Computers, Materials & Continua
DOI:10.32604/cmc.2021.017227
Article

Tech Science Press

Towards Privacy-Preserving Cloud Storage: A Blockchain Approach

Jia-Shun Zhang[1], Gang Xu[2,*], Xiu-Bo Chen[1], Haseeb Ahmad[3], Xin Liu[4] and Wen Liu[5,6,7]

[1]Information Security Center, State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, 100876, China
[2]School of Information Science and Technology, North China University of Technology, Beijing, 100144, China
[3]Department of Computer Science, National Textile University, Faisalabad, 37610, Pakistan
[4]Inner Mongolia University of Science & Technology, School of Information Engineering, Baotou, 014010, China
[5]State Key Laboratory of Media Convergence and Communication, Communication University of China
[6]School of Computer Science and Cybersecurity, Communication University of China
[7]Key Laboratory of Convergent Media and Intelligent Technology Communication University of China, Ministry of Education, Beijing, 102204, China
[*]Corresponding Author: Gang Xu. Email: gangxu_bupt@163.com
Received: 24 January 2021; Accepted: 11 March 2021

Abstract: With the rapid development of cloud computing technology, cloud services have now become a new business model for information services. The cloud server provides the IT resources required by customers in a self-service manner through the network, realizing business expansion and rapid innovation. However, due to the insufficient protection of data privacy, the problem of data privacy leakage in cloud storage is threatening cloud computing. To address the problem, we propose BC-PECK, a data protection scheme based on blockchain and public key searchable encryption. Firstly, all the data is protected by the encryption algorithm. The privacy data is encrypted and stored in a cloud server, while the ciphertext index is established by a public key searchable encryption scheme and stored on the blockchain. Secondly, based on the characteristics of trusted execution of smart contract technology, a control mechanism for data accessing and sharing is given. Data transaction is automatically recorded on the blockchain, which is fairer under the premise of ensuring the privacy and security of the data sharing process. Finally, we analyzed the security and fairness of the current scheme. Through the comparison with similar schemes, we have shown the advantages of the proposed scheme.

Keywords: Blockchain; privacy protection; smart contract; public encryption keyword search

1 Introduction

In recent years, the vigorous development of cloud computing technology has emerged as a new vitality in the information industry. Compared with traditional IT system, cloud computing provides various convenient and reliable services by deploying servers in the cloud, allowing users to manage their business flexibly and efficiently. However, at the same time, user's weak

control of cloud server has brought security problem of data stored in the cloud server. In this context, encrypting has become an effective way to enhance the privacy and security of cloud-stored data. Schemes such as homomorphic encryption [1], provable data possession [2], proof of retrievability [3] and some related schemes [4–7] mainly focus on the data integrity and confidentiality, but the data sharing process is inefficient. With the development of cryptography, Searchable Encryption (SE) technology has received widespread attention which can balance the security of cloud storage data and convenience of management.

According to the encryption method, Searchable Encryption can be divided into Symmetric Searchable Encryption (SSE) and Public key Encryption with Keyword Search (PEKS). In 2000, Song et al. [8] firstly constructed an SSE scheme using a linear scan method of keyword ciphertext based on symmetric encryption. Since SSE uses a symmetric key to encrypt ciphertext, it is more suitable for single-user scenarios. In 2004, Boneh et al. [9] switched symmetric encryption to public-key encryption technology and proposed a PEKS scheme in the background of email data encryption. Boneh et al. used a public-key encryption algorithm in their scheme and upgraded the single-user model to a multi-user model. They also defined data owners, data users, and cloud storage servers in their scheme. This change makes searchable encryption technology more suitable for data storage and exchange in the multi-user scenario, thereby increasing the value of the searchable encryption technology.

In schemes [8,9], there is only one index keyword for retrieving ciphertext, and it should be accurate and correct, which makes data search easily fails due to the existence of synonyms. To improve the usability of data searching, Li et al. [10] gave a symmetric fuzzy keyword searchable encryption scheme using a keyword edit distance method. When the keyword spelling or format is wrong, their scheme can compare the similarity of keywords and obtain correct search results, thereby enhancing the practicality of keyword search. In 2013, based on homomorphic encryption technology, Dong et al. [11] proposed an interactive fuzzy keyword PEKS scheme. Compared with the fuzzy search scheme of Li et al., the scheme of Dong et al. improves the computational efficiency.

Another method to enhance the accuracy of the search process is the searchable encryption technology that supports joint keyword queries. This idea was first proposed by Park et al. [12]. They constructed two Public Key Encryption with Conjunctive Field Keyword Search (PECK) schemes. Adding a single query keyword to multiple keyword queries effectively improve the accuracy of the search, but there are some shortages in schemes of Park et al. The first scheme of Park et al. requires multiple bilinear pairing calculations to encrypt keywords, while the second scheme requires a private key proportional to the number of keywords, increasing the complexity of schemes. Later in 2007, Hwang et al. [13] proposed an efficient PECK scheme based on Decision Linear Diffie-Hellman (DLDH) problem, which can effectively implement multi-keyword joint query by using only one key. In 2013, Hu et al. [14] designed a PECK scheme to sort the search results. By sorting, only the most similar results are returned, which guarantees the multi-keyword query while improving the performance of the query. Wang et al. [15] gave a sortable PECK scheme based on the inverted index technology in 2015. Using multiplication and exponentiation operations, this scheme performs better than previous bilinear calculation schemes. It causes less computational overhead in practical applications. Wu et al. [16] realized a searchable encryption scheme with revocable data access rights while supporting joint keyword query, thereby improving the management ability of PEKS.

