Hindawi
Security and Communication Networks
Volume 2022, Article ID 2410455, 13 pages
https://doi.org/10.1155/2022/2410455

Research Article

Blockchain-Aided Searchable Encryption-Based Two-Way Attribute Access Control Research

Zhigang Xu,[1] **Shiguang Zhang**,[1] **Hongmu Han**,[1] **Xinhua Dong**,[1] **Zhiqiang Zheng**,[2] **Haitao Wang**,[2] **and Wenlong Tian**[3]

[1]*School of Computer Science, Hubei University of Hubei University of Technology, Wuhan 430068, China*
[2]*Narcotics Control Bureau of Department of Public Security of Guangdong Province, Guangzhou 510050, China*
[3]*School of Computer Science and Technology, University of South China, Hengyang 421001, China*

Correspondence should be addressed to Zhigang Xu; 20181100@hbut.edu.cn

Received 15 April 2022; Accepted 25 August 2022; Published 29 September 2022

Academic Editor: Jiewu Leng

In the Internet of Things (IoT), data sharing security is important to social security. It is a huge challenge to enable more accurate and secure access to data by authorized users. Blockchain access control schemes are mostly one-way access control, which cannot meet the need for ciphertext search, two-way confirmation of users and data, and secure data transmission. Thus, this paper proposes a blockchain-aided searchable encryption-based two-way attribute access control scheme (STW-ABE). The scheme combines ciphertext attribute access control, key attribute access control, and ciphertext search. In particular, two-way access control meets the requirement of mutual confirmation between users and data. The ciphertext search avoids information leakage during transmission, thus improving overall efficiency and security during data sharing. Moreover, user keys are generated by the coalition blockchain. Besides, the ciphertext search and pre-decryption are outsourced to cloud servers, reducing the computing pressure on users and adapting to the needs of lightweight users in the IoT. Security analysis proves that our scheme is secure under a chosen-plaintext attack and a chosen keyword attack. Simulations show that the cost of encryption and decryption, keyword token generation, and ciphertext search of our scheme are preferable.

1. Introduction

In Industry 4.0, the IoT is commonly used in industrial environments and often requires processing large amounts of data. Due to the limited resources of IoT devices, we often store large amounts of data from IoT devices on cloud servers. However, this outsourced storage approach may cause many privacy and security problems, such as identity leakage, illegal access to private data, and data tampering. The solution to these problems is to store the ciphertext in the cloud server. Symmetric encryption can guarantee data confidentiality but cannot achieve fine-grained access control and secure data sharing.

Attribute access control is an access control mechanism proposed by Sahai and Waters [1] to ensure effective and secure data sharing and fine-grained access. Technically, attribute access control is mainly divided into two types: ciphertext-policy attribute-based encryption (CP-ABE) [2] and key-policy attribute-based encryption (KP-ABE) [3]. In the CP-ABE scheme, each data user obtains the corresponding attribute secret key from the authorization agency according to their attributes, and the access structure of the file is determined by the data owner. Only when the attribute set in the secret attribute key of the data acquirer meets the access structure of the file can the file be viewed correctly. In the KP-ABE, by contrast, files can only be viewed when the access structure of the identity key satisfies the ciphertext properties. However, these two methods of attribute access control are only a single method of authentication. They address the need for one-way control of data sharing but do not meet the need for two-way confirmation of users and data. For this reason, Attrapadung and Imai [4] proposed a two-policy attribute access control scheme whose core idea is to combine ciphertext access control with key access control. On the one hand, the ciphertext is obtained by associating the plaintext with the corresponding user access structure and plaintext attributes. On the other hand, the user's private key is computed by associating its attribute set with the ciphertext access control structure. The plaintext can only be decryption if both the ciphertext access control and the key access control match. However, this solution is a centrally authorized agent prone to a single point of failure. Han et al. [5] proposed a distributed bidirectional attribute access control strategy. However, the scheme does not consider users' security requirements for personal data queries and transmissions in the IoT environment.

Blockchain is increasingly used in non-transactional scenarios such as supply chains, the IoT, smart healthcare, and public security, where data often contain users' private data. The data cannot be fully disclosed to everyone as a transaction and can only be shared to a limited extent. Through blockchain research, the use of blockchain to manage users' keys ensures secure data sharing for the development of the Industrial Internet of Things (IIoT).

In this paper, we propose a blockchain-aided searchable encryption-based two-way attribute access control scheme (STW-ABE) to manage massive IoT data and meet people's demand for data access control of private data. The main contributions of our scheme can be summarized as follows:

(1) *Blockchain-Aided Key Generation*. Blockchain consensus nodes jointly execute the DKG to generate the secret key. It avoids the problem of secret key leakage caused by a single point of failure.

(2) *Blockchain-Cloud-Aided Keyword Search*. The combination of attribute encryption technology and searchable encryption achieves fine-grained two-way access control of transaction ciphertexts in the blockchain. The blockchain sends a token containing a single keyword to the CS. The CS uses the token to perform a ciphertext search to avoid leakage of private data during transmission.

(3) *Cloud-Aided Pre-Decryption*. The CS provides the pre-decryption service for users with access permission, and the user only needs to perform one exponential operation to decrypt the ciphertext. It reduces computational pressure for users and meets the needs of resource-constrained IoT devices.

The rest of this article is organized as follows. Section 2 reports the most related work. Section 3 introduces relevant knowledge, including linear secret-sharing schemes, distributed key generation protocols, searchable encryption, and blockchain technology. Section 4 presents the system definition, including the system model, the STW-ABE scheme, and the security model. In Section 5, we reveal the detailed construction of the STW-ABE scheme. Section 6 analyzes the security of our scheme and compares the time cost with other schemes in encryption, decryption, and ciphertext search. Finally, we conclude in Section 7.

2. Related Work

In Industry 4.0, access control technology is essential to build trust and sustainability in a distributed context of the IoT. Leng et al. [6] proposed a blockchain model with chemical signature access under a distributed context, Makerchain, which binds unique signature data to the blockchain and automatically executes smart contracts set between manufacturers to achieve service trust between manufacturers. Rahman et al. [7] proposed a distributed multi-signature technology based on blockchain to realize multi-party identity authentication and guarantee the trust between multiple parties in the Industry 4.0 system. However, it does not consider the resource limitation of IoT terminal devices. Most data encryption techniques in use today are based on bilinear mapping encryption, which means that the computational cost of decryption is high. Most lightweight devices do not adapt to attribute-based access control. Therefore, many attribute access control schemes propose the method of outsourcing decryption. Li et al. [8] proposed an outsourcing ABE scheme search based on keyword search. However, the search method used in this scheme is a common public key encryption of keywords, which cannot achieve a fine-grained searchable encryption scheme. Ziegler et al. [9] proposed an outsourcing decryption scheme based on a prime order group to bridge the gap between the highly dynamic IIoT environment and resource-constrained devices. The IoT includes a core network and an edge network, and data security problems will be encountered in data sharing. Liu et al. [10] proposed a privacy-protecting multi-keyword searchable encryption scheme in a distributed system. Through a multi-server architecture, authorized servers can jointly search whether the token matches the ciphertext, thus improving the search efficiency. Miao et al. [11] put forward a multi-keyword search scheme based on attributes and transformed attributes into 0 and 1 codes for attribute judgment comparison, thus improving the efficiency of strategy judgment.

