, our scheme can support on-demand secure forwarding with low cost.

Our main contributions can be summarized as follows:

(i) We propose a new NDN architecture with a secure and cached-enabled NDN forwarding plane based on programmable switches, which can support high-availability in-network caching and lightweight securing forwarding

(ii) We present a decoupled cache module in the architecture, which can provide high-availability in-network caching capability for the data plane without affecting the forwarding performance as much as possible

(iii) We enhance the design of the PIT and egress pipeline in traditional NDN forwarding plane to support retransmission of interest packets and multicast forwarding of data packets

(iv) We integrate the extended content permutation algorithm into the NDN forwarding plane. The programmable switch can encrypt and decrypt data packets with security requirements against eavesdropping attacks

The rest of this paper is organized as follows: Section 2 surveys the related work. Section 3 introduces the example scenario. Section 4 presents the proposed architecture. In Section 5, we describe the design of the secure and stateful forwarding plane. Several experiments based on prototype system are done in Section 6 to evaluate the performance. Security analysis is discussed in Section 7. Finally, Section 8 concludes this paper.

## 2. Related Work

This section reviews the related work on traditional schemes for protecting data confidentiality in NDN and P4-based schemes for NDN and securing forwarding.

*2.1. P4-Based NDN Solutions.* In 2016, Signorello et al. [3] preliminarily implemented an NDN router for the first time by using P4, which contains most of the functions of NDN. Due to the limitations of stateful storage in $P4_{14}$, the implementation can not support cache requests. On this basis, Miguel et al. [9] presented an extended design of an NDN router in $P4_{16}$, which especially supports cache requests implemented with P4 externs. Due to the reference software target, BMv2-ss, having little support for it, the implementation is at the software logic level. Hou et al. [6] proposed a new P4-based NDN solution in which the cache function is supported by using an NFD-based cache server. Due to the use of a separate cache server, it is difficult to fully demonstrate the performance improvement brought by the cache. In addition, the security issues brought by data plane eavesdropping are not considered in this architecture. Karrakchou et al. [10] designed an enhanced NDN architecture by using P4, which extends the design of the traditional NDN forwarding pipeline and the existing FIB and PIT tables to provide support for several content delivery patterns. In addition, the architecture supports the execution of multiple P4 forwarding functions. Guo et al. [5] proposed an NDN routing mechanism in the P4 environment based on the NLSR protocol, which supports resource-location management and routing calculation. However, these

mechanisms [5, 10] do not take the functionality and performance of caching into account, which is an important feature for NDN.

In addition, the existing P4-based NDN schemes pay little attention to the security of the network layer. Compared with traditional NDN data encryption at the application layer, P4 provides the potential for NDN network layer data encryption.

*2.2. Traditional Access Control Schemes in NDN.* Different from the channel-based security model in traditional TCP/IP networks, NDN adopts the content-based security model. To enhance the confidentiality of data, researchers have designed many data access control schemes in NDN. Due to space limitations, we review the relevant work in the past five years. Other works can refer to the survey [11].

Xue et al. [12] designed a secure, efficient, and accountable edge-based access control framework for NDN, which adopts group signature to achieve anonymous authentication performed at the network edge. In addition, this framework uses the hash chain technique to reduce the overhead when users make continuous requests for the same file. However, the data is still transmitted in plain text in the network, and it is difficult to guarantee the confidentiality of the data.

Zhang et al. [13] presented a name-based access control scheme, which ensures data confidentiality in NDN. Combined with attributed-based encryption, the scheme supports fine-grained access control policies. By leveraging extended NDN naming conventions, the scheme also automates cryptographic key management. Misra et al. [14] proposed a data access control scheme based on broadcast encryption. In this scheme, even if the authentication entity is offline, the content producer can still allow legitimate consumers to access the data. Ning-Chun et al. [15] designed a data access control mechanism, which introduces the precalculated and cached auxiliary key block to reduce the retrieval delay of the auxiliary key block and the decryption cost for consumers. Besides, combining a two-dimensional one-way function, the mechanism ensures the uniqueness of the consumer's secret share.

The above-mentioned schemes [13–15] are conceptually similar. In these schemes, content producers encrypt their content before distributing them to the network. Consumers need to authenticate themselves and obtain the content decryption keys to be able to decrypt and access the content. Considering that the overhead of a symmetric encryption algorithm is much smaller than that of an asymmetric encryption algorithm, the content is usually encrypted using a symmetric encryption algorithm, and the symmetric key (also called the content key) is encrypted using an asymmetric encryption algorithm. However, the security cost is still a challenge for resource-constrained edge network scenarios. Also, most of these schemes are deployed at the application layer, which can support encryption and decryption of application data with security requirements by an extra specific development. Obviously, deployment can become cumbersome for multiple applications with security requirements.

*2.3. P4-Based Securing Forwarding and Cache Solutions.* For P4-based securing forwarding, Oliveira et al. [16] introduced traditional AES algorithm into programmable data plane, which allows nodes in data plane to establish secure channels between each other. However, this encryption mechanism still imposes high computational costs in the data plane, due to the interaction process including multiple key agreements between nodes. Liu et al. [17] proposed a P4-based network immune scheme (P4NIS) against the eavesdropping attacks, which is equipped with three lines of defenses. In the third line, the scheme encrypts all packet payloads in the application layer using traditional cryptographic mechanisms. In addition, the scheme reencrypts all packet headers in the transport layer using S-BOX matrix-based cryptographic schemes in the second line. However, the operation of S-BOX matrix-based cryptographic schemes is cumbersome and cannot meet the requirements of encoding and decoding data packets at line rates. Lin et al. [18, 19] designed an enhanced content permutation algorithm (eCPA) with programmable switches that can ensure the confidentiality of data in the ramable data plane. eCPA can perform encoding and decoding operations of data packets at line rate. Nevertheless, since NDN adopts a content-centric communication mode, eCPA cannot be directly integrated into P4-based NDN architecture.

In terms of P4-based cache, Jin et al. [7] used the software switch register unit to achieve cache, which has the best performance on P4. The size of the register unit of the software switch is limited. It is impractical to support packet buffering in MTU 1500 scenarios without modifying the source code. Qu et al. [8] proposed a P4-based queue cache. Queue cache refers to the use of a virtual port to store packets. When the packet leaves the queue, by retrieving the packet, the packet enters the queue again to realize packet cache. However, the cache capacity required by NDN is larger than the capacity provided by the above solutions.

## 3. Example Scenario

To facilitate an understanding the rest of this paper, we present a typical SDN-based battlefield application scenario in Figure 1. In this scenario, there are four types of entities: (i) command center ("/military/task/control") that sends control commands, (ii) battle units ("/military/task/uav" and "/military/task/squad(1/2)") that receive commands from the command center, (iii) forwarders (Programmable Switch(1/2) and Gateway) that forward packets among battle units and command center, and (iv) SDN controller that drives the secure forwarding of the forwarders by configuring the flow table.

This battlefield scenario ensures that eavesdropping attackers cannot obtain certain pieces of data sent by the command center to the combat units. As an example, when the command center sends a command intended to squad 1, all the other entities or attackers in the wide area network (WAN) should not be able to see the content even if they have retrieved the data packets. In the rest of the paper, we

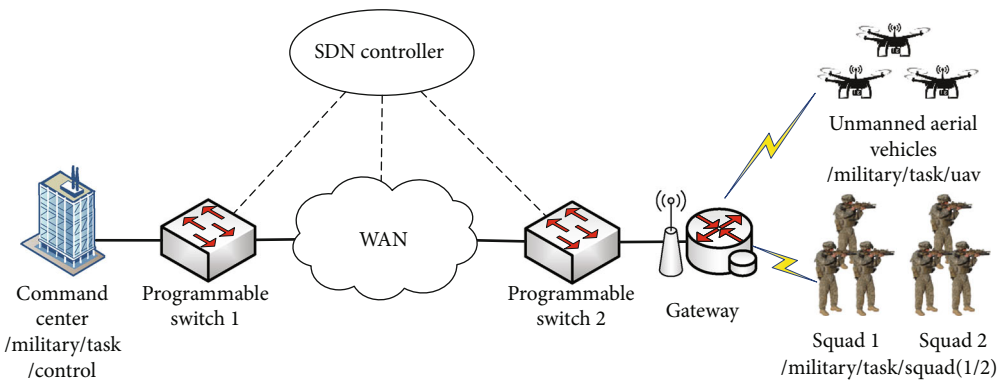



Figure 1: A SDN-based battlefield communication scenario.

illustrate how our architecture guarantees the confidentiality of communications.

## 4. Requirements, Proposed Architecture, and Security Assumptions

*4.1. Requirements.* We fully analyze the requirements for current P4-based NDN architecture as follows.

*4.1.1. Caching Availability.* Due to in-network caching being one of the important features of NDN, supporting high-availability caching is necessary for the proposed architecture. Specifically, the programmable data plane should support large-capacity caching. In addition, the support of the in-network caching feature should not affect the forwarding performance of the programmable data plane for other packets as much as possible.

*4.1.2. Data Confidentiality.* To mitigate eavesdropping attacks against packets, the proposed architecture should protect the confidentiality of data. Furthermore, the protection for data confidentiality should be lightweight in edge network scenarios.

*4.2. Proposed Architecture.* To meet the above requirements, we present a new P4-based NDN architecture, which supports more practical caching functionality and lightweight confidentiality protection of data at the network layer. Following the principles of SDN, the proposed architecture separates the control plane and the programmable data plane. The controller uses the control link between the control plane and the programmable data plane to configure the programmable switches. Figure 2 shows an overview of the proposed architecture.

The control plane is designed for topology discovery and management and data plane configuration containing table configuration, security configuration, and caching management. More specifically, the controller perceives the network topology through the link layer discovery protocol, including programmable switches and communication links. The controller also provides a content name registration service to establish resource-location mapping and store it in the database. The information in the database will subsequently be converted into configuration parameters for the programmable switches in the programmable data plane. Besides, the controller needs to set the cache policy and configure the security parameters of programmable switches.

In the programmable data plane, programmable switches focus on the packet forwarding for interest and data packets sent by consumers and producers, which mainly contains parser, PIT and FIB module, CS module, and encryption and decryption module. Since the NDN packets adopt the type-length-value (TLV) format, the traditional parser module needs to be extended to parse the variable-length NDN packet. Besides, according to the security requirements of the application, the encryption and decryption module performs encryption and decryption operations on the corresponding NDN packets. In ingress pipeline and egress pipeline, traditional PIT, FIB, and CS functionalities are included. In order to adapt to the environment of programmable switches, we split the traditional CS into CS-list and CS-server to improve caching performance. In addition, the CS module is integrated into the programmable switches via the virtual face. The design details of the programmable data plane are presented in Section 5.

*4.3. Security Assumptions.* We elaborate on the security assumptions of the proposed architecture as follows:

(i) The controller configures the programmable switches by using the control link between the control plane and the programmable data plane, which can usually be encrypted with TLS/SSL. We assume that the control link is secure; that is, the information transmitted in the control link cannot be eavesdropped on and tampered with

(ii) We assume that there are potential attackers in the network, and the attackers may launch man-in-the-middle attacks or eavesdropping attacks. In the man-in-the-middle attack, the attacker tampers with the content information of interest and data packets. In the eavesdropping attack, the attacker eavesdrops on the content information

(iii) Considering that the designed architecture is mainly to mitigate man-in-the-middle attacks or

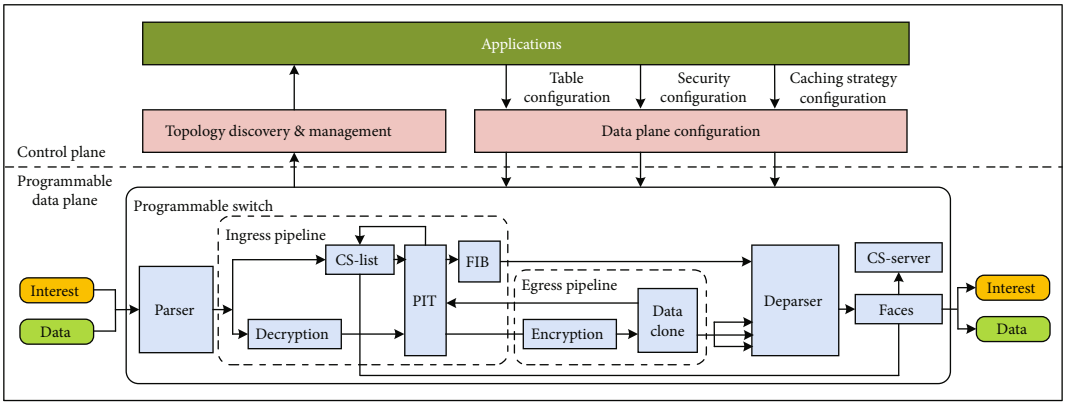



Figure 2: Overview of the architecture.

eavesdropping attacks in the network, we assume that consumers and content producers are trusted in the proposed architecture

## 5. Programmable Data Plane Design

Based on the previous work, we enhance the design of the programmable data plane, which support high-availability in-network caching, lightweight secure forwarding against eavesdropping attacks, and packet retransmission. In addition, the native multicasting function of NDN is realized in the proposed scheme.

In this section, we introduce the programmable data plane design from the following four main parts: (1) parser module, (2) PIT and FIB module, (3) CS module, and (4) encryption and decryption module. Finally, we show the workflow of our mechanism. It should be noted that PIT and FIB module, CS module, and encryption and decryption module are all main components of ingress pipeline and egress pipeline.

*5.1. Parser Module.* The main function of the parser module is to parse the header field of the packet from the outside to the inside according to the parsing sequence specified by the P4 target. The parsing process of each header field is called a parser method. When the parser module is working, after performing a parser method, the packet will enter the next parser method according to the byte offset of the currently processed header field and the next pending header field specified by the P4 target.

Since the NDN packet is composed of multiple continuous TLV blocks, we use the header stack data structure to simplify this series of the same type of adjacent header parsing. For continuous TLV blocks, P4 target will repeat the TLV header parsing process several times until all TLV block parsing is completed.

In addition, we extend the parsing process of the parser module to support version discovery of the NDN protocol and parsing of three types of NDN packets, including interest packet ($0 \times 05$), data packet ($0 \times 06$), and NACK packet ($0 \times 0a$).

*5.2. PIT and FIB Module.* Firstly, the variable name is converted into a fixed-length hash string by the hash function to be stored, parsed, and matched. FIB and PIT of NDN match the routing interest packet by the longest prefix of the content name. All packets must be converted from name to hash to enter the forwarding process. Hash is stored in the metadata of P4 target, which is only used for internal processing of the switch. Hash index and PIT table stored in cache will not be sent at the port and will be released after the packet processing process.

In the programmable switch, PIT and CS-list are stored in the register, which is completely handled by the switch itself. FIB is flow table form and issued by the control plane. The advantage of this form is that unnecessary communication between the control plane and the data plane is minimized to meet the decoupling requirements of the control plane and the data plane.

Then, as shown in Figure 3, different processing is performed depending on the type of the packet.

*5.2.1. NACK Packet.* When receiving an NACK packet, the programmable switch will forward it to the controller in the control plane for processing.

*5.2.2. Interest Packet.* When receiving an interest packet, the programmable switch will process it as follows:

*Step 1.* Match in the CS-list to determine whether there is a cache. The interest packet is directly forwarded to the face of CS-server when the cache is hit.

*Step 2.* Enter the PIT (recording the previously forwarded interest packets and their faces entering the switch) for matching. If the corresponding entry is found in the PIT, the routing node has forwarded the same requested interest packet but has no response yet. It is only necessary to discard the interest packet and update the face to the PIT. If the interest packet is received multiple times on the same face, it is regarded as a retransmitted packet, so the matching action table FIB is applied.

*Step 3.* If the corresponding matching table entry is not found in the PIT, then the content request is processed by

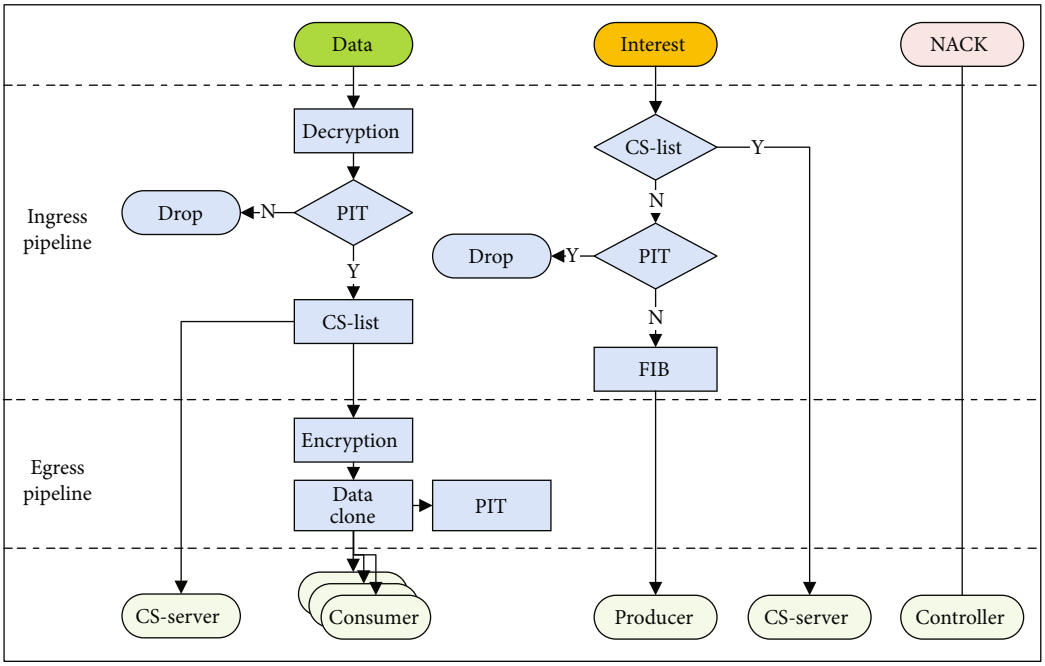



Figure 3: Packet processing flow in programmable switches.

the routing node for the first time and needs to be searched in the FIB (recording the forwarding rules between different nodes). The request is forwarded to other routing nodes according to the interface list corresponding to the content name in the FIB, and the PIT entry corresponding to the content name is modified to indicate to the next same content request that the interest packet of the content name is being requested to be resolved.

*5.2.3. Data Packet.* When receiving a data packet, the programmable switch will process it as follows:

*Step 1.* First, it is determined whether the data packet is an encrypted packet. If it is an encrypted data packet, decrypt the data packet. If it is not an encrypted data packet, transfer the data packet to PIT for subsequent matching operations. Date packets enter the PIT for matching. No matching entry means it is an unsolicited data packet and should be dropped.

*Step 2.* Next, match the data packet in PIT. If there is no hit, the data packet is unsolicited and should be dropped. If there is a hit in the PIT, the name of the data packet should be stored in the CS-list to wait for other consumer requests.

