1. Introduction

In the current medical scenarios, medical institutions generate a large amount of patient medical data. These data are difficult to supervise, lack necessary technical support, and cost medical institutions many resources. To solve this problem, many medical institutions have adopted EHR systems to reduce the burden and cost of maintaining medical information [1]. The EHR system is a digital health file with medical information as the main body and information sharing as the core. It aims to realize that patients can manage their medical data, and doctors can also access the patient’s medical data if they have permission. However, outsourcing management of the EHR system is not an ideal choice. Because the third-party organization responsible for storing the EHR system has too much power, once a malicious attacker buys it, it can launch a collision attack on the medical data in the system to threaten the privacy of medical data. To avoid this situation, medical institutions usually encrypt medical data through various encryption schemes [2] and store it in the EHR system. Therefore, how to realize the sharing of medical data between patients and doctors in the ciphertext state is a problem to be solved. Thus, Public Key Encryption with Keyword Search (PEKS) [3] is a marvelous candidate in cloud-assisted E-healthcare scenarios, realizing medical data retrieval without privacy leakage. As efficient encryption primitive, it ensures searchable encrypted medical data through keywords.

Although the existing proposed PEKS schemes [4–6] have brought significant benefits to the Internet of Things, there are four significant obstacles to the widespread PEKS in systems in recent decades. Initially, most PEKS schemes were established based on traditional hardness cryptography problems. Nevertheless, with the advent of quantum computers [7] and quantum information [8], the PEKS scheme will be threatened exponentially. Recent breakthrough articles [7] indicate that shortly, it is possible to adopt quantum computers in a realistic view, putting forward higher
requirements for postquantum cryptographic searchable encryption schemes than before. Secondly, the most computational cost of cloud servers is to search target data from the third-party service agency since cloud servers need to execute a verification procedure for the corresponding keyword. Due to the exorbitant public-key encryption operations, the existing PEKS scheme introduces a significant calculation overhead. In the E-healthcare scenarios, the cloud server can work with medical data from mobile medical detection devices simultaneously to retrieve the data of multiple doctors. Therefore, it has a performance bottleneck on the medical cloud servers. Thirdly, with the explosive utilization of mobile medical detection equipment, most schemes have key exposure problems [9]. The existing PEKS scheme cannot guarantee the forward privacy of the key. The existing PEKS scheme cannot guarantee the forward privacy of the key. Once the doctor’s secret key is compromised, the attacker can trace the trapdoor content previously submitted by the doctor, thereby further infringing on the confidentiality of the outsourced data [10].

In this regard, we optimize the lattice cryptography in our scheme to make the key have relations with period to ensure that the key exposure at the previous period will not affect the medical data confidentiality at the later period and achieve the forward security of the key [11]. Last but not least, the search function of the traditional PEKS scheme is generally delivered to the service party. However, the untrustworthiness of the service party will cause attackers to generate Keyword Guess Attacks (KGA) on medical information. Fortunately, blockchain can effectively solve this problem [12–17]. Blockchain is a new database technology that can realize decentralized distributed architecture design. Its core technical concept was proposed by Satoshi Nakamoto [18] in 2008. Blockchain, as a distributed public ledger, records all transactions packaged in the block without the need for third-party control and ensures the safety and traceability of each transaction record [19]. After a single block is generated, all nodes in the blockchain network use a consensus algorithm to determine whether the block is on the chain, and each block is connected by a hash function, thereby effectively ensuring the immutability of transaction information. Therefore, using blockchain technology to replace the service party in PEKS is an effective way to solve the problem of the untrustworthiness of the service party. For example, [20] replaces the traditional centralized server with a decentralized blockchain system, supports forward and backward privacy, and realizes privacy protection. [21] proposed a novel PEKS scheme, which eliminates the reliance on third-party institutions and makes the entire program completely decentralized. Therefore, to solve the above-mentioned hindrances, we propose a postquantum public-key searchable encryption on blockchain for cloud-assisted E-healthcare scenarios, called PPSEB, based on lattice cryptography [22, 23], one of the postquantum cryptographic primitives, ensuring a robust security level. In addition, we reduce the security of PPSEB to the Learning WithError (LWE) hardness assumption, which can oppose keyword guessing attacks based on quantum computing launched by malicious attackers effectively.

In our proposed scheme, the patient initially encrypts medical data and its keywords under the public key of the doctor and transmits the corresponding ciphertext to the cloud server for storage. Then, the medical doctor will utilize his/her secret key to compute a trapdoor corresponding to the keyword and then uploads it to the blockchain. Further, the smart contracts on blockchain search for the keyword ciphertext corresponding to the trapdoor and return its number to the cloud server. Finally, the cloud server sends the ciphertext of medical information matching the keyword to the doctor. In summary, we elaborate our main contributions as follows:

1. We propose a postquantum Public-key Searchable Encryption on Blockchain (PPSEB) for the E-healthcare scenarios. PPSEB is constructed on lattice-based public-key searchable encryption based on the LWE hardness assumption.

2. We then introduce blockchain technology into our proposed scheme in response to the untrustworthiness of third parties during the search process. Therefore, we achieve the decentralization architecture of the PPSEB oracle and enhance the security level.

3. PPSEB achieves forward security in order to solve the key leakage of various existing public-key searchable encryption algorithms.

4. We give the computational proof of the correctness and forward security of PPSEB. Furthermore, the comprehensive implementation performance evaluation represents that our scheme is efficient in terms of testing time and computational cost compared with existing outperforming E-healthcare schemes and is suitable for medical scenarios.

The structure of our paper is organized as follows. In Section 2, we propose the design goals and security models of our scheme, considering three existing challenges for the proposed PPSEB scheme and the solution to make PPSEB work better in the medical scenarios. In Section 3, we propose our preliminaries of lattice and trapdoor. In Section 4, we present our PPSEB scheme and the main steps of our scheme, including, PPSEB Initialization, PPSEB.KeyExt, PPSEB.Encrypt, PPSEB.PEKS, PPSEB.Trapdoor, PPSEB.Verification, and PPSEB.Decrypt. In Section 5, we provide the security analysis of PPSEB based on correctness and provable security. In Section 6, a precise performance evaluation is proposed by our paper. Finally, we conclude this paper in Section 7.

2. Design Goals and Security Models

2.1. Design Goals. In this paper, we propose three existing challenges for the proposed PPSEB scheme:

1. How to make PPSEB resistant to the untrustworthy problem of the service party. In the traditional
searchable encryption scheme, a third-party organization is generally responsible for searching medical information, which makes malicious attackers collude with third-party organizations to provide unreliable search results. Therefore, we use blockchain to replace traditional third-party agencies.