In the IIoT, blockchain is a new generation of security technology with immutability and traceability characteristics. Leng et al. [12] discussed how blockchain promotes the sustainable development of manufacturing and product life management in Industry 4.0. Mehta et al. [13] proposed a blockchain-based copyright contract transaction scheme for the Industry 4.0 supply chain, which ensured the security of copyright transactions for different stakeholders in the industry. But the blockchain has its potential security problems. Leng et al. [14] proposed the PDI model and divided blockchain security issues into process level, data level, and infrastructure level. This paper mainly studies data access control to solve the data-level security sharing problem to improve blockchain systems' data security. In Industry 4.0, blockchain provides key technology for the secure intelligent manufacturing of IIoT, but distributed Industry 4.0 needs to realize collaborative trust. Leng et al. [15] put forward eight network security obstacles in the intelligent manufacturing of blockchain. The cybersecurity barriers include device deception, false authentication, and trust in data sharing among participants. Therefore, implementing blockchain identity authentication in the IIoT is of great significance for the multi-party trust and sustainability of Industry 4.0. Li et al. [16] proposed a multi-keyword encrypted search scheme applicable for blockchain, which implements ciphertext information search and data access control through smart contracts. Before ciphertext search, the smart contract automatically determines data access permission to enhance trust among IoT users. Feng et al. [17] proposed a data privacy protection scheme based on blockchain searchable attribute access control. The user's permission authentication is implemented by the user's local server, avoiding the security risk of submitting the user's private key and access structure to the blockchain network. Gao et al. [18] proposed a trusted secure ciphertext policy and attributed a hiding access control scheme based on blockchain. The scheme hides the ciphertext policy and attribute information and reduces accidental leakage of data information. Therefore, Liu et al. [19] proposed a searchable attribute-based encryption scheme in which a coalition blockchain replaces the traditional centralized server to be responsible for the generation and storage. Qin et al. [20] proposed a lightweight IoT access control scheme based on attribute encryption and blockchain to verify the accuracy of outsourced decrypted data in IoT through a smart contract. In addition, some schemes use the distributed feature of blockchain to distribute secret keys as the authority. Lewko and Waters [21] proposed a multi-authority attribute-based encryption scheme. In this scheme, the secret user key consists of multiple components, each from a different organization, to prevent collusion attacks among users. Qin et al. [22] proposed a blockchain multi-attribute access control scheme for cloud data sharing. Smart contracts on blockchain manage attribute tokens across domains to solve the trust problem between multiple users. Shi et al. [23] proposed a blockchain-based distributed access control scheme for IoT. The solution uses blockchain nodes as the addresses of IoT devices. It uses blockchain to complete the data authorization, cancelation, access control, and auditing process to ensure data security in the distributed IoT system.

The access control scheme mentioned above compensates for the deficiency of the blockchain access control mechanism in the IIoT environment. However, combining the existing access control scheme with blockchain is not enough to meet users' demand for secure sharing access control of private data. Currently, most blockchain access control solutions only implement user access policy settings for the data and do not address the need for two-way policy confirmation between the user and the data. Furthermore, the security of the data during sharing and the usability of users of lightweight devices were not considered. Therefore, this paper proposes a blockchain-aided searchable encryption-based two-way attribute access control scheme (STW-ABE).

3. Preliminaries

3.1. Linear Secret-Sharing Schemes (LSSS). The linear secret-sharing scheme [24] is defined as follows.

Definition 1. Let P be a set of parties. Let M be a $l \times k$ matrix. Let ρ: $\{1, \ldots, l\} \longrightarrow P$ be a function that maps a row to a party for labelling. Let (M, ρ) represent a linear secret-sharing scheme with access structure A, which usually consists of two polynomial-time algorithms:

(1) For $x = 1, \ldots, l$, the x-th row of matrix M is labeled by a party $\rho(i)$, where ρ: $\{1, \ldots, l\} \longrightarrow P$ is a function that maps a row to a party for labelling. The algorithm takes as input the secret value $s \in \mathbb{Z}_p$ that is shared. $y_2, \ldots, y_k \in \mathbb{Z}_p$ are randomly chosen, and $\overrightarrow{v} = (s, y_2, s \in \mathbb{Z}_p \ldots, y_k)^T$. The share $\lambda_{p(i)} = M_i \cdot \overrightarrow{v}$ belongs to party $\rho(i)$.

(2) Let $S \in A$ be input. Let $I = \{\{i | \rho(i) \in S\}$, and randomly select $c_i \in \mathbb{Z}_P^*$. The output is a constant with linear reconstruction characteristics: $\sum_{i\in I} c_i \sigma_i = s$.

3.2. Distributed Key Generation (DKG) Protocol. Traditional key generation is performed by the central server. This centralized management approach is prone to a single point of failure problem. To solve this problem, researchers proposed the distributed key generation (DKG) protocol [25]. In the DKG protocol, the generation of secret values is done by multiple parties, not by an authoritative center, and does not rely on trusted third parties. Multiple nodes jointly generate a secret value α, and each node has a corresponding secret value α to share. The secret value can be restored only when the sharing rule (t, n) is met, where t is the number of nodes authorized by participants and n is the threshold. The secret value generated by t nodes must be shared by at least n participants to complete the sharing of secret value α.

3.3. Searchable Encryption. Song et al. [26] proposed the practical technology of encrypted data search. In this technique, the scheme for searching the encrypted data is described, and the security of the generated encryption system is proved. The third-party server can only obtain the matching ciphertext results if only the ciphertext data are provided. Nevertheless, it cannot obtain the data information in plaintext, which implements query isolation. In addition, a hidden query is supported. Data users only need to send the search token containing the query keyword to the third-party server for ciphertext search without disclosing the detailed information of the keyword to the server.

3.4. Blockchain Technology. Generally, there are three types of blockchain: public blockchain, private blockchain, and coalition blockchain. A public blockchain allows any node to generate transaction information and view all information in the block. In a private blockchain, all nodes on the network are controlled by a single organization, and only a small number of authorized nodes have access to the data information. In the coalition blockchain, authorized nodes can join the blockchain network and participate in transactions and information synchronization with strong controllability and high privacy.

This paper uses a coalition blockchain. The blockchain is controlled by a group of trusted nodes that control the consensus protocol. Other authorized nodes can generate data and send them to the blockchain for storage. Then, the consensus node runs the consensus protocol to complete the ledger update in the coalition blockchain so that all nodes keep the whole state consistent. In this paper, the specific functions of the coalition blockchain are as follows. (1) The consensus node in the blockchain initializes system parameters using the distributed secret key generation protocol. (2) The consensus node is responsible for generating, storing, and distributing global public keys, public and private key pairs of users, and user identity keys. (3) The consensus node responds to the keyword searched by the user, generates a ciphertext index through the blockchain, and sends it to the cloud server.

4. System Definition

4.1. System Model. The STW-ABE scheme contains four participants presented as follows. The detailed structural components of the scheme are shown in Figure 1.

(1) *Data Publisher (DP).* Any IoT device can generate data. The plaintext data containing ciphertext attributes and user access structure are encrypted on the local service. Then, the ciphertext and ciphertext index are uploaded to the cloud server. Data publishers can be people and any IoT device.

(2) *Data Acquirer (DA).* The data acquirer receives the user identity key from the blockchain, which contains the user attributes and the ciphertext access structure. The DA can only capture the ciphertext if the DA attribute meets the user access structure of the DP and the ciphertext attribute meets the ciphertext access structure of the DA. The DA obtains the ciphertext that meets the individual's conditions and decrypts it with its user identity key.

(3) *Blockchain (BC).* A coalition blockchain comprises trusted consensus nodes. The blockchain is responsible for initializing the global public key and generating users' public and private key pairs, user identity secret keys, and tokens.

(4) *Cloud Server (CS).* Cloud servers are used to store large amounts of ciphertext and ciphertext indexes that are uploaded by DP. In addition, CS responds to users' search requests, verifies access control permission, provides pre-decryption services for DA who meets the permission, and returns the pre-decrypted intermediate ciphertext to the DA.

The STW-ABE scheme is divided into three parts. The first part is encryption. First, the DP obtains the global public key and users' public and private key pairs from the blockchain. Then, the DP encrypts the plaintext data through the ciphertext attribute set and user access structure. The DP then sends the ciphertext and ciphertext index to CS. The second part is the ciphertext search. DA searches for ciphertext information by keyword. First, DA sends a keyword to the blockchain network. Second, the blockchain network encrypts a keyword into a token and sends the token to CS, which conducts a ciphertext search through the ciphertext index and search tokens. Finally, the retrieved ciphertext is stored. The third part is decryption. The CS verifies the access control permissions of the set of users, that is, whether the user attributes meets the user access structure and whether the ciphertext attribute set meets the ciphertext access structures. The CS provides a pre-decryption service to generate intermediate ciphertext for the DA, satisfying the two-way access structure. When the DA receives the intermediate ciphertext from CS, the DA uses the user identity key to decrypt the intermediate ciphertext into plaintext.