*Step 3.* Then, forward the data packet to consumers and the CS-server through faces according to the PIT. Before that, it is also necessary to encrypt and clone the data packet. If the data packet is a plain data packet, the data packet needs to be encrypted. In the data clone operation, the original data packet is first forwarded to the face with the smallest sequence number, and then, a loop is constructed through the clone operation in PIT. When a data packet is sent to each face, the corresponding bit in the PIT is set to zero. When the corresponding bits in the PIT are all zeros, the operation of sending data packets to the face stops. We specify the CS-server face to a specific value.

When the programmable switch performs the clone operation in the egress pipeline, a new thread is created to reexecute the egress process each time a new data packet is cloned. During this period, the switch can continue to process the newly arrived packets to make full use of face resources.

*5.3. CS Module.* Considering that in the traditional P4-based NDN solution, the CS is directly deployed in the pipeline of the programmable switch, which will have an impact on the forwarding performance of the programmable switch. In the proposed architecture, the decoupled cache module consists of two parts, CS-list and CS-server, as shown in Figure 4.

The CS-list is deployed in the registers and adopts a bloom filter-based structure, which only determines whether there is a cache, and does not perform a cache lookup operation. The CS-server is an independent process of the programmable switch, which will perform a name lookup operation on the received interest packets and also respond to the corresponding data packets.

In the decoupled cache module, the received interest packet will enter the CS-list based on the bloom filter for matching. If it hits, it will be forwarded to the CS-server for cache search and response. If there is no hit, the interest packet will enter the FIB for matching and forwarding. In CS-server, the data package corresponding to the interest package will be searched, and the data package will be returned along the reverse path. If the corresponding data packet is not found, the CS-server will return a NACK

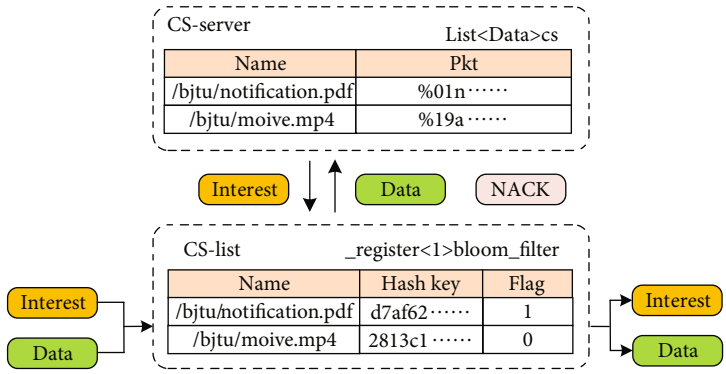



Figure 4: The schematic diagram of the cache module.

packet, which will be reencapsulated as an interest packet and sent to the FIB for matching and forwarding.

*5.4. Encryption and Decryption Module.* The encryption and decryption module encrypts and decrypts the payload of the data packets according to the security level field and flag bit in the header.

After receiving the data packets to be encrypted, the encryption module will process the data packets according to the procedure, as shown in Figure 5.

First, the payload of the packet is parsed as the header and divided into $n$ blocks. Next, the $n$ blocks will be secretly permuted according to the CPA-based encryption algorithm to complete the encryption process. Finally, the encryption module generates an encrypted packet.

The principle of CPA-based encryption algorithm is shown in Algorithm 1. For $n \geq 2$, we divide the payload field to be encrypted into $n$ codewords. Denote $n$ codewords for the $i^{\text{th}}$ state as $R_n^i = \{r_1^i \| r_2^i \| \cdots \| r_n^i\} (i = 0)$, where $r_n$ is the $n^{\text{th}}$ codeword of the field. $R_n^i = \{r_1^i \| r_2^i \| \cdots \| r_n^i\} (i = n-1)$ represents the result after $n-1^{\text{th}}$ permutation. Denote a binary cypher of size $n-1$ as $K_{n-1} = \{x_1 \| x_2 \| \cdots \| x_{n-1}\}$, where $x_i \in \{0, 1\}$ for $1 \leq i \leq n-1$. The $R_n^0$ is permuted with the key $K_{n-1}$, and the output result is $R_n^{n-1}$. Besides, we use the Buffer to store the intermediate value in secret permutation. In the Algorithm 1, line 4 to line 14 are a loop that is executed $n-2$ times. We divide the encoding actions by key value. If $x_i = 1$, we shift code words from $\{r_1^i \| \cdots \| r_{i-1}^i \| r_i^i \| r_{i+1}^i \| \cdots \| r_{n-1}^i \| r_n^i\}$ to $\{r_1^i \| \cdots \| r_{i-1}^i \| r_n^i \| r_i^i \| r_{i+1}^i \| \cdots \| r_{n-1}^i\}$, and the latter is $n$ codewords for the $i+1^{\text{th}}$ state. If $x_i = 0$, do not change the position of codeword $r_i$. If $i = n-1$, we execute lines 15 to 23. Finally, each codeword is disordered and located in a different position.

Since the operation of CPA-based decryption is opposite to the encryption operation, we will not elaborate further in this paper.

*5.5. Workflow.* The communication flow in the network is shown in Figure 6, which contains the system initialization phase and content retrieval phase. According to the different network entities responding to the interest packet, the content retrieval phase can be divided into two categories: content retrieval phase (from the producer) and content retrieval phase (from cache).

*5.5.1. System Initialization.* The control plane first generates a symmetric key and delivers the symmetric key to the programmable switch on the programmable data plane through a TLS-encrypted secure channel. After receiving the symmetric key, the programmable switch stores the symmetric key in its own register. In addition, the controller preconfigures the cache replacement policy in the programmable switch.

*5.5.2. Content Retrieval.* In the content retrieval phase, after the programmable switch 1 receives the interest packet carrying the security granularity requirement, it will first match its CS-list. If the CS-list hits, the interest packet will be sent to the CS-server through the face based on interprocess communication, and the CS-server will respond to the data packet corresponding to the interest packet. If there is no hit in the CS-list, the programmable switch will encrypt the interest packet according to the requirements of the security granularity. After that, the programmable switch sends the encrypted interest packet to the next-hop programmable switch 2.

After the programmable switch 2 receives the interest packet carrying the security granularity requirement, the switch first determines whether the interest packet is an encrypted interest packet according to the flag bit. If it is an encrypted interest packet, first perform the decryption operation and match its CS-list. If it is not an encrypted interest, it will directly match its CS-list. The execution operation in the CS-list is similar to that of the programmable switch 1, except that the encryption operation is no longer performed.

Contrary to the forwarding process of the interest packet, after the programmable switch 2 receives the responding data packet, the switch encrypts the data packet and forwards it to the programmable switch 1. After receiving the encrypted data packet, the programmable switch 1 decrypts the data packet and forwards the data packet to consumer 1.

## 6. Deployment and Performance Evaluation

In this section, we first implement the proposed scheme and build the prototype system based on it. Then, we evaluate the performance of the proposed scheme from four metrics.

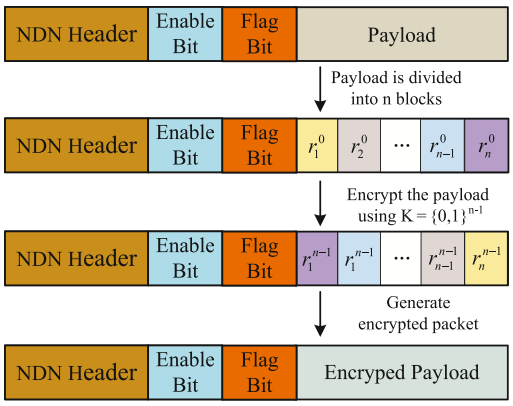



Figure 5: The encryption procedure for NDN packet.

**Input:**
RawPacket: $R_n^i = \{r_1^i \| r_2^i \| \cdots \| r_n^i\} (i = 0)$.
**Output**:
EncPacket: $R_n^i = \{r_1^i \| r_2^i \| \cdots \| r_n^i\} (i = n - 1)$.
//generate the key
key = KeyGenerate() = $K_{n-1} = \{x_1 \| x_2 \| \cdots \| x_{n-1}\}$;
//encrypt the RawPacket using the key $K_{n-1}$.
//step 1 to step n-2
**for** $i = 1$ to $n - 2$ **do**
  **if** $x_i = 1$ **then**
    Buffer = $r_n^{i-1}$;
    **for** $k = n - 1$ to $i$ **do**
      $r_{k+1}^i = r_k^{i-1}$;
    **end**
    $r_i^i$ = Buffer;
  **else**
    NoAction();
  **end**
**end**
//step n-1
**while** $i = n - 1$ **do**
  **if** $x_i = 1$ **then**
    Buffer = $r_n^{i-1}$;
    $r_{i+1}^i = r_i^{i-1}$;
    $r_i^i$ = Buffer;
  **else**
    NoAction();
  **end**
**end**

Algorithm 1: The CPA-based encryption algorithm.

*6.1. Deployment Environment.* We implemented the proposed scheme by combining the simplified NDN Forwarding Daemon (NFD) [20] with the enhanced NDN.p4 [3] using libraries from the behavioral model (BMv2) software switch [21]. More specifically, the traditional NFD is simplified to only retain the function of content store, and NDN.p4 is extended to support packet retransmission and secure forwarding. In addition, interest/data packets are exchanged between NDN.p4 and NFD through inter-process communication. The version of the P4 language accepted by the proposed scheme's P4 function target is the standard $P4_{16}$ version.

The topology of the prototype system is shown in Figure 7, which contains two consumers (consumer 1 and consumer 2), two programmable switches (programmable switch 1 and programmable switch 2), and a content producer. To more realistically show the network environment when the content provider provides services, we add a SPIRENT network emulator to the prototype system, which is used to simulate the latency of the Internet. The producer is connected to the programmable switch 2 and serves two consumers connected to a programmable switch 1. The latency of the network emulator is set to obey the delay characteristics of Internet ranging from 2.5 ms to 5 ms. Table 1 provides a summary of device parameter information in the prototype system.

*6.2. Experiment Result and Analysis.* We evaluate the performance of the proposed scheme from the following metrics: (1) security cost, (2) content retrieval latency, (3) packet processing latency, and (4) received throughput.

(i) Security cost refers to the total time of encryption operation and decryption operation of interest/data packets

(ii) Content retrieval latency refers to the time interval between when consumer sends the first interest packet and receives the last corresponding data packet

(iii) Packet processing latency refers to the time of each blocks in the programmable switches

(iv) Received throughput refers to the real-time throughput of data packets received by consumers, which reflects the forwarding performance of programmable switches

*6.2.1. Security Cost.* We evaluate the encryption and decryption time of the proposed scheme, the representative AES-based scheme [16], and S-BOX matrix-based scheme [17]. In the comparative experiment, we set the payload length of the NDN data packet to 1024 bits, and the key lengths used are all 128 bits. Correspondingly, in the proposed scheme, the codeword is 8 bits.

As shown in Figure 8, the encryption and decryption time of the proposed scheme in the programmable switch has a decrease of 96.9% and 84%, respectively, compared to the AES-based scheme and S-BOX matrix-based scheme.

*6.2.2. Content Retrieval Latency.* To demonstrate the impact the proposed scheme with high-availability caching and lightweight on-demand secure forwarding functions on the forwarding performance of programmable switches, we first compare the content retrieval latency with and without in-network caching in different schemes. In our experiment, the consumer sends 10,000 interest packets at a constant rate where the interest packets carry names with four name

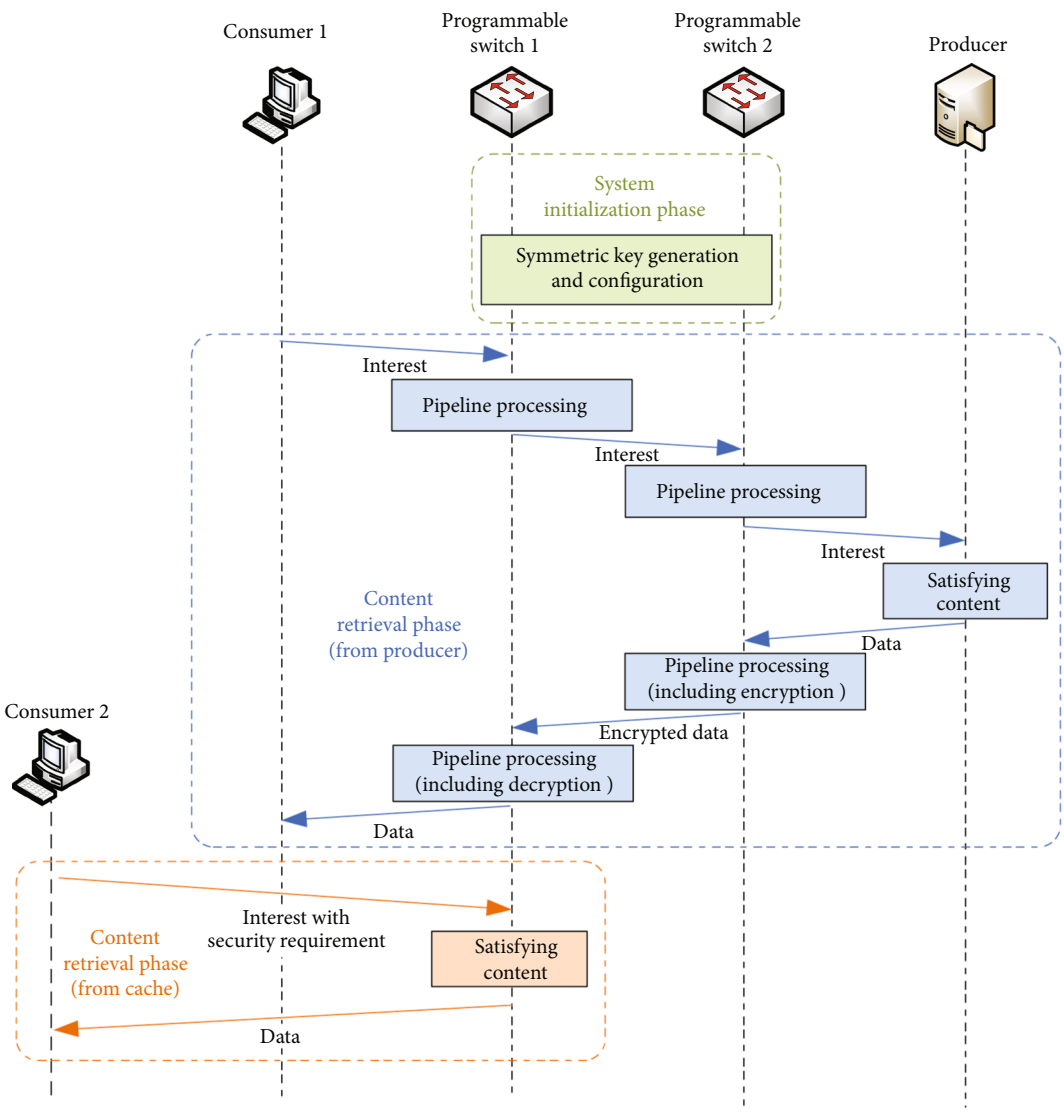



Figure 6: Communication flow in the network.

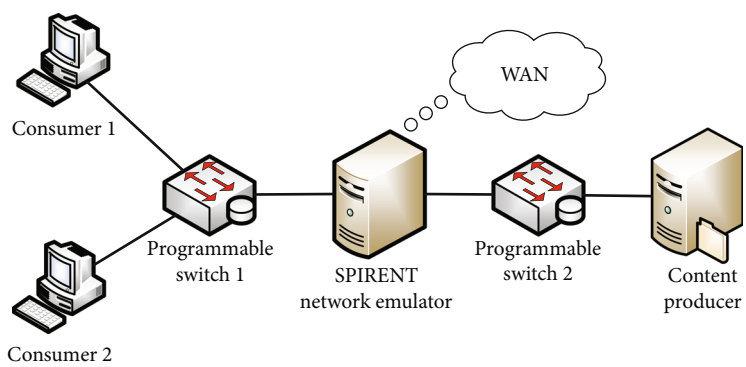



Figure 7: Topology of the prototype system.

components. In addition, the payload of the response data packet is set to 1024 bits.

Table 2 shows the content retrieval latency in different schemes, which are averaged over 10,000 experiments.

Compared with the traditional NDN.p4 solution [3], the proposed scheme is approximately equal in content retrieval latency when the in-network caching is disabled. When the in-network caching function is enabled, the proposed

Table 1: The summary of device parameter information in the prototype system.

| Num | Device | Parameter | Count |
|---|---|---|---|
| 1 | BMv2 switch | Intel Xeon Gold 5218@2.30 GHz 64.0 GB RAM | 2 |
| 2 | Consumer | Intel Xeon E3-1230v2@3.30 GHz 4.0 GB RAM | 2 |
| 3 | Producer | Intel Xeon E5-2620@2.00 GHz 16.0 GB RAM | 1 |
| 4 | Network emulator | SPIRENT 2*M series V2 2*1G-copper (GEM mode) | 1 |

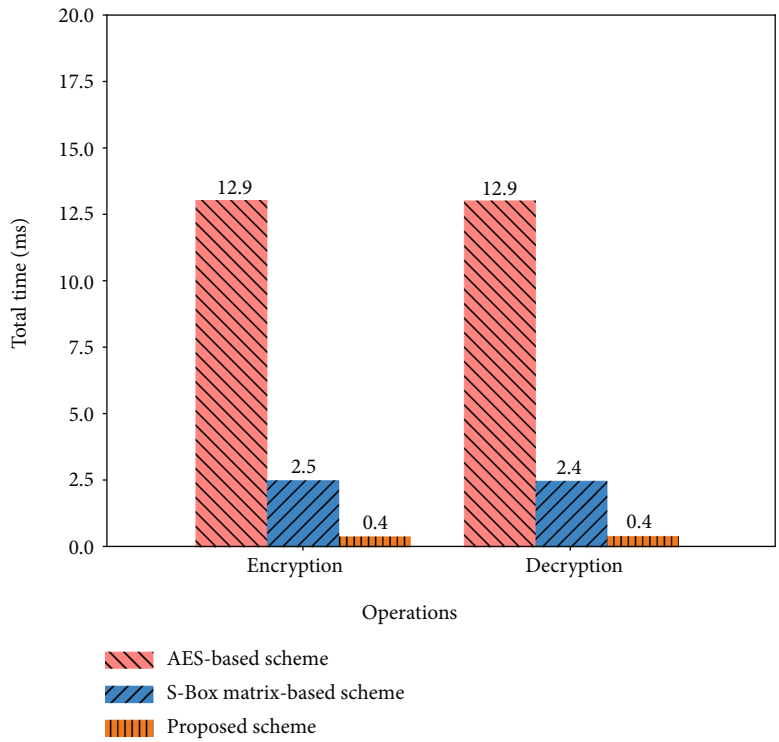



Figure 8: The comparison for encryption and decryption time.

scheme reduces the content retrieval latency by 57%. It is worth mentioning that the traditional NDN.p4 solution does not support the in-network caching function.

In addition, when the secure forwarding function is enabled, the content acquisition delay will increase by 0.8 ms (not supporting in-network caching) and 0.002 ms (supporting in-network caching), which is acceptable.

