AAQ-PEKS: An Attribute-based Anti-Quantum Public-Key Encryption Scheme with Keyword Search for E-healthcare Scenarios

Gang Xu, Shiyuan Xu, *Student Member, IEEE*, Yibo Cao, *Student Member, IEEE*, Ke Xiao, Xiu-Bo Chen, Mianxiong Dong , *Senior Member, IEEE*, and Shui Yu , *Fellow, IEEE*

***Abstract*—Electronic Medical Records (EMRs) have been utilized in plentiful medical institutions due to their superior convenience and low storage overhead. Nevertheless, it is difficult for medical departments with disparate management regulations to share EMRs through secure communication channels since sensitive EMRs are prone to be tampered with. Therefore, the EMRs should be encrypted before being outsourced to the network servers. Public key Encryption with Keyword Search (PEKS) has the ability for doctors to search encrypted EMRs, but traditional PEKS algorithms are susceptible to quantum computing attacks and without considering access control. To address the aforementioned issues, we proposed AAQ-PEKS scheme, named an attribute-based anti-quantum public-key encryption scheme with keyword search. Initially, based on the LWE hardness, we first introduce the attribute-based PEKS that can resist quantum attacks in E-health scenarios. Secondly, we combine Attribute-Based Encryption (ABE) into AAQ-PEKS to realize access control for sensitive EMRs. Thirdly, the computational security analysis illustrates that our scheme achieves correctness, Indistinguishability against Chosen Plaintext Attack (IND-CPA) and Indistinguishability against Chosen Keyword Attack (IND-CKA). Lastly, comprehensive performance evaluation in practice elaborates that our AAQ-PEKS is more efficient compared with other existing top-tier schemes. To conclude, our scheme has the characteristics of resisting quantum attacks and fine-grained access control for E-health scenarios.**

***Index Terms*—Electronic medical records, public-key encryption with keyword search, access control, lattice-based cryptography, privacy-preserving.**

I. INTRODUCTION

In recent decades, with the wide application of cloud storage, more and more medical institutions and hospitals store EMRs in the cloud, significantly enhancing the service efficiency of hospitals and reducing the storage overhead of medical data [1]. Nevertheless, different medical institutions always have different management regulations for EMRs, and it is difficult to share them with each other [2]. EMRs are also vulnerable to malicious exploitation due to their containing private information and sensitive data [3]. There are several threat models for the leakage of EMRs, including chosen-plaintext attack (CPA) and chosen keyword attack (CKA) from malicious attackers [4]. Electronic medical records are usually encrypted through cryptographic algorithms (such as RSA and ECC) and then uploaded to medical institutions to protect their privacy [5]. However, encrypted EMRs are tough for doctors to retrieve and for medical institutions to access control. In addition, when doctors look for EMRs, they must fetch the data and decrypt it to get the corresponding data. Thus, it will waste numerous storage costs, seriously affecting the working efficiency and excessive occupancy of public space.

To solve the above-mentioned hindrances, searchable encryption is considered a proper cryptographic initiative [6]. Searchable encryption enables data sharing between communication parties by searching ciphertext based on keywords. However, most conventional searchable encryption, based on bilinear pairing, is not capable of resisting quantum computing attacks [4, 6, 7, 20, 48], which will cause EMR data tempered by adversaries. In addition to this, numerous schemes cannot achieve access control due to design flaws and not being considerate. Without access control, all physicians can retrieve a patient's EMR, and the privacy of the EMR cannot be guaranteed thoroughly. Furthermore, as for the E-health scenarios, we also need to achieve the authorization keyword retrieval of encrypted electronic medical records.

In summary, we conclude the potential problems of security in E-health scenarios as below. When doctors search for EMRs ciphertext, the problems of quantum computing attacks and leakage of access control need to be solved urgently.

A. Contribution

To circumvent these threats, we propose an attribute-based anti-quantum public-key encryption scheme with a keyword search named AAQ-PEKS in the E-healthcare scenarios.

This work was finished when Shiyuan Xu and Yibo Cao are at North China University of Technology. 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. 62271070, 92046001), the Fundamental Research Funds for Beijing Municipal Commission of Education, Beijing Urban Governance Research Base of North China University of Technology, Inner Mongolia Natural Science Foundation (2021MS06006), 2023 Inner Mongolia Young Science and Technology Talents Support Project (NJYT23106), 2022 Fund Project of Central Government Guiding Local Science and Technology Development (2022ZY0024) *(Corresponding author: Xiu-Bo Chen; Mianxiong Dong)*

Gang Xu and Ke Xiao are with the School of Information Science and Technology, North China University of Technology, Beijing, 100144, China (e-mail: gx@ncut.edu.cn; xiaoke@ncut.edu.cn).

Shiyuan Xu is with Department of Computer Science, The University of Hong Kong, Hong Kong. (e-mail: syxu2@cs.hku.hk).

Yibo Cao and Xiu-Bo Chen are with the Information Security Center, State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, 100876, China; School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing, China. (e-mail: a18613361692@163. com; flyover100@163.com).

Mianxiong Dong is with the Department of Sciences and Informatics, Muroran Institution of Technology, Muroran 050-8585, Japan (e-mail: mx.dong@csse.muroran-it.ac.jp).

Shui Yu is with the School of Computer Science, University of Technology Sydney, Sydney, NSW 2007, Australia (e-mail: shui.yu@uts.edu.au).

Firstly, as an EMR owner, the patient encrypts his/her own EMR and sets the access condition using his/her attribute. Then, the patient extracts the keyword index and uploads it to the Trusted Servicer with EMR ciphertext. After that, the doctor utilizes their secret key and attribute to generate a trapdoor for EMR search. When the doctor submits a trapdoor, the Trusted Servicer checks whether the doctor's attributes meet the access conditions. If true, the EMR ciphertext corresponding to the trapdoor will be returned to the doctor. Last but not least, the doctor decrypts the EMR ciphertext with his/her secret key and obtains the correct plaintext. Specifically, we illustrate our contributions as follows:

(1) Due to the fact that quantum computing has threatened many traditional cryptographic primitives, we need to achieve quantum-safe in our scheme. Besides, EMR data can be accessed arbitrarily by doctors, which will cause privacy leakage of patients. Thus, we construct an AAQ-PEKS, integrating lattice-based cryptographic primitive to resist quantum attacks from malicious attackers, which is the first scheme to introduce attribute-based character into PEKS while.

(2) Furthermore, we also introduce attribute-based encryption into AAQ-PEKS to achieve fine-grained access control for sensitive EMRs, enabling specific roles that meet access conditions to search and decrypt EMRs to enhance security and privacy. In this way, only the patient's attending physician can access the EMR.

(3) As for the security issues, our scheme achieves the correctness of Search and Decrypt algorithms, IND-CPA, and IND-CKA under the LWE assumptions. Through the security reduction, we give the provable theorems in theory and provide several corollaries for the practical scenarios. We also compare security primitives and assumptions with other existing schemes to illustrate the superiority of our AAQ-PEKS.

(4) We conduct comprehensive performance evaluations both in theory and practice, which specify communication efficiency, computation efficiency, the computation overhead of the encryption and search phase, and operational times, respectively, elaborating the practical of our scheme in E-healthcare scenarios as expected.