In the above schemes, the data retrieval process is performed by the cloud server that stores target data. Therefore, the cloud server in the above literature is specified as honest but

curious, that is, it will not maliciously and intentionally perform search operations by mistake. In other words, the above schemes are difficult to resist attacks from malicious servers due to credibility defects.

In the last decade, blockchain technology has become a research hotspot. As an emerging distributed database technology, blockchain technology has the characteristics of credibility and unforgeability. With the development of technology, more blockchain applications with unique features are gradually documented [17–20]. Compared to those blockchain-based encrypted digital currencies [21], one of the major breakthroughs is the blockchain application that implements smart contract. With the help of smart contract, blockchain supports fixed-step trusted transactions without any third party. These transactions are traceable and undeniable. This feature makes blockchain qualified as a reliable manager in the data management process. Wang et al. [22] proposed a blockchain-based data protection method in the Internet of Things environment through a new type of RSA accumulator. While Deng et al. [23] designed an electronic medical record protect scheme based on proofs of retrievability and blockchain. In 2020, Gao et al. [24] gave a blockchain-based data protection scheme, which effectively provides a reliable control method for data privacy in the multimedia field.

The trusted, traceable, and non-tamperable characteristics of the blockchain coincide with a fair and trustworthy server environment in searchable encryption. With the development of blockchain technology, SE starts combining blockchain technology. For example, Hu et al. [25] combined SSE with Ethereum and designed a scheme for effective management and encryption of E-mail data. Chen et al. [26] designed and implemented a medical record transaction scheme based on SE technology. In 2020, based on blockchain technology, Yan et al. [27] gave a SE scheme that can achieve more fine-grained access control by combining attributes encryption and symmetric searchable encryption, effectively restricting malicious server.

Considering the lack of the cloud server's credibility in the existing SE schemes, this article combines public-key searchable encryption and blockchain technology to construct a PECK scheme based on smart contracts. This scheme aims to solve the privacy and credibility issues in the context of cloud storage. The main contributions of this article are as follows:

(1) We propose BC-PECK, a cloud storage data privacy protection scheme based on blockchain and smart contracts. Blockchain and smart contract are fair and credible to ensure the data sharing process. It demonstrates the potential of blockchain technology in enhancing data privacy protection.
(2) We introduce a PECK technology in our scheme. The data sharing process in multi-user scenarios is realized with the help of PECK. Moreover, our scheme allows a more complex and accurate query process using multi-keyword retrieval. This effectively enhances the practicability of our scheme.
(3) We prove the security of BC-PECK, and ensured that our scheme could effectively prevent the leakage of private data and realize ciphertext query simultaneously. We also ensure the feasibility of our scheme in practical applications through performance analysis.

The rest of our article is organized as follows. In Section 2, background knowledge is introduced. The model and algorithm flow of the scheme are given in Section 3. In Section 4, we describe the detailed execution process of the scheme. In Section 5, we analyze and evaluates the performance of our scheme. Finally, in Section 6, the whole article is summarized, and the future research directions are discussed.

2 Preliminaries

2.1 Ethereum & Smart Contract

Ethereum is a blockchain platform that uses proof of work consensus mechanism and generates chain structure. The Ethereum client constructs a turing complete virtual machine called Ethereum virtual machine [17]. It is associated with a programming language called Solidity. Therefore, Ethereum is programmable, and can deploy smart contract programs for distributed trusted applications.

Smart contract is defined by [28] as a method to automatically execute pre-defined electronic contracts. With the development of blockchain technology, smart contract has received extensive attention in the field of blockchain. In Ethereum, smart contract is programmed according to the predetermined process, and then deployed to the Ethereum blockchain network. Smart contract ensures that its content is fixed and cannot be changed, which coincides with the characteristics of Ethereum. Through smart contract, the fairness and credibility of Ethereum can be effectively enhanced.

2.2 Bilinear Pair

Supposing $\mathbb{G}_1, \mathbb{G}_2$ are two cyclic groups of order p, and g is a generator of $\mathbb{G}_1$. There is a mapping $\hat{e}: \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$, then this set of mappings should satisfy:

(1) Bilinear. For mapping $\hat{e}: \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$, for any $x, y \in \mathbb{G}_1, a, b \in \mathbb{Z}_p^*$, it satisfies in a mapping $\hat{e}\hat{e}\left(x^a, y^b\right) = \hat{e}(x, y)^{ab}$.
(2) Non-degenerate. There exists $x \in \mathbb{G}_1$ that $\hat{e}(x, x) \neq 1$.
(3) Computable. There are efficient algorithms making it possible to compute $\hat{e}(x, y)$ for any $x, y \in \mathbb{G}_1$.