4.2. System Procedure. The composition of the STW-ABE scheme is as follows.

4.2.1. Initialization. $\mathrm{Setup}(\lambda) \longrightarrow \mathrm{GP}$. Setup: the process runs on blockchain consensus nodes participating in authorization and outputs global public key GP.

$\mathrm{Authority\,Setup}(\mathrm{GP}) \longrightarrow \mathrm{SK}, \mathrm{PK}$. User public key and private key generation: the process runs in the blockchain consensus nodes, with global public key GP as input, and outputs user public key PK and user private key SK.

4.2.2. User Identity Key Generation. $\mathrm{KeyGen}(\mathrm{GP}, \mathrm{SK}, \mathrm{PK}, \mathrm{UID}, \mathrm{K}, (P, \eta)) \longrightarrow \mathrm{UK}_{\mathrm{UID}}$. User identity key generation: the process is run consensus nodes in blockchain that execute the distributed key generation protocol, taking the global public key GP, the user public key PK, the user private keySK, the user attributes set K, the ciphertext access structure (P, η), and user's identity UID as input, and outputs the user identity key $\mathrm{UK}_{\mathrm{UID}}$.

4.2.3. Encryption. $\mathrm{Encrypt}(\mathrm{GP}, \mathrm{SK}, \mathrm{PK}, \mathrm{UID}, (\mathrm{F}, \rho), \Lambda, \mathrm{M}) \longrightarrow \mathrm{D}, \mathrm{KW}$. Encryption: this process is run by the DP, taking the global public key GP, the user public key PK, the user private key SK, the user's identity UID, the user access structure (F, ρ), ciphertext attribute set Λ, and plaintext M as input, and outputs ciphertext D and keywords of ciphertext KW.

4.2.4. Index Generation. $\mathrm{IndexGen}(\mathrm{GP}, \mathrm{PK}, \mathrm{KW}) \longrightarrow \mathrm{Index}$. Index generation: this process is run by the DP, with the global public key GP, the user public key PK, and the keywords of ciphertext KW as input, and outputs the ciphertext index Index.

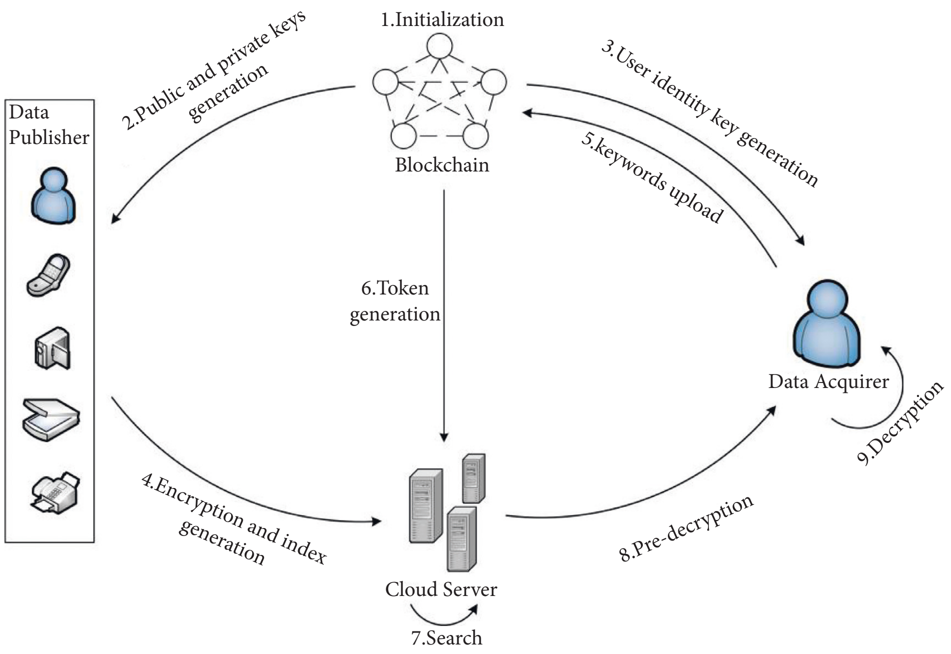

Figure 1: Scheme structure. The structure contains a specific implementation process for access control.

4.2.5. Token Generation. TokenGen$(\mathrm{GP}, \mathrm{UK}_{\mathrm{UID}}, \mathrm{kw}) \longrightarrow \mathrm{Tok}$. Token generation: this process is run by the blockchain consensus nodes, with the global public key GP, the user identity key $\mathrm{UK}_{\mathrm{UID}}$, and the keywords of the data user kw as input, and outputs user search token Tok.

4.2.6. Search. CipherTextSearch$(GP, \mathrm{Tok}, \mathrm{Index}) \longrightarrow D$. Search: this process is run by the CS, taking the global public key GP, the user search token Tok, and the ciphertext index Index as input to output the matching ciphertext D.

4.2.7. Decryption. ProxyDecrypt$(GP, D, UK_{\mathrm{UID}}) \longrightarrow D'$. Proxy decryption: this process is run by the CS, taking the global public key GP, the ciphertext D, and the user identity key UK_{UID} as input. If the ciphertext attribute set Λ satisfies the ciphertext access structure (P, η) and the user attribute set K satisfies the user access structure (F, ρ), the ciphertext is pre-decrypted and sends the intermediate ciphertext D' returned to the DA.

userDecrypt$(GP, D, D', UK_{\mathrm{UID}}) \longrightarrow M$. User decryption: this process is run by the DA, taking the global public key GP, the ciphertext D, the intermediate ciphertext D', and the user identity key UK_{UID} as input, and outputs plaintext M.

The notations used in our scheme are summarized in Table 1.

4.3. Security Model

4.3.1. Ciphertext Indistinguishability. The indistinguishability security under chosen-plaintext attack (IND-CPA) of an STW-ABE scheme is defined by the following game between a challenger C and a probabilistic polynomial-time (PPT) adversary A. Let A_u be the authority universe of size t. We define adversary A as a (t, n) adversary who can compromise at most $t-1$ authority. This security model adopts the (t, n) key generation protocol. The description of the game is as follows:

(1) Initialization: C runs the *Initialization* of STW-ABE and returns the global public key GP, user public key PK, and user private key SK to A.

(2) Query phase I: adversary A queries the following oracles adaptively.

   (a) *User Identity Key Oracle.* A submits an identity UID to C. C runs the KeyGen$(GP, SK, PK, UID, K, (P, \eta)) \longrightarrow UK_{\mathrm{UID}}$. Finally, it returns UK_{UID} to A.

   (b) *Encryption Oracle.* A sends $((F, \rho), \Lambda, M)$ to C. C runs the Encrypt$(GP, SK, PK, (F, \rho), \Lambda, M) \longrightarrow D, KW$ to generate the ciphertext D. Notice that the user access structure (F, ρ) does not satisfy the challenge user attribute set K, and the ciphertext attribute set Λ does not satisfy the challenge ciphertext access structure (P, η).

(3) Challenge: A submits two plaintexts of equal length M_0, M_1 and sends them to C. C selects a random number $\partial \in \{0, 1\}$ and encrypts the selected plaintext with user access structure (F^*, ρ^*) and ciphertext attribute set Λ^*. The final ciphertext will be generated (D^*) and sent to A.

(4) Query phase II: A still can make queries adaptively as in *Query Phase I.*

(5) Guess: A outputs a guess b' for b.

Table 1: Notations.

Notation	Meaning
λ	A security parameter
GP	The global public key
SK	User's private key
PK	User's public key
UID	User's identity
K	User's attribute set
(P, η)	The ciphertext access structure
UK_{UID}	User's identity key
(F, ρ)	The user access structure
Λ	The ciphertext attribute set
KW	The keywords of ciphertext
Index	The ciphertext index
kw	User search keywords
Tok	Search token
D/D'	Ciphertext of the data/pre-decrypted ciphertext

The advantage of A in this game is defined as follows:

$$\mathrm{Adv}_A^D = \left| Pr\left[b' = b\right] - \frac{1}{2} \right|. \qquad (1)$$

Definition 2. An STW-ABE scheme is IND-CPA secure if the advantage defined above for any $(t,\ n)$ PPT adversary A is negligible.