B. Outline

Section 2 gives the literature reviews compared with other research outputs. After that, we cover the preliminary part in Section 3. Our system model, threat model, and design goals will be specified in Section 4. While the concrete procedures of our proposed schemes and algorithms are elaborated in Section 5. In Sections 6 and 7, we give the comprehensive experimental performance evaluation as well as the theoretical and practical security analysis, respectively. Ultimately, we will conclude this paper in the final section.

II. Related Work

Public key encryption with keyword search (aka. PEKS), initially proposed by Boneh et al. in 2004, has been utilized to retrieve encryption messages without disclosing private information [6]. From then on, numerous researchers devoted themselves to enhancing the security level and the manifold functionality [8-20]. Specifically, Jiang et al. proposed the PEAKS scheme, achieving the authority to search keywords when users do not have the corresponding secret key [8]. However, its practical efficiency is inadequate and without considering multi-role. Then, Chen et al. put forward a novel PEKS scheme based on the dual-server architecture [9] to make it more suitable for real schemes. In [10], Chen et al. incorporated the advantage of [8] and [9], comping up with the Dual-Server architecture together with the authentication function for the traditional PEKS scheme. Furthermore, Yuan et al. introduced another PEKS method for similarity search when the data is high-dimensional, alleviating the computational efficiency from a practical point of view [11]. From reference [12], we notice that researchers initialized a method simplifying the key management through an identity-based key exchange measure. After that, Song et al. proposed the FAST and FASTIO, achieving the searchable encryption together with forward privacy and enhancing the I/O efficiency through symmetric cryptographic primitives [13]. In 2021, Cui et al. introduced the multiuser searchable encryption (MUSE) protocol for the industrial internet of things (IIoT) through key-insulation to promote the tolerance to key exposure [14]. Moreover, Hoang et al. presented a novel Incidence Matrix (IM)-DSSE scheme based on the conventional Dynamic Searchable Symmetric Encryption (DSSE), with the characteristic of secure, efficient, and low storage on real cloud settings [15]. In addition to this, there are several hotspots in terms of security and privacy for the log systems [16] and also for the E-health scenarios [17]. Zhang et al. has proposed a survey regarding to the PEKS in healthcare, specifying four actual scenarios to utilize searchable encryption [19]. Since then, a lot of researchers are keen on introducing PEKS into practical E-health schemes, mitigating the inside and outside attacks as well as achieving access control [19-21].

Nevertheless, the aforementioned traditional PEKS scheme will be tampered with by the quantum information and computation initiatives, such as the Grover algorithm and Shor algorithm [22]. Auspiciously, some researchers have made substantial contributions to post-quantum cryptology to resist quantum attacks, including lattice-based cryptography [23, 49-51]. From reference [24], scholars initialized the first probabilistic PEKS scheme from the lattice assumption, which is a milestone for the post-quantum PEKS. Then, Wang et al. made contributions to conjunctive keyword search to realize the conjunctive PEKS based on lattice hardness [25]. Meanwhile, scholars proposed an NTRU-based PEKS protocol, enhanced the computational efficiency of the Test algorithm [26], and also give a concrete experiment for their scheme. Recently, Xu et al. indicating the first lattice-based PEKS in E-health scenarios, without considering the access control [27].

TABLE I
Cryptographic Property Comparison

Schemes	Post-quantum property	Searchable encryption	Attribute-based encryption
Jiang et al. [8]	×	√	×
Chen et al. [9]	×	√	×
Chen et al. [10]	×	√	×

Yuan et al. [11]	×	√	×
Qin et al. [12]	×	√	×
Song et al. [13]	×	√	×
Cui et al. [14]	×	√	×
Hoang et al. [15]	×	√	×
Chen et al. [17]	×	×	×
Xu et al. [19]	×	√	×
Wang et al. [20]	×	√	√
Bao et al. [21]	×	√	√
Hou et al. [24]	√	√	×
Wang et al. [25]	√	√	×
Behnia et al. [26]	√	√	×
Yin et al. [29]	×	√	√
Li et al. [30]	×	√	√
Zhao et al. [31]	√	×	√
Our scheme	√	√	√

Attribute-based encryption (ABE) has been designed for several years due to the flexibility of access control for cryptographic protocols [28]. Since then, Yin et al. presented a multi-keyword search PEKS scheme for E-health, but their efficiency is not utopian [29]. In addition, Li et al. put forward an EMK-ABSE scheme, enhancing the communication efficiency as well as achieving the attribute-based multiple keyword search algorithm in 2022 [30]. However, their schemes will be threatened by quantum attackers. Zhao et al. considered this problem and proposed the RL-ABE scheme, elaborating the ABE on lattice assumption and the revocability [31]. Nevertheless, they did not consider securing EMR data. Lately, Chen et al. put forward the AQ-ABS protocol, introducing the first attribute-based signature capable of resisting quantum attacks into E-health scenarios [5]. We compared the above-mentioned schemes in Table I.

III. Preliminary

Definition 1 (Lattice) In n-dimensional space, let $\mathbf{B}=[\mathbf{b}_1,\mathbf{b}_2,...,\mathbf{b}_n]\in\mathbb{R}^m$ are n linearly independent vectors. The lattice $L(\mathbf{B})$ is defined that:

$$L(\mathbf{B})=\{x_1\mathbf{b}_1+x_2\mathbf{b}_2+...+x_n\mathbf{b}_n : x_i\in\mathbb{Z}, i=1,2,...,n\}$$

$\mathbf{b}_1,\mathbf{b}_2,...,\mathbf{b}_n$ is known as a basis of L.

Definition 2 (Statistical Distance) Here, we show the definition of statistical distance between X, Y that:

$$\Delta(X,Y)=\frac{1}{2}\sum_{a\in D}\left|\Pr[X=a]-\Pr[Y=a]\right|$$

, which X, Y are two random variables over a distribution D.

Definition 3 (Discrete Gaussian Distribution) Given the standard Gaussian function $\rho_{c,\sigma}(x)=\exp\frac{-\pi\|x-c\|^2}{\sigma^2}$, which c is the center and σ is the standard deviation. Then, the discrete Gaussian distribution over lattice L is defined that: $D_{L,c,\sigma}(x)=\frac{\rho_{c,\sigma}(x)}{\rho_{c,\sigma}(L)}$.

Definition 4 (Decisional LWE) Suppose there exists one prime q, a matrix $\mathbf{B}\in\mathbb{Z}_q^{n\times m}$, the error distribution D on $\mathbb{Z}_q$, and vectors $\mathbf{s}\in\mathbb{Z}_q^n$, $\mathbf{v}\in\mathbb{Z}_q^m$, we distinguish $(\mathbf{B},\mathbf{B}^T\mathbf{s}+\mathbf{e})$ from $(\mathbf{B},\mathbf{v})$, where $\mathbf{e}\in D^m$.