(2) How to achieve the forward security of PPSEB. Key exposure is a thorny problem faced by existing searchable encryption schemes. Once the private key of the doctor is lost, the attacker can forge the doctor to initiate an inquiry for medical information, and the privacy of medical information cannot be guaranteed. Therefore, how to use lattice-based cryptography to ensure that the leakage of the master key used at this time will not result in the leakage of the past session key is a problem to be solved.

(3) How to realize PPSEB to resist KGA under quantum computing. The existing searchable encryption scheme cannot guarantee the security of the search process under the attack of quantum computing, and there is a significant commonality between the keywords of medical information. Once the attacker is equipped with a quantum computer, it is possible to launch KGA on medical information through quantum computing, which severely threatens the security of the existing searchable encryption schemes. Once the private key of the doctor is lost, the attacker can forge the doctor to initiate an inquiry for medical information. Consequently, resisting KGA launched by quantum opponents is also a challenging problem. In order to make PPSEB work better in the medical scenarios, the solution in this article should have the following characteristics:

(1) Postquantum KGA: PPSEB can resist KGA attacks under quantum computing.

(2) Forward security: PPSEB achieves forward security to solve the problem of private key exposure.

(3) Efficiency: PPSEB has a higher computational efficiency by reducing the size of the trapdoor.

2.2. Security Model. In this section, we show the ciphertext indistinguishability of our scheme. We can describe several scenarios through games between challenger S and adversary A, in which S generates system security public parameters, initializes the public keys of patient and doctor. A will receive them from S and is permitted to access the oracles as below.

Hash Oracle(HO): A has been permitted to access all values of HO in time t, where t = 1, 2, . . . , η and is the total number in the period. Then, A will receive the corresponding hash value.

Break-in phase: After obtaining the query about SKt−1, the doctor in time t by A, S will return the corresponding SKt−1 in t time to A. We note that t* is the break-in period, which satisfies t > t*.

Trapdoor Oracle(TO): A inputs a keyword w to ask S for a trapdoor T_w. Then, we make the restriction t > t* in order to make sure the forward security, where t* is break-in period.

Challenge phase: A takes (w'_t, w_t') in t* and then submits them to S to be the challenge keywords. S then selects b at random and obtains CT_b. Consequently, S returns CT_b to A.

Guess phase: At last, A will output b' ∈ {0, 1}. It wins the game if and only if b = b'. We define A dv^b_A(k) = |Prob[b' = b] − 1/2|, which means the benefit of A to distinguish ciphertexts in t* successfully.

3. Preliminary

Definition 1 (Lattice). Let A = [a_1, a_2, . . . , a_n] ∈ R^m be n linearly independent vectors in m-dimensional space. A lattice L is composed of the linear combination of all integer coefficients of a_1, a_2, . . . , a_n, and we can define:

L(A) = {∑_{i=1}^{n} x_i a_i : i = 1, 2, . . . , n, x_i ∈ ℤ}, a_1, a_2, . . . , a_n is known as a basis of L. Given a prime number q, a matrix A ∈ Z_q^n×m, we define

L_q(A) = {y ∈ Z_q^m : y = A^T x mod q, x ∈ ℤ},

L_q^∗(A) = {y ∈ Z_q^m : Ay = 0 mod q}.

Definition 2 (LWE). Assume q be a prime number, given a random matrix A ∈ Z_q^m×n, a vector b ∈ Z_q^n, and the error distribution D on Z_q^n, find that the vector x ∈ Z_q^m satisfies b = A^T x + e mod q, where e ∈ D^n.

Definition 3 (Statistical Distance). Given two variables X, Y over a domain D, we define the statistical distance of X and Y: D(X, Y) = 1/2 · ∑_{b∈D} |Pr[X = b] − Pr[Y = b]|.

Definition 4 (Discrete Gaussian Distribution). Let ρ_c,o(x) = exp(−π∥x−c∥2/σ^2) be the standard Gaussian function c represents the center and σ represents the standard deviation. Then we define:

D_{L_c,o}(x) = ρ_c,o(x)/ρ_c,o(L), which is a Gaussian Distribution over Lattice L.

Lemma 1 (TrapGen) [24]. Let q ≥ 3, m ≥ 2n log q. There is a polynomial-time algorithm TrapGen, which outputs a matrix A ∈ Z_q^{m×n} statistically close to the uniform distribution and a trapdoor base T_{A} ∈ Z_q^{m×n}, such that ∥T_{A}∥ ≤ O(n log q) and ∥T_{A}∥ ≤ O(√n log q).

Lemma 2 (SamplePre) [25]. Given L_q^∗(A), a trapdoor base T_{A} ∈ Z_q^{m×n}, a parameter s ≥ ∥T_{A}∥ (log m + m) , and a vector v ∈ Z_q^n. Then, the SamplePre algorithm outputs a vector w statistically close to D_q^s(A,v), such that Aw = v mod q.

Lemma 3 (SampleL) [26]. Set a positive integer m > n, q ≥ 3. Given L_q^∗(A) and its trapdoor base T_{A}, matrix B ∈ Z_q^{m×n}, parameter s ≥ ∥T_{A}∥ (log (m + m') + log (m + m')), and a vector u ∈ Z_q^n. The SampleL algorithm computes e ∈ Z_q^{m×n} statistically close to D_q^s(A,B,v), such that (A|B)e = u mod q.
Lemma 4 (SampleR) [26]. Set a positive integer \( m > n, q \geq 3 \).
Given \( L_q^\perp (B) \) and its trapdoor base \( T_{B} \in \mathbb{Z}_{m,q}^{m \times m} \), matrix \( A \in \mathbb{Z}_{m,q}^{n \times m} \), \( R \in \mathbb{Z}_{m,q}^{n \times q} \), \( s \geq \|T_{B}\|/\omega(\sqrt{\log m}) \) and vector \( u \in \mathbb{Z}_{3, q}^{m} \). The SampleR algorithm outputs a vector \( e \in \mathbb{Z}_{m,q}^{m} \)
over \( D_{\pm 1(A[R + B])s} \) and satisfies \( A[R + B]e = u \mod q \),
where \( s' = \max_{R \subseteq R} \|Rx\| \).

Lemma 5 (NewBasisDel) [27]. Set a positive integer \( m > 2n, q \geq 3 \). Given \( L_q^\perp (A) \) and a trapdoor base \( T_A \in \mathbb{Z}_{m,q}^{m \times m} \), an invertible matrix \( R \in D_{m,q} \), \( D_{m,q} \) is invertible on \( \mathbb{Z}_{m,q}^{m \times m} \), \( s \geq \|T_A\|/\sqrt{n} \log q \cdot \omega(\sqrt{\log m}) \cdot \sqrt{m} \cdot \omega(\log^{1.5} m) \). The NewBasisDel algorithm outputs \( L_q^\perp (B) \) and a trapdoor base \( T_{B} \in \mathbb{Z}_{m,q}^{m \times m} \), where \( B = AR^{-1} \mod q \).