4.3.2. Index Indistinguishability. Index indistinguishable security (IND-CKA) under chosen access structure and chosen keyword attack is defined as the security game of challenger C and a probabilistic polynomial-time (PPT) adversary A for the STW-ABE scheme. In this scheme, only single keyword ciphertext retrieval is considered. The description of the game is as follows:

(1) Initialization: A defines a user access structure (F^*, ρ^*) and ciphertext attribute set Λ^*.

(2) Setup: C runs the *Initialization* of STW-ABE and returns the global public key GP, user public key PK, and user private key SK to A.

(3) Query phase I: adversary A queries the following oracles adaptively.

  (a) *User Identity Key Oracle.* A submits an identity UID to C. C runs the $\mathrm{KeyGen}(GP, SK, PK, UID, \theta, (P,\ \eta)) \longrightarrow UK_{\mathrm{UID}}$. Finally, it returns UK_{UID} to A.

  (b) *Token Oracle.* A send (kw) to C. C runs the $\mathrm{TokenGen}(GP, SK, kw) \longrightarrow \mathrm{Tok}$ to generate the token Tok. Notice that the user access structure (F, ρ) does not satisfy the challenge user attribute set K, and ciphertext attribute set Λ does not satisfy the challenge ciphertext access structure (P, η). We assume that all query results (Tok) have at least one matched index that can be searched out.

  (c) *Index Oracle.* A submits $(UK_{\mathrm{UID}}, \{KW\})$ to C, and C runs the $\mathrm{IndexGen}(GP, PK, UK_{\mathrm{UID}}, KW) \longrightarrow \mathrm{Index}$ to generate the index.

(4) Challenge: A submits two keywords of equal length kw_0, and kw_1 to C. C chooses randomly number $b \in \{0, 1\}$ and runs the $\mathrm{IndexGen}(GP, PK, UK_{\mathrm{UID}}, KW) \longrightarrow \mathrm{Index}$ with the challenge user access structure (F^*, ρ^*) and ciphertext attribute set Λ^* to return Index^* to A.

(5) Query phase II: A still can make queries adaptively as in *Query Phase I* after receiving the challenge index. Similarly, A cannot query on the user access structure, which satisfies the challenge user attribute set, and ciphertext attribute set, which satisfies the ciphertext access structure.

(6) Guess: A outputs a guess b' for b.

The advantage of A in this game is defined as follows:

$$\mathrm{Adv}_A^{kw} = \left| Pr\left[b' = b\right] - \frac{1}{2} \right|. \qquad (2)$$

Definition 3. An STW-ABE scheme is IND-CKA secure if the advantage defined above for any PPT adversary A is negligible.

5. Construction

This section presents a detailed construction of our STW-ABE scheme, including initialization, user identity key generation, encryption, decryption, token generation, and search.

5.1. Initialization. This stage is divided into two parts. First, the blockchain consensus node executes the distributed key generation protocol to generate the global public key. Then, the blockchain consensus nodes generate user public and private keys.

Part One. $\mathrm{Setup}(\lambda) \longrightarrow GP$. First, the q-order bilinear group $\mathbb{G}_0$ with generator g and bilinear mapping $\mathbb{G}_0 \times \mathbb{G}_0 \longrightarrow \mathbb{G}_T$ is selected in the setup. In addition, the description of a hash function $H^{\circ}\{0, 1\}^* \longrightarrow \mathbb{G}_0$ that maps user identity UID to elements of $\mathbb{G}_0$ is published. Finally, the global public key is generated.

$$GP = \{g, H\}. \tag{3}$$

Part Two. Authority Setup $(GP) \longrightarrow SK, PK$. The authorization center $CN_i\,(i = 1, 2, \ldots, n)$ manages the set of user attributes $\widetilde{U}_i$ and ciphertext attributes $\widehat{U}_i$ of all users. CN_irandom selection of parameters $\alpha_q \in Z_P^*\,(q \in \overline{U}_i)$, $\beta_d \in Z_P^*\,(d \in \widehat{U}_i)$ according to the attribute set. Then, blockchain consensus nodes generate user public key $PK = \{e(g, g)^{\alpha_i}, g^{\beta_i}\}$ and user private key $SK = \{\alpha_i, \beta_i\}$.

5.2. User Identity Key Generation. KeyGen $(GP, SK, PK, \text{UID}, K, (P, \eta))) \longrightarrow UK_{\text{UID}}$. This KeyGen is run by the consensus nodes that execute the distributed key generation protocol, taking the global public key GP, the user public key PK, the user private key SK, the user attributes set K, the ciphertext access structure (P, η), and user's identity UID as inputs to output the user identity key UK_{UID}.

(1) Let P be a $l_o \times k_o$ matrix. The process randomly selects $\varphi \in \mathbb{Z}_P^*$, $y_i \in \mathbb{Z}_P^*\,(i = 2, \ldots, k_o)$ and constructs the vector $\overrightarrow{v}_{k_o} = (\varphi, y_2, \ldots, y_{k_o})^T$ and vector $\overrightarrow{w}_{k_o} = (0, y_2, \ldots, y_{k_o})^T$. φ is the secret value to be shared.

(2) Let P_x be the x-th row of the matrix P, and calculate $\sigma_x = P_x \cdot \overrightarrow{v}_{k_o}$, $\tau_x = P_x \cdot \overrightarrow{w}_{k_o}$.

(3) Select $\mu_x \in \mathbb{Z}_P^*, x = 1, 2, \ldots, l_o$ for each P_x to calculate the following equation:

$$\begin{gathered} U = e(g, g)^{\varphi}, \\ U_{x,1} = e(g, g)^{\sigma_x} e(g, g)^{\alpha_{\eta(x)}\mu_x}, U_{x,2} = g^{\mu_x}, U_{x,3} = g^{\beta_{\eta(x)}\mu_x} g^{\tau_x}. \end{gathered} \tag{4}$$

(4) Create a key that belongs to a primary attribute t $(t \in K)$ for the user identity UID and do the following calculation: $U_t = g^{\alpha_t} H(UID)^{\beta_t}$.

(5) Finally, the user identity key is generated $(UK_{\text{UID}} = \{(P, \eta), U, \{U_{x,1}, U_{x,2}, U_{x,3}\}_{x=}\ 1, 2, \ldots, l_o, \{U_t\}_{t\in K}\})$ and sent to the DA.

5.3. Encryption. The encryption consists of two processes, namely, encryption Encrypt $(GP, SK, PK, \text{UID}, (F, \rho), \Lambda, M) \longrightarrow D, KW$ and the index generation IndexGen $(GP, SK, PK, KW) \longrightarrow$ Index.

Encrypt $(GP, SK, PK, \text{UID}, (F, \rho), \Lambda, M) \longrightarrow D, KW$. This process is run by the *DP*, taking the global public key *GP*, the user public key *PK*, the user private key *SK*, the user identity UID, the user access structure (F, ρ), ciphertext attribute set Λ, and plaintext M as input.

(1) Let F be a $l_e \times k_e$ matrix. The process first randomly selects $s \in Z_P^*$, $t_j \in Z_P^*\,(j = 2, \ldots, k_e)$. Let vector $\overrightarrow{v}_{k_e} = (s, t_2, \ldots, t_{k_e})^T$, $\overrightarrow{w}_{k_e} = (0, t_2, \ldots, t_{k_e})^T$, and s be the secret value to be shared.

(2) Let F_x be the x-th row of the matrix F, and $\lambda_x = F_x \cdot \overrightarrow{v}_{k_e}$, $\mu_x = F_x \cdot \overrightarrow{w}_{k_e}$.

(3) Select $r_x \in \mathbb{Z}_P^*, x = 1, 2, \ldots, l_e$ for each F_x to calculate the following equation:

$$\begin{gathered} D = Me(g, g)^{s}, \\ D_{x,1} = e(g, g)^{\lambda_x} e(g, g)^{\alpha_{\rho(x)} r_x}, D_{x,2} = g^{r_x}, D_{x,3} = g^{\beta_{\rho(x)} r_x} g^{\mu_x}. \end{gathered} \tag{5}$$

(4) Create a key that belongs to the corresponding subattribute $i\,(i \in \Lambda)$ for the encrypted file, and the following calculation is performed:

$$D_k = g^{\alpha_k} H(\text{UID})^{\beta_k}. \tag{6}$$

(5) Finally, the ciphertext is generated:

$$D = \left\{(F, \rho), D, \left\{D_{x,1}, D_{x,2}, D_{x,3}\right\}_{x=1,2,\ldots,l_e}, \{D_k\}_{k\in\Lambda}\right\}, \tag{7}$$

and sent to the CS.