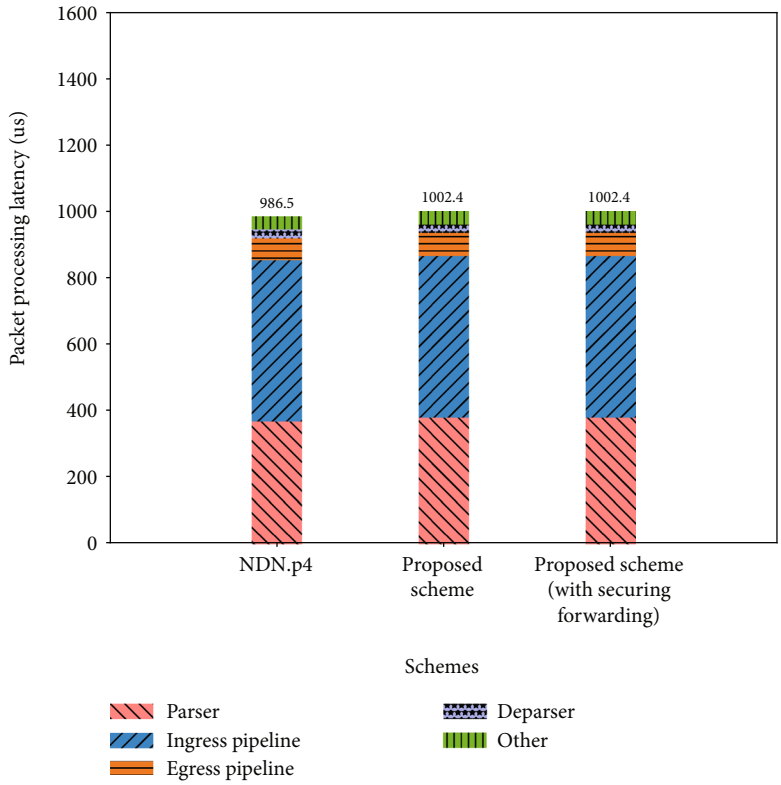
*6.2.3. Packet Processing Latency.* On the basis of content retrieval latency, we evaluate the packet processing latency between in different schemes, as shown in Figure 9.

Figures 9(a) and 9(b) show the comparison for main block processing latency of interest packets. The parser and ingress pipeline blocks account for the vast majority of the packet processing latency. Compared with the traditional NDN.p4 solution, the proposed scheme slightly increases the average packet processing latency. This is because the proposed scheme adds extra operations in the parser and pipeline in order to support the function of in-network caching and the retransmission of interest packets. In addition, enabling secure forwarding has no significant effect on the packet processing latency of interest packets. This is because the object of our encryption and decryption operation is the payload of the data packets, so it will not affect the packet processing latency of the interest packets.

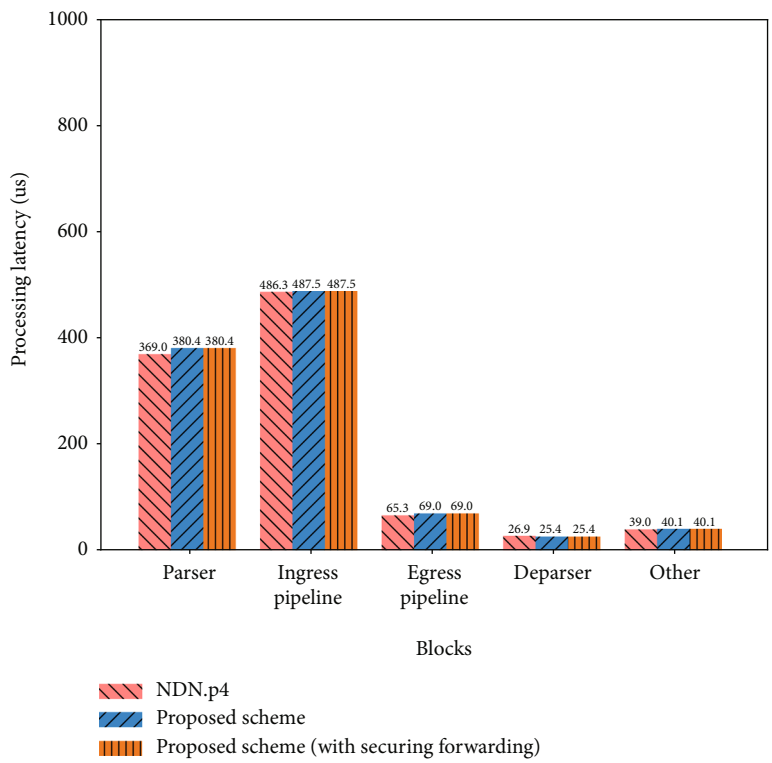
Table 2: The comparison of content retrieval latency.

| Different schemes | From content producer | From cache |
|---|---|---|
| NDN.p4 [3] | 6.350 ms | — |
| Proposed scheme | 6.370 ms | 2.727 ms |
| Proposed scheme (with secure forwarding) | 7.172 ms | 2.729 ms |

Figures 9(c) and 9(d) show the comparison for main block processing latency of data packets. Compared with the traditional NDN.p4 solution, the proposed scheme slightly reduces the average processing latency in ingress and egress pipelines. This is thanks to our optimization of the traditional forwarding pipeline. In addition, when the secure forwarding function is enabled, the average packet processing latency increases by about 24.5%, and most of the added extra latency is in the ingress pipeline. This is because programmable switches need to perform additional

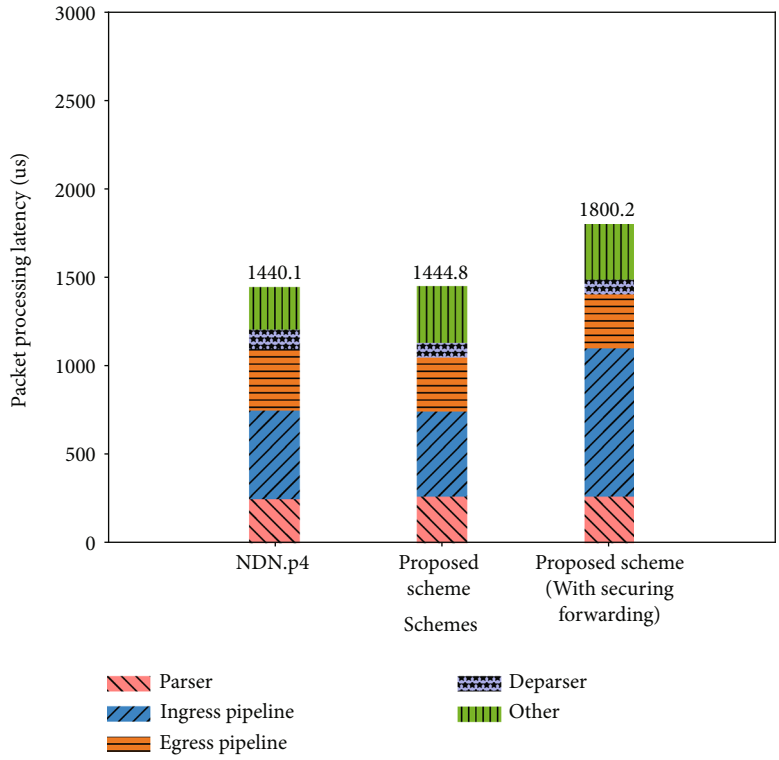



(a) Packet processing latency of interest packets

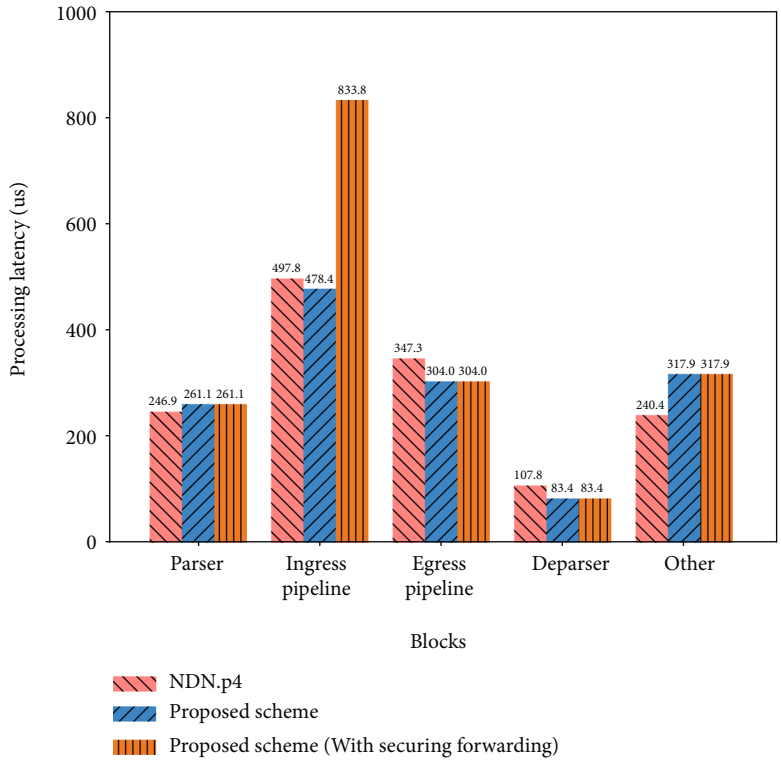



(b) Main block processing latency of interest packets

Figure 9: Continued.

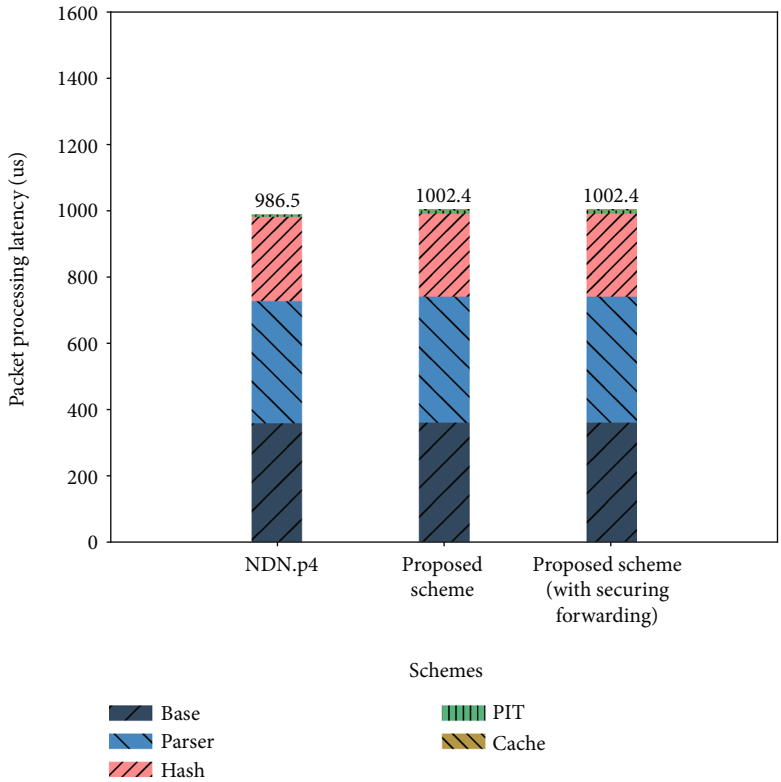



(c) Packet processing latency of data packets

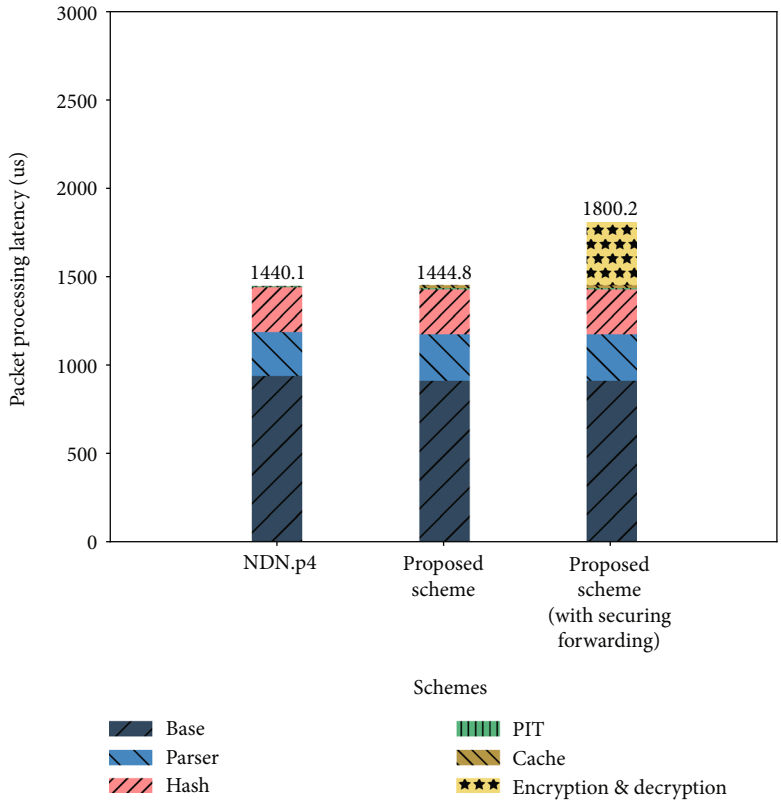



(d) Main block processing latency of data packets

Figure 9: Continued.




(e) Main operation processing latency of interest packets




(f) Main operation processing latency of data packets

Figure 9: The comparison for processing latency of P4 target.

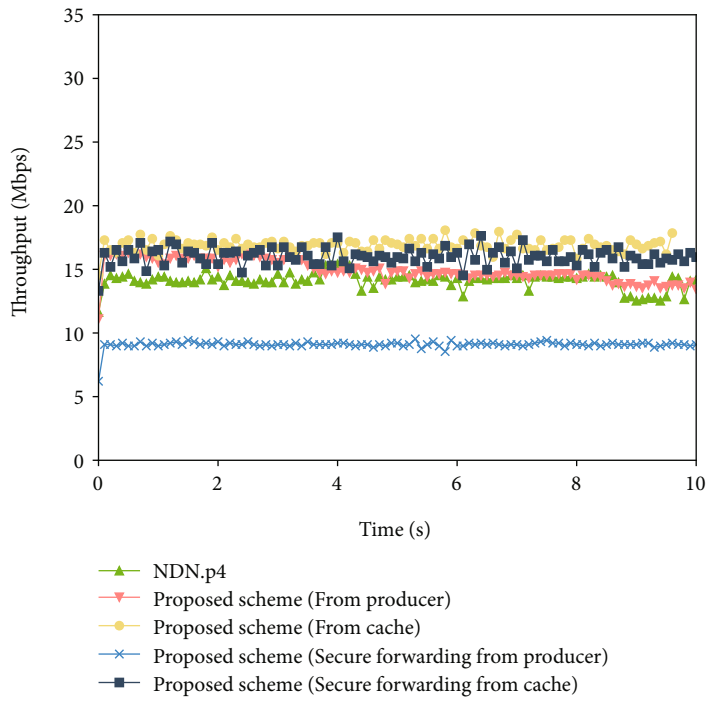



Figure 10: Comparison for received throughput.

operations in the ingress pipeline to complete the encryption and decryption of data packets.

Figures 9(e) and 9(f) show the main operation processing latency of interest/data packets in different schemes. The additional packet processing latency caused by enabling the in-network caching function is less than 0.5% of the total packet processing latency. This is due to the separation of the CS-list and the CS-server architecture. The in-network caching function is no longer a restriction on forwarding performance. In addition, when the secure forwarding function is enabled, an additional overhead of about 355 $\mu$s is added compared to the case where the secure forwarding function is not enabled, which is generated when the content permutation algorithm performs operations on the payload of the data packets. Since it only accounts for 20% of the total packet processing latency, the additional time cost is acceptable.

*6.2.4. Received Throughput.* Finally, we evaluate the received throughput of the proposed scheme. In our environment, the consumer requests a traffic of about 16 MB files. Furthermore, the proposed mechanism deploys a slow-start algorithm and a congestion avoidance algorithm for congestion control. We evaluate the receive throughput in different cases separately.

As shown in Figure 10, in the case of only enabling the cache function, the proposed scheme outperforms in the receive throughput compared with the traditional NDN.p4 solution, which is due to the fact that the proposed scheme supports the retransmission of interest packets and the multicast forwarding of data packets. In addition, when the secure forwarding function is enabled, the average received throughput drops to about 9 Mbps (from producer). It is worth noting that the traffic fluctuations at the end of the curve are related to retransmitted packets over a period of time, which is mainly due to the multithreaded design of BMv2. Every time a new data packet is cloned, a new thread will be created, which may lead to continuous processing of multicast forwarding. Multiple clone queues in a process cause high queuing delays for individual clone packets.

## 7. Security Analysis and Discussion

In this section, we describe the two main attacks and threats in the packet forwarding process. Next, we analyze how the proposed mechanism responds to these two attacks.

*7.1. Man-in-the-Middle Attack.* The man-in-the-middle attack is to hijack interest packets or the response data packets sent by the consumer through the intermediate switch and tamper with the content, to achieve the purpose of making the consumer unable to obtain the required information normally.

In the proposed scheme, the control plane periodically sends control signaling to the programmable switches located in the data plane through a TLS-encrypted control channel. The switch completes the pipeline operation (including encryption and decryption operations) of the data packets through control signaling. Even if a malicious device is connected to the network, since it does not have the control signaling periodically issued by the controller, it cannot complete the pipeline operation of the data packet to obtain the information of the data packet. Therefore, man-in-the-middle attacks can be mitigated.

*7.2. Eavesdropping Attacks.* Eavesdropping attacks mean that devices maliciously accessing the network can obtain rich content information in data packets forwarded in the network.

In the proposed scheme, we combine the content replacement algorithm to encrypt data packets with security requirements in the network. Since an attacker cannot brute force a key with a length of 128 bits or more in a short period of time, he cannot obtain the original content information, thereby mitigating the threat of eavesdropping attacks.

## 8. Conclusion

In this paper, we present the design of a secure and cached-enabled NDN forwarding plane based on programmable switches. To mitigate the impact on the forwarding plane performance, our scheme adopts a separate architecture of cache query and cache response. With the advantage of network programmability of P4 technology, we extend the content permutation algorithm and integrate it into NDN forwarding plane, which makes our scheme also support lightweight secure forwarding. In addition, we enhance the design of traditional NDN forwarding plane to support interest retransmission and multicast forwarding of data packets. Experiment results show that our scheme outperforms in terms of content retrieval latency and received throughput. Besides, our scheme can support on-demand secure forwarding with low cost.

Compared with software BMv2 switches, physical programmable switches have more powerful packet processing capabilities. Next, we will evaluate our proposed scheme on physical programmable switches.

## Data Availability

All data generated or used during the study appear in the submitted paper.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported by the National Key Research and Development Program of China (No. 2019YFB1802503) and the National Natural Science Foundation of China (Nos. 61972026 and 61802014).

## References

[1] T. Barnett, S. Jain, U. Andra, and T. Khurana, "Cisco visual networking index (VNI) complete forecast update, 2017–2022," *APJC Cisco Knowledge Network (CKN) Presentation*, 2018, Available online: https://www.cisco.com/c/dam/m/en_us/network-intelligence/service-provider/digital-transformation/knowledge-network-webinars/pdfs/1213-business-services-ckn.pdf. Accessed 12 October 2022.