Lemma 1 (TrapGen) [32] Given the integer $q\geq 2$, $m\geq 2n\log q$. The polynomial time algorithm TrapGen outputs a matrix $\mathbf{A}\in\mathbb{Z}_q^{n\times m}$ which close to uniform distribution statistically and a basis $T_{\mathbf{A}}\in\mathbb{Z}^{m\times m}$, such that $\|T_{\mathbf{A}}\|\leq O(n\log q)$ and $\|T_{\mathbf{A}}\|\leq O\left(\sqrt{n\log q}\right)$.

Lemma 2 (SamplePre) [33] Let the lattice $L_q^{\perp}(\mathbf{A})$ and its trapdoor basis $T_{\mathbf{A}}\in\mathbb{Z}^{m\times m}$, then given a parameter $s\geq\|T_{\mathbf{A}}\|\omega\left(\sqrt{\log m}\right)$ and a vector $\mathbf{v}\in\mathbb{Z}_q^n$. There is a SamplePre algorithm which outputs a vector $\boldsymbol{\varepsilon}$ such that $A\boldsymbol{\varepsilon}=\mathbf{v}\bmod q$.

Lemma 3 (SampleBasis) [34] Let the integer $q\geq 2$, $m\geq 2n\log q$, then given matrix $\mathbf{A}\in\mathbb{Z}_q^{n\times km}$, a set $S\subseteq[k]$, a basis $\mathbf{B}_S$, and a parameter $\eta\geq\|\mathbf{B}_S\|\sqrt{km}\omega\left(\sqrt{\log km}\right)$. The polynomial time algorithm SampleBasis outputs a matrix $\mathbf{B}\in\mathbb{Z}_q^{n\times km}$ such that $\|\mathbf{B}\|\leq\eta$ with overwhelming probability.

Lemma 4 (SampleL) [35] Let lattice $L_q^{\perp}(\mathbf{A})$ and its trapdoor basis $T_{\mathbf{A}}$, then given a positive integer $m>n$, $q>2$, matrix $\mathbf{B}\in\mathbb{Z}_q^{n\times m'}$, parameter $s\geq\|T_{\mathbf{A}}\|\omega\left(\sqrt{\log(m+m')}\right)$ and vector $\mathbf{v}\in\mathbb{Z}_q^n$. There is a SampleL algorithm which outputs a vector $\boldsymbol{\varepsilon}\in\mathbb{Z}^{m+m'}$ closed to $D_{L_q^{\mathbf{v}}(\mathbf{A}|\mathbf{B}),s}$ such that $(\mathbf{A}|\mathbf{B})\boldsymbol{\varepsilon}=\mathbf{v}\bmod q$.

Lemma 5 (SampleR) [35] Let lattice $L_q^{\perp}(\mathbf{B})$ and its trapdoor basis $T_{\mathbf{B}}\in\mathbb{Z}^{m\times m}$, then given a positive integer $m>n$, $q>2$, matrix $\mathbf{A}\in\mathbb{Z}_q^{n\times m'}$, $\mathbf{R}\in\mathbb{Z}_q^{m'\times m}$, and vector $\mathbf{v}\in\mathbb{Z}_q^n$. Set parameter $s\geq\|T_{\mathbf{B}}\|s'\omega\left(\sqrt{\log m}\right)$ which $s'=\max_{\|x\|=1}\|\mathbf{R}x\|$. There is a SampleR algorithm which outputs a vector $\boldsymbol{\varepsilon}\in\mathbb{Z}^{m+m'}$ closed to $D_{L_q^{\mathbf{v}}(\mathbf{A}|\mathbf{AR}+\mathbf{B}),s}$ such that $(\mathbf{A}|\mathbf{AR}+\mathbf{B})\boldsymbol{\varepsilon}=\mathbf{v}\bmod q$.

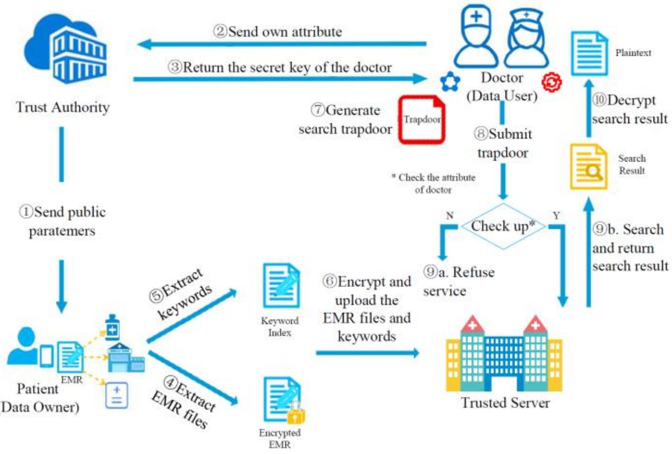

Fig. 1. System architecture.

IV. System Models, Threat Models and Design Goals

A. System Models

In this section, the architecture of our scheme is illustrated in Figure 1. In order to achieve the design goals excellently, we introduce four participating entities, including Trust Authority (TA), patient, doctor, and Trusted Server (TS).

Trusted Authority (TA): TA is the authoritative center in our scheme, and its main functions are specified as follows: (1) Parameter generating: TA is responsible for generating the security parameters and master key required by our scheme. (2) The secret key generating: TA receives the attribute of the doctor and then generates a secret key for each doctor.

Patient: The patient has the following functions: (1) EMR encryption: The patient will encrypt EMR using his/her attribute to obtain the corresponding ciphertext and uploads it to the TS. (2) Keyword extraction: The patient extracts keywords to obtain a keyword index and upload it to the TS.

Doctor: He/She has the following functions: (1) Secret key obtaining: The doctor submits his/her attribute to request TA for a secret key. (2) Trapdoor generating: Doctors generate search trapdoors using secret keys and send them to the TS. If his/her attribute meets the access condition, TS will return the corresponding EMR ciphertext to the doctor. (3) EMR decryption: After receiving the EMR ciphertext, the doctor decrypts it to calculate the EMR plaintext.

Trusted Server (TS): TS is the storage center. (1) EMR storage: TS performs trusted storage of EMR ciphertext and keyword index uploaded by the patient. (2) Access control: TS can check the attribute of the doctor. If the access condition is met, it will accept the search trapdoor sent by the doctor. Otherwise, TS rejects this request. (3) EMR search: TS accepts the search trapdoor, and executes the search algorithm to match the EMR ciphertext, and returns the search result to the doctor.

$Initialize(\lambda, l) \to (pp, \mathbf{T}_{\mathbf{M}_0})$: After input security parameter λ , the length l , it outputs pp and the master secret key $\mathbf{T}_{\mathbf{M}_0}$.

$KeyExt(pp, \mathbf{T}_{\mathbf{M}_0}, a) \to sk_a$: Having input the public parameter pp , the master secret key $\mathbf{T}_{\mathbf{M}_0}$, and attribute a . Then it will output secret key sk_a of doctor.

$Encrypt(pp, w, PT, a') \to CT$: With inputting the public parameter pp , the keyword w , the plaintext PT , and the attribute a' of the patient. Then, it outputs the ciphertext CT .

$Trapdoor(sk_a, a, w', pp) \to Trap_{w'}$: On input the secret key sk_a of doctor, the attribute a of doctor, the keyword w' , and the public parameter pp , it will output search trapdoor $Trap_{w'}$.

