Computers, Materials & Continua
DOI: 10.32604/cmc.2022.028562
Article

Tech Science Press

A Searchable Encryption Scheme Based on Lattice for Log Systems in Blockchain

Gang Xu[1], Yibo Cao[1], Shiyuan Xu[1], Xin Liu[2,*], Xiu-Bo Chen[3], Yiying Yu[1] and Xiaojun Wang[4]

[1]School of Information Science and Technology, North China University of Technology, Beijing, 100144, China
[2]School of Information Engineering, Inner Mongolia University of Science and Technology, Baotou, 014010, China
[3]Information Security Center, State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, 100876, China
[4]School of Electronic Engineering, Dublin City University, Dublin, Ireland
*Corresponding Author: Xin Liu. Email: lx2001.lx@163.com
Received: 12 February 2022; Accepted: 14 March 2022

Abstract: With the increasing popularity of cloud storage, data security on the cloud has become increasingly visible. Searchable encryption has the ability to realize the privacy protection and security of data in the cloud. However, with the continuous development of quantum computing, the standard Public-key Encryption with Keyword Search (PEKS) scheme cannot resist quantum-based keyword guessing attacks. Further, the credibility of the server also poses a significant threat to the security of the retrieval process. This paper proposes a searchable encryption scheme based on lattice cryptography using blockchain to address the above problems. Firstly, we design a lattice-based encryption primitive to resist quantum keyword guessing attacks. Moreover, blockchain is to decentralize the cloud storage platform's jurisdiction of data. It also ensures that the traceability of keyword retrieval process and maintains the credibility of search result, which malicious platforms are prevented as much as possible from deliberately sending wrong search results. Last but not least, through security analysis, our proposed scheme satisfies the credibility and unforgeability of the keyword ciphertext. The comprehensive performance evaluates that our scheme has certain advantages in terms of efficiency compared with others.

Keywords: Lattice cryptography; searchable encryption; blockchain; privacy protection; log system; information security; applied cryptography

1 Introduction

With the advancement of the big data period, more and more log files containing private data are being stored by data owners in the cloud, facing significant privacy threats and challenges. Searchable encryption is a technology for searching the log ciphertext based on keyword trapdoors. In this technology, data users can obtain the search trapdoors based on the searching keywords provided to the servicer. Then, the servicer executes a search algorithm to search for the matching ciphertext

to correctly retrieve the data required by the user without recovering the plaintext, which significantly protects data privacy. At the beginning of the 21st century, Song et al. [1] put forward the new academic concept of searchable encryption for the first time. They completed the first research plan for the search problem of encrypted data. Further, searchable encryption can be divided into two categories, whether the encryption key and the decryption key are the same. The encryption key and decryption key of symmetric searchable encryption are the same, which cannot guarantee data security. Generally, it is only used when the owner and user of the log file are the same people, and it fails to satisfy most applications.

In 2004, Boneh et al. [2] focused on searching for specific encrypted mailboxes to define the concept of Public-key Encryption with Keyword Search (PEKS) and gave a specific implementation. PEKS has different encryption and decryption keys, which can achieve the effect of sharing data information between data owners and data users so that searchable encryption can be applied to more practical scenarios. Many researchers have since improved and optimised the PEKS scheme to achieve faster search efficiency. Xu et al. [3] proposed Searchable Public Key Ciphertext with Hidden Structure (SPCHS), enabling the fastest possible search of keywords without compromising the encrypted keywords' contextual security. Cui et al. [4] proposed the concept of key aggregation searchable encryption and adopted the Key-Aggregate Searchable Encryption (KASE) scheme. In this scheme, the data owner is only required to issue a public key to the data users who share many files, which is helpful for the effective management of the key and makes this scheme easier to use in practical situations. Song et al. [5] proposed two searchable encryption schemes, FAST and FASTIO, to achieve forward privacy, dramatically improving I/O efficiency and reducing communication overhead.

However, traditional PEKS faces the problem of the untrustworthiness of the servicer, which is fortunately solved to a certain extent by the emergence of blockchain technology. Blockchain, proposed by Nakamoto [6] in 2008, is a distributed public ledger that records all transactions into the block without third-party control and ensures the security and fairness of each transaction. At present, the vigorous development of blockchain technology is favoured by many researchers [7–16]. Wang et al. [17] proposed a scheme that uses blockchain technology to store the hash value of users' private data and the attribute set of third-party applications, which realises secure one-to-many transmission of personal data. Xu et al. [18] avoided the intervention of third-party agencies through blockchain technology and established a multi-party security system. However, blockchain can solve the problem of the untrustworthiness of third-party organisations. Based on this idea, Li et al. [19] proposed a searchable encryption scheme (SSE-using-BC) based on blockchain technology, storing encrypted data on a decentralised blockchain. It avoids the intervention of centralised service providers and ensures the privacy of encrypted data. In 2019, Chen et al. [20] proposed a searchable encryption scheme for Electronic Health Records (EHR). Researchers used logical expressions to generate an EHR index and store it in the blockchain to ensure the EHR index's immutability, integrity, and traceability. In 2020, Nie et al. [21] used searchable encryption to safeguard the privacy of data information and applied the Ethereum blockchain to solve the fairness problem in the search process. Chen et al. [22] designed a new Blockchain-based Searchable Public-key Encryption Scheme with Forward and Backward Privacy (BSPEFB), which largely avoids the adaptive leakage-exploiting attacks of searchable encryption technology in the Vehicle Social Network (VSN) and reflects the practicality of the scheme.