Definition 5. (PEKS scheme): One general PEKS scheme includes five algorithms as PEKS = (Initialization, KeyExt, PEKS, Trapdoor, Verification), these algorithms are defined in the following sentences:

\((s, \sigma) \leftarrow \text{Initialization}(\cdot)\): In this step, it generally initializes some security parameters \( s \) and parameters regard to the Gaussian Distribution \( X \) in one time period \( j \). The output is just these parameters which will utilize in the next step.

\((pk, sk) \leftarrow \text{KeyExt}(s)\): After inputting the parameter \( s \), it will output the public key \( pk \) and secret key \( sk \), which consist \((pk, sk)\).

\(s, \epsilon \leftarrow \text{PEKS}(pk, \epsilon)\): The algorithm takes a public key \( pk \) and one word \( \epsilon \) as input, and outputs a ciphertext \( s_{\epsilon} \).

\(t_{\epsilon}, t'_{\epsilon} \leftarrow \text{Trapdoor}(sk, s_{\epsilon})\): Having input the secret key \( sk \) and one keyword \( \epsilon \), it outputs one trapdoor \( t_\epsilon \) in this algorithm.

\((1 \text{ or } 0) \leftarrow \text{Verification}(t_\epsilon, s_{\epsilon})\): With the input of a trapdoor \( t_\epsilon \) and a searchable ciphertext \( s_{\epsilon} \), this algorithm designs to output the comparison decision bit 1 if \( \epsilon \cdot t_{\epsilon} = \epsilon \), or 0 otherwise.

4. Our Proposed Scheme

4.1. Blockchain Architecture. Blockchain is essentially a decentralized database, which is a string of blocks that are associated using cryptography methods. Each transaction includes hash function, Merkle tree, and so on. In this paper, we replace the search party in searchable encryption with blockchain to ensure the credibility of the search process. As shown in Figure 1, our paper optimizes and adjusts the five-layer architecture of the original blockchain and adds a data retrieval function to the application layer to ensure that the blockchain network can base on the algorithm written in the smart contract realizing the retrieval of the keyword ciphertext.

4.2. System Model. In this section, we give an introduction to the system model of our PPSEB scheme in Figure 2, with four main entities, including patient, doctor, a cloud server, and blockchain network.

(1) Patient: The patient integrates Electronic Health Record (EHR), including various medical information such as drug-using records as a patient. Moreover, the patient encrypts the EHR and uploads it to the Cloud Server. Then the patient generates a set of keywords [keywords, sequence number] related to the specified keyword and adds blocks to the blockchain.

(2) Doctor: The doctor needs to generate a trapdoor to search for information about patients. The doctor submits the corresponding trapdoor to the blockchain.

(3) Blockchain: After receiving the trapdoor from the doctor, the blockchain network will start chain code retrieval to search the corresponding sequence number and submit it to the CloudServer.

(4) Cloud Server: After receiving the query request, the Cloud Server can use trapdoor to search for all encrypted data and return the query results of the ciphertext corresponding to the keywords to the doctor. During the entire process, the server is unable to obtain any information about the data and keywords.

4.3. The Scheme of PPSEB. In this section, we present our proposed scheme in detail. There are mainly seven steps of our scheme, including PPSEB.Initialize, PPSEB.KeyExt, PPSEB.Encrypt, PPSEB.PEKS, PPSEB.Trapdoor, PPSEB.Verification, and PPSEB.Decrypt, which are elaborated in the following paragraphs and algorithms.

\((X, \sigma, \delta, \mu, H_1, H_2, sk_r, \sigma) \leftarrow \text{PPSEB.Initialize}((k, X, \sigma, \delta))\): Firstly, we have to input one security parameter \( k \), the discrete Gaussian Distribution \( X \) and its parameters \( \delta = (\delta_1, \delta_2, \ldots, \delta_q) \), \( \sigma = (\sigma_1, \sigma_2, \ldots, \sigma_n) \) in one period \( j \), where \( j = 1, 2, \ldots, \eta \). After that, the initialization step is shown as follows.

\((SK_{\|\cdot\|}, pk_{\|\cdot\|}) \leftarrow \text{PPSEB.KeyExt}((X, \sigma, \delta, \mu, H_1, H_2, sk_r, \sigma, i)\|i)\): After inputting the set Algorithm 1.

\((X, \delta, \sigma, \mu, H_1, H_2, sk_r, \sigma) \leftarrow \text{PPSEB.Trapdoor}((X, \delta, \sigma, \mu, H_1, H_2, sk_r, \sigma, i)\|i)\): Obtained from the initialization step, we also have to input the current period \( j \) together with the secret key \( sk_r\|i \) in the previous period \( i \). Then, the doctor will procedure the following operations, which shows in Algorithm 2.

\((N, W, I_M) \leftarrow \text{PPSEB.Encrypt}(M, pk_{\|\cdot\|})\): Firstly, the patient divides the medical data \( M \) into groups, named \( M = (M_1, M_2, \ldots, M_q) \), and generates an index \( N = (1, 2, \ldots, n) \) for each group. After that, the patient extracts keywords from each group of medical data and records them as \( W = (w_1, w_2, w_3, \ldots, w_n) \). Finally, the patient
encrypts each group of medical data with the doctor’s public key $pk_r$ at time $j$, obtains a ciphertext set $CM = (CM_1, CM_2, \ldots, CM_n)$, and generates an index set of the medical data ciphertext $I_M = \{(1, CM_1), (2, CM_2), \ldots, (n, CM_n)\}$, and it will be stored in the cloud server.