IndexGen $(GP, PK, KW) \longrightarrow$ Index. This process was conducted by DP on local devices, with the global public key *GP*, the user public key *PK*, and the keywords of ciphertext *KW* as inputs. U_w is the number of data keywords. The following calculations are performed to encrypt each keyword into a ciphertext index.

$$Idx_{1,\omega} = e(g, g)^{\alpha_\omega \cdot \varphi \cdot H(\text{kw}_k)}, I\text{d}x_{2,i} = g^{\beta_i \cdot \varphi}. \tag{8}$$

Finally, the ciphertext index $\text{Index} = \left\{\left\{I\text{d}x_{1,\omega}\right\}_{\omega\in U_w}, I\text{d}x_{2,i}\right\}$ is obtained and sent to the CS.

5.4. Token Generation. TokenGen $(GP, UK_{\text{UID}}, kw) \longrightarrow Tok$. This process is run by the consensus nodes that execute the distributed key generation protocol, with the global public key *GP*, the user identity key UK_{UID}, and the keywords of the data users *kw* as input. The following calculations are performed.

$$\text{tok}_i = \left(\frac{\alpha_i}{g^{\beta_i}}\right)^{H(kw)}. \tag{9}$$

Finally, the user tokens are generated $(\text{Tok} = \{\text{tok}_i\})$ and sent to the CS.

5.5. Search. CipherTextSearch $(GP, \text{Tok}, \text{Index}) \longrightarrow D$. The search is conducted by the CS. This process takes the global public key *GP*, the user search token Tok, and the ciphertext index Index as input. Suppose the ciphertext search is successful, output the ciphertext. Otherwise, the process is terminated.

(1) Judge if the following equation holds:

$$Idx_{1,\omega} = \prod_{i\in U_i} e\left(Idx_{2,i}, \mathrm{tok}_i\right). \tag{10}$$

(2) If yes, output the storage ciphertext D; else, abort.

5.6. Decryption. The decryption consists of two processes, namely, the proxy decryption process $\mathrm{ProxyDecrypt}(GP, D, UK_{\mathrm{UID}}) \longrightarrow D'$ and the user decryption process $\mathrm{userDecrypt}(GP, \mathrm{D}, D', UK_{\mathrm{UID}}) \longrightarrow M$.

$\mathrm{ProxyDecrypt}(GP, D, UK_{\mathrm{UID}}) \longrightarrow D'$. Proxy decryption is run by the CS, taking the global public key GP, the ciphertext D, and the user identity key UK_{UID} as input. Determine whether the user attributes satisfy the file access permission, whether the ciphertext attribute set Λ satisfies the ciphertext access structure (P, η), and whether the user attribute set K satisfies the user access structure (F, ρ).

Verify that the user attribute set satisfies the user access structure; randomly selected $c_x \in \mathbb{Z}_P^*$ makes $\sum_{x\in\theta} c_x\lambda_x = s, \sum_{x\in\theta} c_x\mu_x = 0$. Similarly, verify that the ciphertext attribute set satisfies the ciphertext access structure; randomly selected $d_y \in \mathbb{Z}_P^*$ makes $\sum_{y\in\omega} d_y\sigma_y = \varphi, \sum_{y\in\omega} d_y\tau_y = 0$. If the authentication succeeds, perform the following calculation for the ciphertext pre-decryption.

Pre-decryption equation:

$$\begin{aligned} D' &= \frac{\prod_{x\in K}\left(D_{x,1}\cdot e\left(H(\mathrm{UID}), D_{x,3}\right)/e\left(\check{K}_{\rho(x)}, D_{x,2}\right)\right)^{c_x}}{\prod_{y\in\Lambda}\left(U_{y,1}\cdot e\left(H(\mathrm{UID}), U_{y,3}\right)/e\left(\check{C}_{\eta(y)}, U_{y,2}\right)\right)^{d_y}} \\ &= \frac{\prod_{x\in K}\left(e(g,g)^{\sigma_x} e(H(UID), g)^{\tau_x}\right)^{c_x}}{\prod_{y\in\Lambda}\left(e(g,g)^{\lambda_y} e(H(UID), g)^{\mu_y}\right)^{d_y}} \\ &= \frac{e(g,g)^s}{e(g,g)^{\varphi}}. \end{aligned} \tag{11}$$

$\mathrm{userDecrypt}(GP, D, D', UK_{\mathrm{UID}}) \longrightarrow M$. The user decryption is run by DA, taking the global public key GP, the ciphertext D, the intermediate ciphertext D', and the user identity key UK_{UID} as input.

Decryption equation:

$$M = \frac{D}{D'\cdot U}. \tag{12}$$

6. Security and Performance Analysis

6.1. Security Analysis. The STW-ABE simplifies the security problem to a decisional bilinear Diffie–Hellman (DBDH) problem.

Theorem 1. *The STW-ABE scheme is IND-CPA secure if the decisional bilinear Diffie–Hellman (DBDH) problem is hard.*

Proof. If adversary A can break the STW-ABE scheme with a non-negligible advantage, adversary A can solve the DBDH problem with a non-negligible advantage. Aq-order bilinear group $\mathbb{G}_0$ with generator g and bilinear mapping $\mathbb{G}_0 \times \mathbb{G}_0 \longrightarrow \mathbb{G}_T$ exists. C plays as the challenger in the following steps. Given an instance of the DBDH problem (g, g^a, g^b, g^c, Z), where a, b, c, $z \in \mathbb{Z}_q$ are randomly selected. $\rho \in \{0, 1\}$; when $\rho = 0$, $Z = e(g,g)^{\mathrm{abc}}$; when $\rho = 1$, $Z = e(g,g)^z$.

Initialization. C runs the *Initialization* of STW-ABE and returns the global public key GP, user public key PK, and user private key SK to A.

Query Phase I. Adversary A queries the following oracles adaptively.

User Identity Key Oracle. A submits an identity UID to C. C runs the $\mathrm{KeyGen}(GP, SK, PK, \mathrm{UID}, K, (P, \eta)) \longrightarrow UK_{\mathrm{UID}}$. Finally, it returns UK_{UID} to A.

Encryption Oracle. A sends $((F, \rho), \Lambda, M)$ to C. C runs the $\mathrm{Encrypt}(GP, SK, PK, (F, \rho), \Lambda, M)$ to generate the ciphertext D. Notice that the primary access structure (F, ρ) does not satisfy the challenge primary attribute set K, and ciphertext attribute set Λ does not satisfy the challenge secondary access structure (P, η).

Challenge. A submits two plaintexts of equal length M_0, M_1 and sends them to C. C selects a random number $\partial \in \{0, 1\}$ and then encrypts the selected plaintext with user access structure (F^*, ρ^*) and ciphertext attribute set Λ^*. The final ciphertext D^* will be generated and sent to A.

Query Phase II. A still can make queries adaptively as in *Query Phase I* after receiving the challenge ciphertext D. Similarly, A cannot query the user access structure that satisfies the challenge user attribute set and the ciphertext attribute set that satisfies the ciphertext access structure.

Guess. A outputs a guess ∂' for ∂. If $\partial' = \partial$, C outputs $\rho' = 0$, and C receives a tuple $(g, g^a, g^b, g^c, e(g,g)^{\mathrm{abc}})$. Otherwise, C outputs $\rho' = 1$, and C receives a tuple $(g, g^a, g^b, g^c, e(g,g)^z)$. The advantage of A is analyzed as follows.

When $\rho = 1$, $Z = e(g,g)^z$, A cannot obtain the information of D. Thus, $\Pr[\partial' \neq \partial | \partial = 1] = 1/2$. When $\rho' = 1$, $\Pr[\rho' = \rho | \rho = 1] = 1/2$.