Although the above scheme has improved the problem of the untrustworthiness of the service party in the searchable encryption process, with the rapid development of quantum computing [23], the Shor algorithm realises the rapid decomposition of large prime factors [24]. As a result, malicious users with a quantum computer can launch a keyword guessing attack based on quantum

computing, causing searchable encryption schemes based on traditional cryptography to lose security and privacy. As a result, more researchers have devoted themselves to proposing a post-quantum method. Nabil et al. [25] proposed that the traditional blockchain scheme is vulnerable to attacks by quantum adversaries. It designed the first post-quantum security signature scheme with public key re-randomisation by providing a deterministic wallet scheme with a post-quantum structure. However, lattice-based cryptography has the highest efficiency and low communication overhead among many post-quantum schemes, and is thus the most promising technology.

In a nutshell, we propose a searchable encryption scheme based on lattice for log systems in the blockchain. Then, we describe our main contributions as follows:

1. Firstly, we propose a searchable encryption scheme based on lattice cryptography to resist keyword guessing attacks under quantum computing.
2. Secondly, blockchain technology is applied to replace the authoritative and trusted party in the scheme to ensure the honesty and credibility of the server.
3. Finally, we conduct a security analysis of our scheme and compare it with other schemes, demonstrating its feasibility and efficiency.

2 Preliminary

Definition 1 (Lattice) Suppose that $a_1, a_2, \ldots, a_m \in \mathbb{R}^n$ are m linearly independent vectors, then the set of linear combinations is called lattice L, denoted by $L = L(A) = \{x_1 \cdot a_1 + x_2 \cdot a_2 + \cdots + x_m \cdot a_m | x_i \in \mathbb{Z}\}$, where the matrix $A = \{a_1, a_2, \ldots, a_m\} \in \mathbb{R}^{n \times m}$ is called a basis of L, m is called the rank of L, and n is called the dimension of L. When $m = n$, L is called full rank.

Definition 2 (Dual Lattice) Suppose that $L(A)$ is the lattice composed of bases $A \in \mathbb{R}^{n \times m}$, then define the dual lattice as: $L^* = \{x \in \mathbb{R}^n | \langle x, y \rangle \in \mathbb{Z}, \forall y \in L\}$.

Definition 3 (q-ary Lattice) Set q is a positive integer, given a matrix $A \in \mathbb{Z}^{n \times m}$ and vector $u \in \mathbb{Z}_q^n$, we define:

$$L_q^{\perp}(A) = \{e \in \mathbb{Z}^{\mathrm{m}} | Ae = 0 \bmod q\} \tag{1}$$

$$L_q^{u}(A) = \{e \in \mathbb{Z}^{\mathrm{m}} | Ae = u \bmod q\} \tag{2}$$

$$L_q(A) = \left\{e \in \mathbb{Z}^{\mathrm{m}} | A^T s = e \bmod q, \exists s \in \mathbb{Z}^n\right\} \tag{3}$$

Obviously, $L_q(A)$ and $L_q^{\perp}(A)$ are dual lattices of each other, and $L_q^{u}(A)$ can be obtained by translation $L_q^{\perp}(A)$.

Definition 4 (Discrete Gaussian Distribution) Suppose that $D_{L,\sigma,c}$ is a discrete Gaussian distribution defined on lattice L with a vector c as the center and σ as a parameter, and the specific expression form is: $D_{L,\sigma,c}(x) = \frac{\rho_{\sigma,c}(x)}{\rho_{\sigma,c}(L)}$ such that $\rho_{\sigma,c}(x) = \exp\left(-\frac{\pi \|x-c\|^2}{\sigma^2}\right)$. When $c = 0$, it records as $D_{L,\sigma}$.

Lemma 1 (TrapGen) [26] Suppose $q \geq 2$, and $m \geq 2n \log q$, TrapGen algorithm outputs a matrix $A \in \mathbb{Z}_q^{n \times m}$ which is a statistically approximation to a uniform distribution and the basis $T_A \in \mathbb{Z}^{m \times m}$ of $L_q^{\perp}(A)$ satisfying $\|T_A\| \leq O(n \log q)$ and $\|T_A\| \leq O\left(\sqrt{n \log q}\right)$.

Lemma 2 (SamplePre) [27] Set T_A is the basis of $L_q^{\perp}(A)$, the parameters $\sigma \geq \|T_A\| \cdot \omega\left(\sqrt{\log m}\right)$ and the vector $u \in \mathbb{Z}_q^n$, and then SamplePre algorithm outputs a vector ε that is statistically close to $D_{L_q^{\perp},s}$, satisfying $A\varepsilon = u \bmod q$.

Lemma 3 (SampleL) [28] Suppose a positive integer $m > n$, $q > 2$, given a lattice $L_q^{\perp}(A)$, set T_A as the basis of $L_q^{\perp}(A)$, matrix $B \in \mathbb{Z}_q^{n \times m'}$, parameters $\sigma \geq \|T_A\| \cdot \omega\left(\sqrt{\log(m+m')}\right)$, and vector $u \in \mathbb{Z}_q^n$, then SampleL algorithm outputs a vector ε, which is statistically close to $D_{L_q^{\perp},s}$, satisfying $(A|B)\,\varepsilon = u \bmod q$.

Lemma 4 (SampleR) [28] Suppose a positive integer $m > n$, $q > 2$, given a lattice $L_q^{\perp}(B)$, set T_B as the basis of $L_q^{\perp}(B)$, matrix $A \in \mathbb{Z}_q^{n \times m'}$, $R \in \mathbb{Z}_q^{m' \times m'}$, parameters $\sigma \geq \|T_B\| \cdot \omega\left(\sqrt{\log(m+m')}\right) \cdot \max_{\|x\|=1} \|Rx\|$ and vector $u \in \mathbb{Z}_q^n$, then SampleR algorithm outputs a vector ε which is close to $D_{L_q^{\perp},s}$ satisfying $(A|AR+B)\,\varepsilon = u \bmod q$. In particular, if $R \in \{-1,1\}^{m \times m}$, then we obtain $s' < O\left(\sqrt{m}\right)$.