$(CT_j) \leftarrow PPSEB.PEKS((X, \delta, \sigma, \mu, H_1, H_2, sk_r, sk_s), j, SK_r, w)$: The patient will procedure $PPSEB.PEKS$ algorithm and input the set $(X, \delta, \sigma, \mu, H_1, H_2, sk_r, sk_s)$, the public key $pk_r$, the current time $j$, and keyword $w$. This Probabilistic Polynomial Time (PPT) algorithm shows in detail as below. For each keyword $w_i \in W$, the patient executes $PPSEB.PEKS$ algorithm, obtains $CT_W = (CT_{j_1}, CT_{j_2}, \ldots, CT_{j_n})$, and pairs each keyword ciphertext with the number to generate keyword index set $I_W = \{(1, CT_{j_1}), (2, CT_{j_2}), \ldots, (n, CT_{j_n})\}$. When we get $I_W$, the patient calculates the hash value $H_j$ of $I$ with his own

Figure 1: Blockchain architecture.
ALGORITHM 1: \((X, \delta, \sigma, \mu, H_1, H_2, sk_r, sk_s) \leftarrow \text{PPSEBInitialization}(k, X, \delta, \sigma)\).

\[
\begin{align*}
\text{Input:} & \text{ set } (X, \delta, \sigma, \mu, H_1, H_2, sk_r, sk_s), \text{ current time period } j, \text{ secret key } sk_{r,j} \text{ in previous time period } i \\
\text{Output:} & \text{ } SK_{r,j} \text{ and } pk_{r,j}, \text{ where is the secret key during this period } j \\
& (1) \text{ Compute } H_1(pk_r[i]H_1(pk_r[i]-1)\ldots H_1(pk_r[1]) \in \mathbb{Z}_q^{m} \\
& (2) \text{ Set } R_{i,j} = H_1(pk_r[i]H_1(pk_r[i]-1)\ldots H_1(pk_r[1])) \\
& (3) \text{ Compute } pk_r(R_{i,j})^{-1} = pk_r(H_1(pk_r[i]H_1(pk_r[i]-1)\ldots H_1(pk_r[1]))^{-1} \in \mathbb{Z}_q^{m} \\
& (4) \text{ Set } pk_{r,j} = pk_r(R_{i,j})^{-1} = pk_r(H_1(pk_r[i]H_1(pk_r[i]-1)\ldots H_1(pk_r[1]))^{-1} \\
& (5) \text{ Compute } H_1(pk_r[i]H_1(pk_r[j-1])\ldots H_1(pk_r[i]+1) \in \mathbb{Z}_q^{m} \\
& (6) \text{ Set } R_{i,j-1} = H_1(pk_r[i]H_1(pk_r[j-1])\ldots H_1(pk_r[i]+1)) \\
& (7) \text{ Call } \text{NewBasisDel}((pk_{r,j}, R_{i,j-1}, sk_r, sk_s, \delta)) \text{ to compute } SK_{r,j} \text{ and } sk_{r,j} \\
& (8) \text{ Compute } pk_r(R_{i,j-1})^{-1} = pk_r(R_{i,j})^{-1} \in \mathbb{Z}_q^{m} \\
& (9) \text{ Set } pk_{r,j} = pk_r(R_{i,j})^{-1} \\
& (10) \text{ Return } SK_{r,j} \text{ and } pk_{r,j}.
\end{align*}
\]

private key to generate a digital signature, writes down the transaction ID and timestamp, generates the corresponding transaction, and submits it to the master node for verification. After that, all nodes of the blockchain network execute the consensus algorithm, and the master node jointly packs the transaction orders in a period of time to form a block and then sends it to the affiliate node. Then, the affiliate node receives the block sent by the master node and verifies the transaction slip contained in the block. Firstly, the affiliate node extracts the public key of the patient stored in the
transaction sheet from the node and decrypts the digital signature and get the hash value $H_2$ of $I_w$. If $H_1 = H_2$, the affiliate node declares that the verification is successful. Otherwise, it means that the data may be tampered with and return this transaction to the patient. Assuming that the maximum number of malicious nodes that can exist in the consensus algorithm is $f$, if the number of verifications passes $Num = f + 1$, the block will be stored in each node of the blockchain network Algorithm 3.

$\text{Trap}_{w,j} \leftarrow \text{PPSEB}.\text{Trapdoor}(\{X, \delta, \sigma, \mu, H_1, H_2, s_k, s_k\}, (pk\|j, sk\|j), j, w)$: The doctor will procedure this algorithm after inputting the set $(X, \delta, \sigma, \mu, H_1, H_2, s_k, s_k)$, the public key and secret key pair $(pk\|j, sk\|j)$ of the medical doctor during this period $j$, and one keyword $w \in W$. The detailed description is shown in Algorithm 4.

Finally, the doctor will send $\text{Trap}_{w,j}$ to the blockchain through an efficient and secure communication channel.

$N_0$ or False $\leftarrow \text{PPSEB}.\text{Verification}(\{X, \delta, \sigma, \mu, H_1, H_2, s_k, s_k\}, CT_j, t\|j\}$: This PPT algorithm produced by the blockchain inputs including the set $(X, \delta, \sigma, \mu, H_1, H_2, s_k, s_k)$, the ciphertext $CT_j$, one trapdoor $\text{Trap}_{w,j}$ in this period $j$ of the doctor. If it outputs true; it means that the trapdoor $\text{Trap}_{w,j}$ and the ciphertext $CT_j$ contain the uniform keyword $w$. Then, the blockchain returns the number $N_0$ of the ciphertext corresponding to the keyword to the cloud server. The cloud server finds the ciphertext of the keyword according to $N_0$ and returns it to the doctor Algorithm 5.

$M_0 \leftarrow \text{PPSEB}.\text{Decrypt}(CM_0, j, SK\|j)$: After the doctor obtains the ciphertext $CM_0$ of the medical data returned by the cloud server, he/she decrypts it with his/her $SK\|j$ at time $j$ to obtain the plaintext of medical data $M_0$.

5. Security Analysis

In this section, we will demonstrate our scheme’s correctness and provable security to achieve the security of the keyword ciphertext in our scheme under random oracle.

5.1. Correctness. In this section, we suppose that the key pair at time $j$ of doctors and patients are $(pk\|j, sk\|j)$, $(pk\|j, sk\|j)$, respectively. Then, we set $w$ as the keyword of the ciphertext $CT_j$ and then $w'$ is a keyword that matches the trapdoor $\text{Trap}_{w,j}$. It is well known that the cloud server can use $\text{Trap}_{w,j}$ at a time $j$ to recover $(y^j_1, y^j_2, \ldots, y^j) = CT_j - \text{Trap}_{w,j} CT_j$ in PPSEB. Verification. Since the relationship between $w$ and $w'$ is uncertain, we divide the discussion into the following two situations:

Case 1: If $w \neq w'$, then $CT_j - \text{Trap}_{w,j} CT_j$ and $CT_j - \text{Trap}_{w',j} CT_j$, we can decrypt the ciphertext $CT_j$ and obtain that: for $i = 1, 2, \ldots, l$, there must be $y_i \neq 1$.

Case 2: If $w = w'$, then is $CT_j - \text{Trap}_{w,j} CT_j = CT_j - \text{Trap}_{w',j} CT_j$, no $y_i \neq 1$. Among them, $\text{Trap}_{w,j} CT_j$ is a noise vector. According to [25], we need to ensure that the error vector is less than $q/5$, so that the decryption process does not make mistakes. Consequently, we can compute that: for $i = 1, 2, \ldots, l$, $y_i^j = 1$.