$Search(Trap_{w'}, a, CT) \to (\text{TURE } or \text{ FALSE})$: On input the search trapdoor $Trap_{w'}$, the attribute a of doctor, and the ciphertext CT , it will output $\text{TURE } or \text{ FALSE}$.

$Decrypt(sk_a, CT) \to (\text{TRUE } or \text{ FALSE})$: Having input the secret key sk_a of doctor, and the ciphertext CT , this algorithm will output $\text{TURE } or \text{ FALSE}$.

B. Threat Models

In a traditional searchable encryption scheme, EMR is vulnerable to quantum computing attack, chosen plaintext attack (CPA) and chosen keyword attack (CKA) from malicious attackers, posing a serious threat to the privacy.

(1) **Quantum computing attack**: In our architecture, EMR of patients confront security issues under quantum attacks.

(2) **CPA and CKA attacks**: To ensure that our scheme has secure property under CPA and CKA, two indistinguishability games are defined as follows.

Definition 5: To construct the indistinguishability game against CPA, we introduce challenger C and adversary A to interact in this game.

Setup: Challenger C executes *Initialize* algorithm to generate the public parameter, then C returns it to adversary A.

Secret-key query: A sends an attribute to C for the secret key query. Then, C executes the *KeyExt* algorithm to compute the secret key and returns it to A. Further, A conducts a trapdoor query and submits the keyword, and C runs *Trapdoor* algorithm and computes the trapdoor to A.

Challenge: A sends two keywords without a trapdoor query and an attribute to C. After that, C selects either of the two keywords and executes *Encrypt* algorithm to obtain the ciphertext. Finally, C sends this ciphertext to A.

Guess: A gives a guess for a keyword-based on ciphertext.

Definition 6: To construct the indistinguishability game against CKA, we introduce the simulator S and the adversary A to interact in this game.

Initialize: Simulator S generates the public parameter and returns it to adversary A.

Hash query: S maintains a list to record the hash query by A. Facing each hash query from A, S adds the list and returns the corresponding result to A.

Trapdoor query: Firstly, A sends an attribute and a keyword to S for the trapdoor query. Then, S generates a trapdoor and sends it to adversary A.

Challenge: A sends two keywords without a trapdoor query and an attribute to S. After that, S chooses either of the two keywords, performs a hash query, obtains the ciphertext, and then submits it to adversary A.

Guess: A sends a guess for the ciphertext to S and receives the corresponding result.

C. Design Goals

We design a lattice-based searchable encryption scheme for EMR in the E-health scenario, named AAQ-PEKS, which not only resists quantum computing attacks but also implements access control to data users (doctors) to ensure EMR security and privacy. The main features of our scheme are as follows:

(1) **Anti-quantum**: To resist quantum computing attacks initiated by malicious attackers, we apply lattice cryptography to design an anti-quantum searchable encryption scheme.

(2) **Fine-grained access control**: In our scheme, the EMR of a patient can be searched or decrypted when the doctor's attribute meets the access conditions, which realizes fine-grained access control of EMR.

(3) **Indistinguishability of against CPA and CKA**: In the traditional searchable encryption scheme, the server is vulnerable to CPA and CKA initiated by malicious attackers, affecting the privacy of EMR extremely. Consequently, in AAQ-PEKS, we reduce the difficulty of breaking the indistinguishability of CPA and CKA to LWE hardness.

(4) **Efficiency**: Instead of large number operations, we adopt matrix operations in our scheme, improving the computational efficiency in terms of encryption time and search time.

V. Our Proposed Scheme: AAQ-PEKS

We illustrate the concrete attribute-based anti-quantum public-key encryption with keyword search scheme for E-health, AAQ-PEKS, mainly incorporating six Probabilistic Polynomial-Time (PPT) algorithms (Initialize, KeyExt, Encrypt, Trapdoor, Search, Decrypt). The explanation of main variables involved in these algorithms are shown in Table II.

TABLE II
Nomenclature in Algorithms

Acronyms	Descriptions
λ	The security parameter
l, d	The length parameter
pp	The public parameter
$\mathbf{T}_{\mathbf{M}_0}$	The master secret key
sk_a	The secret key of doctor
PT	The EMR plaintext
a	The attribute of doctor
a'	The attribute of patient
w	The EMR keyword
w'	The keyword to be searched
CT	The ciphertext of EMR and keywords
$Trap_{w'}$	The search trapdoor

A. Initialize

$Initialize(pp,\mathbf{T}_{\mathbf{M}_0})$: To begin with, the system will initialize the security parameter λ and the length l as input, and then TA executes Algorithm 1 to obtain the public parameter pp, and the master secret key $\mathbf{T}_{\mathbf{M}_0}$.

```
Algorithm 1  $Initialize(pp,\mathbf{T}_{\mathbf{M}_0})$
Input: The security parameter $\lambda$ , and the length parameter $l$
Output: The public parameter $pp$ , and the master secret key $\mathbf{T}_{\mathbf{M}_0}$
Select $n,m$ ▷ $n,m$ are positive integers
Select $q$ $s.t.$ $q \geq 2$ AND $m > 6n\log q$ ▷ $q$ is a prime
Calculate $H_i : \mathbb{Z}_q^n \xleftarrow{\$} \{0,1\}^*$ ▷ $i \in [1,d]$
Generate $\mathbf{M}_0$ through calling $\mathrm{TrapGen}(q,n)$ ▷ $\mathbf{M}_0 \in \mathbb{Z}_q^{n\times m}$
Generate $\mathbf{T}_{\mathbf{M}_0}$ $s.t.$ $\|\mathbf{T}_{\mathbf{M}_0}\| \leq O(\sqrt{n\log q})$ ▷ $\mathbf{T}_{\mathbf{M}_0}$ is the master secret key
for $(i=1,2,...,l)$ do
  Select $\mathbf{M}_i$
end for
Select $\mathbf{B} \in \mathbb{Z}_q^{n\times m}$ and vector $\mathbf{v} \in \mathbb{Z}_q^n$
Set $pp = (\mathbf{M}_0,\mathbf{M}_1,...,\mathbf{M}_l,\mathbf{B},\mathbf{v})$
Return $(pp,\mathbf{T}_{\mathbf{M}_0})$
```

```
Algorithm 2  $KeyExt(sk_a)$
Input: The public parameter $pp$ , the master secret key $\mathbf{T}_{\mathbf{M}_0}$ , and attribute $a$
Output: The secret key $sk_a$ of doctor
Select attribute $a \in \{0,1\}^l$
Calculate $\mathbf{M}_a = \sum_{j\in a}(\mathbf{B}+\mathbf{M}_j)$
Call $\mathrm{SampleBasis}(\mathbf{M}_0|\mathbf{M}_a,\mathbf{T}_{\mathbf{M}_0})$ to generate $\mathbf{T}_{\mathbf{M}_0|\mathbf{M}_a}$ ▷ $\mathbf{T}_{\mathbf{M}_0|\mathbf{M}_a} \in \mathbb{Z}^{m\times m}$
Call $\mathrm{SampleL}(\mathbf{M}_0,\mathbf{M}_a,\mathbf{T}_{\mathbf{M}_0},\mathbf{v})$ to generate $\boldsymbol{\varepsilon}_a$ $s.t.$ $(\mathbf{M}_0|\mathbf{M}_a)\boldsymbol{\varepsilon}_a = \mathbf{v}$
Set $sk_a = (\boldsymbol{\varepsilon}_a,\mathbf{T}_{\mathbf{M}_0|\mathbf{M}_a})$
Return $sk_a$
```