[2] B. Nour, S. Mastorakis, R. Ullah, and N. Stergiou, "Information-centric networking in wireless environments: security risks and challenges," *IEEE Wireless Communications*, vol. 28, no. 2, pp. 121–127, 2021.

[3] S. Signorello, R. State, J. Franois, and O. Festor, "NDN.p4: programming information-centric data-planes," *2016 IEEE NetSoft Conference and Workshops (NetSoft)*, pp. 384–389, IEEE Institute of Electrical and Electronics Engineers, Seoul, Korea, 2016.

[4] S. Signorello, *A Multifold Approach to Address the Security Issues of Stateful Forwarding Mechanisms in Information-Centric Networks*, University of Lorraine, Nancy, France, 2018.

[5] X. Guo, N. Liu, X. Hou, S. Gao, and H. Zhou, "An efficient NDN routing mechanism design in P4 environment," in *2021 2nd Information Communication Technologies Conference (ICTC)*, pp. 28–33, IEEE, Nanjing, China, 2021.

[6] S. Hou, Y. Hu, L. Tian, and Z. Dang, "NNFD.P4: NDN in-networking cache implementation scheme with P4," *IEICE Transactions on Information and Systems*, vol. E105.D, no. 4, pp. 820–823, 2022.

[7] X. Jin, X. Li, H. Zhang et al., "NetCache: balancing key-value stores with fast in-network caching," in *Proceedings of the 26th Symposium on Operating Systems Principles*, pp. 121–136, Association for Computing Machinery, New York, NY, United States, 2017.

[8] T. Qu, R. Joshi, M. C. Chan, B. Leong, D. Guo, and Z. Liu, "SQR: in-network packet loss recovery from link failures for highly reliable datacenter networks," in *2019 IEEE 27th International Conference on Network Protocols (ICNP). IEEE*, pp. 1–12, IEEE, Chicago, IL, USA, 2019.

[9] R. Miguel, S. Signorello, and F. M. V. Ramos, "Named data networking with programmable switches," in *2018 IEEE 26th International Conference on Network Protocols (ICNP)*, pp. 400–405, IEEE, Cambridge, UK, 2018.

[10] O. Karrakchou, N. Samaan, and A. Karmouch, "ENDN: an enhanced NDN architecture with a p4-programmabie data plane," in *Proceedings of the 7th ACM Conference on Information-Centric Networking, ser. ICN '20. New York, NY, USA: Association for Computing Machinery*, Association for Computing Machinery, p. 111, 2020, [Online]. Available.

[11] R. Tourani, S. Misra, T. Mick, and G. Panwar, "Security, privacy, and access control in information-centric networking: a survey," *IEEE Communications Surveys Tutorials*, vol. 20, no. 1, pp. 566–600, 2018.

[12] K. Xue, P. He, X. Zhang et al., "A secure, efficient, and accountable edge-based access control framework for information centric networks," *IEEE/ACM Transactions on Networking*, vol. 27, no. 3, pp. 1220–1233, 2019.

[13] Z. Zhang, Y. Yu, S. K. Ramani, A. Afanasyev, and L. Zhang, "NAC: automating access control via named data," in *MILCOM 2018 – 2018 IEEE Military Communications Conference (MILCOM)*, pp. 626–633, IEEE, Los Angeles, CA, USA, 2018.

[14] S. Misra, R. Tourani, F. Natividad, T. Mick, N. E. Majd, and H. Huang, "AccConF: an access control framework for leveraging in-network cached data in the ICN-enabled wireless edge," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 1, pp. 5–17, 2019.

[15] L. Ning-Chun, G. Shuai, H. Xin-Di, and G. Xin-Chang, "A data access control scheme in information-centric mobile ad hoc networks," *Journal of Beijing University of Posts and Telecommunications*, vol. 44, no. 2, pp. 54–60, 2021.

[16] I. Oliveira, E. Neto, R. Immich et al., "Dh-aes-p4: on-premise encryption and in-band key-exchange in p4 fully programmable data planes," in *2021 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, pp. 148–153, IEEE, Heraklion, Greece, 2021.

[17] G. Liu, W. Quan, N. Cheng et al., "Softwarized IoT network immunity against eavesdropping with programmable data planes," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6578–6590, 2021.

[18] T.-T. Lin, S.-C. Tsai, and W.-G. Tzeng, "Efficient encoding and decoding with permutation arrays," in *2008 IEEE International Symposium on Information Theory*, pp. 211–214, IEEE, Toronto, ON, Canada, 2008.

[19] Y.-B. Lin, T.-J. Huang, and S.-C. Tsai, "Enhancing 5g/IoT transport security through content permutation," *IEEE Access*, vol. 7, pp. 94 293–94 299, 2019.

[20] A. Afanasyev, J. Shi, B. Zhang et al., "NFD developers guide," *Dept. Comput. Sci., Univ. California, Los Angeles, Los Angeles, CA, USA, Tech. Rep. NDN-0021*, 2014, https://named-data.net/publications/techreports/ndn-0021-11-nfd-guide/.

[21] P4lang/behavioral-model, 2022, https://github.com/p4lang/behavioral-model.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 6719302, 15 pages
https://doi.org/10.1155/2022/6719302

*Research Article*

# CD-ABSE: Attribute-Based Searchable Encryption Scheme Supporting Cross-Domain Sharing on Blockchain

**Kaiyang Guo**,[1,2] **Yiliang Han**,[1,2] **Riming Wu**,[1,2] **and Kai Liu**[1,2]

[1]*Engineering University of the PAP, Xi'an 710086, China*
[2]*Key Laboratory for Network and Information Security of the PAP, Xi'an 710086, China*

Correspondence should be addressed to Yiliang Han; yilianghan@hotmail.com

Received 4 May 2022; Revised 7 September 2022; Accepted 15 September 2022; Published 6 October 2022

Academic Editor: Haitao Xu



The network security situation is grim, and the problem of "information isolated island" is becoming increasingly prominent. In view of the low efficiency and insufficient security of data cross-domain sharing in the open network environment, a searchable data sharing scheme supporting cross-domain is proposed based on attribute encryption technology. Firstly, different types of nodes on the blockchain are used to realize the data sharing of users in different domains. Secondly, the flexible ciphertext-search function is realized through the search form of keyword strategy. Moreover, the scheme adopts the mode of storage under the chain, which reduces the operation pressure of the blockchain. At the same time, according to the characteristics of the blockchain, the traceability and tamper-proof of the access process can be realized. Finally, the analysis shows that the scheme can resist quantum attack and collusive attack while avoiding complex bilinear operation and meet the security of trapdoor search and indistinguishability under chosen-plaintext attack. Compared with other searchable attribute-based encryption schemes, the scheme has certain advantages in function and performance.

## 1. Introduction

With the increasing data resources in cyberspace, the security and efficiency problems have attracted much attention. How to use information safely and efficiently to create greater value has become one of the urgent issues to be solved in this era. With the increase of the amount of individual data, there are more and more network attacks, and the data security situation is severe, which makes the cross-domain access that is already difficult to maintain permissions and low access efficiency more difficult. The failure to share data safely and efficiently will greatly reduce the value of data, resulting in a waste of resources and restricting development. However, traditional access control models, such as Discretionary Access Control, Mandatory Access Control model, and Role-Based Access Control model, have some limitations on the face of current needs. In order to ensure that the data in cyberspace can be shared more safely and efficiently, more and more scholars begin to study new access control models that are more in line with the actual needs.

Wang et al. proposed a cross-domain access control method for large organizations by applying ABAC model in distributed authoritative domain [1]. Yang and Wang proposed a new cross-domain access control model based on trust measurement [2] that can realize dynamic authorization and fine-grained access in a simple way. Shuang and Chen had built an efficient trusted cross-domain access control system by combining role mapping technology and blockchain [3]. Blockchain is used to record user roles, mapping rules, and access policies and rely on efficient smart contracts to make access decisions; Bai et al. proposed a multidomain access control service for intelligent city service system [4], which transmits data based on attribute encryption and improves the mapping efficiency through the combination of digital attribute table and B + tree. The scheme can also rely on third-party outsourcing to reduce the computational burden. Ullah et al. designed a lightweight provable cross-domain access control scheme based on the wireless body area network on the Internet of Things [5]; the computing and communication costs are reduced under the condition of ensuring security.

As a new functional public key encryption technology, it has unique advantages in data security sharing, the biggest feature of attribute-based encryption is to integrate data confidentiality and access control and determine the object of data sharing through the matching of attributes and policies. The concept of attribute-based encryption was first proposed by Sahai and Waters on the basis of identity-based encryption in 2005 [6]. Later, it is usually divided into ciphertext-policy attribute-based encryption (CP-ABE) and key-policy attribute-based encryption (KP-ABE) [7]. The ciphertext-policy is formulated by the data owner, and it is more flexible in data sharing and more widely used in research and application compared with the key-policy. In 2007, Bethencourt et al. proposed the first CP-ABE scheme [8], but it does not have provable security; in the same year, Cheung et al. proposed the first scheme that can prove security under the standard model [9], but the expression ability of access structure of AND gate is limited; Waters constructed a CP-ABE scheme with flexible strategy based on linear secret sharing scheme (LSSS) in 2011 [10]. After that, many scholars put forward schemes with more perfect functions, but most of these schemes are based on bilinear map, and the complex bilinear pairing operation restricts the efficiency of the scheme. Therefore, some scholars began to try to construct attribute-based encryption schemes based on other mathematical systems. In 2012, Agrawel et al. discussed the possibility of constructing attribute-based encryption on lattice scheme [11]; in 2013, Wang proposed the CP-ABE scheme based on the learning with errors (LWE) problem on the basis of Agrawel's theory [12]. In 2015, Tan and Azmasn proposed the CP-ABE scheme based on the learning with errors over ring (RLWE) problem [13], which is significantly improved in size and efficiency compared with the scheme on the LWE problem. The research of ABE on lattice has been paid more and more attention by scholars, and the traditional problems such as access structure [14], attribute revocation, and key abuse have been deeply discussed.

With the deepening of the research on attribute-based encryption, its potential in data protection and access control has gradually attracted extensive attention in the academic community. Basu and Tripathy improved the efficiency by using CP-ABE scheme based on the security multicast requirements in the Internet of Things (IOT) [15]. In 2019, Yao and Wang protected the security of data exchanged with IOT devices based on ABE and equality testing technology [16]; Challagidad and Birje proposed a multiauthority access control scheme [17], which combined the Role Hierarchy Algorithm with the ABE, and the hierarchical access structure significantly improves the efficiency. Tian et al. applied ABE to blockchain to protect transaction privacy and realize traceability information sharing [18]. Sandoval et al. proposed a data storage method based on ABE in the cloud [19], which supports the sharing and search of encrypted data. Niu et al. used the characteristics of blockchain to improve the security of CP-ABE scheme [20]; Zhang et al. proposed an accountable data sharing model combining blockchain and ABE [21]. Based on the need of medical data, Niu et al. designed a data sharing scheme that can protect users' privacy by using ABE [22]. Kanimozhi and Victoire proposed a scheme for data sharing of the IOT based on attribute-based encryption [23]. By performing clustering and the collected data and then encrypting it in the cloud, the confidentiality and integrity of the data are guaranteed. Li and Tan proposed an electronic certificate sharing scheme based on blockchain and attribute-based searchable encryption to achieve fine-grained access control [24].

*1.1. Security and Function Requirements.* A complete access control system should provide corresponding functions and security services to ensure data sharing among entities.

(1) Fine-grained access control. Users can freely decide who can access the data they own and can also access the data shared by other users as needed

(2) Data security and user privacy protection. Users' data in the process of data sharing should be safe and effective, and their personal identity information should be in a safe state

(3) Security of index and trapdoor. During the search process, the index and trapdoor should be safe and reliable. Attackers cannot obtain more information through the index and trapdoor, nor can they destroy the system through the search process

(4) Tailored forensics. The system shall provide certain evidence collection mechanism to ensure that the transaction has certain integrity and traceability

*1.2. Contribution.* In order to solve the problem of data sharing between different domains, this paper proposes a cross-domain access control scheme, which is based on CP-ABE to ensure data security and fine-grained access control. The cross-domain sharing of data is realized by connecting blockchains of different domains through cross-domain nodes. At the same time, the scheme also supports flexible ciphertext-search function. The main contributions of this paper are listed as follows.

(1) Through the combination of blockchain and CP-ABE, users in the same domain and users in different domains can share data safely

(2) The scheme supports ciphertext-search function before data access. By generating search traps in the form of keyword policy, the search of multiple keywords can be realized while ensuring privacy, which improves the flexibility of search

(3) Using the way of ciphertext off chain storage, only a small part of the data needs to be uploaded to the blockchain, which reduces the calculation and storage pressure of the blockchain. Through encrypted storage, even if there is data leakage, it can ensure the security of information, and according to the characteristics of the blockchain, the traceability and tamper-proof of the access process can be realized

(4) The scheme is constructed based on RLWE, without complex bilinear pairing, and has the characteristics of antiquantum attack

*1.3. Paper Structure.* The remainder of this paper is organized as follows. In Section 2, we review some mathematical knowledge and define the security model. In Section 3, we give the system model, definition of scheme, and construction. The scheme is analyzed in Section 4, mainly including security analysis and performance analysis. Finally, we conclude our paper in Section 5.

## 2. Preliminaries

*2.1. Lattice*

*Definition 1* (lattice). $\Lambda$ is called lattice if there are $m$ linearly independent $n$-dimensional vectors in $\Lambda$, such that any vector in $\Lambda$ is an integer linear combination of $B = \{b_1, b_2, \cdots, b_m\}$, that is, $\Lambda = \Lambda(b_1, b_2, \cdots, b_m) = \{\sum_{i=1}^{m} s_i b_i, s_i \in \mathbb{Z}\}$, $n$ is the dimension of lattice $\Lambda$, $m$ is the rank of lattice $\Lambda$, and $B$ is a set of bases of lattice $\Lambda$.

*Definition 2* (ideal lattice). There is a ring $R = [x]/\langle f \rangle$ and an ideal $I \subseteq R$; a lattice $\Lambda \in Z^n$ is an ideal lattice if $\Lambda$ is associated with $I$.

*Definition 3* (Decision R-LWE$_{d,q,\chi}$ Problem [25]). Given the security parameter $\lambda$, select the integer $d, q$ based on $\lambda$, let $R = Z[x]/f(x)$, where $f(x) = x^d + 1$ and $R_q = R/q$. Given discrete distribution $\chi \subset R_q$ based on $\lambda$, there is an unspecified challenge model $O$ in the Decision R-LWE$_{d,q,\chi}$ Problem, that is, to determine whether the challenge model is a noisy pseudorandom sampler $O_s$ or a real random sampler $O_s{}'$ for random secret key, $K \in R_q$, which perform, respectively, as follows:

$O_s$: outputs $(\omega, v) = (\omega, \omega K + e) \in R_q \times R_q$. The element $\omega$ is uniformly random from $R_q$, where $\omega \longleftarrow R_q$ and the $K \longleftarrow R_q$ fixed for all samples. The element $e \longleftarrow R_q$ is a small error term that generated with a distribution $\chi$.

$O_s{}'$: outputs truly random samples $(\omega, v) \in R_q \times R_q$.

*2.2. Access Control Structure*

*Definition 4* (Monotone Access Structure). Let $U = \{u_1, u_2, \cdots, u_n\}$ be a set of attributes. A collection $D \subseteq U$ is monotone if $\forall B, C : B \in D, B \subseteq C \Rightarrow C \in D$. The sets in $A$ are called as authorized sets, and the sets not in $D$ are called as unauthorized sets.

*Definition 5* (linear secret sharing scheme (LSSS) [13]). The $\prod$ is a secret sharing scheme over a set of attributes $U$ if the following properties are met:

(1) All sharers have a secret sharing vector based on $R_q$

(2) There is a share-generating matrix $F \in R_q^{n \times m}$ for $\prod$, with row labels $\rho(i) \in U, \forall i \in [n]$. Given a column vector, $\vec{v} = (s, r_2, \cdots, r_m)$, where $s \in R_q$ is the secret to be shared and $r_2, \cdots, r_m \longleftarrow R_q$ are randomly chosen. Let $\delta_i = F_i \times v \in R_q, i \in (1, n)$ represent attribute $\rho(i)$, where $\rho(i)$ is a function from $i$ to $U$

Linear secret sharing scheme has linear reconstruction characteristics. Suppose that $\prod$ is an LSSS that represents the access structure $A$. Let $A \in A$ be an authorized set, and $I \subset \{1, \cdots, n\}$, $I = \{i : \rho(i) \in D\}$. There exist constants $\{\omega_i \in R_q\}_{i \in I}$ then $\sum_{i \in I} \delta_i \omega_i = s$ such that of $\delta_i$ are valid shares of a secret $s$ according to $\prod$. Furthermore, these constants $\omega_i$ can be calculated through the share-generating matrix $F$ in polynomial time. For unauthorized sets, it cannot be calculated, that is, any information of secret sharing value cannot be obtained.

## 3. Attribute-Based Searchable Encryption Scheme Supporting Cross-Domain Sharing on Blockchain

*3.1. System Model.* The model in this scheme can be divided into three layers, such as storage layer, blockchain service layer, and application layer from bottom to top. The model is shown in Figure 1.

The storage layer is responsible for providing data storage, which is divided into blockchain data storage and IPFS (Inter Planetary File System) data storage. Blockchain data mainly includes system initialization parameters, relevant information applied by users, indexes, and initial ciphertext, etc., and these data will be stored in the form of transactions; IPFS mainly stores the encrypted data uploaded by users. In the blockchain service layer, it is mainly divided into Unit-chain and Region-chain, in which Unit-chain is mainly responsible for internal data services, including data recording and access services; The Region-chain is mainly responsible for cross domain data services between different units. Based on the weak credit environment, this model is based on the Consortium Blockchain, and only licensed nodes can operate. At the same time, the credit consensus mechanism is adopted, and the nodes with violations will be revoked and removed from the system. The nodes in this model are mainly divided into general nodes and cross domain nodes. General nodes mainly maintain blockchain services within their own units, and cross domain nodes are responsible for connecting blockchains between two different domains, providing cross domain access services, and deploying the authority on cross domain nodes to improve work efficiency and resource utilization. The application layer provides various functional applications.

The proposed system includes five entities: *Authority*, *Data Owner (DO)*, *Inter Planetary File System (IPFS)*, *Data User (DU)*, and *Blockchain*. The *Authority* is deployed on the *Blockchain*. The relationship among the entities is shown in Figure 2.