So, the cloud server can ensure that the keyword $w$ corresponds to the ciphertext $CT_j = (CT_j, CT_j)$ and the trapdoor $\text{Trap}_{w,j}$ that is, PPSEB can achieve correctness.

Last but not least, the cloud server sends the encrypted medical data corresponding to the keyword $w$ to the doctor, and the doctor obtains the corresponding plaintext data after decrypting it according to its key.

5.2. Provable Security

Theorem 1. In the PPSEB, the difficulty of the attacker to crack the indistinguishability of the ciphertext can be reduced to the difficulty of the LWE problem.

Proof. Suppose that there is an attacker $A$ under the random oracle model, which can crack the indistinguishability of the ciphertext in polynomial time. On this basis, we have created a challenger $C$ having the ability to solve the LWE problem.

5.2.1. Setup. To begin with, challenger $C$ sends $(u_k, v_{k1}, v_{k2}, \ldots, v_{kl})$, $k = 0, 1, \ldots, m$ from a random oracle machine. Then, $C$ guesses $\tau = \ast$ as a point in time when $A$ breaks the indistinguishability of the ciphertext. After that, $C$ creates two lists, named $L_1$ and $L_2$. Finally, $C$ interacts with attacker $A$. The steps are as follows:

1. Challenger $C$ runs the SampleR algorithm to obtain $R$, then $C$ selects $\tau + 1$ vectors from $R^* \tau \times R^*$ and assembles it into a matrix $F^* \in \mathbb{Z}_q^{m \times \ast}$, making $u_k$ the $k$-th column of $F^*$.

2. Challenger $C$ obtains $pk \subset F^* R^* \tau \times R^*$. Because $F^*$ is independent of $\mathbb{Z}_q^{m \times \ast}$ and $R^* \tau \times R^*$ are irreversible matrices, $pk \subset$ is independent of $\mathbb{Z}_q^{m \times \ast}$.

Then, $C$ selects a matrix as $pk \subset \in \mathbb{Z}_q^{m \times \ast}$ and sets $\mu = u_0 \in \mathbb{Z}_q^\ast$ to get a set $(pk\|j, pk\|j, \mu, H_1, H_2)$. Last but not least, we send $(pk\|j, pk\|j, \mu, H_1, H_2)$ to attacker $A$. After receiving the set $(pk\|j, pk\|j, \mu, H_1, H_2)$, $A$ executes $H_1$ query and $H_2$ query.

$H_1$ query: $A$ initiates an inquiry to each $pk\|j$, where $j = 1, 2, \ldots, \tau$. $C$ computes $R^* \mu = H_1(pk\|j)$ and sends $R^* \mu$ to $A$.

Case 1: $j = \tau + 1$. Challenger $C$ gets $pk\|j-1 \subset pk\|j \cdot (R^* \tau \times R^* \tau \tau)^{-1}$ and runs Sample R with Basis algorithm to get $R$, and the basis $sk\|j$ of lattice $L_q(A_j\|j)$, where $A_j\|j = R^* \mu \cdot A_j\|j$. Then, $C$ appends $(pk\|j, pk\|j, R, sk\|j)$ to the list $L_1$. Consequently, $C$ transmits $R^*$ to attacker $A$.

Case 2: $j > \tau + 1$. Challenger $C$ finds $(pk\|j-1 \subset pk\|j-1, R_{j-1}, sk_{j-1})$ from the list $L_1$. Then, $C$ selects a
Input: set \((X, \delta, \alpha, \mu, H_1, H_2, sk_r, sk_s)\), current time period \(j\), secret key \(SK_{A_j}\) in current period \(j\).

Output: \(CT_j\)

1. Set a binary string \(B_j \leftarrow Z_q^{\|l\|}\), where \(l\) is the security level of test in medical data cloud storage.
2. Select a unitary matrix \(B_j \leftarrow Z_q^{\|l\|\times (n \times l)}\) of \((n \times l)\) dimension.
3. Select noise \(noiv_{j1}, noiv_{j2}, \ldots, noiv_{jn} \leftarrow Z_q\) through \(X\).
4. Set \(noi_j = (noiv_{j1}, noiv_{j2}, \ldots, noiv_{jn})\).
5. Select each noise vector \(noiv_{j1}, noiv_{j2}, \ldots, noiv_{jn} \leftarrow Z_q\) on the basis of \(X^n\).
6. Set the noise vector matrix \(noiv_j = (noiv_{j1}, noiv_{j2}, \ldots, noiv_{jn}) \in Z_q^{\|n\|}\).
7. Assume \(\beta_j = H_2(u_j^w)\) and then compute \(CT_j = \mu^1 B_j + noiv_j + y_j[q/2]\) and \(CT_{j+1} = (pk_{\|j\|} R_j^{-1})^T B_j + noiv_j\) as ciphertext.
8. Set ciphertext \(CT_j = (CT_{j1}, CT_{j2}) = (\mu^1 B_j + noiv_j + y_j[q/2], (pk_{\|j\|} R_j^{-1})^T B_j + noiv_j)\).
9. Return \(CT_j\) to doctor.

**Algorithm 3: \((CT_j) \leftarrow PPSEPKS((X, \delta, \alpha, \mu, H_1, H_2, sk_r, sk_s), j, SK_{A_j}, w)\).**

Input: set \((X, \delta, \alpha, \mu, H_1, H_2, sk_r, sk_s)\), current period \(j\), public-secret key pair \((pk_{\|j\|}, sk_{\|j\|})\), one keyword \(w\).

Output: \(sk_{A_{\|j\|}}\) and \(Trap_{w_{\|j\|}}\).

1. Compute \(\beta_j = H_2(u_j^w)\) to encrypt the message.
2. Set \(R_{\|j\|} = H_1(pk_{\|j\|} H_1(pk_{\|j\|} - 1) \ldots H_1(pk_{\|j\|} - 1))\).
3. Call NewBasicDel \((pk_{\|j\|}, \beta_j, sk_{\|j\|}, \delta_j)\) to generate one short lattice basis \(sk_{A_{\|j\|}} \in Z_q^{\|mson\}}\) in random.
4. Call SamplePre \((pk_{\|j\|}, R_{\|j\|}^1, sk_{\|j\|}, \mu, \sigma)\) to generate the trapdoor \(Trap_{w_{\|j\|}} \in Z_q^{\|m\}}\).
5. Return \(Trap_{w_{\|j\|}}\).

**Algorithm 4: \(Trap_{w_{\|j\|}} \leftarrow PPSEPKS.Trapdoor((X, \delta, \alpha, \mu, H_1, H_2, sk_r, sk_s), (pk_{\|j\|}, sk_{\|j\|}), j, w)\).**

Input: set \((X, \delta, \alpha, \mu, H_1, H_2, sk_r, sk_s)\), ciphertext \(CT_j\), current period \(j\), trapdoor \(Trap_{w_{\|j\|}}\).