B. KeyExt

$KeyExt(sk_a)$: Firstly, the doctor submits the own attribute to TA. Then, TA will input the parameter pp , the master secret key $\mathbf{T}_{\mathbf{M}_0}$ and perform Algorithm 2 to generate the secret key sk_a of the doctor.

```
Algorithm 3  $Encrypt(CT)$
Input: The public parameter $pp$ , the keyword $w$ , the plaintext $PT$ , the attribute $a'$ of the patient, and the length parameter $l$ , $d$
Output: The ciphertext $CT$
Select keyword $w \in \{0,1\}^*$ and plaintext $PT \in \{0,1\}$
Select the attribute $a' \in \{0,1\}^l$ of the patient
Procedure $Encrypt$
Select vector $\mathbf{c} \in \mathbb{Z}_q^n$
for $(j=1,2,...,l)$ do
  Select matrices $\mathbf{A}_j \in \{-1,1\}^{m\times m}$ randomly
end for
for $(i=1,2,...,l)$ do
  Set noise vectors $noi$ , $noi_i \leftarrow \chi$ and $\mathbf{y} \leftarrow \chi^m$
end for
Calculate $\mathbf{u} = \mathbf{v}^T\mathbf{c} + noi + PT\left\lfloor \frac{q}{2} \right\rfloor \in \mathbb{Z}_q$
for $(i=1,2,...,d)$ do
  Calculate $\mathbf{u}_i = H_i(w)^T\mathbf{c} + noi_i$ ▷ $\mathbf{u}_i \in \mathbb{Z}_q$
end for
Calculate $\boldsymbol{\mu} = {\mathbf{M}_0}^T\mathbf{c} + \mathbf{y}$ ▷ $\boldsymbol{\mu} \in \mathbb{Z}_q^m$
for $(j=1,2,...,l)$ do
  if $j \in a'$
    Calculate $\boldsymbol{\mu}_j = (\mathbf{B}+\mathbf{M}_j)^T\mathbf{c} + {\mathbf{A}_j}^T\mathbf{y}$ ▷ $\boldsymbol{\mu}_j \in \mathbb{Z}_q^m$
  else
    Select $\boldsymbol{\mu}_j$ randomly ▷ $\boldsymbol{\mu}_j \in \mathbb{Z}_q^m$
  end if
end for
Set and return $CT = (\mathbf{u},\mathbf{u}_1,\mathbf{u}_2,...,\mathbf{u}_d,\boldsymbol{\mu}_0,\boldsymbol{\mu}_1,...,\boldsymbol{\mu}_l)$
```

Encrypt

$Encrypt(CT)$: The patient takes his/her attribute a' , the keyword w, the plaintext PT, and the public parameter pp as input, and then executes Algorithm 3. After that, it will output the ciphertext CT .

D. Trapdoor

In this process, for a keyword w' to be searched, the doctor will execute Algorithm 4, which generates a corresponding search trapdoor $Trap_{w'}$.

E. Search

$Search(\text{TRUE } or \text{ FALSE})$: In the search phase, the doctor sends a search trapdoor $Trap_{w'}$ and attribute a to the trusted server. After that, the trusted server will check the attribute of the doctor who submitted a search request. If $a \subseteq a'$, the trusted server will input the ciphertext CT and perform Algorithm 5 to generate the search result $\text{TRUE } or \text{ FALSE}$. If the output is TRUE , which indicates that the search result is valid, the trusted server will return the ciphertext to doctor. Otherwise, the TS will refuse the search request of the doctor.

```
Algorithm 4  $Trapdoor(Trap_{w'})$
Input: The secret key $sk_a$ of doctor, the attribute $a$ of doctor, the keyword $w'$ searched by doctor, and the public parameter $pp$
Output: The search trapdoor $Trap_{w'}$
1: Calculate $\mathbf{M}_a = \sum_{j \in a}^{j=1} (\mathbf{B} + \mathbf{M}_j)$
2: for $(i = 1, 2, ..., d)$ do
3:     Call $\text{SamplePre}(\mathbf{M}_0 | \mathbf{M}_a, \mathbf{T}_{\mathbf{M}_0|\mathbf{M}_a}, H_i(w'))$ to generate vector $\boldsymbol{\varepsilon}_i \in \mathbb{Z}_q^{2m}$
    $s.t.$ $(\mathbf{M}_0 | \mathbf{M}_a)\boldsymbol{\varepsilon}_i = H_i(w')$ ▷ $(\mathbf{M}_0 | \mathbf{M}_a) \in \mathbb{Z}_q^n$
4: end for
5: Calculate trapdoor $Trap_{w'} = (\boldsymbol{\varepsilon}_1, \boldsymbol{\varepsilon}_2, ..., \boldsymbol{\varepsilon}_d)$
6: Return $Trap_{w'}$ to the doctor
```

```
Algorithm 5  $Search(\text{TRUE } or \text{ FALSE})$
Input: The search trapdoor $Trap_{w'}$, the attribute $a$ of doctor, the ciphertext $CT$, and the length parameter $d$
Output: $\text{TRUE } or \text{ FALSE}$
1: for $(i = 1, 2, ..., d)$ do
2:   Calculate $\boldsymbol{\delta}_i = \mathbf{u}_i - (\boldsymbol{\varepsilon}_{i,0}^T \boldsymbol{\mu}_0 + \boldsymbol{\varepsilon}_{i,1}^T \sum_{j \in a}^{j=1} \boldsymbol{\mu}_j)$
3:   if $\boldsymbol{\delta}_i < \left\lfloor \frac{q}{4} \right\rfloor$
4:     Return TRUE
5:   else
6:     Return FALSE
7:   end if
8: end for
```

F. Decrypt

$Decrypt(\text{TRUE } or \text{ FALSE})$: This algorithm will be launched by the doctor after obtaining the ciphertext from trusted server.

```
Algorithm 6  $Decrypt(\text{TRUE } or \text{ FALSE})$
Input: The secret key $sk_a$ of doctor, and the ciphertext $CT$
Output: $\text{TRUE } or \text{ FALSE}$
1: Calculate $\varpi = \mathbf{u} - (\boldsymbol{\varepsilon}_{a,0}^T \boldsymbol{\mu}_0 + \boldsymbol{\varepsilon}_{a,1}^T \sum_{j \in a}^{j=1} \boldsymbol{\mu}_j)$ ▷ $\varpi \in \mathbb{Z}_q$
2: if $\left| \varpi - \left\lfloor \frac{q}{2} \right\rfloor \right| < \left| \frac{q}{4} \right|$
3:   Return TRUE
4: else
5:   Return FALSE
6: end if
```

VI. Security Analysis

Theorem 1. Our AAQ-PEKS.Search algorithm is correct under the LWE assumptions.