(1) *Authority*. The authority generates the system's public parameters and master key, manages the users in the system, and constructs the private key for each user according to the user's identity and authority, then the authority generates temporary keys for

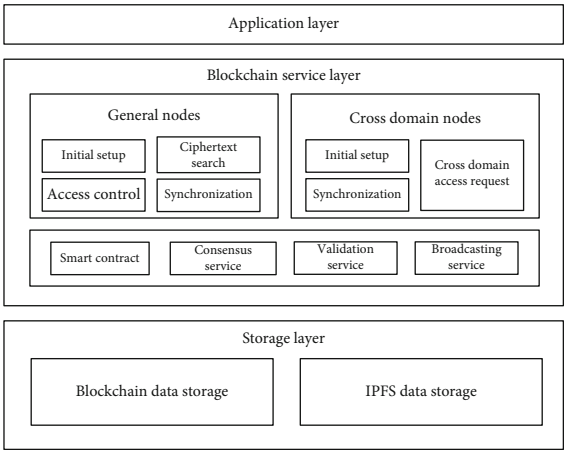



Figure 1: System model.

users in other domains and search traps for users during cross-domain access. We assume that the authority is completely trusted, will faithfully perform various operations, and will not disclose users' personal information. In order to facilitate operation and data processing, we deploy the authority to the cross-domain nodes of the blockchain

(2) *Data Owner (DO)*. The data owner generates keyword index $I_w$ based on data and encrypts data with symmetric key $k$, then uploads encrypted data to the IPFS. After that, DO sets the access policy of the data and encrypts the symmetric key and address, then DO uploads this ciphertext $ct$ and index $I_w$ to blockchain

(3) *Inter Planetary File System (IPFS)*. The IPFS is responsible for storing data and returning an address. IPFS is honest but curious, always correctly implement the requirements put forward by all entities in the scheme, but attempts to decrypt the ciphertext content

(4) *Data User (DU)*. The data user can access data according to their needs. Apply to the authority for a search trapdoor as needed and send it to the blockchain node. After obtaining the returned initial ciphertext, DU decrypts the ciphertext according to private key. After obtaining the address, download the corresponding data from IPFS, then DU can decrypt the data according to the symmetric key

(5) *Blockchain*. The blockchain performs smart contract and runs algorithm, and important events in the access process will form blocks in the form of transactions and be saved in the blockchain. Due to the guarantee of trust proof of work, the node will faithfully perform operations. We assume that the entity is not completely trusted and may try to decipher user's data

*3.2. Overview of the Scheme.* Based on ABE and blockchain, the scheme realizes the data access control of users in the same domain or between different domains and can also provide ciphertext-search function. In order to ensure the traceability and tamper-proof of the search process, important events in the access process will be formed into blocks in the form of transactions and stored in the blockchain. The information contained in the release record can be determined according to the specific situation. If the privacy is strong, it can be released in the form of pseudo-ID or other forms, which is not the focus of the scheme and will not be discussed too much. The access process of the scheme is shown in Figure 3. The specific contents of the scheme are as follows.

(1) Initialize accounts, deploy smart contracts, and initialize systems

(2) The user submits the registration information to authority, which verifies and generates the corresponding private key

(3) DO extracts keywords from the data to be shared and generates an index $I_w$, then encrypts the data through a symmetric algorithm, and uploads the encrypted data to IPFS, then gets address $L$, then encrypts the address and symmetric key $k$ according to own strategy to get the initial ciphertext $ct$, and finally, embeds $ct$ and $I_w$ into a transaction, and publishes it to the blockchain

(4) When DU needs to access the data in their own domain, DU sends an application to their authority, which contains the visitor's information, keyword combination, and signature. The authority first verifies the user's identity. If the user is forged or illegal, it will refuse access; if the identity is valid, the trapdoor $T_{w'}$ is generated through keyword combination and is sent to the node for search

(5) When DU needs to access the data of other domains, they first apply to their authority. After receiving the application, the cross-domain node, as the user's agent, submits a temporary access application to the authority of the target domain. After verification, the authority of the target domain assigns a temporary private key. The private key has time or times limit when used, and then follow step (4)

(6) According to the incentive mechanism, after receiving the search trapdoor $T_{w'}$, the node runs the algorithm for matching search that to get reward. When the keywords match, the node will return the corresponding initial ciphertext $ct$; otherwise, it will return $\perp$

(7) When DUs receive the initial ciphertext $ct$, they decrypt it with their own private key. If their own attributes meet the policies formulated by the DO, it can be decrypted smoothly to obtain the address $L$ and symmetric key $k$. Once decrypted successfully, the user's "wallet" will publish the access record to the blockchain; otherwise, it will return $\perp$

(8) Finally, DU submits the address to IPFS, downloads the corresponding encrypted data, and then, decrypts it with symmetric key $k$ to obtain the data

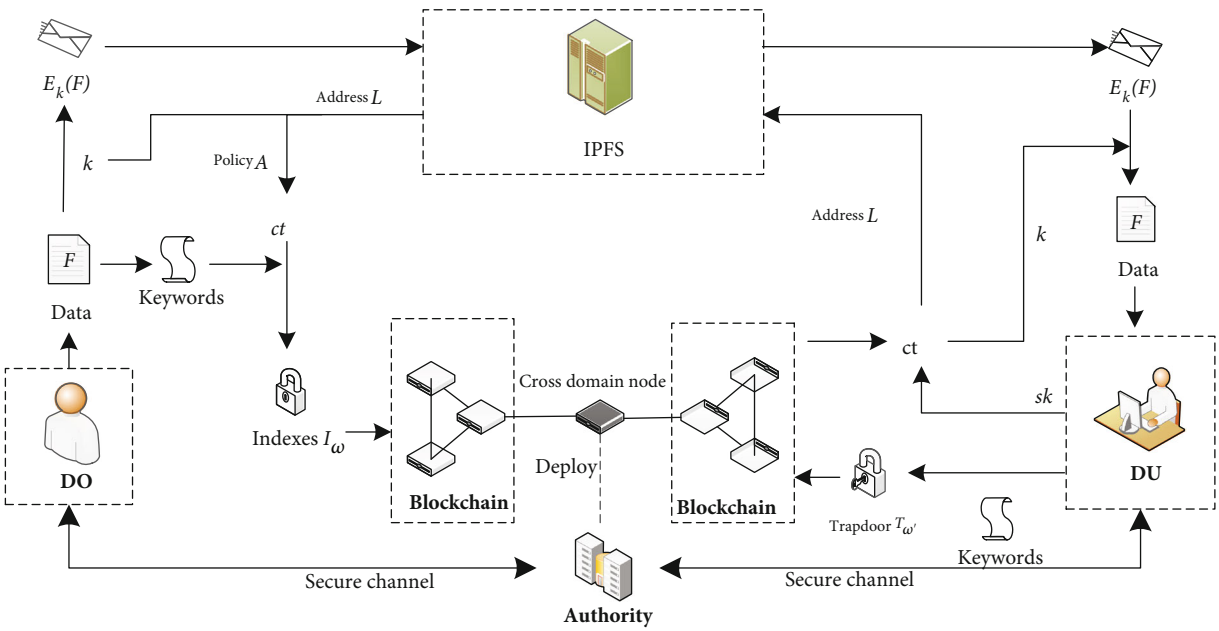



Figure 2: System architecture.

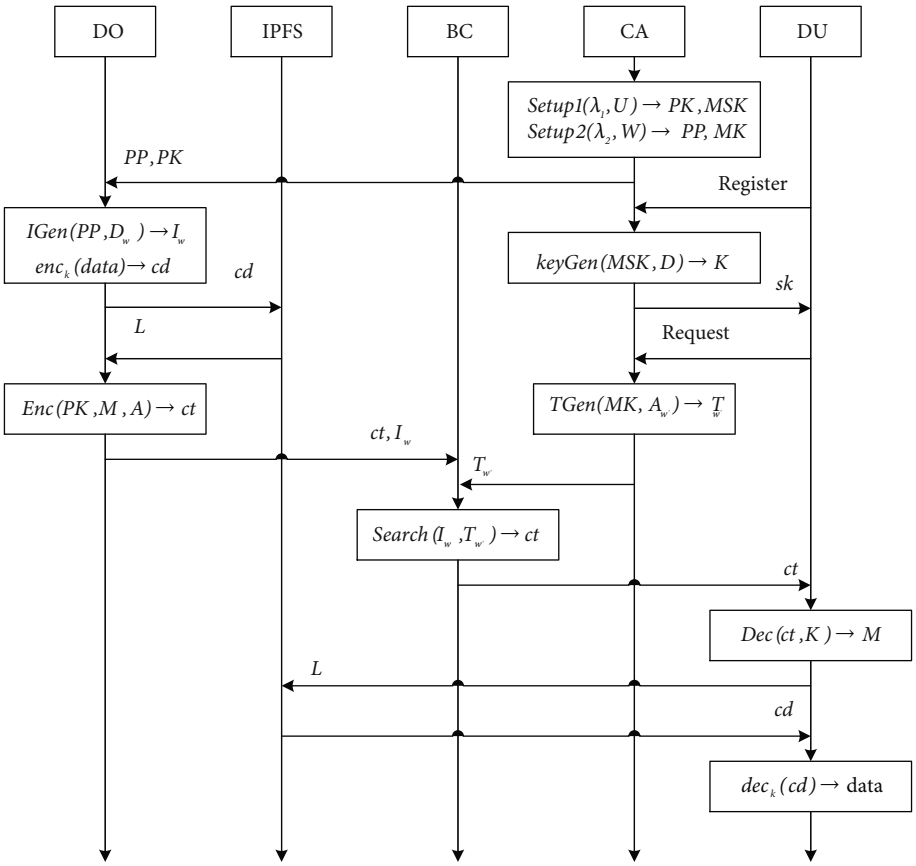



Figure 3: Access flow chart.

The scheme consists of the following eight algorithms.

*Setup1* $(\lambda_1, U) \longrightarrow PK, MSK$. The algorithm is executed by authority. Given the security parameter $\lambda_1$, and the collection of all attributes $U$ in the system, this algorithm outputs public parameters $PK$ and master secret key $MSK$.

*Setup2* $(\lambda_2, W) \longrightarrow PP, MK$. The algorithm is executed by authority. Given the security parameter $\lambda_2$, and the collection of all attributes $W$ in the system, this algorithm outputs public parameters $PP$ and master key $MK$.

$IGen(PP, D_w) \longrightarrow I_w$. The algorithm is executed by DO. Input public parameters $PP$, a set of keywords $D_w$ used to describe data. This algorithm outputs index $I_w$.

$Enc(PK, M, A) \longrightarrow ct$. The algorithm is executed by DO. Input public parameters $PK$, the message $M$ about address and symmetric key $M = k\|L$, and user's access policy $A$. This algorithm outputs the ciphertext $ct$.

$keyGen(MSK, D) \longrightarrow sk$. The algorithm is executed by authority. Input master secret key $MSK$ and user's attribute set $D$. This algorithm outputs the secret key $sk$ for the user.

$TGen(MK, A_{w'}) \longrightarrow T_{w'}$. The algorithm is executed by authority. Input master key $MK$ and user's keyword policy $A_{w'}$. This algorithm outputs a trapdoor $T_{w'}$.

$Search(PP, I_w, T_{w'}) \longrightarrow ct$. The algorithm is executed by a node of blockchain. Input public parameters $PP$, index $I_w$, and a trapdoor $T_{w'}$; if keywords match the corresponding data, the ciphertext $ct$ is returned; otherwise, it return $\perp$.

$Dec(PK, ct, sk) \longrightarrow M$. The algorithm is executed by DU. Input public parameters $PK$, ciphertext $ct$, and user's secret key $sk$. This algorithm outputs $M = k\|L$, then the DU can download the data through the address $L$ and decrypt it with the symmetric key $k$ to obtain the data.

*3.3. Security Model.* It is assumed that the authority is a fully trusted entity. IPFS and blockchain are semitrusted entities. They will faithfully perform operations, but they may try to decipher user data; IPFS and blockchain may collude with attackers. Assuming that the channel between users and authority is a secure channel, consider the following attacker and security models.

(1) The scheme should meet the basic data security requirements and ensure the confidentiality of the data in the sharing process. The attacker 1 mainly focuses on the security problems in the system of ABE and attempts to decrypt the encrypted data

(2) Based on the characteristics of ABE, the scheme should be able to resist collusion attack. We define attacker 2 as malicious legitimate users, who can obtain any number of keys and attempt to collude to expand their decryption ability. It is defined that if the advantage of attacker 2 can be ignored in any polynomial time, the scheme meets the security of anticollusion attack

(3) The scheme should meet the privacy security of the index, and the attacker should not be able to distinguish the index corresponding to different keywords. Define that attacker 3 attempts to obtain information from the index

(4) The scheme should meet the privacy security of the trapdoor, and the attacker should not be able to distinguish the trapdoor corresponding to different keywords. Define that attacker 4 attempts to obtain information from the trapdoor

*Definition 6* (IND-CPA security). The definition is given by describing the game between adversary $\mathscr{A}$ and simulator $\mathscr{B}$. The scheme satisfies the security of chosen-plaintext attack if all polynomial algorithm adversaries' advantage is negligible in the game. The specific process of the game is as follows.

*Initialization.* The adversary $\mathscr{A}$ selects an access structure $A^*$ and sends it to $\mathscr{B}$.

*Setup.* The simulator $\mathscr{B}$ generates public parameters $PK$ and master key $MSK$ and sends $PK$ to $\mathscr{A}$.

*Inquiry Phase 1.* The adversary $\mathscr{A}$ asks the simulator $\mathscr{B}$ for the private key, but $\mathscr{A}$'s attribute set does not meet the access structure. The simulator runs the $KeyGen$ algorithm to generate the private key and send it to $\mathscr{A}$.

*Challenge.* The adversary $\mathscr{A}$ chooses two messages $M_0$, $M_1 \in \{0, 1\}$ and sends them to simulator $\mathscr{B}$, then $\mathscr{B}$ randomly selects $b \in \{0, 1\}$ to calculate the challenge ciphertext and sends it to $\mathscr{A}$.

*Inquiry Phase 2.* $\mathscr{A}$ asks for the key as in phase 1.

*Guess.* Adversary $\mathscr{A}$ outputs his guess $b'$ about $b$. The advantage of $\mathscr{A}$ in this game is defined as $\mathrm{adv}^{\mathscr{A}} = \Pr[b' = b] - (1/2)$.

*Definition 7* (IND-CKA security). The definition is given by describing the game between adversary $\mathscr{A}_3$ and simulator $\mathscr{B}_3$. The scheme satisfies the security of chosen-keyword attack if all polynomial algorithm adversaries' advantage is negligible in the game. The specific process of the game is as follows.

*Initialization.* The adversary $\mathscr{A}_3$ selects $D_{w0}{}^*$ and $D_{w1}{}^*$ as two keywords with the same length and sends them to $\mathscr{B}_3$.

*Setup.* The simulator $\mathscr{B}_3$ generates public parameters $PP$ and master key $MK$ and sends $PP$ to $\mathscr{A}_3$.

*Inquiry Phase 1.* $\mathscr{A}_3$ sends $D_{wi}{}^*$ keywords to $\mathscr{B}_3$, then $\mathscr{B}_3$ runs the $IGen(PP, D_w) \longrightarrow I_w$ algorithm to generate $I_{wi}$ and send it to $\mathscr{A}_3$. Note that the keyword set of the query cannot be the same as the keyword set of the challenge.

*Challenge.* $\mathscr{B}_3$ randomly selects $b \in \{0, 1\}$ to calculate the challenge index $I_{wb}$ and send it to $\mathscr{A}_3$.

*Inquiry Phase 2.* $\mathscr{A}_3$ asks for the index as in phase 1.

*Guess.* Adversary $\mathscr{A}_3$ outputs his guess $b'$ about $b$. The advantage of $\mathscr{A}_3$ in this game is defined as $\mathrm{adv}^{\mathscr{A}_3} = \Pr[b' = b] - (1/2)$.

*Definition 8* (IND-IKGA security). The definition is given by describing the game between adversary $\mathscr{A}_4$ and simulator $\mathscr{B}_4$. The scheme satisfies the security of internal-keyword

guessing attack if all polynomial algorithm adversaries' advantage is negligible in the game. The specific process of the game is as follows.

*Initialization.* The adversary $\mathscr{A}_4$ selects $A_{w0}$ and $A_{w1}$ as two keyword policies with the same length and sends them to $\mathscr{B}_4$.

*Setup.* The simulator $\mathscr{B}_4$ generates public parameters $PP$ and master key $MK$ and sends $PP$ to $\mathscr{A}_4$.

*Inquiry Phase 1.* $\mathscr{A}_4$ sends $A_{wi}{}^*$ keyword policy to $\mathscr{B}_4$, then $\mathscr{B}_4$ runs the $TGen(MK, A_{w'}) \longrightarrow T_{w'}$ algorithm to generate $T_{wi}$ and send it to $\mathscr{A}_4$. Note that the keyword set of the query cannot be the same as the keyword set of the challenge.

*Challenge.* $\mathscr{B}_4$ randomly selects $b \in \{0, 1\}$ to calculate the challenge keyword-policy $A_{wb}$ and send it to $\mathscr{A}_4$.

*Inquiry Phase 2.* $\mathscr{A}_4$ asks for the trapdoor as in phase 1.

*Guess.* Adversary $\mathscr{A}_4$ outputs his guess $b'$ about $b$. The advantage of $\mathscr{A}_4$ in this game is defined as $\mathrm{adv}^{\mathscr{A}_4} = \Pr[b' = b] - (1/2)$.

### 3.4. Construction of the CD-ABSE Scheme

*3.4.1. Initialization Phase.* This phase mainly includes the initialization of the authority and the blockchain system, in which the blockchain system completes the setting of the corresponding accounts and nodes, etc. The initialization of the authority mainly includes the following two algorithms.

$Setup1(\lambda_1, U) \longrightarrow PK, MSK$. Given the security parameter $\lambda_1$ and the collection of all attributes $U$ in the system, randomly select a large prime number $q_1 = 1 \bmod (2\lambda_1)$ and a small positive integer $p_1$, where $p_1 \ll q_1$ and gcd $(p_1, q_1) = 1$. Let $f(x) = (x^d + 1)$, where $d$ is a power of 2. Let $R_{q_1} = Z_{q_1}[x]/<f(x)>$ be the ring of integer polynomials modulo both $f(x)$ and $q_1$. Let $\chi_1 = \chi_1(\lambda)$ be an error distribution over $R_{q_1}$. Select a uniformly random $SK_0 \longleftarrow R_{q_1}$ and random element $a_1 \longleftarrow R_{q_1}$, then choose a small noise term $e_0 \longleftarrow \chi_1$. Compute $PK_0 = aSK_0 + pe_0 \in R_{q_1}$. Next, select a pair of uniformly random $(SK_i, SK_i{}^{-1}) \longleftarrow R_{q_1}$ for each attribute in $U$, where $SK_i{}^{-1}$ is the inverse of $SK_i$ in $R_{q_1}$, and select a small noise term $e_i \longleftarrow \chi_1$, then compute $PK_i = SK_i + pe_i \in R_{q_1}$. Lastly, output the public parameters $PK = \{a, PK_0, \{PK_i\}_{i=1}^n\}$ and the master secret key $MSK = \{SK_0, \{SK_i\}_{i=1}^n, \{SK_i{}^{-1}\}_{i=1}^n\}$.