Output: \(N_0\) or False.

1. Compute \((y_{j1}, y_{j2}, \ldots, y_{jn}) = CT_j - Trap_{w_{\|j\|}} CT_{j+1}\).
2. Set \(y_j = (y_{j1}, y_{j2}, \ldots, y_{jn})\).
3. Select integer \(q\) satisfies \([1, 2, \ldots, q] \in Z^+_\). 
4. for \((i = 1, 2, \ldots, l)\) do 
5. if \(|y_{ji} - \lfloor q/2 \rfloor| \geq \lfloor q/4 \rfloor\) then 
6. The medical cloud server will abort it and Return False.
7. else 
8. Set \(y_{ji} = 1\) up to \(y_{j1} = 1\)
9. end if 
10. endfor 
11. if \(y_j = (1, 1, \ldots, 1) \in \{1\}^l\) then 
12. Return \(N_0\)
13. else 
14. Return False.
15. end if 

**Algorithm 5: \(True or False \leftarrow PPSEPKS.Verification((X, \delta, \alpha, \mu, H_1, H_2, sk_r, sk_s), CT_j, tw_{\|j\|})\).**

matrix \(R_j\), and carries out the New Basis Del algorithm to compute \(sk_{A_{\|j\|}}\) as the basis of \(L_q^n(pk_{\|j\|})\), where \(pk_{\|j\|} = pk_{\|j\| - 1} \cdot R_j^{-1}\). Consequently, the challenger \(C\) appends \((pk_{\|j\|}, j, pk_{\|j\|}, R_j, sk_{\|j\|})\) to \(L_1\), and transmits \(R_j\) to attacker \(A\).

\(H_2\) query: The attacker \(A\) queries \(w\), at the same time challenger \(C\) performs the following operations:

Case 1: \(w = w^*\) and \(j = j^*\). The challenger \(C\) calculates \(R^* = H_2(u_{\|j\|})\) and sends \(R^*\) to \(A\).

Case 2: \(w \neq w^*\) or \(j \neq j^*\). The challenger \(C\) looks for \((pk_{\|j\|}, j, pk_{\|j\|}, R_j, sk_{\|j\|})\) in \(L_1\), selects a matrix \(R_{w_{\|j\|}}\), and executes the Newbasis del algorithm to generate a basis \(sk_{A_{\|j\|}}\) of \(L_q^n(pk_{\|j\|} R_{w_{\|j\|}}^{-1})\). Finally, \(C\) saves \((u_{\|j\|}, j, pk_{\|j\|} \cdot R_{w_{\|j\|}}^{-1}, R_{w_{\|j\|}}^1, sk_{A_{\|j\|}})\) in \(L_2\), and sends \(R_{w_{\|j\|}}\) to \(A\).

5.2.2. Trapdoor Query. When \(C\) receives a query for a keyword \(w\) from \(A\), \(C\) first looks at \(L_2\), and if there is no \((u_{\|j\|}, j, pk_{\|j\|} \cdot R_{w_{\|j\|}}^{-1}, R_{w_{\|j\|}}, sk_{A_{\|j\|}})\) in \(L_2\), then this process will be restarted.
5.2.3. Break-In Phase. In this process, attacker A can query the private key of the doctor in the $j > j'$ period, and $j' = r$ is set a break-in time. After A queries $H_1$ on $pk_{\|j}$, C sends the private key $sk_{\|j}$ to A.

In time $i$, which is the prior period, we can find $(pk_i\|j, pk_{i\|j}, R_{i, sk_{\|j}})$ from $L_i$ because the attacker A will perform $H_1$ queries on $pk_{i\|j}$. Further, we calculate $pk_{\|j} = pk_{j\|r+1} = pk_y \cdot (R_y^1 \cdots \cdot R_y^r)^{-1} \cdot H_1 (pk_y, \|r+1)^{-1}$, which $sk_{\|j}$ is the basis of the lattice $L^2_y (pk_{r\|j})$. After that, challenger C calculates $R_{\|j} = H_1 (pk_{\|j}) \cdots H_1 (pk_{\|j} \cdot \|1)$ and runs the NewBasisDel algorithm to obtain $pk_{\|j} = pk_{\|j} \cdot R_{\|j}^{-1}$ and $sk_{\|j}$ in time $j$. Consequently, C sends $sk_{\|j}$ to attacker A.

5.2.4. Challenge Phase. Assuming that $w_0^i$ and $w_1^i$ are two keywords, challenger C randomly selects a quantity from $\{0, 1\}$ and assigns it to $b$. Then we need to divide into the following cases according to the value of $b$.

Case 1: $b = 0$. The challenger C sends ciphertext $(CT^*_{r1}, CT^*_{r2})$ of $w_0^i$ to A.

Case 2: $b = 1$. We create $v_0 = (v_0^1, v_0^2, \ldots, v_0^q)$, $v^f = (v_1^1, v_2^1, \ldots, v_1^q)$, and $v^f = (1, 1, \ldots, 1)$. Then, $CT_{r1}^* = v_0 + [q/2] \cdot y^f$ and $CT_{r2}^* = v^f$ can be obtained. Consequently, C sends the ciphertext $(CT_{r1}^*, CT_{r2}^*)$ of $w_1^i$ to A.

5.2.5. Guess Phase. In this process, attacker A outputs $b' = 0$ or $b' = 1$ as the response of the Challenge phase.

Analysis: To begin with, according to the basic probability knowledge, the probability of C outputting the ciphertext of the keyword $w_1$ is 1/2.

Suppose that A can break the indistinguishability of the ciphertext with the probability $p$. In addition, the probability that challenger C can correctly obtain the break time is $1/m$. Consequently, C can solve the LWE hardness with the probability of $p/2m$. In a nutshell, the difficulty of the attacker to crack the indistinguishability of the ciphertext can be reduced to the difficulty of the LWE-hardness.

6. Performance Evaluation

In this section, to guarantee the forward security, anti-quantum KGA, and suitability in the medical scenarios of our PPSEB scheme, we analyze the computational expense, security property, and network communication costs of our scheme and compare our scheme with existing PEKS schemes [3, 5, 28, 29] on the actual performance in the medical background through experiments and numerical simulation technique. The experiments evaluating and testing the actual performance of our scheme are operated on a MacOS with an Intel Core i7 CPU and 16 GB RAM. The implementation of schemes is based on the C++ language, and we use medical data extremely close to actual applications of daily life to complete the experiments. Meanwhile, in order to realize the security of the $q$-ary lattices, the parameters satisfy $m > 2n \log q$, $q \geq 3$, since the algorithms counting on lattice-based cryptography are relied on $q$, $m$, $n$.