Proof of Theorem 1:

Initially, we have to choose several parameters to satisfy $\text{TrapGen, SampleL, SampleR}$ algorithms, which specifies as below. $m = 6n\lceil \log q \rceil$, $q = \omega\left(\sqrt{\log n}\right) l \sqrt{m^5}$, $\sigma = m\omega\sqrt{\log n}$ and $\alpha = \frac{1}{m^2 \omega l \sqrt{\log n}}$.

For random input secret key sk_a , the doctor's attribute $a \subseteq a'$, ciphertext $CT = (\mathbf{u}, \mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_d, \boldsymbol{\mu}_0, \boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_l)$, for each $j \in a$, it always finds $\exists \boldsymbol{\mu}_j$. We also set $\boldsymbol{\varepsilon}_{a,0}, \boldsymbol{\varepsilon}_{a,1} \in \mathbb{Z}^m$.

If $w = w'$,We compute:

$$\boldsymbol{\delta}_i = \mathbf{u}_i - (\boldsymbol{\varepsilon}_{i,0}^T \mu_0 + \boldsymbol{\varepsilon}_{i,1}^T \sum_{j \in a}^{j=1} \boldsymbol{\mu}_j) = H_i(w)^T \mathbf{c} + noi_i - (\boldsymbol{\varepsilon}_{i,0}^T \mu_0 + \boldsymbol{\varepsilon}_{i,1}^T \sum_{j \in a}^{j=1} \boldsymbol{\mu}_j)$$

$$= H_i(w)^T \mathbf{c} + noi_i - (\boldsymbol{\varepsilon}_{i,0}^T (\mathbf{M}_0 \mathbf{c} + \mathbf{y}) + \sum_{j \in a}^{j=1} \boldsymbol{\varepsilon}_{i,1}^T ((\mathbf{B} + \mathbf{M}_j)^T \mathbf{c} + \mathbf{A}_j \mathbf{y}))$$

$$= H_i(w)^T \mathbf{c} + noi_i - (\boldsymbol{\varepsilon}_{i,0}^T (\mathbf{M}_0 \mathbf{c} + \mathbf{y}) + \sum_{j \in a}^{j=1} \boldsymbol{\varepsilon}_{i,1}^T (\mathbf{B} + \mathbf{M}_j)^T \mathbf{c}) - \sum_{j \in a}^{j=1} \boldsymbol{\varepsilon}_{i,1}^T \mathbf{A}_j \mathbf{y}$$

$$= H_i(w)^T \mathbf{c} + noi_i - (\mathbf{M}_0 \boldsymbol{\varepsilon}_{i,0} + \mathbf{M}_a \boldsymbol{\varepsilon}_{i,1})^T \mathbf{c} - \boldsymbol{\varepsilon}_{i,0}^T \mathbf{y} - \sum_{j \in a}^{j=1} \boldsymbol{\varepsilon}_{i,1}^T \mathbf{A}_j \mathbf{y}$$

$$= H_i(w)^T \mathbf{c} + noi_i - ((\mathbf{M}_0 | \mathbf{M}_a) \boldsymbol{\varepsilon}_i)^T \mathbf{c} - \boldsymbol{\varepsilon}_{i,0}^T \mathbf{y} - \sum_{j \in a}^{j=1} \boldsymbol{\varepsilon}_{i,1}^T \mathbf{A}_j \mathbf{y}$$

$$= (H_i(w)^T \mathbf{c} - H_i(w)^T \mathbf{c}) + noi_i - \boldsymbol{\varepsilon}_{i,0}^T \mathbf{y} - \sum_{j \in a}^{j=1} \boldsymbol{\varepsilon}_{i,1}^T \mathbf{A}_j \mathbf{y}$$

$$= noi_i - \boldsymbol{\varepsilon}_{i,0}^T \mathbf{y} - \sum_{j \in a}^{j=1} \boldsymbol{\varepsilon}_{i,1}^T \mathbf{A}_j \mathbf{y} = noi_i - (\boldsymbol{\varepsilon}_{i,0} + \sum_{j \in a}^{j=1} \mathbf{A}_j \boldsymbol{\varepsilon}_{i,1})^T \mathbf{y}$$

Thus, we get the error term of the AAQ-PEKS.Search algorithm that is $noi_i - (\boldsymbol{\varepsilon}_{i,0} + \sum_{j \in a}^{j=1} \mathbf{A}_j \boldsymbol{\varepsilon}_{i,1})^T \mathbf{y}$.

It is easy to bound $\left| noi_i - (\boldsymbol{\varepsilon}_{i,0} + \sum_{j \in a}^{j=1} \mathbf{A}_j \boldsymbol{\varepsilon}_{i,1})^T \mathbf{y} \right| \le \frac{q}{5}$ to satisfy the correctness of our proposed scheme [35]. Then, due to the fact that $\boldsymbol{\varepsilon}_a = (\boldsymbol{\varepsilon}_{a,0}, \boldsymbol{\varepsilon}_{a,1}) \xleftarrow{\$} \text{SampleLeft}$, we established $\|\boldsymbol{\varepsilon}_{a,0}\| \le \sigma\sqrt{m}$ and $\|\boldsymbol{\varepsilon}_{a,1}\| \le \sigma\sqrt{m}$ with overwhelming probability.

Theorem 2. Our AAQ-PEKS.Decrypt algorithm is correct under the LWE assumption.

Proof of Theorem 2:

For random input secret key sk_a , the doctor's attribute

$a \subseteq a'$, ciphertext $CT = (\mathbf{u}, \mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_d, \boldsymbol{\mu}_0, \boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_l)$, for each $j \in a$, we compute:

$$\varpi = \mathbf{u} - (\boldsymbol{\varepsilon}_{a,0}^T \mu_0 + \boldsymbol{\varepsilon}_{a,1}^T \sum_{j\in a}^{j=1} \boldsymbol{\mu}_j) = \mathbf{v}^T \mathbf{c} + noi + PT \left\lfloor \frac{q}{2} \right\rfloor - (\boldsymbol{\varepsilon}_{a,0}^T \mu_0 + \boldsymbol{\varepsilon}_{a,1}^T \sum_{j\in a}^{j=1} \boldsymbol{\mu}_j)$$

$$= \mathbf{v}^T \mathbf{c} + noi + PT \left\lfloor \frac{q}{2} \right\rfloor - \mathbf{v}^T \mathbf{c} - \boldsymbol{\varepsilon}_{a,0}^T \mathbf{y} - \sum_{j\in a}^{j=1} \boldsymbol{\varepsilon}_{a,1}^T \mathbf{A}_j \mathbf{y}$$

$$= noi + PT \left\lfloor \frac{q}{2} \right\rfloor - (\boldsymbol{\varepsilon}_{a,0} + \sum_{j\in a}^{j=1} \mathbf{A}_j \boldsymbol{\varepsilon}_{a,1})^T \mathbf{y}$$