Lemma 5 (Gaussian Sampling) [27] Knowing the centre c, parameter σ, and an implicit safety parameter p of a distribution, the algorithm randomly selects an integer $x \leftarrow \mathbb{Z} \cap [c - \sigma \cdot t, c + \sigma \cdot t]$ and outputs x with a certain probability and x is close to $D_{\mathbb{Z},\sigma,c}$.

Definition 5 (ISIS problem) Suppose it is an integer q, matrix $A \in \mathbb{Z}_q^{n \times m}$, a real number $\beta > 0$ and a vector v, and there is a non-zero vector ε satisfying $A\varepsilon = v \bmod q$ and $\|\varepsilon\| \leq \beta$.

3 Design of Our Scheme

3.1 Blockchain Architecture and Transaction Design

The structure of the blockchain and transaction constructed in this paper is shown in Fig. 1. The keyword ciphertext attribute and number attribute are added to the transaction. The keyword ciphertext attribute is formed by the data owner encrypting the keyword with the data user's public key, and the number attribute records the number corresponding to the keyword. Based on this, the smart contract traverses the ciphertext of each keyword according to the search trapdoor uploaded by the data user and returns the number corresponding to the qualified keyword to the cloud storage platform. In this way, data owners and data users can share data.

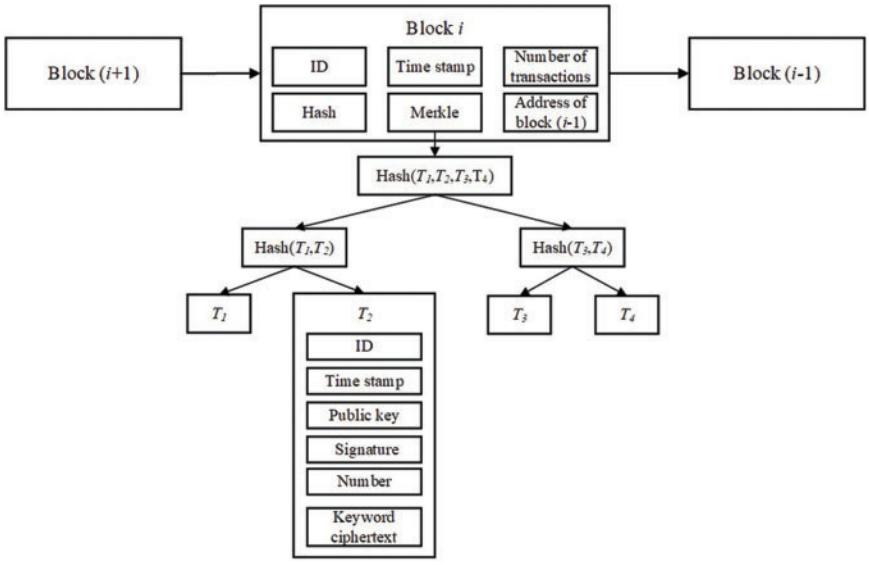

Figure 1: Blockchain and transaction structure

3.2 Our Scheme Architecture

The architecture of our scheme shows in Fig. 2. The roles participating in this scheme include data owner, cloud storage platform, blockchain network, and data user. Their roles in the system are as follows:

(1) Data owner. A data owner is the owner of the log file. He/she divides the log file, generates a number set, and extracts a valid keyword set to store the encrypted log file information in the cloud storage platform. In addition, he/she calculates the keyword ciphertext set corresponding to the log and then uploads the data index to the blockchain. The main problem faced by our scheme is that malicious users perform keyword guessing attacks under quantum computing on the keyword ciphertext, so our scheme focuses on the description of the encryption process of keywords.

(2) Cloud storage platform. Cloud storage platform receives and stores the encrypted log file uploaded by the data owner. After getting the permission of the smart contract, the cloud storage platform returns the specific log file ciphertext to the data user.

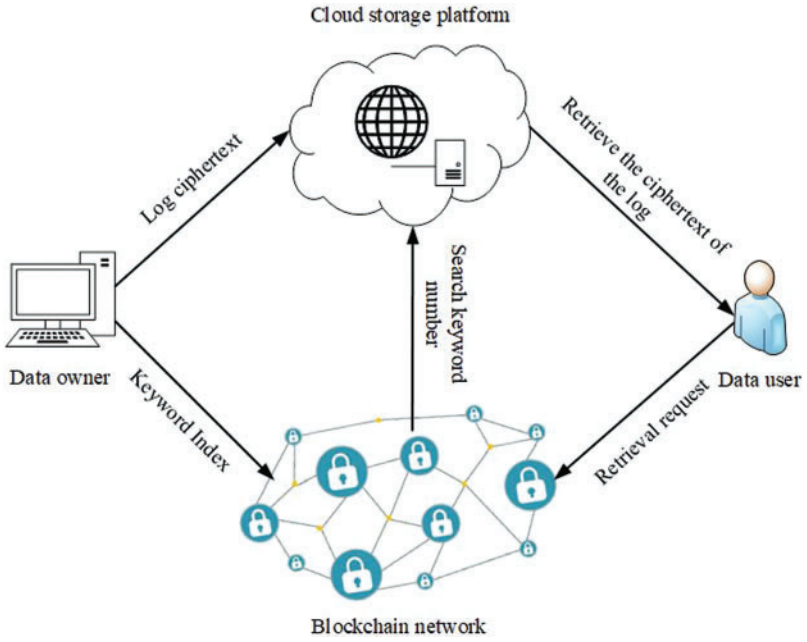

Figure 2: Scheme architecture

(3) Blockchain network. By designing the algorithm in the smart contract, it receives the keyword ciphertext uploaded by the data owner and the trapdoor transmitted by the data user. After receiving a query request from a user, the query is performed according to a specific algorithm, and then the query result is returned to the cloud storage platform. The cloud storage platform is notified whether to send the ciphertext of the log file to the data user.