2.3 Decision Linear Diffie-Hellman Assumption

The DLDH hypothesis was proposed by Boneh et al. [29]. It is assumed that there are groups $\mathbb{G}_1, \mathbb{G}_2$ and map $\hat{e}: \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$, and g_1, g_2, g_3 as a random element of $\mathbb{G}_1$, a, b, c are random elements of $\mathbb{Z}_p^*$. Then the DLDH problem in $\mathbb{G}_1$ is defined as, for two six-tuples $\left(g_1, g_2, g_3, g_1^a, g_2^b, g_3^c\right)$ and $\left(g_1, g_2, g_3, g_1^a, g_2^b, g_3^{a+b}\right)$, the adversary without polynomial time can distinguish the above two six-tuples with non-negligible advantages.

3 System Model

3.1 Definition of System Structure

The structure of our scheme is shown in Fig. 1. There are four major participants, namely data owner (DO), data user (DU), cloud server (CS) and smart contract (SC). The relationship between each participant is shown in Fig. 1, and the specific operations of each participant are as follows:

(1) Data Owner. DO holds documents to be stored on CS and these data can be selling to other data users. The document owned by DO contains multiple keywords. DO will extract keywords contained in the file to construct ciphertext index and ciphertext file. Then the corresponding content will be sent to the smart contract and cloud server. In addition, DO needs to generate trapdoors for searching data after submitting requests for data users.

(2) Data User. DU wants to pay a certain price to use the data uploaded by DO, so DU needs to submit a search request to DO through SC. If the search results are correct and the data is complete, DU will pay to DO a bid for the data. Otherwise, there is no need for DU to pay.
(3) Smart Contract. SC stores ciphertext index and verification data. It will verify the correctness of search results through a predefined process. In addition, smart contracts can also use blockchain for fair payments.
(4) Cloud Server. The cloud storage server will store the encrypted files of DO. After SC verifies above data sharing process, CS transmits the corresponding ciphertext file to DU.

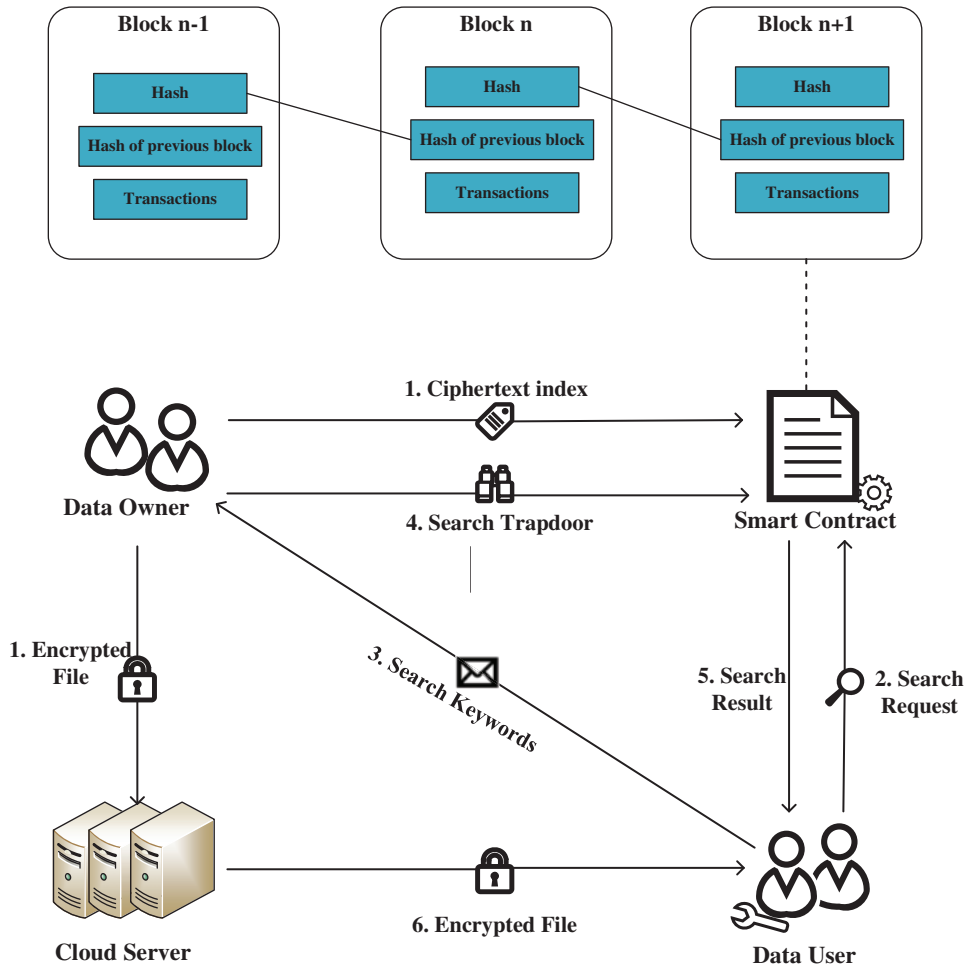

Figure 1: The structure of BC-PECK

3.2 System Work Flow

Based on the PECK algorithm [13], we construct a PECK scheme based on blockchain called BC-PECK. The process can be described by a five-tuple (Setup, Index, Trapdoor, Test, Verify):