$Setup2(\lambda_2, W) \longrightarrow PP, MK$. Given the security parameter $\lambda_2$, and the collection of all keywords $W$ in the system, randomly select a large prime number $q_2 = 1 \bmod (2\lambda_2)$ and a small positive integer $p_2$, where $p_2 \ll q_2$ and gcd $(p_2, q_2) = 1$. Let $f(x) = (x^d + 1)$, where $d$ is a power of 2. Let $R_{q_2} = Z_{q_2}[x]/<f(x)>$ be the ring of integer polynomials modulo both $f(x)$ and $q_2$. Let $\chi_2 = \chi_2(\lambda)$ be an error distribution over $R_{q_2}$. Select a uniformly random $pk_0 \longleftarrow R_{q_2}$ and random element $a_2 \longleftarrow R_{q_2}$, then choose a small noise term $e\_0 \longleftarrow \chi_2$. Compute $sk_0 = a_2 pk_0 + pe\_0 \in R_{q_2}$. Next, select a pair of uniformly random $(pk_i, pk_i{}^{-1}) \longleftarrow R_{q_2}$ for each keyword in $W$, where $pk_i{}^{-1}$ is the inverse of $pk_i$ in $R_{q_2}$, and select a small noise term $e\_i \longleftarrow \chi_2$, then compute $sk_i = pk_i + pe\_i \in R_{q_2}$. Lastly, output the public parameters $PP = \{pk_0, \{pk_i\}_{i=1}^n, \{pk_i{}^{-1}\}_{i=1}^n\}$ and the master secret key $MK = \{a_2, sk_0, \{sk_i\}_{i=1}^n\}$.

*3.4.2. Registration Phase.* This phase mainly refers to that the user submits a registration application to the authority, and the authority runs the following algorithm to generate a key for the user.

$keyGen(MSK, D) \longrightarrow sk$. Input master key $MSK$, user's attribute set $D$, then choose small noise term $e'', e_i{}'' \longleftarrow \chi_1$, and select a pair of uniformly random $(t, t^{-1}) \longleftarrow R_{q_1}$ for each attribute in $D$. Compute $K_0 = SK_0 t^{-1} + pe'' \in R_q$, $K_i = SK_i{}^{-1} t + pe_i{}'' \in R_q, \forall i \in D$; output the secret key $sk = (K_0, K_i)$.

*3.4.3. Data Preparation Phase.* This phase mainly refers to the operation when the data owner shares the data, including symmetrically encrypting the data, sending the encrypted data to IPFS and obtaining the address, and then, encrypting the address and the symmetric key to obtain the ciphertext. In addition, the user also needs to generate a ciphertext index for this data. The algorithm for index generation and encryption is as follows:

$IGen(PP, D_w) \longrightarrow I_w$. Input public parameters $PP$ and a keyword set $D_w$ of data. Select a pair of uniformly random $(t', t'^{-1}) \longleftarrow R_{q_2}$, and choose small noise term $e\_'', e\_i{}'' \longleftarrow \chi_2$ for each keyword in $D_w$. Compute $I_0 = pk_0 t^{-1} + pe\_'' \in R_{q_2}$, $I_i = pk_i{}^{-1} t + pe\_i{}'' \in R_{q_2}, \forall i \in D_w$; output an index $I_w = (I_0, I_i)$.

$Enc(PK, M, A) \longrightarrow ct$. Input public parameters $PK$, the message $M \in \{0, 1\}^n$ about $k \| L$, and set access policy $A = (F, \rho)$, $F \in R_{q_1}^{n \times m}$ with row labels $\rho(j) \in H$, $\forall j \in [n]$, $H \in A$. Generate a vector $v = (s_1, r_2, \cdots, r_m)$, where $r_2, \cdots, r_m \longleftarrow R_{q_1}$, and $s_1 \in R_{q_1}$ is the secret to be shared. $\delta_i = F_i \times v \in R_{q_1}, i \in (1, n)$, where $F_i$ is the vector corresponding to $i$th row of $F$. Then, choose a uniformly random element $r_1 \longleftarrow R_{q_1}$ and noise terms $e', e_i{}' \longleftarrow \chi_1$; compute $c_0 = PK_0 r_1 s_1 + M + p \cdot e' \in R_{q_1}$, $c_i = ar_1 PK_i \delta_i + pe_i{}' \in R_{q_1}$, and output $ct = (c_0, c_i)$.

After completing the above steps, DO embeds the ciphertext $ct$ and index $I_w$ into the transaction $T_X$ and signs it to $T_Y$, then broadcasts $T_X$ to the whole blockchain. After the transaction is verified, it is recorded on the blockchain by the miner. The data structure is as shown in Table 1.

*3.4.4. Access Preparation Phase.* The user sends the data keywords to be accessed to the authority, and the authority executes the following algorithm to generate a search trapdoor for the user.

$TGen(MK, A_{w'}) \longrightarrow T_{w'}$. Input master key $MK$, a keyword set $D_w$ of data, and set keyword policy $A_{w'} = (F_w, \rho)$, $F_w \in R_{q_2}^{n \times m}$ with row labels $\rho(j) \in H, \forall j \in [n], H \in A_{w'}$. Generate a vector $v = (s_2, r_2, \cdots, r_m)$, where $r_2, \cdots, r_m \longleftarrow R_{q_2}$, and $s_2 \in R_{q_2}$ is the secret to be shared. $\delta_i{}' = F_{wi} \times v \in R_{q_2}, i \in (1,$

Table 1: Blockchain data structure in data preparation phase.

| Identification | From | To | Action | Timestamp | Signature | Transaction |
|---|---|---|---|---|---|---|
| ID_1 | DO | BC | Publish | Timestamp1 | Sig1 | $T_X, T_Y$ |

$n$), where $F_{wi}$ is the vector corresponding to $i$th row of $F_w$. Then, choose a uniformly random element $r_1{}' \longleftarrow R_{q_2}$, and noise terms $e\_{}'.e\_{i}{}' \longleftarrow \chi_2$; compute $T_0 = sk_0 r_2 s_2 + pe\_{}' \in R_{q_2}$, $T_i = a_2 sk_i r_2 \delta_i{}' + pe\_{i}{}' \in R_{q_2}$, and output trapdoor $T_{w'} = (T_0, T_i)$.

*3.4.5. Search Phase.* The search phase mainly involves two parts. The first is that DU embeds trapdoor $T_{w'}$ into $T_X$, then publishes it to the smart contract address of the blockchain, and then invokes the search contract for calculation and retrieval. After the search is completed, the blockchain returns the data to DU through the user address. The two data structures are shown in Table 2.

$Search(PP, I_w, T_{w'}) \longrightarrow ct$. Input public parameters $PP$, index $I_w$, and trapdoor $T_{w'}$. If the set of keyword meets the keyword policy $A_{w'}$ and $I \subset \{1,\cdots,n\}$, $I = \{i : \rho(i) \in A_{w'}\}$, compute a set of constants $\{\omega_i \in R_{q_2}\}_{i\in I}$ with a linear reconstruction algorithm of LSSS, then $\sum_{i\in I}\delta_i{}'\omega_i = s_2$, and compute $J = T_0 - I_0\sum_{i\in I} I_i\omega_i T_i \bmod p$; if $J = 0$, the search is successful, and $ct$ is returned; otherwise, it return $\perp$. The correctness of the successful search of the scheme is explained as follows.

$$\begin{aligned}
J' &= T_0 - I_0\sum_{i\in I} I_i\omega_i T_i = T_0 - I_0\sum_{i\in I} I_i\omega_i\left(a_2 sk_i r_2\delta_i{}' + pe_{i}{}'\right)\\
&= T_0 - I_0 a_2 r_2 s_2\sum_{i\in I} sk_i I_i - I_0 p\sum_{i\in I}\left(e_{i}{}'\omega_i I_i\right)\\
&= T_0 - I_0 a_2 r_2 s_2\sum_{i\in I}(pk_i + pe_{i})\left(pk_i{}^{-1}t + pe_{i}{}''\right)\\
&\quad - I_0 p\sum_{i\in I}\left(e_{i}{}'\omega_i I_i\right) = T_0 - I_0 a_2 r_2 s_2\sum_{i\in I}\\
&\quad\cdot\left(t + pe_{i}pk_i{}^{-1}t + pk_i pe_{i}{}'' + p^2 e_{i}e_{i}{}''\right)\\
&\quad - I_0 p\sum_{i\in I}\left(e_{i}{}'\omega_i I_i\right) = T_0 - I_0 a_2 r_2 s_2 t\\
&\quad - pI_0 a_2 r_2 s_2\sum_{i\in I}\left(e_{i}pk_i{}^{-1}t + k_i pe_{i}{}'' + pe_{i}e_{i}{}''\right)\\
&\quad - I_0 p\sum_{i\in I}\left(e_{i}{}'\omega_i I_i\right) = sk_0 r_2 s_2 + pe_{}'\\
&\quad - \left(pk_0 t^{-1} + pe_{}''\right)a_2 r_2 s_2 t - pT_0 a_2 r_2 s_2\sum_{i\in I}\\
&\quad\cdot\left(e_{i}pk_i{}^{-1}t + k_i pe_{i}{}'' + pe_{i}e_{i}{}''\right) - T_0 p\sum_{i\in I}\left(e_{i}{}'\omega_i I_i\right)\\
&= pe_{0} r_2 s_2 + pe_{}' - pe_{}'' a_2 r_2 s_2 t - pT_0 a_2 r_2 s_2\sum_{i\in I}\\
&\quad\cdot\left(e_{i}pk_i{}^{-1}t + k_i pe_{i}{}'' + pe_{i}e_{i}{}''\right) - T_0 p\sum_{i\in I}\left(e_{i}{}'\omega_i I_i\right).
\end{aligned} \tag{1}$$

If the conditions are met, then $J = J' \bmod p = 0$. Otherwise, there will be $\sum_{i\in I}\delta_i{}'\omega_i \neq s_2$ and $J = J' \bmod p \neq 0$; the access will be terminated.

The above algorithm will be executed in the smart contract, and the design of smart contract is shown in Table 3.

*3.4.6. Decryption Phase.* After receiving the ciphertext, the user decrypts it according to his own key. The decryption algorithm is as follows.

$Dec(PK, ct, sk) \longrightarrow M$. Input public parameters $PK$, ciphertext $ct$, and user's secret key $sk$. If the DU meets the access control policy $A$, $I \subset \{1,\cdots,n\}$, $I = \{i : \rho(i) \in A\}$, compute a set of constants $\{\omega_i \in R_{q_1}\}_{i\in I}$ with a linear reconstruction algorithm of LSSS, then $\sum_{i\in I}\delta_i \cdot \omega_i = s_1$; compute $M' = C_0 - K_0\sum_{i\in I} C_i\omega_i K_i$, $M = M' \bmod p$; the DU can download the data through the address $L$ and decrypt it with the symmetric key $k$ to obtain data.

The correctness of the successful decryption of the scheme is explained as follows.

$$\begin{aligned}
M' &= C_0 - K_0\sum_{i\in I} C_i\omega_i K_i = C_0 - K_0\sum_{i\in I}\left(arPK_i\delta_i + pe_i{}'\right)\omega_i K_i\\
&= C_0 - K_0 ar_1 s\sum_{i\in I}(PK_i\cdot K_i) - K_0 p\sum_{i\in I}\left(e_i{}'\omega_i K_i\right)\\
&= C_0 - K_0 ar_1 s\sum_{i\in I}\left((SK_i + pe_i)\left(SK_i{}^{-1}t + pe_i{}''\right)\right)\\
&\quad - K_0 p\sum_{i\in I}\left(e_i{}'\omega_i K_i\right) = C_0 - K_0 ar_1 s_1\sum_{i\in I}\\
&\quad\cdot\left(t + SK_i pe_i{}'' + pe_i SK_i{}^{-1}t + p^2 e_i e_i{}''\right)\\
&\quad - K_0 p\sum_{i\in I}\left(e_i{}'\omega_i K_i\right) = C_0 - K_0 ar_1 s_1 t - K_0 ar_1 s_1 p\sum_{i\in I}\\
&\quad\cdot\left(SK_i e_i{}'' + e_i SK_i{}^{-1}t + pe_i e_i{}''\right) - K_0 p\sum_{i\in I}\left(e_i{}'\omega_i K_i\right)\\
&= PK_0 r_1 s_1 + M + pe' - K_0 ar_1 s_1 t - K_0 ar_1 s_1 p\sum_{i\in I}\\
&\quad\cdot\left(SK_i e_i{}'' + e_i SK_i{}^{-1}t + pe_i e_i{}''\right) - K_0 p\sum_{i\in I}\left(e_i{}'\omega_i K_i\right)\\
&= (aSK_0 + pe_0) r_1 s_1 + M + pe' - \left(SK_0 t^{-1} + pe''\right)ar_1 s_1 t\\
&\quad - K_0 ar_1 s_1 p\sum_{i\in I}\left(SK_i e_i{}'' + e_i SK_i{}^{-1}t + pe_i e_i{}''\right)\\
&\quad - K_0 p\sum_{i\in I}\left(e_i{}'\omega_i K_i\right) = pe_0 r_1 s_1 + M + pe' - pe'' ar_1 s_1 t\\
&\quad - K_0 ar_1 s_1 p\sum_{i\in I}\left(SK_i e_i{}'' + e_i SK_i{}^{-1}t + pe_i e_i{}''\right)\\
&\quad - K_0 p\sum_{i\in I}\left(e_i{}'\omega_i K_i\right).
\end{aligned} \tag{2}$$

Table 2: Blockchain data structure in search phase.

| Identification | From | To | Action | Timestamp | Signature | Transaction |
|---|---|---|---|---|---|---|
| ID_2 | DU | BC | Calculation | Timestamp2 | Sig2 | $T_X, T_Y$ |
| ID_3 | BC | DU | Publish | Timestamp3 | Sig3 | $T_X, T_Y$ |

Table 3: Implementation process of contract.

| Search contract | |
|---|---|
| Input | User ID, trapdoor, DU address |
| Output | Ciphertext set ($ct$.row) or $\perp$ |
| 1. | While (true) do |
| 2. | Calculate $J$ by executing .. |
| 3. | If ($J == 0$) then |
| 4. | Add $ct$ to $ct$.row |
| 5. | Else |
| 6. | Return ($\perp$) |
| 7. | End if |
| 8. | Continue |
| 9. | End while |

Then, $M = M' \bmod p$, and in order to ensure the correctness of the scheme, the noise term in the scheme must be small enough compared to the ratio of $q$ to $p$.

After successful decryption, the user obtains the data address and the symmetric key and decrypts the data with the symmetric key after obtaining the data from IPFS to obtain the original data.

## 4. Analysis

*4.1. Security Analysis.* This section will discuss the security of the scheme from four aspects according to the security definition in Section 3.3.

(1) Analysis of IND-CPA security

**Theorem 9.** *If there exists a Probabilistic Polynomial Time (PPT) algorithm adversary $\mathscr{A}$, with the advantage $\varepsilon$ to win the game in Definition 6, then there exists a PPT simulator $\mathscr{B}$ which can decide Decision R-LWE$_{d,q,\chi}$ Problem with advantage $\varepsilon/2$.*

*Proof.* The Decision R-LWE$_{d,q,\chi}$ Problem is to determine whether the oracle $O$ is a noisy pseudorandom $O_s$ or a truly random $O_s{}'$, then the simulator $\mathscr{B}$ differentiates $O$ by adversary $\mathscr{A}$. First, $\mathscr{B}$ queries the oracle and receives $(t+1)$ samples $(\omega_k, \upsilon_k) \in R_q \times R_q$, where $k \in \{0, 1, 2, \cdots, t\}$, then proceed as follows. □

*Initialization phase.* Given a set of attributes $U$, the adversary $\mathscr{A}$ selects an access structure $A^*$ that wishes to be challenged and sends it to $\mathscr{B}$.

*Setup.* $\mathscr{B}$ runs $Setup(\lambda, U) \longrightarrow PP, MSK$, let $PK_0 = p\omega_0 \in R_q$, and select a pair of uniformly random $(SK_i, SK_i{}^{-1}) \longleftarrow R_q$ for each attribute in $U$. Let $PK_i = p\omega_i \in R_q$ if $i \in A^*$; otherwise, let $PK_i = SK_i + pe_i \in R_q$. Then, $\mathscr{B}$ sends $PK = \{a, PK_0, \{PK_i\}_{i=1}^n\}$ to $\mathscr{A}$.

*Inquiry Phase 1.* $\mathscr{A}$ sends private key queries for $D^* = \{D_1{}^*, D_2{}^*, \cdots, D_j{}^*\}$, where $D^*$ does not meet the access policy $A^*$. $\mathscr{B}$ runs $KeyGen$, computes $K_0 = SK_0 t^{-1} + pe'' \in R_q$, $K_i = SK_i{}^{-1} t + pe_i{}'' \in R_q, \forall i \in D^*$, and sends $K = (K_0, K_i)$ to $\mathscr{A}$.

*Challenge.* $\mathscr{A}$ chooses two messages $M_0, M_1 \in \{0, 1\}$ and sends them to simulator $\mathscr{B}$, then $\mathscr{B}$ randomly selects $b \in \{0, 1\}$, if $b = 0$, $\mathscr{B}$ randomly chooses $x \longleftarrow R_q$ and lets $C_0 = px_0 \in R_q$, $C_j = px_j \in R_q$; if $b = 1$, let $C_0 = p\nu_0 + M \in R_q$, $C_j = p\nu_j \in R_q$ for $j \in A^*$.

*Inquiry Phase 2.* $\mathscr{A}$ asks for the key as in phase 1.

*Guess.* Adversary $\mathscr{A}$ outputs his guess $b'$ about $b$ to $\mathscr{B}$. If $b' = b$, output $O' = O_s$; otherwise, output $O' = O_s{}'$. The advantage of $\mathscr{A}$ in this game is defined as $\text{adv}^{\mathscr{A}} = \Pr[b' = b] - (1/2)$, so the oracle $O$ has the following two cases.

$O$ is a noisy pseudorandom $O_s$. The advantage of $\mathscr{A}$ is $\varepsilon$, then $|\Pr[b' = b | O = O_s]| = (1/2) + \varepsilon$ and $|\Pr[O' = O | O = O_s]| = (1/2) + \varepsilon$.

$O$ is a truly random $O_s{}'$. $\mathscr{A}$ has no advantage $\varepsilon$ and unable to get information about $b$, then $|\Pr[b' \neq b | O = O_s{}']| = (1/2)$ and $|\Pr[O' = O | O = O_s{}']| = (1/2)$.

Then, the advantage of simulator $\mathscr{B}$ is as follows.

$$\frac{1}{2}\left|\Pr\left[O' = O | O = O_s\right]\right| + \frac{1}{2}\left|\Pr\left[O' = O | O = O_s{}'\right]\right| - \frac{1}{2} = \frac{1}{2}\left(\frac{1}{2} + \varepsilon\right) + \frac{1}{2}\left(\frac{1}{2}\right) - \frac{1}{2} = \frac{\varepsilon}{2}. \tag{3}$$

Hence, Theorem 9 is proved, and this means that the scheme meets IND-CPA security.

(2) Analysis of anticollusion attack security

The private key generated by the authority to the user contains the randomly selected reciprocal element $(t, t^{-1}) \longleftarrow R_{q_1}$, which ensures the uniqueness of the key. At the same time, from the assumption of learning with error, it is difficult for malicious users to restore effective parameter information from their own key. Even if the attributes of colluding users are combined to contain the attributes of the target they want to attack, it is difficult to generate an effective new private key by effective means.