(4) Data user. A data user is responsible for making a query request to the smart contract. Thus, he/she gets the ciphertext of the corresponding logfile from the cloud storage platform and obtains the plaintext of the log file after decryption.

3.3 Related Algorithm Definition

Definition 6 The searchable encryption scheme based on lattice includes:

(1) $(p, pk_{own}, sk_{own}, pk_{re}, sk_{re}, \$\text{ charge}, \$\text{ fund}) \leftarrow Initialize(\lambda)$: We input the security parameters λ of the architecture, output parameters p, the key held by the data owner (pk_{own}, sk_{own}), the key held by the data user (pk_{re}, sk_{re}), the single search price \$ charge of the blockchain, and the initial deposit value \$ fund of the data user.

(2) $(I, C_F) \leftarrow Encrypt(F, pk_{own}, sk_{own}, pk_{re})$: Firstly, the data owner generates a number set N and then takes the effective keywords $W_i\,(1 \le i \le n)$ in the log file under each number to generate a keyword set $W = \{W_1, W_2, \ldots;, W_n\}$. Then, the data owner inputs the private key sk_{own} and the public key pk_{re} to encrypt the keyword set to obtain the ciphertext C_W. Furthermore, we combine C_W and N to get the data index I, and the data index performs the chain operation. Finally, the log file is encrypted with the data user's public key pk_{re}, and the ciphertext F is obtained and uploaded to the cloud storage platform.

(3) $T \leftarrow TrapGenerate(pk_{re}, W_i)$: The algorithm takes the keyword W_i to be searched as well as the data user's public key pk_{re} as input. Then, the data user calculates and outputs the corresponding trapdoor T and uploads it to the blockchain network.

(4) $N_W \leftarrow Verify(I, T)$: The algorithm is executed by the smart contract in the blockchain. According to the index I transmitted to the blockchain and the trapdoor T generated by the data user, the smart contract executes the *Verify* algorithm to search for the keyword ciphertext that matches the trapdoor T. If the corresponding ciphertext is found, the number N_W of the ciphertext will be returned to the cloud storage platform.

4 Detailed Description of Our Scheme

4.1 Initialise (λ)

Initialise algorithm includes system initialisation, key initialisation, and blockchain network initialisation. In this algorithm, the system sets a series of parameters required for execution and distributes keys to the main participants of the system. The specific process *Initialise* (λ) is defined as follows:

(1) System initialisation. A series of system parameters (n, m, q) are specified, where q is a prime number, and the system runs the initialisation program to generate the system parameters p. In this process, the system uses the input parameters to construct $\mathbb{Z}_q^n$ and $\mathbb{Z}_q^{n\times m}$, selects a random vector v from the $\mathbb{Z}_q^n$, and randomly selects two matrices M_1 and M_2 from the $\mathbb{Z}_q^{n\times m}$. Then, it sets the hash functions $H_1\colon \{0,1\}^* \to \mathbb{Z}_q^n$ and $H_2\colon \{0,1\}^* \times \mathbb{Z}_q^{2m} \to \mathbb{Z}_q^n$ required by the system and outputs $p = \{n, m, q, v, M_1, M_2, H_1, H_2\}$.

(2) Key initialisation. The system inputs the parameter p, Lemma 1 outputs the matrix M and the basis T of the lattice $L(M)$ and satisfies $\|T\| \le O\left(\sqrt{n\log q}\right)$ and $MT = 0 \bmod q$. Then, the system obtains the public-private key pair $(pk, sk) := (M, T)$. Through the above process, the key $(pk_{own}, sk_{own}) := (M_{own}, T_{own})$ of the data owner and the key $(pk_{re}, sk_{re}) := (M_{re}, T_{re})$ of the data user can be obtained, and then the system will transmit the generated key to the data owner and data user securely and confidentially.

(3) Blockchain network initialisation. The data owner initialises the single search cost \$ charge on the blockchain network. The data user uses a unique identity ID to register an identity account on the blockchain network and set the deposit \$ fund of ID.

4.2 Encrypt ($F, pk_{own}, sk_{own}, pk_{re}$)

Encrypt algorithm includes log file encryption and keyword encryption. At this stage, this paper mainly studies the process of keyword encryption. The specific steps *Encrypt* $(F, pk_{own}, sk_{own}, pk_{re})$ are as follows:

(1) Preparation stage. The data owner will divide the log file F, namely $F = (b_1, b_2, \ldots, b_n)$. After that, each division is numbered to generate a number set $N = (1, 2, \ldots, n)$. For $i = 1, 2, \ldots, n$, each keyword W_i are extracted from each division b_i. Finally, we obtain a keyword set $W = (W_1, W_2, \ldots, W_n)$.

(2) Log file encryption. The data owner uses the public key pk_{re} of the data used to encrypt each division of the log file and generates the corresponding log file ciphertext C_{b_i} to form the ciphertext set $C_F = \left\{\left(1, C_{b_1}\right), \left(2, C_{b_2}\right), \ldots, \left(n, C_{b_n}\right)\right\}$. Then, the data owner uploads C_F to the cloud storage platform.