The notations of the following specific descriptions in the experiments are provided in Table 1. The accurate experimental data of 200 trials on average are shown in the following figures, and the results accord with our design objective extremely.

Our PPSEB is highly efficient compared with other PEKS schemes. As is illustrated in Table 2, the theoretical communication costs of each scheme are listed accurately.

We prove the theoretical value, and the experimental result reflects in Figure 3, demonstrating that the trapdoor size of the PPSEB scheme is the least one among the whole schemes. Along with the stabilizing growth in communication costs, our algorithm is superior to the others, indicating a hidden potential to reduce network resource consumption.

As to the actual performance, Figure 3 indicates that the PPSEB scheme reveals a considerable efficiency advantage. The PEKS size of PPSEB is relatively close to the scheme [3, 5, 28] and much less than the scheme [29]. The trapdoor size in our scheme is a quarter of [29]. However, in terms of postquantum, our proposed PPSEB is more secure than the scheme [3, 5, 28] while being applied in medical data encryption protection. Thus, it is pretty sound and acceptable for PPSEB to increase the nominal communication costs corresponding to PEKS size.

In addition, we not only analyze the computational expense and security property of our scheme but also compare it with existing PEKS schemes [3, 5, 28] through experimental medical data. As shown in Figure 4, the testing time of our scheme is also much shorter than the other existing PEKS schemes. Significantly, the more the number of retrieving keywords increases, the more apparent the superiority becomes.

Besides, we test the testing time and computational expense of the PEKS schemes and record the results in Table 3.

Our scheme realizes nearly the same as a scheme [3] in saving the computational expense and searching efficiency according to the comparison in Figure 5. When the number of retrieving keywords is 180, the testing time of [5] is 7.2s, and ours is 0.477s, which is 15.09 times that of PPSEB. As a result, our scheme is not only advantageous in terms of postquantum property, but also relatively efficient than the other schemes. Consequently, although the introduction of blockchain technology has brought a certain amount of complexity and extra overhead to our system, it is certified that our PPSEB scheme can realize the property of postquantum, forward security on maintaining the confidentiality of medical data and superiority in the applications of medical scenarios. From a more practical view, it is both convenient and swift for doctors to master the patient’s physical condition, obtain the patient’s medical records, and make the correct diagnosis promptly in practical medical scenarios. In
addition, the more profound performance of PPSEB on managing medical data of Electronic Health Records systems, such as electronic medical record and electronic prescription, need to be tested experimentally and further study in development.

In Figure 6, we compared the PEKS computational expense of PPSEB with [3, 5, 28, 29]. Among them, the PEKS computational expense of our scheme is much smaller than other schemes, which shows that our scheme has higher efficiency under the same number of retrieving keywords.

### Table 1: Notations of descriptions.

| Notations | Descriptions |
|-----------|--------------|
| $Time_{me}$ | The modular exponentiation time |
| $Time_{sm}$ | The scalar multiplication time |
| $Time_{hp}$ | The hash-to-point time |
| $Time_{pa}$ | The point addition time |
| $Time_{hf}$ | The bilinear pairing time |
| $Time_{m}$ | The hash function time |
| $Time_{m}$ | The multiplication time |
| $S_{l}$ | One element bit size in $G_1$ |
| $S_{r}$ | One element bit size in $G_{r}$ |
| $S_{p}$ | One element bit size in $Z_{p}$ |
| $S_{q}$ | One element bit size in $Z_{q}$ |
| $S_{l}$ | The security level with a value of 10 |

### Table 2: Communication costs.

| Schemes          | Size of PEKS algorithm | Size of trapdoor algorithm |
|------------------|------------------------|-----------------------------|
| Our scheme       | $(S_{l} + mS_{q})S_{q}$ | $mS_{q}$                    |
| Boneh et al. [3] | $S_{p} + S_{l}$        | $S_{l}$                     |
| Ma et al. [5]    | $5S_{k} + 3S_{l}$      | $3S_{l}$                    |
| Ma et al. [28]   | $S_{p} + S_{l}$        | $S_{l}$                     |
| Shao et al. [29] | $S_{l} + S_{l}$        | $S_{l}$                     |

![Figure 3: Communication costs comparison corresponding to trapdoor size.](image-url)
Figure 4: Communication costs comparison corresponding to PEKS size.

Table 3: Testing time and computational expense.

| Schemes         | Testing time                  | PEKS computational expense                   |
|-----------------|-------------------------------|-----------------------------------------------|
| Our scheme      | $mS_t \cdot \text{Time}_{me}$ | $(mnS_t + nS_f + m^2n)\text{Time}_{me} + \text{Time}_{hf}$ |
| Boneh et al. [3]| $\text{Time}_{hf} + \text{Time}_{hp}$ | $2\text{Time}_{pa} + \text{Time}_{hp} + 4\text{Time}_{sm} + \text{Time}_{hp} + 3\text{Time}_{hf}$ |
| Ma et al. [5]   | $5\text{Time}_{me} + 4\text{Time}_{hp} + \text{Time}_{hf}$ | $3\text{Time}_{hf} + 9\text{Time}_{me} + 3\text{Time}_{hp}$ |
| Ma et al. [28]  | $\text{Time}_{hf} + \text{Time}_{hp} + \text{Time}_{sm} + 2\text{Time}_{pa} + 2\text{Time}_{hp}$ | $\text{Time}_{hf} + 3\text{Time}_{hp} + 2\text{Time}_{me} + 4\text{Time}_{sm} + 3\text{Time}_{hp}$ |
| Shao et al. [29]| $\text{Time}_{hf} + \text{Time}_{hp}$ | $2\text{Time}_{hf} + 2\text{Time}_{me} + \text{Time}_{hp}$ |

Figure 5: The testing time comparison.
7. Conclusion

In our paper, we proposed postquantum Public-key Searchable Encryption on Blockchain (PPSEB) for E-healthcare scenarios. PPSEB is capable of resisting keyword-guessing quantum computing attacks. Moreover, our proposed scheme combines public-key searchable encryption and blockchain, avoiding turning over the searching process to a third party and enhancing the security level. Furthermore, we assure forward security, maintaining the confidentiality of medical data. Both security analysis and comprehensive performance evaluation demonstrate that PPSEB can achieve the property of searching efficiency and lightweight of lower computational cost in retrieving keywords and generating trapdoor compared with other existing E-healthcare schemes.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

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, and 62076042), the Foundation of Guizhou Provincial Key Laboratory of Public Big Data (Nos. 2019BDKFJ010 and 2019BDKFJ014), 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 (YP2020013), and Inner Mongolia discipline inspection and supervision big data laboratory open project fund (IMDBD2020020).