(1) Setup: DO generate the public parameter *Params* and the public and private key pair used by each participant in the subsequent steps. In addition, DO also needs to initialize smart contracts for the management process and fair payment.
(2) Index: DO extracts a keyword set W from the data file D and generates an encrypted keyword index S. DO also uses an encryption algorithm to encrypt D and obtain ciphertext file C, calculate the hash value H of C. After sending C to CS, DO obtain the stored serial number N, and send verify data (S, N, H) to SC for fair payment.
(3) Trapdoor: DU requests search authorization from DO through the smart contract. After DO uses SC to confirm the request, DU sends authorized search keywords Q to DO through a secure channel. DO calculates the keyword trapdoor T_q, and sends T_q to SC for retrieval operation.
(4) Test: According to index S generated in previous step and the search trapdoor T_q, SC calculates whether the keyword of the trapdoor query matches the file. When the file matches, SC changes the variable of the data stored in SC as the result of the query. Otherwise, the value of the variable will not be changed.
(5) Verify: According to the query result, DU applies for the ciphertext file C to CS after the query result is the required document, and CS queries SC for query result. After getting the affirmation, send ciphertext to the DU. Then DU calculates the hash value H of the ciphertext after obtaining data to ensure that data has not been tampered with by CS. If the hash H is confirmed to be valid, SC will automatically use the search fee deposited by DU to pay the data fee to DO and storage service fee to CS. Otherwise, DU's money will be refunded to his own account. After verification, the ciphertext file C will be sent to DU. Then DU decrypts the encrypted document and uses it after receiving the ciphertext document from the CS.

3.3 Security Goals

In this subsection, the security goals of our scheme can be divided into following two aspects.

(1) Confidentiality. The confidentiality here means that our scheme does not leak the keyword information during the data storage process and the data query process. Assuming the ciphertext file is safe, then only the confidentiality of ciphertext index and search trapdoor needs to considered. That is to say the adversary cannot obtain extra information through the analysis of ciphertext index and trapdoor.
(2) Fairness. Fairness requires that each role in the system correctly executes the algorithm in accordance with the regulations and completes the content of the specific steps to obtain the corresponding incentives. If the wrong operation of one party leads to failure, he cannot get the corresponding income. Furthermore, the operation that already happened should not be denied. Fairness means every participant is motivated to perform correct operations.

4 The Detail Construction of BC-PECK

In this section, we will introduce the BC-PECK scheme process in detail.

(1) Setup (k): DO generates a public parameter $Params = \left(\mathbb{G}_1, \mathbb{G}_2, \hat{e}, H_1(\cdot), H_2(\cdot), g\right)$ based on the secret parameter k. The elements in *Params* are cyclic group $\mathbb{G}_1$ and $\mathbb{G}_2$, $H_1: \{0,1\}^{\log w} \to \mathbb{G}_1$ and $H_2: \{0,1\}^{\log w} \to \mathbb{G}_1$ are two different anti-collision hash functions, and g is a generator of $\mathbb{G}_1$.

Then DO arbitrarily selects a random number x on $\mathbb{Z}_p^*$, calculates $y = g^x$, and finally obtains a public and private key pair $(pk, sk) = (y, x)$.

In addition, DO also needs to deploy and initialize the smart contract. He also needs to set a purchase price for the data.

(2) Index (pk, W, D): DO extracts the keyword set $W = \{w_1, \cdots w_n\}$ contained in the file D, then uses a common encryption algorithm to encrypt the file $ENC(D) = C$, and calculates the hash value H of the ciphertext C. Then DO generates two random numbers r and s for the keyword set W, and calculates the ciphertext index information $S = \{A, B, C_1, \cdots, C_l\}$ for search, where:

$$h_i = H_1(w_i), f_i = H_2(w_i)$$

$$A = g^r, B = y^s, C_i = h_i^r f_i^s$$

Next, DO transmits the ciphertext data to CS and obtains the stored serial number N of the file, and transmits index and verification information (N, H, S) to SC.

(3) Trapdoor (sk, Q): DU sends a request for searching data to DO through the smart contract. After DO confirms it on SC, DU calculates hash of Q and a random value *salt*, and send $H(Q + salt)$ to SC. Then he sends the search keyword set $Q = \{I_1, \cdots, I_m, w_{I_1}, \cdots, w_{I_m}\}$ to DO through a secure channel. After DO obtains the search keyword set Q, he generates a random number t and calculates:

$$T_{Q,1} = g^t, T_{Q,2} = \left(h_{I_1} \cdots h_{I_m}\right)^t, T_{Q,3} = \left(f_{I_1} \cdots f_{I_m}\right)^{t/x}$$

Finally, DO construct the search trapdoor $T_Q = \{I_1, \cdots, I_m, w_{I_1}, \cdots, w_{I_m}\}$, and upload T_Q to SC to execute the search operation.