(3) Keyword encryption. The data owner randomly samples a vector r and a noise vector y in $\mathbb{Z}_q^n$. Then, for any $W_i \in W$, the data owner calculates: $c_{W_i} = (M_{re} | M_1 + M_2 \cdot W_i)^T r + \begin{pmatrix} y \\ yR^T \end{pmatrix}$ $(1 \le i \le n)$, where $R \in \{-1, 1\}$. It is known that is the hash function H_1 set by the system during the initialisation process. Then, the data owner randomly chooses a bit $\omega \in \{0, 1\}$ and calculates the corresponding hash value $H_1\left(c_{W_i}, \omega\right)$. According to Lemma 2, the data owner can obtain a vector ε, which satisfies $M_{own}\varepsilon = H_1\left(c_{W_i}, \omega\right) \bmod q$. In addition, the data owner calculates a parameter $\xi_i = \left\lfloor \frac{q}{2} \right\rfloor \cdot \omega$ for each keyword ciphertext. The ciphertext corresponding to W_i is $C_{W_i} = \left(c_{W_i}, \varepsilon_i, \xi_i\right)$, and each division in the ciphertext corresponds to the number set to generate a ciphertext index $I = \left\{\left(1, C_{W_1}\right), \left(2, C_{W_2}\right), \ldots, \left(n, C_{W_n}\right)\right\}$.

(4) Keyword ciphertext uploading. The data owner uses the private key to generate a digital signature for the hash value $H_1\left(C_{W_i}, \omega\right)$ of each keyword ciphertext, generates the corresponding transaction, and submits the transaction to the master node. Then all nodes in the blockchain execute the consensus algorithm. The master node packs the transaction in a period together to form a block and then sends it to the slave node, which receives the block delivery by the master node and verifies the transaction in it. The verification process is as follows: the slave node extracts pk_{own} stored in the transaction from the node to decrypt and obtain the hash value $H_1{}'\left(C_{W_i}, \omega\right)$ of the keyword ciphertext. If $H_1\left(C_{W_i}, \omega\right) = H_1{}'\left(C_{W_i}, \omega\right)$, the slave node announces that the verification is successful. Otherwise, the data may have been tampered with, and the slave node returns this transaction to the data owner. Assuming that the maximum number of malicious nodes that can exist in the consensus algorithm is f, if the number of verifications is $N_T > f + 1$, each node on the blockchain will store the block.

4.3 TrapGenerate (pk_{re}, W_i)

The data user inputs their public key pk_{re} and keyword W_i into *TrapGenerate* (pk_{re}, W_i). According to Lemma 3, the algorithm generates a vector T satisfying $(M_{re} | M_1 + M_2 \cdot W_i)\, T = v \bmod q$. Then, T will be outputted as a trapdoor and sent to the consensus node on the blockchain.

4.4 Verify (I, T)

Before running *Verify* (I, T), the smart contract will compare the value of the deposit \$ fund of data user ID and the single search cost \$ charge. If \$ fund $<$ \$ charge, It returnsto the data user that the cost is insufficient. If \$ fund $\geq$ \$ charge, the smart contract automatically executes *Verify* to search for the keyword ciphertext and its number that match the user trapdoor in the data index set. *Verify* takes data index $I = \left\{\left(1, C_{W_1}\right), \left(2, C_{W_2}\right), \ldots, \left(n, C_{W_n}\right)\right\}$ and trapdoor T as input. First, the algorithm enumerates the keyword ciphertext C_{W_i} $(1 \le i \le n)$ in I. For each ciphertext, calculate:

$\delta = v^T r + \xi_i - T^T c_i$, if $\left|\delta - \left\lfloor \frac{q}{2} \right\rfloor\right| < \frac{q}{4}$, then it can be concluded $\omega = 1$, otherwise $\omega = 0$. Then the algorithm calculates the hash value $H_1\left(c_{W_i}, \omega\right)$, if it exists $j \in [1, n]$ and satisfies $M_{re}\varepsilon_j = H_1\left(c_{W_j}', \omega\right) \bmod q$, c_{W_j}' is the target keyword ciphertext corresponding to the trapdoor T. Finally, the number $N_W = j$ of the keyword ciphertext is returned by the algorithm to the cloud storage platform.

The cloud storage platform finds the ciphertext of the corresponding log file according to the index value and transmits it to the data user. To obtain the plaintext of the log file, the data user decrypts the ciphertext.

Proof: For the method of restoring the value during the search and verification process, we give the proof as follow:

$$
\begin{aligned}
\delta &= v^T r + \xi_i - T^T c_i \\
&= v^T r + \left\lfloor \frac{q}{2} \right\rfloor \cdot \omega - T^T \left((M_{re} | M_1 + M_2 \cdot W_i)^T r + \begin{pmatrix} y \\ yR^T \end{pmatrix} \right) \\
&= v^T r + \left\lfloor \frac{q}{2} \right\rfloor \cdot \omega - \left((M_{re} | M_1 + M_2 \cdot W_i)\, T \right)^T r - T^T \begin{pmatrix} y \\ yR^T \end{pmatrix} \\
&= v^T r + \left\lfloor \frac{q}{2} \right\rfloor \cdot \omega - v^T r - T^T \begin{pmatrix} y \\ yR^T \end{pmatrix} \\
&= \left\lfloor \frac{q}{2} \right\rfloor \cdot \omega - T^T \begin{pmatrix} y \\ yR^T \end{pmatrix} \\
&= \left\lfloor \frac{q}{2} \right\rfloor \cdot \omega + E
\end{aligned}
$$

Among this process, E is the error term. According to [29], to meet the algorithm's demands for correct decryption, the error term should be less than $\frac{q}{5}$. When $\omega = 1$, $\left|\delta - \left\lfloor \frac{q}{2} \right\rfloor\right| = \left|\left\lfloor \frac{q}{2} \right\rfloor \cdot (\omega - 1) + E\right| = |E| < \frac{q}{5} < \frac{q}{4}$; when $\omega = 0$, $\left|\delta - \left\lfloor \frac{q}{2} \right\rfloor\right| = \left|\left\lfloor \frac{q}{2} \right\rfloor \cdot (\omega - 1) + E\right| = \left|E - \left\lfloor \frac{q}{2} \right\rfloor\right| > \frac{q}{4}$.