Thus, we get the error term of the AAQ-PEKS.Decrypt algorithm that is $noi - (\boldsymbol{\varepsilon}_{a,0} + \sum_{j\in a}^{j=1} \mathbf{A}_j \boldsymbol{\varepsilon}_{a,1})^T \mathbf{y}$. It is ordinary to bound $\left| noi - (\boldsymbol{\varepsilon}_{a,0} + \sum_{j\in a}^{j=1} \mathbf{A}_j \boldsymbol{\varepsilon}_{a,1})^T \mathbf{y} \right| \le \frac{q}{5}$ to satisfy the correctness of our proposed scheme [35]. Next, because of $\boldsymbol{\varepsilon}_i = (\boldsymbol{\varepsilon}_{i,0}, \boldsymbol{\varepsilon}_{i,1}) \xleftarrow{\$} \text{SamplePre}$, we established $\left\| \boldsymbol{\varepsilon}_{i,0} \right\| \le \sigma\sqrt{m}$, $\left\| \boldsymbol{\varepsilon}_{i,1} \right\| \le \sigma\sqrt{m}$, and $\left\| \boldsymbol{\varepsilon}_{a,0} + \sum_{j\in a}^{j=1} \mathbf{A}_j \boldsymbol{\varepsilon}_{a,1} \right\| \le \left\| \boldsymbol{\varepsilon}_{a,0} \right\| + \sum_{j\in a}^{j=1} \left\| \mathbf{A}_j \boldsymbol{\varepsilon}_{a,1} \right\| \le O(l\sigma m)$ with overwhelming probability.

Then, we have:

$$\left| noi - (\boldsymbol{\varepsilon}_{a,0} + \sum_{j\in a}^{j=1} \mathbf{A}_j \boldsymbol{\varepsilon}_{a,1})^T \mathbf{y} \right|$$

$$\le \alpha\varpi q(\sqrt{\log m}) + (q\alpha\varpi(\sqrt{\log m}) + \frac{\sqrt{m}}{2})O(ml\sigma) + \frac{1}{2}$$

$$\le \alpha\varpi qml\sigma(\sqrt{\log m}) + O\sqrt[3]{m^2 l\sigma}$$

Theorem 3. The IND-CPA of AAQ-PEKS can be reduced to the LWE hardness assumption.

Proof of Theorem 3:

To begin with, the hash function is defined as $H = \left\{ h_{\boldsymbol{\delta}} : (\mathbb{Z}_q^l)^* \to \mathbb{Z}_q \right\}$, and $h_{\boldsymbol{\delta}}(a) = \sum_{j\in a} \boldsymbol{\delta}_j a_j + r$, where $r \in \mathbb{Z}_q$. To prove Theorem 3, we introduce the following games.

Game 0: **Setup**: Challenger C runs $Initialize(pp, \mathbf{T}_{\mathbf{M}_0})$ algorithm to generate the public parameter pp and returns it to Adversary A.

Secret-key query: A sends attribute $a \in \{0,1\}^l$ to C for a secret key query. Then, C executes the $KeyExt(sk_a)$ algorithm to calculate the secret key sk_a and returns it to A. Further, A conducts a trapdoor query and submits the keyword w , and C processes the $Trapdoor(Trap_w)$ algorithm using sk_a and computes the trapdoor $Trap_w$ to A.

Challenge: A sends keywords w_0 and w_1 without trapdoor query and attribute $a^* \in \{0,1\}^l$ to C, such that $a \not\subset a^*$. Moreover, C chooses $b = 0$ or $b = 1$ and executes $Encrypt(CT_b)$ algorithm to obtain the ciphertext CT_b . Finally, C sends CT_b to A.

Guess: A receives CT_b and gives a guess for $b \in \{0,1\}$.

Game 1: Compared with **Game 0**, the difference of **Game 1** is the selection of matrix $\mathbf{M}_j$ and abort operation in **Setup** phase, where $j \in [1, l]$.

In **Setup**, C selects matrices $\mathbf{A}_j^* \in \{-1,1\}^{m\times m}$ randomly and $\boldsymbol{\delta}_j \in \mathbb{Z}_q$ for $j \in [1, l]$. Then, C calculates $\mathbf{M}_j = \boldsymbol{\delta}_j \mathbf{B} + \mathbf{M}_0 \mathbf{A}_j^*$ to generate $\mathbf{M}_j$ in AAQ-PEKS.Initialize algorithm. After that, C chooses $h \in H$ as the hash function in the secret key query from A. In **Guess**, let q_t be the number of private key queries made by A. After receiving A's guess, if $h(a_1) \ne 0, h(a_2) \ne 0, ..., h(a_{q_t}) \ne 0$ and $h(a^*) = 0$, C continues this game. Otherwise, C covers $b \in \{0,1\}$ and halts this game.

According to [35], for $j \in [1, l]$, $\boldsymbol{\delta}_j \mathbf{B} + \mathbf{M}_0 \mathbf{A}_j^*$ closes $\mathbf{M}_j$ statically, which $\mathbf{M}_j$ is selected randomly by C. Furthermore, from A's point of view, it is also random and independent whether the **Guess** phase is halted. Consequently, **Game 0** and **Game 1** are indistinguishable to attacker A.

Game 2: Compared with **Game 1**, the difference of **Game 2** is the selection of matrix $\mathbf{M}_0$ and $\mathbf{B}$.

In **Setup**, C selects $\mathbf{M}_0$ and $\mathbf{B}$ randomly to obtain a lattice $L_q^{\perp}(\mathbf{B})$ and a corresponding trapdoor $\mathbf{T}_{\mathbf{B}}$. Then, in **Phase 1**, C calculates

$$\mathbf{M}_a = \left(\mathbf{M}_0 \middle| \sum_{j\in a} (\mathbf{B} + \mathbf{M}_j) \right) = \left(\mathbf{M}_0 \middle| \mathbf{M}_0 \sum_{j\in a} R_j^* + \left(\sum_{j\in a} \boldsymbol{\delta}_j a_j + r \right) \mathbf{B} \right)$$

. If $\left(\sum_{j\in a} \boldsymbol{\delta}_j a_j + r \right) = h(a) = 0$, C halts this game. Otherwise, C can obtain $\mathbf{T}_{\mathbf{M}_a}$ according to reference [35]. Then, C executes SampleR algorithm to sample a vector $\boldsymbol{\varepsilon}$ and make $sk_a = (\boldsymbol{\varepsilon}, \mathbf{T}_{\mathbf{M}_a})$ as the private key.

Due to the private key sk_a is statistically close to **Game 2**, **Game 1** and **Game 2** are indistinguishable from attacker A.

Game 3: Compared with **Game 2**, the difference of **Game 3** is the selection of ciphertext CT . In **Challenge**, C completely random access to ciphertext CT . In order to prove that **Game 2** and **Game 3** are indistinguishable, we need to introduce a challenger S to complete this proof.