(4) Test (pk, S, T_q): After SC receives the index S and trapdoor T_q, it first checks whether the user who sent the search request is the same DU who send the request. Then SC compares the equation

$$\hat{e}\left(T_{Q,1}, \prod_{i=1}^{m} C_{I_i}\right) = \hat{e}\left(A, T_{Q,2}\right) \cdot \hat{e}\left(B, T_{Q,3}\right)$$

If the search result is correct, SC updates the variable that records the search result in smart contract to confirm the search result.

(5) Verify (C): After the search results are obtained, DU applies for ciphertext data to CS. Then CS verifies the search results stored in the smart contract. If the result is correct, CS sends the ciphertext data C to DU, while DU calculates the hash value of the ciphertext and transmits it with DO on SC. When the result is correct, DU pays to DO. After DO obtains the income, he sends the key for data decryption to DU through a secure channel. Finally, DU can decrypt C, and get the plaintext file D.

5 Security Analysis and Performance Evaluation

5.1 Security Analysis

The security of the scheme relies on the confidentiality and fairness of the data. Therefore, the index confidentiality, trapdoor confidentiality and the fairness of the scheme will be analyzed in this subsection.

5.1.1 Index Confidentiality

Refer to the chosen ciphertext attack game given in [9,13], as shown in Fig. 2, we present the security certification process of our scheme for index confidentiality here:

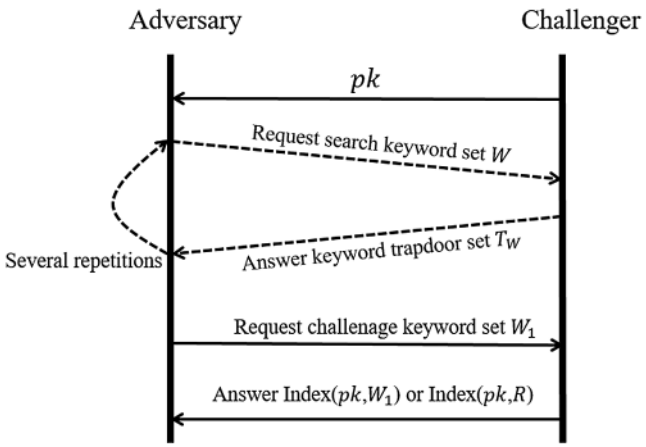

Figure 2: The flow of security game of ciphertext index

(1) Initialize. The challenger initializes relevant information and generates a key pair (pk, sk), and then sends the public key to the adversary.
(2) Phase 1. The adversary sends different query keyword set to challenger, challenger generates the corresponding trapdoor, and then responds to adversary.
(3) Challenge. Adversary sends a query keyword set W_1 that is different from the previous query to challenger. Challenger constructs a random word set R which is similar in structure to W_1, then challenger generates corresponding ciphertext index $S_1 = Index\,(pk, W_1)$ and $S_2 = Index\,(pk, R)$. Then, challenger selects a random index, and sends it to adversary.
(4) Guess. Adversary chooses if he gets the set of keywords sent by himself after getting the index from challenger. If Adversary's choice is right, he wins the game. When the probability of adversary success is greater than 1/2, it means the adversary has the advantage $k = |\Pr(Adv_{win}) - 1/2|$ to break the scheme. When k is infinitely small, it means that the adversary cannot break through the scheme.

When the game comes to the guess phase, the opponent needs to guess whether $S_{challenge} = \{A, B, C_1, \cdots, C_l\}$ is the keyword set sent by himself or randomly generated by the challenger. At this time, according to the previous scheme process, indexes S_1 and S_2 can form two tuples in accordance with the DLDH problem, where R_i is the value of C_i in the index generated by the random keyword set R:

$$\left(g_1 = g, g_2 = y, g_3 = h_i f_i, g_1^a = g^r = A, g_2^b = y^s = B, g_3^{a+b} = C_i\right),$$
$$\left(g_1 = g, g_2 = y, g_3 = h_i f_i, g_1^a = g^r = A, g_2^b = y^s = B, g_3^c = R_i\right)$$

Since this six-tuple is constructed, the adversary needs to break the DLDH problem to win the game. Then the confidentiality of the ciphertext index can be reduced to the DLDH problem,

which is a hard problem according to [29]. That is to say, BC-PECK can ensure the confidentiality of the ciphertext index.