When $\rho = 0$, $Z = e(g,g)^{\mathrm{abc}}$, A obtains ciphertext D. Thus, $\Pr[\partial' = \partial | \partial = 0] = 1/2 + \epsilon$. When $\rho' = 0$, $\Pr[\rho' = \rho | \rho = 0] = 1/2 + \epsilon$.

Thus, C guesses $\rho' = \rho$, and the correct advantage is

$$\begin{aligned} \mathrm{Adv} = \Pr[\rho' = \rho] - \frac{1}{2} &= \frac{1}{2}\Pr[\rho' = \rho | \rho = 1] \\ &+ \frac{1}{2}\Pr[\rho' = \rho | \rho = 0] - \frac{1}{2} = \frac{\epsilon}{2}. \end{aligned} \tag{13}$$

In summary, if adversary A can break the proposed scheme with a non-negligible advantage in polynomial time, a scheme that can solve the DBDH problem with a non-negligible advantage $\epsilon/2$ in polynomial time exists. However, the DBDH problem is difficult, so the STW-ABE scheme is IND-CPA secure. □

Theorem 2. *The STW-ABE scheme is IND-CKA secure if the decisional bilinear Diffie–Hellman (DBDH) problem is hard.*

Proof. Assume that there is a PPT adversary A who can win the index indistinguishability security game defined in Section 4.3.2 with non-negligible advantage ε. Then, we can construct a C to solve the DBDH problem with a non-negligible advantage $(\varepsilon/2)$. C plays as the challenger in the following steps. Given an instance of the DBDH problem (g, g^a, g^b, g^c, Z), where $a, b, c, z \in \mathbb{Z}_q$ are randomly selected. $\rho \in \{0, 1\}$; when $\rho = 0$, $Z = e(g, g)^{\mathrm{abc}}$; when $\rho = 1$, $Z = e(g, g)^z$.

Initialization. A defines a user access structure (F^*, ρ^*) and ciphertext attribute set Λ^*.

Setup. C runs the *Initialization* of STW-ABE and returns the global public key GP, user public key PK, and user private key SK to A.

Query Phase I. Adversary A queries the following oracles adaptively.

User Identity Key Oracle. A submits an identity UID to C. C runs the $\mathrm{KeyGen}(GP, SK, PK, UID, K, (P, \eta)) \longrightarrow UK_{\mathrm{UID}}$. Finally, it returns UK_{UID} to A.

Token Oracle. A sends (kw) to C. C runs the $\mathrm{TokenGen}(GP, SK, kw) \longrightarrow \mathrm{Tok}$ to generate the token Tok. Notice that the user access structure (F, ρ) does not satisfy the challenge user attribute set K, and the ciphertext attribute set Λ does not satisfy the challenge ciphertext access structure (P, η). We assume that all query results (Tok) have at least one matched index that can be searched out.

Index Oracle. A submits $(UK_{\mathrm{UID}}, \{KW\})$ to C, and C runs the $\mathrm{IndexGen}(GP, PK, UK_{\mathrm{UID}}, KW)$ to generate the index.

Challenge. A submits two keywords of equal length kw_0 and kw_1 to C. C chooses number $b \in \{0, 1\}$ randomly and runs the $\mathrm{IndexGen}(GP, PK, UK_{UID}, KW) \longrightarrow \mathrm{Index}$ with the challenge user access structure (F^*, ρ^*) and ciphertext attribute set Λ^* to return $Inde\ x^*$ to A.

$$Idx^*_{1,kw_b} = Z^{H(\mathrm{kw}_b)}, Idx^*_2 = A^b. \tag{14}$$

The advantage of A is analyzed as follows.

When $Z = e(g, g)^{\mathrm{abc}}$, we set $s = a, \alpha = bc$; then, the index presented as follows is identical to an actual index:

$$\begin{aligned} Idx^*_{1,kw_b} &= \left(e(g, g)^{\mathrm{abc}}\right)^{H(\mathrm{kw}_b)} \\ &= e(g, g)^{\alpha\cdot\varphi\cdot H(\mathrm{kw}_b)} \\ Idx_2 &= g^{b\cdot\varphi}. \end{aligned} \tag{15}$$

When $Z = e(g, g)^z$, due to the randomness of z, this index is random to the adversary and contains no information about b.

Query Phase II. A still can make queries adaptively as in *Query Phase I* after receiving the challenge *Index*. Similarly, A cannot query the user access structure that satisfies the challenge user attribute set and the ciphertext attribute set that satisfies the ciphertext access structure.

Guess. A outputs a guess b' for b. If $Z = e(g, g)^{\mathrm{abc}}$, the probability of A outputs $b' = b$ is $1/2 + \epsilon$. If $Z = e(g, g)^z$, the probability of A outputs $b' = b$ is $1/2$. Thus, the advantage of C solving the DBDH problem is

$$\begin{aligned} \mathrm{Adv} &= \left|\frac{1}{2}\Pr\left[b' = b|Z = e(g, g)^{abc}\right]\right. \\ &\quad \left. + \frac{1}{2}\Pr\left[b' = b|Z = e(g, g)^z\right] - \frac{1}{2}\right| \\ &= \left|\left[\frac{1}{2}\left(\frac{1}{2} + \epsilon\right) + \frac{1}{2}\cdot\frac{1}{2}\right] - \frac{1}{2}\right| \\ &= \frac{\epsilon}{2}. \end{aligned} \tag{16}$$

Because the DBDH problem is hard, we can get that $\epsilon/2$ is negligible. In other words, the advantage of A breaking our scheme is negligible, and our scheme achieves chosen keyword security. □

6.2. Performance Analysis. In this section, we analyze the performance and computational efficiency of STW-ABE. We compare the performance of STW-ABE with other schemes in Table 2, where "√" indicates that the solution supports this method. "×" indicates that the solution does not support this method. In Table 3, we compare the computational efficiency of STW-ABE with other schemes, in which E represents an exponential operation, P represents a pairing operation, H represents a hash operation, i represents the number of attributes in the authorized institution, n represents the number of keywords in each document, m is the number of keywords searched by the user, M_i is the number of the ciphertext attributes, and M_e is the number of the user attributes.

As seen in Table 2, our scheme not only realizes two-way access control of ciphertext search but also uses CS to provide outsourced decryption service, reducing the computational pressure on users.

Table 3 compares the computational efficiency of encryption, decryption, index generation, token generation,

Table 2: Comparison of functions.

Scheme	DP-ABE	PAB-MSK	D-ABE	BC-SABE	STW-ABE
Two-way	√	×	×	×	√
Searchable encryption	×	√	×	√	√
Proxy encryption	√	√	√	√	√

Table 3: Comparison of computational methods.

Scheme	DP-ABE	PAB-MSK	D-ABE	BC-SABE	STW-ABE
Encryption	$(6M_i + 5)E + H$	$(3n + 3)E + H$	$(5i + 1)E$	$(4i + 3)E$	$(3M_i + 2)E + H$
Index generation	—	$(3n + 3)E$	—	$(3n + 1)E + H$	$(2i + n)E + H$
Token generation	—	$(2m + 3)E$	—	$(2m + 2)E + H$	$(i + m)E + H$
Search	—	$E + (2n + 4)P$	—	$iE + (2i + 1)P$	$M_iE + (i + m)P$
Decryption	User: $2E$ Cloud: $2E + ((M_i + M_e)P$	$iE + 2iP$	$iE + 2iP$	User: E Cloud: $iE + (3i + 2)P$	*User*: E *Cloud*:$(M_i + M_e)$P

and ciphertext search. In our scheme, first, the user needs to perform a $(3M_i + 2)$ exponential operation and a hash operation to encrypt the data, in which only one exponential operation is required for each ciphertext attribute. The user performs a hash operation on each ciphertext keyword and $(2i + n)$ exponential operation to generate a ciphertext index. Secondly, the cloud server performs a hash operation and $(i + m)$ exponential operation for each keyword to be searched to generate a token for data users. Then, the cloud server performs a ciphertext search by an exponential operation of M_i and pairing operation time of $(i + m)$. In decryption, the scheme divides the decryption cost into the user part (denoted as User) and the cloud server part (denoted as Cloud). The cloud server performs the pairing operation $(M_i + M_e)$. The user then only needs to perform the exponential operation once to decrypt the ciphertext into plaintext. Furthermore, the STW-ABE scheme is compared with two multi-permission ABE schemes, DP-ABE [5] and D-ABE [21], and two searchable encryption schemes, PAB-MSK [11] and BC-SABE [19], in Table 3. The cost of linear secret-sharing protocol is ignored in efficiency analysis.