5 Security Analysis

5.1 Credibility

This solution is based on the blockchain network, which can ensure the honesty and credibility of search results and primarily resist malicious attacks by illegal users on the server. The keyword search process does not involve any third parties in the decentralised blockchain network, and the nodes conduct open and transparent interactions based on transactions. All transactions and operations are recorded on the block. The characteristics of traceability and non-tampering can ensure the fairness and credibility of each search operation. In addition, each transaction initiated in the blockchain network requires the payment of a particular cost, which effectively avoids the possibility of illegal users undermining the program's regular operation through malicious and exhaustive means.

5.2 Unforgeability of Keyword Ciphertext

In this scheme, the keyword ciphertext of the log file is unforgeable, its security can be reduced to ISIS hardness, and it effectively resists keyword guessing attacks initiated by illegal users equipped with quantum computers, thereby ensuring privacy of the plaintext.

Theorem 1 For any adversary A in polynomial time, the difficulty of forging the ciphertext of a keyword is equal to the difficulty of solving the difficult problem of ISIS currently.

Proof: Assuming that adversary A cracks the unforgeability of the ciphertext with a non-negligible probability, it is equivalent to constructing a challenger C capable of solving ISIS problems.

(1) Initialisation. Suppose adversary A initiates a keyword guessing attack on the system by forging the keyword ciphertext. In that case, challenger C executes the Setup algorithm to generate a series of parameters required by the system and sends the parameters p to adversary A. Then, C initiates an inquiry to the random prophecy and obtains the parameters $M_{own}{}^{*}$, which will be used as the public key of the data owner. Finally, C randomly selects a matrix $M_{re}{}^{*} \in \mathbb{Z}_q^{n\times m}$ as the data user's public key and simultaneously dispatches two public keys to A.

(2) Inquiry phase 1. For any ciphertext C_{W_i} with different $\omega \in \{0,1\}$ keywords, if $C_{W_i} = C_{W_i}{}^{*}$ and $w = w^{*}$, the challenger C calculates $H_1(c_W, \omega)$ and returns it to adversary A, and adds $\left(C_W{}^{*}, \omega^{*}, H_1(c_W{}^{*}, \omega^{*})\right)$ to the list. Otherwise, C randomly obtains ε which obeys $D_{L_q^u(M_{own}{}^{*}),\sigma}$ according to Lemma 5, then calculates $H_1(c_W, \omega) = M_{own}{}^{*}\varepsilon \bmod q$ and adds $(C_W, \omega, H_1(c_W, \omega))$ to the list.

(3) Inquiry phase 2. For the keyword $W_i\,(1 \le i \le n)$ that adversary A initiates an inquiry, challenger C calculates $c_{W_i} = (M_{re}{}^{*}|M + B \cdot W_i)^T r + \begin{pmatrix} y \\ yR^T \end{pmatrix}$ and generates the first part c_{W_i} of the ciphertext. After obtaining ε which obeys $D_{L_q^u(M_{own}^{*}),\sigma}$ according to Lemma 5, challenger C will send $\left(c_{W_i}, \varepsilon\right)$ to adversary A.

(4) Forgery phase. In this phase, adversary A will forge a ciphertext $\left(c_{W_i}{}^{*}, \varepsilon^{*}, \xi^{*}\right)$ related to the keyword $W_i^{*}\,(1 \le i \le n)$. To begin with, A gets a trapdoor T^{*} through Lemma 2 and sends it to challenger C. Then C calculates $\delta = v^T r + \xi_i - T^T c_i$, if $\left|\delta - \left\lfloor \frac{q}{2} \right\rfloor\right| < \frac{q}{4}$, then $\omega^{*} = 1$, otherwise $\omega^{*} = 0$, and returns a second part ε^{*} of the ciphertext of the key set forged by A. In this process, C cannot obtain information related to this ciphertext by asking a random oracle. Therefore, the keyword ciphertext $C_W{}^{*} = \left(c_{W_i}{}^{*}, \varepsilon^{*}\right)$ is forged by A, which is a solution to ISIS hardness.

Analysis: Assuming that ε^{*} is a part of the effective keyword ciphertext C_{W_i}, and satisfying $M_{own}{}^{*}\varepsilon^{*} = H_1\left(c_{W_i}{}^{*}, \omega^{*}\right) \bmod q$. If adversary A forges the keyword ciphertext $C_W{}^{*} = \left(c_{W_i{}^{*}}, \varepsilon^{*}\right)$ with probability p, it notices that N is the number of times which the Inquiry phase 1 is executed in polynomial time. So, we obtain the probability that C successfully obtains the satisfying condition $H_1\left(c_{W_i}{}^{*}, \omega^{*}\right)$ is at least $\frac{1}{N}$. Therefore, in the current situation, Challenger C has at least the advantage $\frac{p}{N}$ to break the assumption of ISIS hardness. In summary, the difficulty of adversary A forging the correct keyword ciphertext can be reduced to the difficulty of solving the ISIS hardness.

6 Comprehensive Evaluation Analysis

Our paper proposes a searchable encryption scheme based on lattice for log systems combined with blockchain. The test environment is a 64-bit Windows system with 16GB of memory. And the experimental process is completed by the local virtual machine Ubuntu 16.04.6. As shown in Tab. 1, we compare the scheme in this paper with the schemes [30,31] in terms of methodology and hardness assumptions.

Table 1: Comparison of methodologies and hardness assumption

	Searchable encryption	Blockchain	Post-quantum	Hardness assumption
[30]	√			Diffie–Hellman
[31]	√		√	LWE
Our scheme	√	√	√	ISIS

Through comparison, the common point between the literature [30] and our scheme is that they both use searchable encryption to ensure the fairness and credibility of the system environment. However, our scheme still provides the security of keyword ciphertexts facing quantum computing attacks. Consequently, the privacy of log files is well protected.