5.1.2 Trapdoor Confidentiality

Similarly, the process of trapdoor confidentiality is shown in Fig. 3:

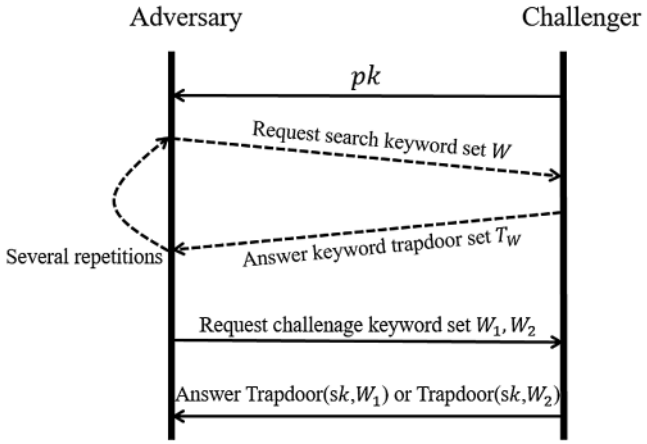

Figure 3: The flow of security game of search trapdoor

(1) Initialize. The challenger initializes relevant information and generates a key pair (pk, sk), and then sends the public key to the adversary.
(2) Phase 1. The adversary sends different query keywords set to challenger, challenger generates the corresponding trapdoor, and then responds to adversary.
(3) Challenge. Adversary sends two query keyword sets W_1 and W_2 different from previous query to challenger. Challenger generates corresponding search trapdoors $T_{q1} = Trapdoor(sk, W_1)$ and $T_{q2} = Trapdoor(sk, W_2)$. Then, challenger sends a random trapdoor to the adversary.
(4) Guess. The adversary guesses which keyword generated the trapdoor he received. If adversary's choice is right, he wins the game. When the adversary's success probability is greater than 1/2, it means that the adversary has the advantage $k = |\Pr(Adv_{win}) - 1/2|$ to break the scheme.

In our scheme, the calculation of trapdoor is divided into three parts. Since each time of the trapdoor generation process uses a new random number, even the same trapdoor generated each time is different. Therefore, it is difficult for adversary to attack the trapdoor content through statistical analysis attacks. That is to say, the adversary can only randomly guess the keyword by coin flipping, and it is difficult to obtain information from the trapdoor with a non-ignorable advantage. Moreover, the request for data generation trapdoor needs to be recorded on blockchain, many repeated search requests can also be expressed through the blockchain, thereby ensuring data security from another perspective.

5.1.3 Fairness

Compared with traditional searchable encryption schemes, BC-PECK adjusts the operation authority of cloud server to smart contract deployed on the blockchain. It stores the operation

records of data retrieval on the blockchain by calling the smart contract. Since blockchain is a distributed database whose data cannot be tampered with, once the data is confirmed and recorded in the blockchain, it cannot be modified and denied. In this way, the operation records of each role in the implementation process of the scheme can be trusted to implement the audit process. Therefore, relying on the credibility of blockchain and smart contract, BC-PECK can effectively verify the non-compliant operation of a certain role and avoid to get the income that is not its due. On the other hand, cryptocurrency in blockchain provides corresponding point-to-point payments after the process is executed, which also effectively enhances the fairness of the scheme.

5.2 Performance Evaluation

In this subsection, we will compare the features and computation cost of BC-PECK with schemes in [16,27].

5.2.1 Feature Comparison

As shown in Tab. 1, both BC-PECK and scheme of reference [16] are PEKS schemes and support multi-keyword query. However, compared to scheme of reference [16], BC-PECK uses blockchain and smart contracts to search ciphertext. That is to say there is no need to introduce a third party, which can effectively reduce risks and enhance credibility. BC-PECK can also make fair payments for the data transmission process based on the cryptocurrency of blockchain. Compared with the scheme of [27], our scheme uses public key searchable encryption technology which is more suitable for actual multi-user scenarios. Moreover, our scheme supports multi-keyword search. This means that our scheme can effectively enhance the accuracy of keyword search, and is more practical.

Table 1: Comparison of scheme features

Scheme	Encrypt type	Keyword number	Third party participation	Verification method	Fair payment
Reference [16]	PECK	Multiple keywords	√	Audit agency	×
Reference [27]	SSE	Single keyword	×	Smart contract	√
Our scheme	PECK	Multiple keywords	×	Smart contract	√

5.2.2 Computation Cost

Compared with general data storage scheme, searchable encryption adds several cryptographic calculation processes to enhance the privacy and security of data. Therefore, we judge the cost of calculation of three schemes. The symbols used in calculation in these schemes are shown in Tab. 2.

As shown in Tab. 3, it is not difficult to find that different schemes require relatively more calculations in the index generation stage. Among these schemes, [16] need to perform multiple exponential calculations and pseudo-random function calculations for the system attributes of

the access tree. While [27] requires the number of attributes of the encrypted access strategy and pseudo-random function calculations for document keywords. While BC-PECK only needs exponential calculation and hash calculation that are multiples of the number of document keywords. The amount of calculation is relatively small. In the trapdoor generation stage, although BC-PECK requires three exponentials and hash calculation with twice number of query keywords, which is more complicated than only one pseudo-random function in [27]. Nevertheless, the calculation is performed locally, the impact on our scheme efficiency is relatively small. In the search stage, the calculation advantage of our scheme compared to the scheme of [16] is obvious, but for the realization of multi-keyword query, the efficiency is not as good as that in [27]. Finally, in the verification phase, compared with the three bilinear pairing calculation operations required in [16], both BC-PECK and [27] only need one hash calculation ensuring that the verification phase does not require excessive calculations and effectively reduces the computational overhead.