Figures 2 and 3 contain the simulation results of the five processes. We simulated this on an Ubuntu 16 desktop system. The system has an Intel Core i7-8700 CPU and 4GB RAM. All programs were developed using Charm (version 0.50) [27], a rapid prototyping framework based on the Python encryption scheme.

Figure 2(a) shows the encryption time cost of three ABE schemes with multiple authority agencies. As seen in the figure, the time cost of all the resulting schemes has a linear relationship with the number of attributes contained in the encrypted access structure. Figure 2(b) shows the decryption time cost of schemes D-ABE, BC-SABE, and STW-ABE. As seen in Figure 2, the time cost of cloud decryption of D-ABE, BC-SABE, and STW-ABE has a linear relationship with the number of attributes, and the user decryption cost in STW-ABE is independent of the number of attributes. Since STW-ABE outsources most of the decryption work to cloud servers, the computing pressure on users is greatly reduced. This scheme is more suitable for using lightweight devices in the IoT environment.

Figure 3(a) shows the time cost of the generated ciphertext index. It can be seen from the figure that STW-ABE has better computing performance than scheme PAB-MKS and scheme BC-SABE. Figure 3(b) shows the simulation results of the time cost required for the generation of the Token. The time cost of the schemes is linearly related to the number of attributes, but it can be seen that the time cost of STW-ABE is much shorter than that of scheme BC-SABE. Figure 3(c) shows the time cost of the search process under simulation. In this scheme, the file index and the number of files are fixed as simulated constants. The results of the search process represent only the performance of the search process and do not include the time cost of searching the actual database. As seen in Figure 3(c), the time cost of the STW-ABE search process is similar to that of the search process in the PAB-MKS scheme and the BC-SABE scheme. Moreover, they are all linearly related to the number of attributes.

Discussion. There are two concerns when designing searchable encryption access control schemes. *(1) Security*. In Section 4.1, the two-way access control scheme based on searchable encryption has been proven to be IND-CPA security and IND-CKA security, which is also achieved by most searchable encryption schemes. In this paper, we use the distributed feature of blockchain and change the central authorization model in traditional access control to a blockchain consensus node with DKG that generates the relevant secret key. Where DKG follows the(t, n) $(n \leq t)$-sharing principle,t is the total number of blockchain consensus nodes involved in key generation, and$n$is the minimum number of consensus nodes involved in key generation. The secret key sharing must be participated by more than n consensus nodes, thus improving the robustness of the scheme. At the same time, a blockchain is a distributed ledger that ensures data integrity, immutability, and traceability of the information stored in it such as global public keys and user keys. *(2) Efficiency*. In this paper, the simulation experiment simulated the efficiency of the scheme and compared it with other schemes. It can be found that this scheme has certain advantages in implementation

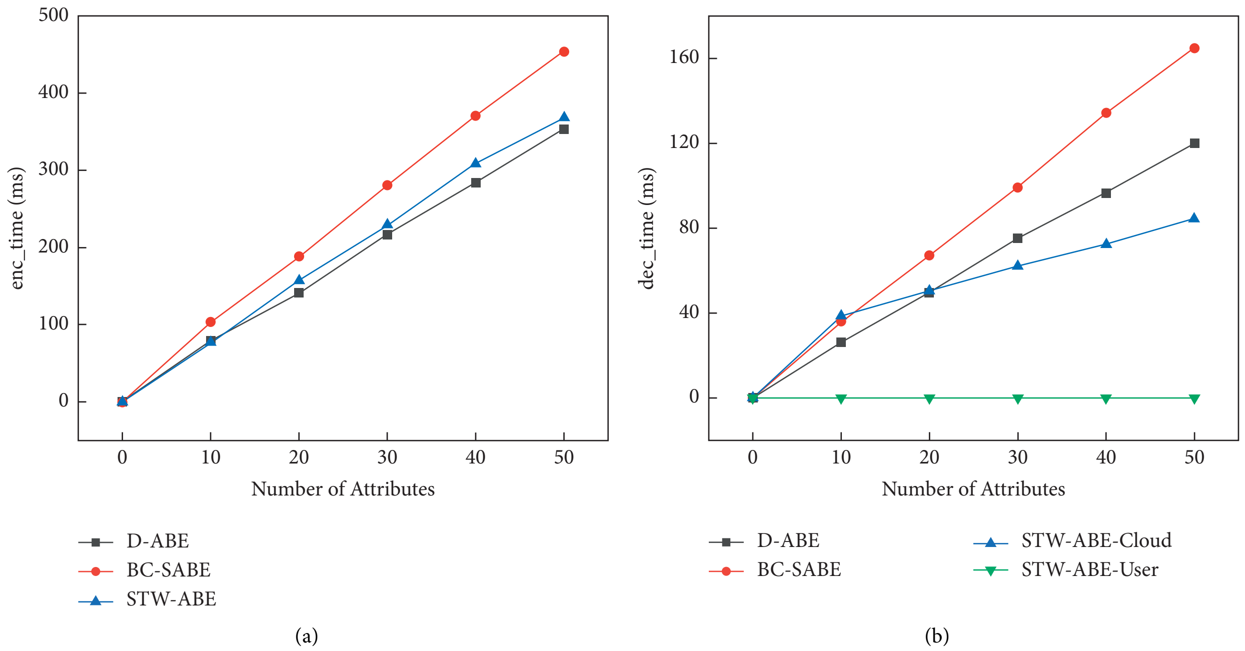

FIGURE 2: Time cost of encryption and decryption. (a) Encryption time. (b) Decryption time.

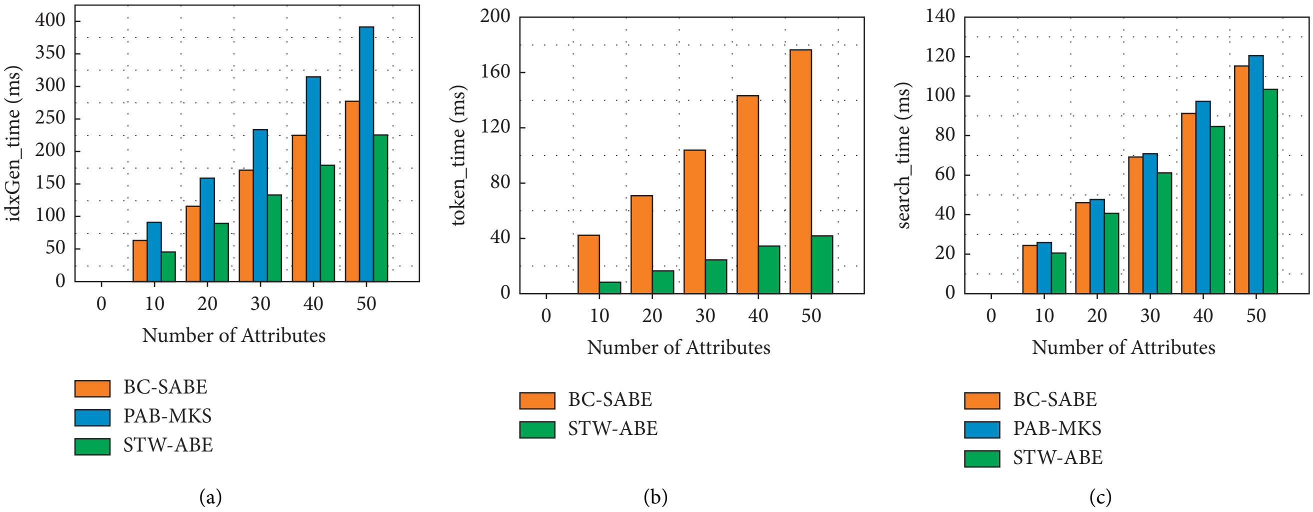

FIGURE 3: Time costs of ciphertext search. (a) Ciphertext index generation time. (b) Token generation time. (c) Ciphertext search time.

efficiency. In the decryption process, considering that most IoT devices have limited resources and cannot perform efficient decryption calculations, this scheme uses cloud servers to assist users in decryption. A large amount of decryption computation is outsourced to CS, thus reducing the computational pressure on users. This scheme adopts a distributed key generation protocol, and multiple blockchain consensus nodes participate in generating users' public and private key pairs, which will not affect the security and robustness of the scheme. Meanwhile, the secret keys generated by blockchain nodes do not need to respond to user requests in real time, so the time cost of secret key generation is not simulated in this paper. Among them, in the BC-SABE scheme, the cloud server is used to complete the generation of tokens with the user jointly, and the user does not need to perform the calculation related to the number of attributes. In the BC-SABE scheme, the token generation time by the blockchain consensus node is not given, so there is no comparison between them in Figure 3(b). Similarly, the generation of a token in STW-ABE is completed by the blockchain consensus node. The user does not need to calculate the consumption in the generation of tokens.