After that, in order to compare the differences between the literature [31] and our proposed scheme in the trapdoor cost, verify algorithm cost, ciphertext size, and trapdoor size, Tab. 2 defines the acronyms used in the experimental process, and Tab. 3 shows the performance comparison results.

Table 2: Glossary

Acronyms	Descriptions
T_H	Time of hash
T_L	Time of SampleL
T_S	Time of SamplePre
T_N	Time of NewBasisDel [27]
T_{BE}	Time of modular exponentiation
T_{MT}	Time of matrix transpose
$\lvert Z_q \rvert$	Length of $\mathbb{Z}_q$
$\lvert K \rvert$	Length of keyword

Table 3: Performance comparison

Performance	Zhang et al. [31]	Our scheme
Trapdoor cost	$T_S + T_N + T_{BE} + T_H$	$T_L + 2T_{BE}$
Verify cost	$4T_{BE} + T_H + T_S + T_N + T_{MT}$	$3T_{BE} + T_H$
Ciphertext size	$2m\lvert Z_q \rvert + \lvert K \rvert$	$2m\lvert Z_q \rvert$
Trapdoor size	$m\lvert Z_q \rvert$	$2m\lvert Z_q \rvert$

Since our scheme uses the trapdoor generation algorithm based on lattice theory and mainly relies on the addition and multiplication of vectors, the overall efficiency of our scheme is superior to traditional cryptography.

Fig. 3 shows the changing trend of the keyword search time under not introducing the blockchain and introducing the blockchain. Since the search operation in the blockchain requires more cost for on-chain transactions, the keyword search time is increased, but the credibility and traceability of the search results can be guaranteed. However, with the linear growth in the number of keywords, the time spent in the blockchain transaction will become insignificant compared to the time used to search for keywords, so the time is less and less affected by the blockchain transaction time.

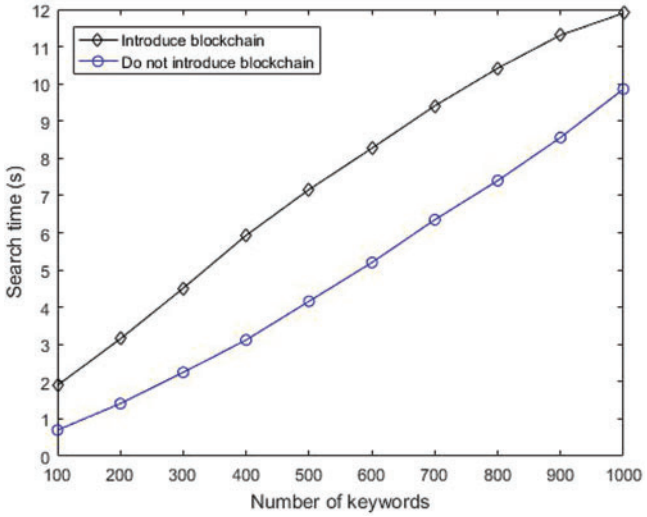

Figure 3: Comparison of schemes

7 Conclusion

To solve the keyword guessing attacks launched by quantum attackers and the untrustworthiness of service providers, this paper designs and proposes a searchable encryption scheme based on lattice for log systems in the blockchain. The application of a lattice-based encryption algorithm makes the scheme resist quantum computing and ensures the security of the keyword ciphertext. At the same time, blockchain technology is employed to separate the keyword ciphertext search from the log file storage. Due to the keyword search is performed by designing a smart contract that ensures the reliability of the search results when the credibility of the servicer is unknown. According to the security analysis and experimental simulation, our scheme is secure in quantum attacks while being highly efficient. In the future, we will introduce forward security and optimize computational cost.

Funding Statement: This work was supported by the Open Fund of Advanced Cryptography and System Security Key Laboratory of Sichuan Province (Grant No. SKLACSS-202101), NSFC (Grant Nos. 62176273, 61962009, U1936216), the Foundation of Guizhou Provincial Key Laboratory of Public Big Data (No.2019BDKFJJ010, 2019BDKFJJ014), the Fundamental Research Funds for Beijing Municipal Commission of Education, Beijing Urban Governance Research Base of North China University of Technology, the Natural Science Foundation of Inner Mongolia (2021MS06006), Baotou Kundulun District Science and technology plan project (YF2020013), and Inner Mongolia discipline inspection and supervision big data laboratory open project fund (IMDBD2020020).

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

References

[1] D. X. Song, D. Wagner and A. Perrig, "Practical techniques for searches on encrypted data," in *Proc. of 2000 IEEE Symp. on Security and Privacy (S&P 2000)*, Berkeley, CA, USA, pp. 44–55, 2000.

[2] D. Boneh, G. D. Crescenzo, R. Ostrovsky and G. Persiano, "Public key encryption with keyword search," in *Proc. of 23 th Advances in Cryptology (EUROCRYPT 2004)*, Interlaken, Switzerland, pp. 506–522, 2004.

[3] P. Xu, Q. Wu, W. Wang, W. Susilo, J. Domingo-Ferrer *et al.,* "Generating searchable public-key ciphertexts with hidden structures for fast keyword search," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 9, pp. 1993–2006, 2015.

[4] B. Cui, Z. Liu and L. Wang, "Key-Aggregate Searchable Encryption (KASE) for group data sharing via cloud storage," *IEEE Transactions on Computers*, vol. 65, no. 8, pp. 2374–2385, 2016.

[5] X. Song, C. Dong, D. Yuan, Q. Xu and M. Zhao, "Forward private searchable symmetric encryption with optimized I/O efficiency," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 5, pp. 912–927, 2020.

[6] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf.

[7] F. Baothman, K. Saeedi, K. Aljuhani, S. Alkatheri, M. Almeatani *et al.,* "Computational intelligence approach for municipal council elections using blockchain," *Intelligent Automation & Soft Computing*, vol. 27, no. 3, pp. 625–639, 2021.