Table 2: Symbols of cryptographic calculations

E	Exponential operation on group G_0 or G_1
M	Modular multiplication
P	Bilinear pair operation
H	Hash function
F	Pseudorandom function
$\|\mathrm{m}\|$	Number of keywords extracted by DO
$\|l\|$	Number of keywords queried by DU
$\|U\|$	Number of system attributes
$\|S\|$	Number of attributes owned by DU
$\|Y\|$	Number of leaf nodes of tree access strategy Γ

Table 3: Cryptographic calculations costs of different schemes

Scheme	Index phase	Trapdoor phase	Search phase	Verify phase
Reference [16]	$(2\|U\|+2\|\mathrm{m}\|+13)\,E+\|U\|H$	$(2\|S\|+\|l\|+4)\,E$	$(2\|S\|+1)\,E+5P$	$3P$
Reference [27]	$M+(2Y+2)\,E+\|m\|F$	F	1	H
Our scheme	$(2\|l\|+2)\cdot E+2\|l\|H$	$3E+2\|l\|H$	$3P$	H

In summary, BC-PECK is practical and feasible in terms of computational overhead. Meanwhile, it can rely on PECK to ensure data privacy and security. In addition, by combining blockchain and smart contract, the fairness and credibility of the data sharing process can be realized.

6 Conclusion

Cloud computing service is becoming the technical support for the transformation and upgradation of various industries worldwide, but the privacy of data in the cloud storage environment has become a serious threat to privacy information. This article resorts to blockchain technology

to construct a credible and reliable cloud storage data privacy management scheme. The data is stored and transmitted in the form of ciphertext to ensure data privacy and security. Moreover, through the PECK technology, the accuracy and efficiency of the multi-keyword query of the data sharing process is realized. In addition, the use of blockchain technology and smart contract technology helps to achieve fair and credible access control of data in the multiple users' scenario. Finally, the feasibility and effectiveness of the scheme are proved through security analysis and performance evaluation. In the future, more fine-grained control capabilities will be considered in the data retrieval process, while simplifying the execution steps to further optimize the efficiency of our scheme.

Funding Statement: This work is supported by the NSFC (Grant Nos. 92046001, 61671087, 61962009), the Fundamental Research Funds for the Central Universities (Grant No.2019XD-A02), the Open Foundation of Guizhou Provincial Key Laboratory of Public Big Data (Grant Nos. 2018BDKFJJ018, 2019BDKFJJ010, 2019BDKFJJ014), the High-quality and Cutting-edge Disciplines Construction Project for Universities in Beijing (Internet Information, Communication University of China), the Open Research Project of the State Key Laboratory of Media Convergence and Communication, Communication University of China, China (Grant No. SKLMCC2020KF006). Inner Mongolia Major science and technology projects (2019ZD025), Baotou Kundulun District Science and technology plan project (YF2020013), Inner Mongolia discipline inspection and supervision big data laboratory open project fund (IMDBD2020020), the Natural Science Foundation of Inner Mongolia (2021MS0602), Huawei Technologies Co. Ltd (No. YBN2020085019), and the Scientific Research Foundation of North China University of Technology.

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

References

[1] X. B. Chen, Y. R. Sun, G. Xu and Y. X. Yang, "Quantum homomorphic encryption scheme with flexible number of evaluator based on (k, n)-threshold quantum state sharing," *Information Sciences*, vol. 501, no. 1, pp. 172–181, 2020.

[2] H. Wang, D. He, J. Yu and Z. Wang, "Incentive and unconditionally anonymous identity-based public provable data possession," *IEEE Transactions on Services Computing*, vol. 12, no. 5, pp. 824–835, 2016.

[3] C. B. Tan, M. H. A. Hijazi, Y. Lim and A. Gani, "A survey on proof of retrievability for cloud data integrity and availability: Cloud storage state-of-the-art, issues, solutions and future trends," *Journal of Network and Computer Applications*, vol. 110, no. 1, pp. 75–86, 2018.

[4] G. Xu, K. Xiao, Z. P. Li and M. Ryan, "Controlled secure direct communication protocol via the three-qubit partially entangled set of states," *Computers, Materials & Continua*, vol. 58, no. 3, pp. 809–827, 2019.

[5] Z. Qu, S. Chen and X. Wang, "A secure controlled quantum image steganography algorithm," *Quantum Information Processing*, vol. 19, no. 10, pp. 1–25, 2020.

[6] Z. Qu, S. Wu, W. Liu and X. Wang, "Analysis and improvement of steganography protocol based on bell states in noise environment," *Computers, Materials & Continua*, vol. 59, no. 2, pp. 607–624, 2019.

[7] Z. Dou, G. Xu, X. Chen and K. Yuan, "Rational non-hierarchical quantum state sharing protocol," *Computers, Materials & Continua*, vol. 58, no. 2, pp. 335–347, 2019.

[8] D. X. Song, D. Wagner and A. Perrig, "Practical techniques for searches on encrypted data," in *Proc. 2000 IEEE Sym. on Security and Privacy*, Berkeley, CA, USA, pp. 44–55, 2000.