(3) Analysis of IND-CKA security

**Theorem 10.** *If there exists a Probabilistic Polynomial Time (PPT) algorithm adversary $\mathscr{A}_3$, with the advantage $\varepsilon$ to win the game in Definition 7, then there exists a PPT simulator $\mathscr{B}_3$ which can decide Decision R-LWE$_{d,q,\chi}$ Problem with advantage $\varepsilon/2$.*

*Proof.* The Decision R-LWE$_{d,q,\chi}$ Problem is to determine whether the oracle $O$ is a noisy pseudorandom $O_s$ or a truly random $O_s{}'$, then the simulator $\mathscr{B}_3$ differentiates $O$ by adversary $\mathscr{A}_3$. First, $\mathscr{B}_3$ queries the oracle and receives $(t+1)$ samples $(\omega_k, \upsilon_k) \in R_q \times R_q$, where $k \in \{0, 1, 2, \cdots, t\}$, then proceed as follows. □

*Initialization phase.* Given a set of keywords $W$, the adversary $\mathscr{A}_3$ selects $D_{w0}{}^*$ and $D_{w1}{}^*$ as two keywords with the same length that wishes to be challenged and sends them to $\mathscr{B}_3$.

*Setup.* $\mathscr{B}_3$ runs $Setup(\lambda_2, W) \longrightarrow PP, MK$, let $sk_0 = p\omega_0 \in R_{q_2}$, and select a pair of uniformly random $(pk_i, pk_i{}^{-1}) \longleftarrow R_{q_2}$ for each keyword in $W$. Let $sk_i = pk_i + pe\__i \in R_{q_2}$. Then, $\mathscr{B}_3$ sends $PP = \{pk_0, \{pk_i\}_{i=1}^n, \{pk_i{}^{-1}\}_{i=1}^n\}$ to $\mathscr{A}_3$.

*Inquiry Phase 1.* $\mathscr{A}_3$ sends index queries for $D_{wi}{}^*$, where the keyword set of the query cannot be the same as the keyword set of the challenge. $\mathscr{B}_3$ runs $IGen(PP, D_{wi}) \longrightarrow I_{wi}$, computes $I_0 = pk_0 t^{-1} + pe\_'' \in R_{q_2}$, $I_i = pk_i{}^{-1} t + pe\__i{}'' \in R_{q_2}$, $\forall i \in D_{wi}{}^*$, and sends $I_{wi} = (I_0, I_i)$ to $\mathscr{A}_3$.

*Challenge.* $\mathscr{B}_3$ randomly selects $b \in \{0, 1\}$; if $b = 0$, $\mathscr{B}_3$ randomly chooses $x \longleftarrow R_q$ and lets $I_0 = px_0 \in R_q$, $I_i = px_i \in R_q$; if $b = 1$, let $I_0 = pk_0 t^{-1} + pe\_'' \in R_{q_2}$, and $I_i = pk_i{}^{-1} t + pe\__i{}'' \in R_{q_2}$, then send $I_{wb} = (I_0, I_i)$ to $\mathscr{A}_3$.

*Inquiry Phase 2.* $\mathscr{A}_3$ asks for the index as in phase 1.

*Guess.* Adversary $\mathscr{A}_3$ outputs his guess $b'$ about $b$ to $\mathscr{B}_3$. If $b' = b$, output $O' = O_s$; otherwise, output $O' = O_s{}'$. The advantage of $\mathscr{A}_3$ in this game is defined as $\text{adv}^{\mathscr{A}} = \Pr[b' = b] - (1/2)$, so the oracle $O$ has the following two cases.

$O$ is a noisy pseudorandom $O_s$. The advantage of $\mathscr{A}_3$ is $\varepsilon$, then $|\Pr[b' = b | O = O_s]| = (1/2) + \varepsilon$, and $|\Pr[O' = O | O = O_s]| = (1/2) + \varepsilon$.

$O$ is a truly random $O_s{}'$. $\mathscr{A}_3$ has no advantage $\varepsilon$ and unable to get information about $b$, then $|\Pr[b' \neq b | O = O_s{}']| = (1/2)$, and $|\Pr[O' = O | O = O_s{}']| = (1/2)$.

Then, the advantage of simulator $\mathscr{B}_3$ is as follows.

$$\frac{1}{2}\left|\Pr\left[O' = O | O = O_s\right]\right| + \frac{1}{2}\left|\Pr\left[O' = O | O = O_s{}'\right]\right| - \frac{1}{2} = \frac{1}{2}\left(\frac{1}{2} + \varepsilon\right) + \frac{1}{2}\left(\frac{1}{2}\right) - \frac{1}{2} = \frac{\varepsilon}{2}. \tag{4}$$

Hence, Theorem 10 is proved, and this means that the scheme meets IND-CKA security.

(4) Analysis of IND-IKGA security

**Theorem 11.** *If there exists a Probabilistic Polynomial Time (PPT) algorithm adversary $\mathscr{A}_4$, with the advantage $\varepsilon$ to win the game in Definition 8, then there exists a PPT simulator $\mathscr{B}_4$ which can decide Decision R-LWE$_{d,q,\chi}$ Problem with advantage $\varepsilon/2$.*

*Proof.* The Decision R-LWE$_{d,q,\chi}$ Problem is to determine whether the oracle $O$ is a noisy pseudorandom $O_s$ or a truly random $O_s{}'$, then the simulator $\mathscr{B}_4$ differentiate $O$ by adversary $\mathscr{A}_4$. First, $\mathscr{B}_4$ queries the oracle and receives $(t+1)$ samples $(\omega_k, \upsilon_k) \in R_q \times R_q$, where $k \in \{0, 1, 2, \cdots, t\}$, then proceed as follows. □

*Initialization phase.* Given a set of keywords $W$, the adversary $\mathscr{A}_4$ selects $A_{w0}$ and $A_{w1}$ as two keyword policies with the same length that wishes to be challenged and sends them to $\mathscr{B}_3$.

*Setup.* $\mathscr{B}_4$ runs $Setup(\lambda_2, W) \longrightarrow PP, MK$, let $sk_0 = p\omega_0 \in R_{q_2}$, and select a pair of uniformly random $(pk_i, pk_i{}^{-1}) \longleftarrow R_{q_2}$ for each keyword in $W$. Let $sk_i = pk_i + pe\__i \in R_{q_2}$. Then, $\mathscr{B}_4$ sends $PP = \{pk_0, \{pk_i\}_{i=1}^n, \{pk_i{}^{-1}\}_{i=1}^n\}$ to $\mathscr{A}_4$.

*Inquiry Phase 1.* $\mathscr{A}_4$ sends trapdoor queries for $A_{wi}{}^*$, where the keyword set of the query cannot be the same as the keyword set of the challenge. $\mathscr{B}_4$ runs $TGen(MK, A_{w'}) \longrightarrow T_{w'}$, set keyword policy $A_{wi}{}^* = (F_w, \rho)$, $F_w \in R_{q_2}^{n \times m}$ with row labels $\rho(j) \in H$, $\forall j \in [n]$, $H \in A_{w'}$. Generate a vector $v = (s_2, r_2, \cdots, r_m)$, where $r_2, \cdots, r_m \longleftarrow R_{q_2}$, and $s_2 \in R_{q_2}$ is the secret to be shared. $\delta_i{}' = F_{wi} \times v \in R_{q_2}$, $i \in (1, n)$, where $F_{wi}$ is the vector corresponding to $i$th row of $F_w$. Then, choose a uniformly random element $r_1{}' \longleftarrow R_{q_2}$, and noise terms $e\_{}'.e\__i{}' \longleftarrow \chi_2$ computes $T_0 = sk_0 r_2 s_2 + pe\_{}' \in R_{q_2}$, $T_i = a_2 sk_i r_2 \delta_i{}' + pe\__i{}' \in R_{q_2}$, and sends $T_{wi} = (T_0, T_i)$ to $\mathscr{A}_4$.

*Challenge.* $\mathscr{B}_4$ randomly selects $b \in \{0, 1\}$; if $b = 0$, $\mathscr{B}_4$ randomly chooses $x \longleftarrow R_q$ and lets $T_0 = px_0 \in R_q$, $T_i = px_i \in R_q$; if $b = 1$, let $I_0 = pk_0 t^{-1} + pe\_'' \in R_{q_2}$, $I_i = pk_i{}^{-1} t + pe\__i{}'' \in R_{q_2}$, then send $T_{wb} = (T_0, T_i)$ to $\mathscr{A}_4$.

*Inquiry Phase 2.* $\mathscr{A}_4$ asks for the trapdoor as in phase 1.

*Guess.* Adversary $\mathscr{A}_4$ outputs his guess $b'$ about $b$ to $\mathscr{B}_4$. If $b' = b$, output $O' = O_s$; otherwise, output $O' = O_s{}'$. The advantage of $\mathscr{A}_4$ in this game is defined as $\text{adv}^{\mathscr{A}} = \Pr[b' = b] - (1/2)$, so the oracle $O$ has the following two cases.

$O$ is a noisy pseudorandom $O_s$. The advantage of $\mathscr{A}_4$ is $\varepsilon$, then $|\Pr[b' = b | O = O_s]| = (1/2) + \varepsilon$, and $|\Pr[O' = O | O = O_s]| = (1/2) + \varepsilon$.

$O$ is a truly random $O_s{}'$. $\mathscr{A}_4$ has no advantage $\varepsilon$ and unable to get information about $b$, then $|\Pr[b' \neq b | O = O_s{}']| = (1/2)$, and $|\Pr[O' = O | O = O_s{}']| = (1/2)$.

Then, the advantage of simulator $\mathscr{B}_4$ is as follows.

$$\frac{1}{2}\left|\Pr\left[O' = O | O = O_s\right]\right| + \frac{1}{2}\left|\Pr\left[O' = O | O = O_s{}'\right]\right| - \frac{1}{2} = \frac{1}{2}\left(\frac{1}{2} + \varepsilon\right) + \frac{1}{2}\left(\frac{1}{2}\right) - \frac{1}{2} = \frac{\varepsilon}{2}. \tag{5}$$

Hence, Theorem 11 is proved, and this means that the scheme meets IND-IKGA security.

Table 4: Comparison with other schemes.

| Scheme | Problem | Access structure | Searchable | Keyword | Cross-domain | Antiquantum attack | Blockchain |
|---|---|---|---|---|---|---|---|
| [20] | DBDH | Access tree | √ | 1 | × | × | √ |
| [26] | SDP | LSSS | √ | 1 | × | × | × |
| [27] | DBDH | Access tree | × | — | × | × | √ |
| [28] | DBDH | Access tree | × | — | × | × | × |
| [29] | DBDH | LSSS | √ | 1 | × | × | √ |
| [30] | LWE | LSSS | √ | 1 | × | √ | × |
| [31] | LWE | AND | √ | 1 | × | √ | √ |
| Ours | RLWE | LSSS | √ | n | √ | √ | √ |

Table 5: Storage cost.

| Scheme | Public key | Master key | Private key | Index | Trapdoor |
|---|---|---|---|---|---|
| [30] | $(nm + m^2N + 2n + m^2)\log q$ | $m^2 \log q$ | $2nm \log q$ | $2nmA_w \log q$ | $2m \log q$ |
| [31] | $(nm + m^2N + n + m^2)\log q$ | $m^2 \log q$ | $2nm \log q$ | $(m+1)l \log q$ | $m \log q$ |
| Ours | $n(N_1 + 2N_2 + 3)\log q$ | $n(2N_1 + N_2 + 3)\log q$ | $n(A_u + 1)\log q$ | $n(A_w + 1)\log q$ | $n(A_{w'} + 1)\log q$ |

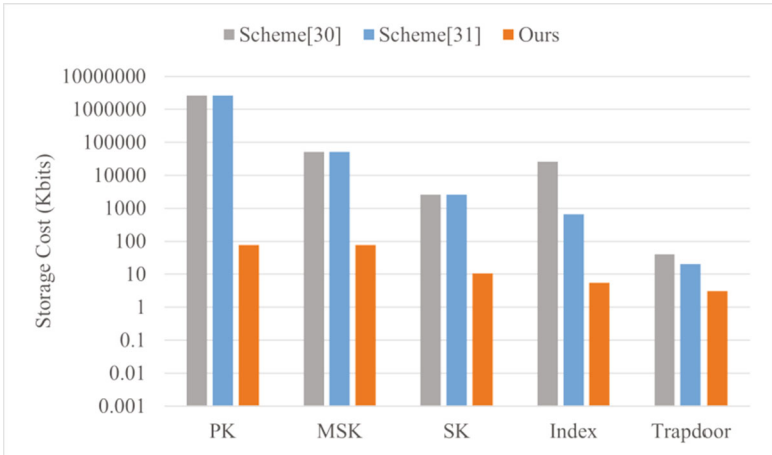



Figure 4: The comparison of storage cost.

Table 6: Calculation cost.

| Scheme | Cost of index generation | Cost of trapdoor generation | Cost of single matching |
|---|---|---|---|
| [30] | $2mn$mul | $2mn$mul | $(4m+2)$mul |
| [31] | $l(m+1)$mul | $2m$mul | $ml$mul |
| Ours | $(A_w + 1)$mul | $(3A_{w'} + 2)$mul | $(2A_d + 1)$mul + mod |

*4.2. Performance Analysis.* Since the scheme in this paper is mainly constructed on the basis of CP-ABE, some schemes based on attribute-based encryption are selected for comparison, including searchable schemes, schemes combined with blockchain and lattice schemes. These schemes are selected to compare their functions, the cost of storage, calculation, and communication.

(1) Different attribute-based encryption schemes are selected for function comparison. The results are shown in Table 4

Scheme [20] uses searchable encryption technology to realize the search of a single keyword on the blockchain and implements access control according to CP-ABE. The scheme is constructed by bilinear pairing, which has great application prospects in social networks and medical information fields, but the scheme cannot resist quantum attacks.

Scheme [26] pays attention to the problems of high computing cost and low efficiency of searching data in ABE scheme, reduces the local computing cost of users by using outsourcing technology, and proves that the scheme meets adaptive security. However, the scheme only supports single keyword search and cannot resist quantum attacks.

For medical data protection, scheme [27] combines ABE and blockchain technology to enable data to be shared

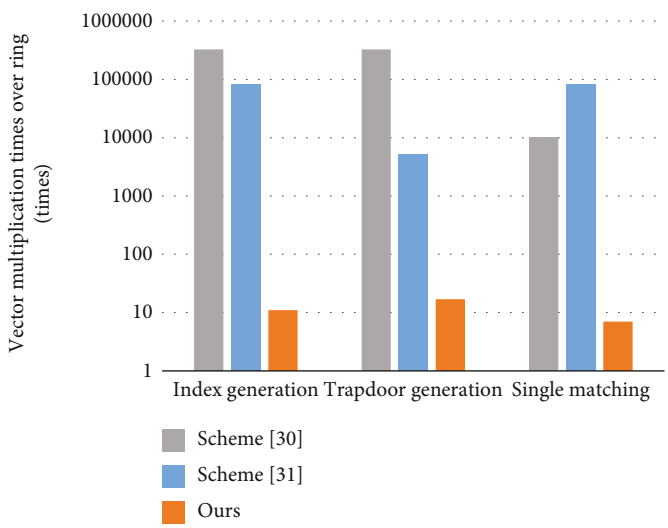



Figure 5: The comparison of calculation cost.

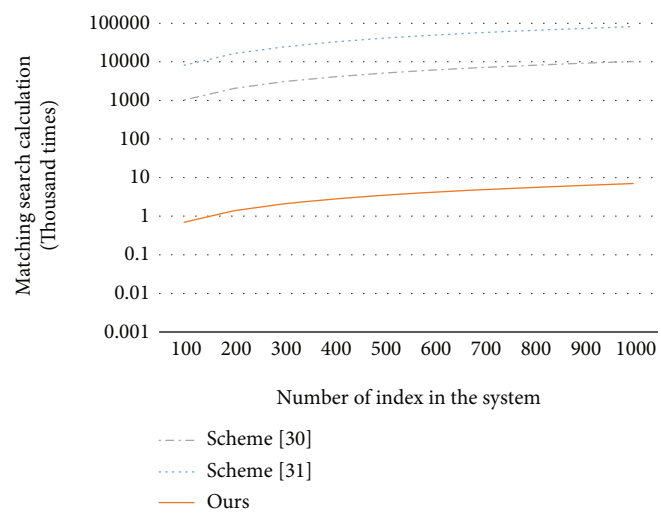



Figure 6: The calculation cost for indexes ranging from 100 to 1000.

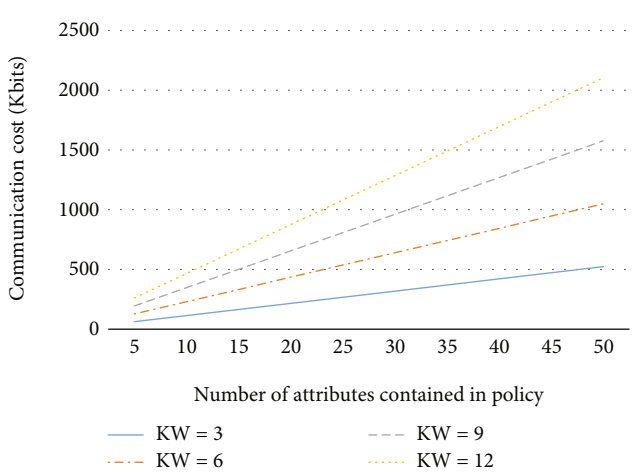



Figure 7: The communication cost for indexes ranging.

efficiently and safely among patients, hospitals, and other entities. The scheme does not support ciphertext search and cannot resist quantum attacks.

In the Internet of Things environment, scheme [28] outsources the decryption operation in content encryption to fog nodes, solves the problem that computing is difficult due to the limited resources of Internet of Things devices and also protects users' privacy by constructing false attributes. The scheme does not support ciphertext search and quantum attack resistance.

Scheme [29] combines blockchain and ABE to realize data sharing. The scheme realizes decentralization and avoids the risk of privacy disclosure by third parties and supports ciphertext search. However, the scheme cannot resist quantum attacks.