[8] Y. Cheng, S. Xu, M. Zang and W. Kong, "LPPA: A lightweight privacy-preserving authentication scheme for the internet of drones," in *Proc. of 2021 IEEE 21st Int. Conf. on Communication Technology (ICCT)*, Tianjin, China, pp. 656–661, 2021.

[9] G. Xu, Y. Cao, S. Xu, K. Xiao, X. Liu *et al.,* "A novel post-quantum blind signature for log system in blockchain," *Computer Systems Science and Engineering*, vol. 41, no. 3, pp. 945–958, 2022.

[10] J. Zhang, S. Zhong, J. Wang, X. Yu and O. Alfarraj, "A storage optimization scheme for blockchain transaction databases," *Computer Systems Science and Engineering*, vol. 36, no. 3, pp. 521–535, 2021.

[11] C. Li, Y. Xu, J. Tang and W. Liu, "Real estate management via a decentralized blockchain platform," *Computers. Materials & Continua*, vol. 66, no. 2, pp. 1813–1822, 2021.

[12] C. T. Li, Y. S. Xu, J. H. Tang and W. J. Liu, "Quantum blockchain: A decentralized, encrypted and distributed database based on quantum mechanics," *Journal of Quantum Computing*, vol. 1, no. 2, pp. 49–63, 2019.

[13] J. Zhang, G. Xu, X. Chen, H. Ahmad, X. Liu *et al.,* "Towards privacy-preserving cloud storage: A blockchain approach," *Computers, Materials & Continua*, vol. 69, no. 3, pp. 2903–2916, 2021.

[14] H. L. Chen, G. Xu, Y. L. Chen, X. B. Chen, Y. X. Yang *et al.,* "Cipherchain: A secure and efficient ciphertext blockchain via mPECK," *Journal of Quantum Computing*, vol. 2, no. 1, pp. 57–83, 2020.

[15] S. Latif, Z. Idrees, Z. E. Huma and J. Ahmad, "Blockchain technology for the industrial Internet of Things: A comprehensive survey on security challenges, architectures, applications, and future research directions," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 11, pp. e4137, 2021.

[16] S. Latif, Z. Idrees, J. Ahmad, L. Zheng and Z. Zou, "A blockchain-based architecture for secure and trustworthy operations in the industrial Internet of Things," *Journal of Industrial Information Integration*, vol. 21, no. 1, pp. 100190, 2021.

[17] Y. Wang, C. Cao and L. You, "A novel personal privacy data protection scheme based on blockchain and attribute-based encryption," *Journal of Cryptologic Research*, vol. 8, pp. 14–27, 2021.

[18] S. Xu, X. Chen and Y. He, "EVchain: An anonymous blockchain-based system for charging-connected electric vehicles," *Tsinghua Science And Technology*, vol. 26, no. 6, pp. 845–856, 2021.

[19] H. Li, F. Zhang, J. He and H. Tian, "A searchable symmetric encryption scheme using blockchain," 2017. [Online]. Available: https://doi.org/10.48550/arXiv.1711.01030.

[20] 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.
[21] M. Nie, X. Pang, W. Chen, S. Gong and T. Yang, "Fair searchable encryption scheme based on ethereum blockchain," *Computer Engineering and Applications*, vol. 56, no. 4, pp. 69–75, 2020.
[22] B. Chen, L. Wu, H. Wang, L. Zhou and D. He, "A blockchain-based searchable public-key encryption with forward and backward privacy for cloud-assisted vehicular social networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 5813–5825, 2020.
[23] S. Xu, X. Chen, C. Wang, Y. He, K. Xiao *et al.,* "A lattice-based ring signature scheme to secure automated valet parking," in *Proc. of 16th Wireless Algorithms, Systems, and Applications (WASA 2021)*, Nanjing, China, vol. 12938, pp. 70–83, 2021.
[24] P. W. Shor, "Algorithm for quantum computation: Discrete logarithms and factoring algorithm," in *Proc. of 35th Annual Symp. on Foundations of Computer Science (FOCS 1994)*, Santa Fe, Mexico, USA, pp. 124–134, 1994.
[25] A. Nabil, D. Poulami, E. Andreas, F. Sebastian, K. Juliane *et al.,* "Deterministic wallets in a quantum world," in *Proc. of 2020 ACM SIGSAC Conf. on Computer and Communications Security (CCS 20)*, New York, NY, USA, pp. 1017–1031, 2020.
[26] M. Ajtai, "Generating hard instances of the short basis problem," in *Proc. of 26th Int. Colloquium on Automata, Languages, and Programming (ICALP 1999)*, Prague, Czech, pp. 1–9, 1999.
[27] C. Gentry, C. Peikert and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *Proc. of 14th annual ACM symp. on Theory of Computing (STOC 2008)*, Victoria, British Columbia, Canada, pp. 197–206, 2008.
[28] S. Agrawal, D. Boneh and X. Boyen, "Efficient lattice (H)IBE in the standard model," in *Proc. of 29th Advances in Cryptology (EUROCRYPT 2010)*, Riviera, France, vol. 6110, pp. 553–572, 2010.
[29] S. Agrawal, D. Boneh and X. Boyen, "Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE," in *Proc. of 30th Advances in Cryptology (CRYPTO 2010)*, Santa Barbara, CA, USA, vol. 6223, pp. 98–115, 2010.
[30] Q. Huang and H. Li, "An efficient public-key searchable encryption scheme secure against inside keyword guessing attacks," *Information Sciences*, vol. 403-404, no. 1, pp. 1–14, 2017.
[31] X. Zhang and C. Xu, "Trapdoor security lattice-based public-key searchable encryption with a designated cloud server," *Wireless Personal Communications*, vol. 100, no. 3, pp. 907–921, 2018.