[9] D. Boneh, G. Di Crescenzo, R. Ostrovsky and G. Persiano, "Public key encryption with keyword search," in *Int. Conf. on the Theory and Applications of Cryptographic Techniques*, Berlin, Heidelberg, Germany, pp. 506–522, 2004.

[10] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren *et al.,* "Fuzzy keyword search over encrypted data in cloud computing," in *Proc. IEEE INFOCOM*, San Diego, CA, USA, pp. 1–5, 2010.

[11] Q. Dong, Z. Guan, L. Wu and Z. Chen, "Fuzzy keyword search over encrypted data in the public key setting," in *Int. Conf. on Web-Age Information Management*, Berlin, Heidelberg, Germany, pp. 729–740, 2013.

[12] D. J. Park, K. Kim and P. J. Lee, "Public key encryption with conjunctive field keyword search," in *Int. Workshop on Information Security Applications*, Berlin, Heidelberg, Germany, pp. 73–86, 2004.

[13] Y. H. Hwang and J. P. Lee, "Public key encryption with conjunctive keyword search and its extension to a multi-user system," in *Int. Conf. on Pairing-Based Cryptography*, Berlin, Heidelberg, Germany, pp. 2–22, 2007.

[14] C. Hu and P. Liu, "Public key encryption with ranked multi-keyword search," in *5th Int. Conf. on Intelligent Networking and Collaborative Systems*, Xi'an, Shaanxi Province, China, pp. 109–113, 2013.

[15] B. Wang, W. Song, W. Lou and Y. T. Hou, "Inverted index based multi-keyword public-key searchable encryption with strong privacy guarantee," in *IEEE Conf. on Computer Communications (INFOCOM)*, Kowloon, HK, China, pp. 2092–2100, 2015.

[16] Q. Y. Wu, J. F. Ma, H. Li, J. W. Zhang, Q. Jiang *et al.,* "Multi-keyword search over encrypted data with user revocation," *Journal on Communications*, vol. 38, no. 8, pp. 183–193, 2017.

[17] D. Vujičić, D. Jagodić and S. Ranić, "Blockchain technology, bitcoin, and Ethereum: A brief overview," in *17th Int. Sym. INFOTEH-JAHORINA*, East Sarajevo, pp. 1–6, 2018.

[18] I. Miers, C. Garman, M. Green and A. D. Rubin, "Zerocoin: Anonymous distributed e-cash from bitcoin," in *IEEE Sym. on Security and Privacy*, Berkeley, CA, USA, pp. 397–411, 2013.

[19] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis *et al.,* "Hyperledger fabric: A distributed operating system for permissioned blockchains," in *Proc. of the Thirteenth EuroSys Conf.*, Porto, Portugal, pp. 1–15, 2018.

[20] C. Li, G. Xu, Y. Chen, H. Ahmad and J. Li, "A new anti-quantum proxy blind signature for blockchain-enabled internet of things," *Computers, Materials & Continua*, vol. 61, no. 2, pp. 711–726, 2019.

[21] M. Tsukerman, "The block is hot: A survey of the state of Bitcoin regulation and suggestions for the future," *Berkeley Technology Law Journal*, vol. 30, no. 4, pp. 1127–1170, 2015.

[22] J. Wang, W. Chen, L. Wang, Y. Ren and R. S. Sherratt, "Blockchain-based data storage mechanism for industrial internet of things," *Intelligent Automation & Soft Computing*, vol. 26, no. 5, pp. 1157–1172, 2020.

[23] Z. Deng, Y. Ren, Y. Liu, X. Yin, Z. Shen *et al.,* "Blockchain-based trusted electronic records preservation in cloud storage," *Computers, Materials & Continua*, vol. 58, no. 1, pp. 135–151, 2019.

[24] Y. L. Gao, X. B. Chen, G. Xu, W. Liu, M. X. Dong *et al.,* "A new blockchain-based personal privacy protection scheme," *Multimedia Tools and Applications*, vol. 4, no. 4, pp. 1–14, 2020.

[25] S. Hu, C. Cai, Q. Wang, C. Wang, X. Luo *et al.,* "Searching an encrypted cloud meets blockchain: A decentralized, reliable and fair realization," in *IEEE Conf. on Computer Communications (INFOCOM)*, Honolulu, HI, USA, pp. 792–800, 2018.

[26] L. Chen, W. K. Lee, C. C. Chang, K. K. R. Choo and N. Zhang, "Blockchain based searchable encryption for electronic health record sharing," *Future Generation Computer Systems*, vol. 95, no. 3, pp. 420–429, 2019.

[27] X. X. Yan, X. H. Yuan, Y. L. Tang and Y. L. Chen, "Verifiable attribute-based searchable encryption scheme based on blockchain," *Journal on Communications*, vol. 41, no. 2, pp. 187–198, 2020.

[28] D. Macrinici, C. Cartofeanu and S. Gao, "Smart contract applications within blockchain technology: A systematic mapping study," *Telematics and Informatics*, vol. 35, no. 8, pp. 2337–2354, 2018.

[29] D. Boneh, X. Boyen and H. Shacham, "Short group signatures," in *Annual Int. Cryptology Conf.*, Berlin, Germany, pp. 41–55, 2004.