Scheme [30] solves the problem of ciphertext search in the cloud environment. The scheme only supports single-

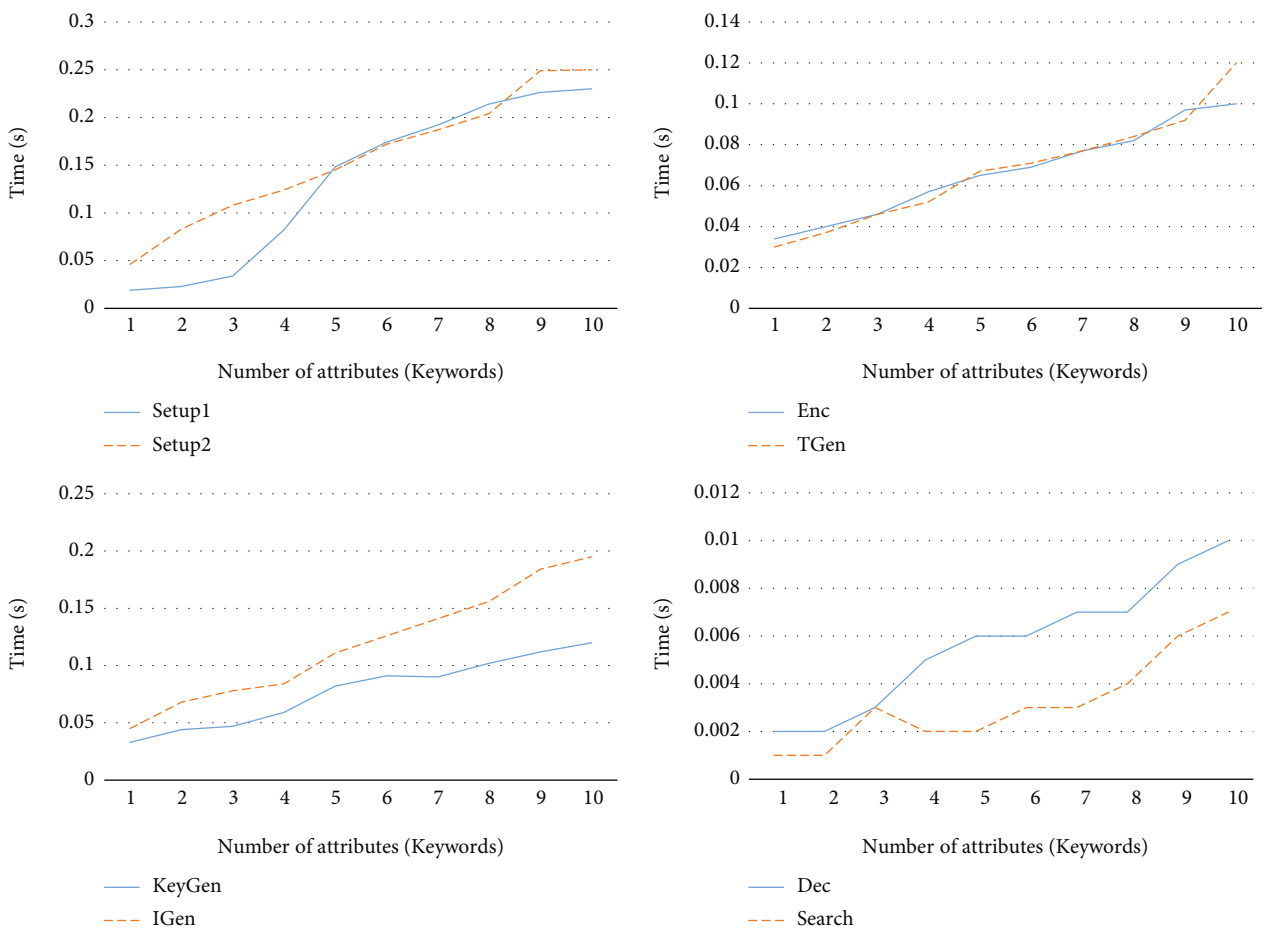



Figure 8: Relationship between the number of attributes or keywords and the running time of algorithm.

keyword search. The biggest feature is that it is based on LWE construction and can resist quantum attacks.

Scheme [31] relies on the technical characteristics of Ethereum to solve the problem of single point of failure in traditional systems, and can resist quantum attacks based on the LWE problem. At the same time, the scheme enables data users to generate private keys for visitors, avoiding key abuse caused by third parties. Due to the use of AND gate multivalue access strategy, its expression ability is slightly weak.

Based on RLWE, our scheme can resist quantum attacks, and LSSS has rich access structure, and the scheme realizes cross-domain access based on blockchain and can realize search of multiple keywords.

(2) Since schemes [20, 26–29] are not based on lattice structure, it is mainly compared with scheme [30, 31]. In schemes [30, 31], $m$ and $n$ are the parameters from lattice, $m \geq 5n \log q$, and $l$ is the security parameter in the keyword matching algorithm in scheme [30]. $N_1$ represents the number of all attributes in the system, $N_2$ represents the number of all keywords in the system, $A_u$ represents the number of attributes in the user attribute set, and $A_w$ represents the number of keywords in index, $A_{w'}$ indicates the number of keywords in the trapdoor. The results are shown in Table 5; this scheme is superior to scheme [30, 31] in size of system public key, master key, user private key, index, and trapdoor. As shown in Figure 4, the three schemes are analyzed by numerical simulation for visual representation, where $n = 64$, $q = 129$, $N_1 = 50$, $N_2 = 50$, $m = 5n \log q$, $A_u = 20$, $A_w = 10$, $A_{w'} = 5, l = 32$

(3) The comparison results with schemes [30, 31] in terms of computational cost are shown in Table 6. Since the cost of addition operation is small, it is not included in the analysis here. $A_d$ represents the number of keywords required for successful matching, the mul represents the multiplication between vectors on the ring, and the mod represents modular operation, and other parameters have the same meaning as (2). For visual representation, numerical simulation analysis is carried out, the calculation amount of index generation, trapdoor generation, and single matching is shown in Figure 5. When

the index's number in the system is 100-1000, the search matching overhead is as shown in Figure 6

(4) The main objects of communication cost include ciphertext and index. The encrypted information in the ciphertext is mainly the address of data storage returned by IPFS and the symmetric key used for symmetric encryption of the original data. Set the sum of the two elements as 1280 bit; the index mainly includes the keyword combination of data. Now, simulate the ciphertext overhead of the number of attributes included in the attribute strategy from 5 to 50 when the keywords are 3, 6, 9, and 12, respectively. The results are shown in Figure 7

(5) Experimental analysis

In order to further analyze the performance of the scheme, we tested 8 algorithms in the scheme. Because there are few simulation experiments related to the lattice attribute-based encryption scheme, it is difficult to effectively compare and analyze with other schemes. Here, the efficiency of the algorithm is mainly tested and analyzed. The experimental environment is AMD ryzen 7-5800H processor 3.20 GHz, 16.0 GB memory, 64-bit Windows 11 operating system. The experimental program is written in c++ language and implemented in QT creator development environment based on NTL library. In this experiment, setting parameters $q = 8380417$, $p = 3$, mainly test the running time of each algorithm when the number of attributes or keywords is from 1 to 10. Since the search part of this scheme is similar to the attribute-based encryption system, two algorithms with similar principles are put into the same diagram for analysis. It can be seen from Figure 8 that the running time of the algorithm is proportional to the number of attributes or keywords contained in the algorithm process, and the experimental results are consistent with the theoretical analysis results.

## 5. Conclusion

In this paper, a searchable attribute-based encryption scheme supporting cross-domain access is constructed based on the RLWE. The whole process can be traced based on the blockchain, and the combined search of multiple keywords is supported at the same time. Through analysis, the scheme meets trapdoor search security, anticollusion attack, and the indistinguishability under chosen-plaintext attack. Compared with other schemes, it has certain advantages in function and performance, but the scheme does not consider the change of user attributes. The next step will study the security and efficiency of attribute revocation and update on this basis.

## Data Availability

All data used during the study are available from the corresponding author upon request.

## Conflicts of Interest

The authors state that there is no conflict of interest.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China (no. 61572521), Engineering University of the PAP Innovation Team Science Foundation (no. KYTD201805), and Natural Science Basic Research Plan in Shanxi Province of China (2021JM252).

## References

[1] C. Wang, J. Z. Chen, Y. J. Liu, and A. Li, "A cross-domain access control method for large organizations," in *Proceedings of 2014 International Conference on Advances in Materials Science and Information Technologies in Industry (AMSITI 2014)*, pp. 28–33, Xi'an,China, 2014.

[2] X. H. Yang and H. Wang, "A cross-domain access control model based on trust measurement," *Wuhan University Journal of Natural Sciences*, vol. 21, no. 1, pp. 21–28, 2016.

[3] S. Shuang and S. D. Chen, "Trusted and efficient cross-domain access control system based on blockchain," *Scientific Programming*, vol. 2020, Article ID 8832568, 13 pages, 2020.

[4] L. Bai, K. Fan, Y. Bai, X. Cheng, H. Li, and Y. Yang, "Cross-domain access control based on trusted third-party and attribute mapping center," *Journal of Systems Architecture*, vol. 116, no. 5, article 101957, 2021.

[5] I. Ullah, S. Zeadally, N. U. Amin, M. A. Khan, and H. Khattak, "Lightweight and provable secure cross-domain access control scheme for the Internet of Things (IoT) based wireless body area networks (WBAN)," *Microprocessors and Microsystems*, vol. 81, no. 2, article 103477, 2021.

[6] A. Sahai and B. Waters, "Fuzzy identity based encryption," in *EUROCRYPT*, pp. 457–473, 2005.

[7] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *CCS '06: Proceedings of the 13th ACM conference on Computer and communications security*, pp. 89–98, Alexandria, Virginia, USA, 2006.

[8] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *2007 IEEE Symposium on Security and Privacy (SP '07)*, pp. 321–334, Berkeley, CA, USA, 2007.

[9] L. Cheung, J. A. Cooley, R. Khazan, and C. Newport, *Collusion-Resistant Group Key Management Using Attribute-Based Encryption*, Cryptology ePrint Archive Report 2007/161, 2007.

[10] B. Waters, "Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization," in *Public Key Cryptography – PKC 2011*, pp. 53–70, Springer, 2011.

[11] S. Agrawal, X. Boyen, V. Vaikuntanathan, P. Voulgaris, and H. Wee, "Functional encryption for threshold functions (or fuzzy IBE) from lattices," in *Public Key Cryptography – PKC 2012*, pp. 280–297, Springer, 2012.

[12] Y. T. Wang, "Lattice ciphertext policy attribute-based encryption in the standard model," *International Journal of Network Security*, vol. 16, no. 6, pp. 444–451, 2014.

[13] S. Tan and S. Azmasn, "Lattice ciphertext-policy attribute-based encryption from ring-LWE," in *2015 International Symposium on Technology Management and Emerging Technologies (ISTMET)*, pp. 258–262, Langkawi, Malaysia, 2015.

[14] G. Kaiyang, H. Yiliang, L. Kai, and W. Riming, "Traceable attribute-based encryption on OBDD access structure from lattice," in *2022 11th International Conference on Communications, Circuits and Systems (ICCCAS)*, pp. 210–215, Singapore, Singapore, 2022.

[15] S. S. Basu and S. Tripathy, "Securing multicast group communication in IoT-enabled systems," *IETE Technical Review*, vol. 36, no. 1, pp. 83–93, 2019.

[16] L. S. Yao and S. P. Wang, "Attribute-based encryption with equality test in the internet of things," *Microelectronics & Computer*, vol. 36, no. 6, pp. 64–69, 2019.

[17] P. S. Challagidad and M. N. Birje, "Efficient multi-authority access control using attribute-based encryption in cloud storage," *Procedia Computer Science*, vol. 167, no. 1, pp. 840–849, 2020.

[18] Y. L. Tian, K. D. Yang, and Z. Wang, "Algorithm of blockchain data provenance based on ABE," *Journal on Communications*, vol. 40, no. 11, pp. 101–111, 2019.

[19] M. M. Sandoval, H. M. Marin-Castro, and J. L. González, "Attribute-based encryption approach for storage sharing and retrieval of encrypted data in the cloud," *IEEE Access*, vol. 8, pp. 170101–170116, 2020.

[20] S. F. Niu, Y. Y. Xie, P. P. Yang, and X. Du, "Cloud-assisted attribute-based searchable encryption scheme on blockchain," *Journal of Computer Research and Development*, vol. 58, no. 4, pp. 811–821, 2021.

[21] X. D. Zhang, T. W. Chen, Y. M. Yu et al., "Model of blockchain data sharing based on ABE," *Application Research of Computers*, vol. 38, no. 8, pp. 2278–2283, 2021.

[22] S. F. Niu, M. Song, L. Z. Fang et al., "Cloud storage data sharing based on attribute encryption in smart healthcare," *Journal of Electronics & Information Technology*, vol. 44, no. 1, pp. 107–117, 2022.

[23] P. Kanimozhi and T. Victoire, "Secure sharing of IOT data in cloud environment using attribute based encryption," *Journal of Circuits, Systems and Computers*, vol. 30, no. 6, article 2150102, 2021.

[24] X. Li and M. Tan, "Electronic certificate sharing scheme with searchable attribute-based encryption on blockchain," *Journal of Physics Conference Series*, vol. 1757, no. 1, article 12161, 2021.

[25] V. Lyubashevsky, C. Peikert, and O. Regev, "A toolkit for ring-LWE cryptography," *EUROCRYPT*, pp. 35–54, 2013.

[26] L. F. Guo and Q. L. Wang, "Adaptive secure outsourced attribute-based encryption scheme with keyword search," *Journal of Computer Applications*, vol. 41, no. 11, pp. 3266–3273, 2020.

[27] S. Pournaghi, B. Majid, and F. Yaghoub, "MedSBA: a novel and secure scheme to share medical data based on blockchain technology and attribute-based encryption," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 11, pp. 4613–4641, 2020.

[28] O. Nouali, A. Abdelouahab, and S. Ahmed, "SHARE-ABE: an efficient and secure data sharing framework based on ciphertext-policy attribute-based encryption and fog computing," *Cluster Computing*, vol. 25, no. 1, pp. 167–185, 2022.

[29] H. Gao, Z. Ma, S. Luo, Y. Xu, and Z. Wu, "BSSPD: a blockchain-based security sharing scheme for personal data with fine-grained access control," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 6658920, 20 pages, 2021.

[30] U. S. Varri, S. K. Pasupuleti, and K. V. Kadambari, "CP-ABSEL: ciphertext-policy attribute-based searchable encryption from lattice in cloud storage," *Peer-to-Peer Networking and Applications*, vol. 14, no. 3, pp. 1290–1302, 2021.

[31] X. Wang and Y. L. Chen, "Attribute-based searchable encryption scheme from lattices on Ethereum," *Journal of Chongqing University of Posts and Telecommunications (Natural Science Edition)*, vol. 33, no. 4, pp. 67–682, 2021.

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 GF$(q)$ are finite fields, and the secret $s$ is selected on 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 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 $x$, $f(x) = s + a_1 x + a_2 x^2 + \cdots + a_{k-1} x^{k-1}$, $d_{i,0} = f(\alpha_i)$, for $i = 1, \cdots, n$, the different participant $d_{i,1}, \cdots, d_{i,k-1}$ randomly selects on GF$(q)$. For any participant $P_j$, the distributor randomly selects different nonnull values $g_{j,i}, i = 1, \cdots, n$ on 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 GF$(p)$.

*4.1.1. Secret Distribution.* $D$ constructs $k - 1$ degree polynomial $f(x) = s + a_1 x + a_2 x^2 + \cdots + a_{k-1} x^{k-1} \bmod p$, where the unknown coefficient $a_1, a_2, \cdots, a_k$ is uniformly randomly selected on GF$(p)$, different values $\alpha_1, \alpha_2, \cdots, \alpha_n$ are randomly selected on 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 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 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 GF$(p)$ of order $p$, and $p(p > n)$ is a large prime number. The select secret $s$ on GF$(p)$ sets the security parameter $v$ and executes the following algorithm:

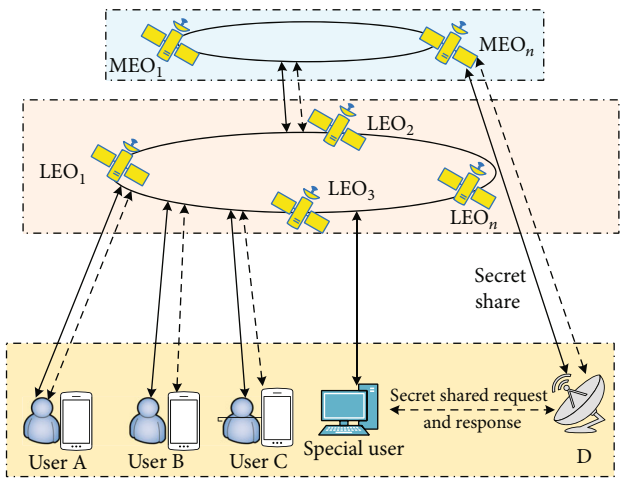



Figure 1: Overview of secret sharing scheme for satellite networks.

(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:

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

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

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

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

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

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

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 non-zero 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 $P_{-i}$.

Step 2: the algorithm execution cycle is round $k$. If $P_i$ receives $m-1$ secret shares of round $k-1$ sent by $P_{-i}$, 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 $P_{-i}$, 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. $\alpha$ 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 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 $\alpha$ 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 $\alpha = t$, $m - \alpha = t - 1$, $m = 2t - 1$. $\alpha$ 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 $s_{1,l+1,1}, s_{2,l+1,1}, \cdots, s_{t-1,l+1,1}, s'_{t,l+1,1}, s'_{t+1,l+1,1}, \cdots, s'_{2t-2,l+1,1}, s'_{l+1}$. 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 $t$th subsecret, and the guessing probability is negligible. The attacker can only show $\alpha$ 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 $\alpha$ 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 $t$:

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

Figure 2 shows the probability distribution of key share leakage with the key update cycle $T$, $x$ is $\lambda$, as can be seen from the figure, $\lambda$ at the same time, the larger the key update cycle, the higher the key share leakage rate. In Figure 2, $\lambda$ 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 | $T_L(m)$ | $T_v(m-1)$ |

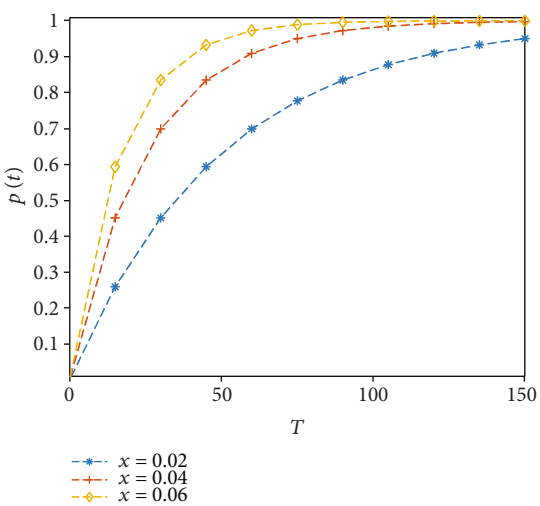



Figure 2: Key share leakage rate graph.

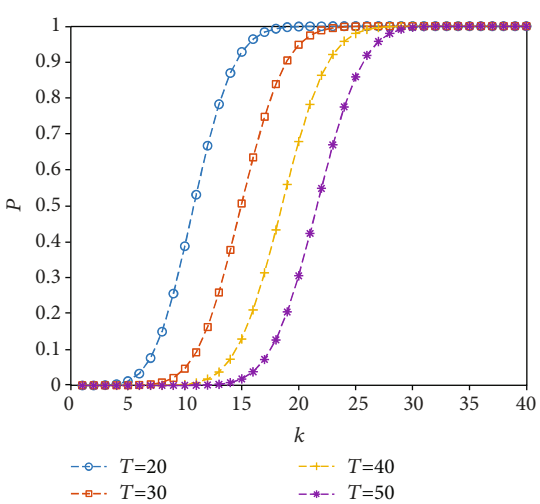



Figure 3: $P$ versus $k$ under different $T$.

*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.