References

[1] I. M. Baytas, K. Lin, F. Wang, A. K. Jain, J. Zhou, and J. Zhou, “PhenoTree: interactive visual analytics for hierarchical phenotyping from large-scale electronic health records,” *IEEE Transactions on Multimedia*, vol. 18, no. 11, pp. 2257–2270, 2016.

[2] Y. Chen, S. Dong, T. Li, Y. Wang, and H. Zhou, “Dynamic multi-key FHE in asymmetric key setting from LWE,” *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 5239–5249, 2021.

[3] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, “Public key encryption with keyword search,” vol. 3027, pp. 506–522, in *Proceedings of the 23rd Annual International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT 2004)*, vol. 3027, pp. 506–522, Springer, Berlin, Heidelberg, May 2004.

[4] Q. Huang and H. Li, “An efficient public-key searchable encryption scheme secure against inside keyword guessing attacks,” *Information Sciences*, vol. 403–404, pp. 1–14, 2017.
[5] M. Ma, D. He, N. Kumar et al., “Certificateless searchable public key encryption scheme for industrial Internet of Things,” IEEE Transactions on Industrial Informatics, vol. 14, no. 2, pp. 759–767, 2017.

[6] R. Chen, Y. Mu, G. Yang et al., “Dual-server public-key encryption with keyword search for secure cloud storage,” IEEE Transactions on Information Forensics and Security, vol. 11, no. 4, pp. 789–798, 2016.

[7] T. D. Ladd, F. Jelezko, R. Laflamme et al., “Quantum computers,” Nature, vol. 464, no. 7285, pp. 45–53, 2012.

[8] A. Galindo and M. A. Martin-Delgado, “Information and quantum computation: classical and quantum aspects,” Reviews of Modern Physics, vol. 74, no. 2, pp. 347–423, 2002.

[9] W. Li, X. Li, J. Gao, and H. Wang, “Design of secure authenticated key management protocol for cloud computing environments,” IEEE Transactions on Dependable and Secure Computing, vol. 18, no. 3, pp. 1276–1290, 2021.

[10] E. Uchiteleva, A. R. Hussein, A. Shami, and A. Shami, “Lightweight dynamic group rekeying for low-power wireless networks in IIoT,” IEEE Internet of Things Journal, vol. 7, no. 6, pp. 4972–4986, 2020.

[11] Y. Cheng, S. Xu, M. Zang et al., “LPPA: a lightweight privacy-preserving authentication scheme for the Internet of drones,” in Proceedings of the 21st International Conference on Communication Technology (ICCT 2021), pp. 656–661, IEEE, Tianjin, China, October 2021.

[12] T. Li, Z. Wang, G. Yang, Y. Cui, Y. Chen, and X. Yu, “Semi-selfish mining based on hidden Markov decision process,” International Journal of Intelligent Systems, vol. 36, no. 7, pp. 3596–3612, 2021.

[13] T. Li, Z. Wang, Y. Chen et al., “Is semi-selfish mining available without being detected?” International Journal of Intelligent Systems, early access, vol. 33, no. 1, 2021.

[14] Y. Chen, J. Sun, Y. Yang, T. Li, X. Niu, and H. Zhou, “PSSPR: a source location privacy protection scheme based on sector phantom routing in WSNs,” International Journal of Intelligent Systems, vol. 37, no. 2, pp. 1204–1221, 2022.

[15] X. Liu, R. Zhang, G. Xu, X.-B. Chen, and N. N. Xiong, “Confidentially judging the relationship between an integer and an interval against malicious adversaries and its applications,” Computer Communications, vol. 180, pp. 115–125, 2021.

[16] G. Xu, Y. Cao, S. Xu 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.

[17] T. Li, Y. Chen, Y. Wang et al., “Rational protocols and attacks in blockchain system,” Security and Communication Networks, vol. 2020, pp. 1–11, Article ID 8839047, 2020.

[18] S. Nakamoto, “Bitcoin: a peer-to-peer electronic cash system,” 2008, https://bitcoin.org/bitcoin.pdf.

[19] 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.

[20] 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.

[21] C. Liu, Y. Xiao, V. Javangula, Q. Hu, S. Wang, and X. Cheng, “NormaChain: a blockchain-based normalized autonomous transaction settlement system for IoT-based E-commerce,” IEEE Internet of Things Journal, vol. 6, no. 3, pp. 4680–4693, 2019.

[22] D. Micciancio and O. Regev, “Lattice-based cryptography,” vol. 4117, pp. 131–141, in Proceedings of the 26th Annual International Cryptology Conference (CRYPTO 2006), vol. 4117, pp. 131–141, Springer, Berlin, Heidelberg, May 2006.

[23] S. Xu, X. Chen, C. Wang et al., “A lattice-based ring signature scheme to secure automated valet parking,” vol. 12938, pp. 70–83, in Proceedings of the International Conference on 16th Wireless Algorithms, Systems, and Applications (WSA 2021), vol. 12938, pp. 70–83, Springer, Nanjing, China, September 2021.

[24] J. Alwen and C. Peikert, “Generating shorter bases for hard random lattices,” Theory of Computing Systems, vol. 48, no. 3, pp. 535–553, 2011.

[25] C. Gentry, C. Peikert, and V. Vaikuntanathan, “Trapdoors for hard lattices and new cryptographic constructions,” vol. 14, pp. 197–206, in Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC 2008), vol. 14, pp. 197–206, ACM, Victoria, British Columbia, May 2008.

[26] S. Agrawal, D. Boneh, and X. Boyen, “Efficient lattice (H)IBE in the standard model,” vol. 6110, pp. 553–572, in Proceedings of the 29th Annual International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT 2010), vol. 6110, pp. 553–572, Springer, Riviera, France, June 2010.

[27] S. Agrawal, D. Boneh, and X. Boyen, “Lattice basis delegation in fixed dimension and shorter-plaintext hierarchical IBE,” vol. 6223, pp. 98–115, in Proceedings of the 30th Annual International Cryptology Conference (CRYPTO 2010), vol. 6223, pp. 98–115, Springer, Santa Barbara, CA, USA, May 2010.

[28] M. Ma, D. He, M. K. Khanand, and J. Chen, “Certificateless searchable public key encryption scheme for mobile healthcare system,” Computers & Electrical Engineering, vol. 65, pp. 413–424, 2017.

[29] Z.-Y. Shao, B. Yang, and B. Yang, “On security against the server in designated tester public key encryption with keyword search,” Information Processing Letters, vol. 115, no. 12, pp. 957–961, 2015.