7. Conclusions

This paper proposes a distributed STW-ABE scheme using coalition blockchain and cloud servers to assist users with accurate and secure data search and sharing. Our solution not only enables two-way confirmation between users and data but also enables the fine-grained search of ciphertext and lightweight decryption for users. In addition, our scheme utilizes a coalition blockchain to replace the centralized key management server. The consensus nodes jointly generate key parameters through the DKG, improving the security of the IoT system. Then, the blockchain is responsible for generating public and private keys, user identity keys, and keyword tokens. Due to the limited resources of IoT devices and the massive pairing operations required for the search and decryption process, we delegate a large amount of computation to CS during the search and decryption process. The user only needs one exponential operation to complete the decryption process from ciphertext to plaintext. The present security and efficiency analysis shows that the scheme has good safety and practicality.

Our ultimate aim is to design a secure and efficient data sharing system for the IIoT. The possible further research direction is to implement dynamic updating of access policies based on the current work.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (grant no. 61772180), the Key-Area Research and Development Program of Guangdong Province (2020B1111420002), the Key-Area Research and Development Program of Hubei Province (2022BAA040), the Science and Technology Project of Department of Transport of Hubei Province (2022-11-4-3), and the Innovation Fund of Hubei University of Technology (BSQD2019027, BSQD2019020, and BSQD2016019).

References

[1] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proceedings of the International Conference on Theory & Applications of Cryptographic Techniques*, Springer, Berlin, Heidelberg, 2005.

[2] B. Waters, "Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization," in *Proceedings of the International Workshop on Public Key Cryptography*, Springer, Berlin Heidelberg, 2008.

[3] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security*, Alexandria, VA, USA, 2006.

[4] N. Attrapadung and H. Imai, "Dual-policy attribute based encryption," in *Proceedings of the 7th International Conference on Applied Cryptography and Network Security*, Springer, Berlin, Heidelberg, 2009.

[5] D. Han, J. Chen, L. Zhang, Y. Shen, and Y. Gao, "Access control of blockchain based on dual-policy attribute-based encryption," in *Proceedings of the 2020 IEEE 22nd International Conference on High Performance Computing and Communications; IEEE 18th International Conference on Smart City; IEEE 6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, Yanuca Island, Cuvu, Fiji, December 2020.

[6] J. Leng, M. Zhou, J. L. Zhao, Y. Huang, and Y. Bian, "Makerchain: a blockchain with chemical signature for self-organizing process in social manufacturing," *Journal of Cleaner Production*, vol. 234, pp. 767–778, 2019.

[7] Z. Rahman, I. Khalil, X. Yi, and M. Atiquzzaman, "Blockchain-based security framework for a critical industry 4.0 cyber-physical system," *IEEE Communications Magazine*, vol. 59, no. 5, pp. 128–134, 2021.

[8] J. Li, X. Lin, Y. Zhang, and J. Han, "KSF-OABE: outsourced attribute-based encryption with keyword search function for cloud storage," *IEEE Transactions on Services Computing*, vol. 10, no. 5, pp. 715–725, 2017.

[9] D. Ziegler and A. Marsalek, "Efficient Revocable Attribute-Based Encryption with Hidden Policies," in *Proceedings of the 2020 IEEE 19th International Conference On Trust, Security And Privacy In Computing And Communications (TrustCom)*, Guangzhou, China, 2020.

[10] X. Liu, G. Yang, W. Susilo, J. Tonien, X. Liu, and J. Shen, "Privacy-preserving multi-keyword searchable encryption for distributed systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 3, pp. 561–574, 2021.

[11] Y. Miao, J. Ma, X. Liu, X. Li, Z. Liu, and H. Li, "Practical attribute-based multi-keyword search scheme in mobile crowdsourcing," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3008–3018, 2018.

[12] J. Leng, G. Ruan, P. Jiang et al., "Blockchain-empowered sustainable manufacturing and product lifecycle management in industry 4.0: a survey," *Renewable and Sustainable Energy Reviews*, vol. 132, Article ID 110112, 2020.

[13] D. Mehta, T. Sudeep, B. Umesh, and S. Arpit, "Blockchain-based royalty contract transactions scheme for industry 4.0 supply-chain management-," *Sciencedirect. Information Processing & Management*, vol. 58, no. 4.

[14] J. Leng, M. Zhou, J. L. Zhao, Y. Huang, and Y. Bian, "Blockchain Security: A Survey of Techniques and Research Directions," *IEEE Transactions on Services Computing*, vol. 15, 2020.

[15] J. Leng, S. Ye, M. Zhou et al., "Blockchain-secured smart manufacturing in industry 4.0: a survey," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 1, pp. 237–252, 2021.

[16] M. Li, C. Jia, and W. Shao, "Blockchain based multi-keyword similarity search scheme over encrypted data Security and Privacy in Communication Networks," in *Security and Privacy in Communication Networks. SecureComm 2020. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, Vol. 336, Springer, Cham, Switzerland, 2020.

[17] T. Feng, H. Pei, R. Ma, Y. Tian, and X. Feng, "Blockchain data privacy access control based on searchable attribute encryption," *Computers, Materials & Continua*, vol. 66, no. 1, pp. 871–890, 2020.

[18] S. Gao, G. Piao, J. Zhu, X. Ma, and J. Ma, "Trustaccess: a trustworthy secure ciphertext-policy and attribute hiding

access control scheme based on blockchain," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 5784–5798, 2020.

[19] S. Liu, J. Yu, Y. Xiao, Z. Wan, S. Wang, and B. Yan, "Bc-sabe: blockchain-aided searchable attribute-based encryption for cloud-iot," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 7851–7867, 2020.

[20] X. Qin, Y. Huang, Z. Yang, and X. Li, "Lbac: a lightweight blockchain-based access control scheme for the internet of things -," *Information Sciences*, vol. 554, pp. 222–235, 2021.

[21] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, Berlin, Heidelberg, 2011.

[22] X. Qin, Y. Huang, Z. Yang, and X. Li, "A blockchain-based access control scheme with multiple attribute authorities for secure cloud data sharing," *Journal of Systems Architecture*, vol. 112, no. 11, Article ID 101854, 2021.

[23] N. Shi, L. Tan, C. Yang, C. He, and H. Xu, "Bacs: a blockchain-based access control scheme in distributed internet of things," *Peer-to-Peer Networking and Applications*, vol. 14, no. 6, 2020.

[24] A. Beimel, *Secure Schemes for Secret Sharing and Key Distribution*, Israel Institute of Technology-Technion, Haifa, Israel, 1996.

[25] T. P. Pedersen, "A threshold cryptosystem without a trusted party (extended abstract)," in *Proceedings of the Advances in Cryptology - EUROCRYPT '91, Workshop on the Theory and Application of of Cryptographic Techniques*, Brighton, UK, 1991.

[26] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proceeding of the 2000 IEEE Symposium on Security and Privacy. Science Progress 2000*, pp. 44–55, Berkeley, CA, USA, May 2000.

[27] J. A. Akinyele, C. Garman, I. Miers et al., "Charm: a framework for rapidly prototyping cryptosystems," *Journal of Cryptographic Engineering*, vol. 3, no. 2, pp. 111–128, 2013.