We make an assumption as attacker A has the ability to distinguish **Game 2** and **Game 3** with overwhelming probability. Further, we utilize adversary A together with challenger C to defeat the LWE hardness, the details elaborated as the following:

Initialization part: (1)

```
for $(i=0,1,...,m)$
    Calculate $\eta_i = \mathbf{u}_i^T\mathbf{c}+noi$, where $\eta_i \in \mathbb{Z}_q^n$ and $\mathbf{u}_i \in \mathbb{Z}_q$
    Challenger $S \xleftarrow{\mathbf{v}_i,\eta_i} C$
end for
```

(2) We set $\mathbf{M}_0 = (\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_m) \in \mathbb{Z}_q^{n\times m}$ and we consider the zero vector in the LWE hardness is $\mathbf{v}_0$ under the circumstance of public parameter pp. Then, we compute:

```
for $(i=1,2,...,l)$
    $\mathbf{M}_i = \mathbf{M}_0\mathbf{A}_i^* + \delta_i\mathbf{B}$, where $\mathbf{B}$ has been initialized in Game 2
end for
```

(3) We set $pp = (\mathbf{M}_0, \mathbf{M}_1, ..., \mathbf{M}_l, \mathbf{B}, \mathbf{v}_0)$ and give it to Attacker A to end this part.

Challenge part:(1) Adversary A initially chooses $a \in \{0,1\}^l$

```
if $h(a) \neq 0$
    Continue the game
else
    Challenger S establishes one secret key through Game 2 and gives it to A
end if
```

(2) Attacker A secondly chooses one attribute for challenge C $a^* \in \{0,1\}^l$ s.t. $a_i \not\subset a^*$ and two plaintext messages $PT_0, PT_1 \in \{0,1\}$.

```
if $h(a^*) = 0$
    Challenger S will firstly select $b \in \{0,1\}$
    Set $\boldsymbol{\eta}^* = (\eta_1, \eta_2, ..., \eta_m)^T \rhd \boldsymbol{\eta}^* \in \mathbb{Z}_q^m$
    Set $u^* = \eta_0 + PT_b\left\lfloor \frac{q}{2} \right\rfloor$ as the ciphertext $\rhd u^* \in \mathbb{Z}_q$
    if $j \notin a^*$
        Select $\boldsymbol{\mu}_j^*$ randomly
    else
        Set $\boldsymbol{\mu}_j^* = (\mathbf{A}_j^*)^T\boldsymbol{\eta}^*$ s.t. $\mathbf{M}_0\mathbf{A}_i^* = \mathbf{M}_i - \delta_i\mathbf{B}$
    end if
    Calculate $CT^* = (u^*, \eta_1, \eta_2, ..., \eta_m, \eta^*, \boldsymbol{\mu}_1^*, \boldsymbol{\mu}_2^*, ..., \boldsymbol{\mu}_l^*)$
    Return $CT^*$ to A
else
    Challenger S will finish the game
end if
```

Guess part:

Adversary A gives a guess for $b^* \leftarrow \{0,1\}$

```
if $b^* \neq b$
    S obtained $\mathbf{v}_i \leftarrow \mathbb{Z}_q^n$ and $\eta_i \leftarrow \mathbb{Z}_q$ randomly.
    To conclude, $CT^*$ can be reduced to Game 3.
else
    $y_i \leftarrow noi$, $\mathbf{v}_i \leftarrow \mathbb{Z}_q^n$ and $\mathbf{v}_i^T\mathbf{c} + y_i \xleftarrow{\$} \mathbb{Z}_q$ sampled by S
    Set $\eta_i = \mathbf{v}_i^T\mathbf{c} + \chi$
    Calculate $u^* = y_0 + PT_b\left\lfloor \frac{q}{2} \right\rfloor + \mathbf{v}_0^T\mathbf{c} \rhd u^* \in \mathbb{Z}_q$
    for $(i=1,2,...,b)$
        Calculate $\eta_i = \mathbf{v}_i^T\mathbf{c} + \chi \rhd \eta_i \in \mathbb{Z}_q$
    end for
    Calculate $\boldsymbol{\eta}^* = \mathbf{M}_0^T\mathbf{c} + \boldsymbol{y}$ s.t. $\boldsymbol{y} = (y_1, y_2, ..., y_m)^T \rhd \boldsymbol{\eta}^* \in \mathbb{Z}_q^m$
    Due to $h\left(a^*\right) = \left(\sum_{j\in a}\boldsymbol{\delta}_j a_j^* + r\right) = 0$
    We get:
    $\left(\mathbf{M}_0 \middle| \sum_{j\in a^*}\left(\mathbf{B}+\mathbf{M}_j\right)\right) = \left(\mathbf{M}_0 \middle| \sum_{j\in a}\left(\mathbf{B}+\boldsymbol{\delta}_j\mathbf{B}+\mathbf{M}_0\mathbf{A}_j^*\right)\right)$
    $= \left(\mathbf{M}_0 \middle| \mathbf{B}\sum_{j\in a}\left(1+\boldsymbol{\delta}_j\right) + \sum_{j\in a}\mathbf{M}_0\mathbf{A}_j^*\right)$
    $= \left(\mathbf{M}_0 \middle| \left(\sum_{j\in a}\left(\boldsymbol{\delta}_j a_j\right)+r\right)\mathbf{B} + \sum_{j\in a}\mathbf{M}_0\mathbf{A}_j^*\right) = \left(\mathbf{M}_0 \middle| \sum_{j\in a}\mathbf{M}_0\mathbf{A}_j^*\right)$
    Calculate $\mathbf{B}+\mathbf{M}_j = \mathbf{M}_0\mathbf{A}_j^*$ and $c_j^* = (\mathbf{B}+\mathbf{M}_j)^T\mathbf{c} + \mathbf{A}_j^T\mathbf{y}$
    Return with $CT^* = (u^*, \eta_1, \eta_2, ..., \eta_m, \eta^*, \boldsymbol{\mu}_1^*, \boldsymbol{\mu}_2^*, ..., \boldsymbol{\mu}_l^*)$
    To conclude, $CT^*$ can be reduced to Game 3.
end if
```

In a nutshell, the difficulty of distinguishing **Game 3** from **Game 4** to break the IND-CPA of our scheme can be reduced to LWE hardness.

Theorem 4. The IND-CKA of AAQ-PEKS can be reduced to the LWE hardness assumption.

Proof of Theorem 4:

We assume that there exists an adversary A under the random oracle model, which will break the IND-CKA of our scheme in polynomial time. According to this, we have created a simulator S having the ability to solve the LWE hardness.

To begin with, for $j \in [0, m]$, S selects the vector $\mathbf{v}_j$ and computes $\eta_j = \mathbf{v}_j^T\mathbf{c} + y_j$, such that y_j is sampled from distribution χ. Then, the following steps will be executed in sequence.

Initialization part:

```
for $(i=0,1,...,m)$
    Calculate $\eta_i = \mathbf{u}_i^T\mathbf{c}+noi$, where $\eta_i \in \mathbb{Z}_q^n$ and $\mathbf{u}_i \in \mathbb{Z}_q$
    Challenger $S \xleftarrow{\mathbf{v}_i,\eta_i} C$
end for
```

We set $\mathbf{M}_0 = (\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_m) \in \mathbb{Z}_q^{n\times m}$ and we consider the zero vector in the LWE hardness is $\mathbf{v}_0$ under the circumstance of public parameter pp. Then, we compute

```
for $(i = 1, 2, ..., l)$
    $\mathbf{M}_i = \mathbf{M}_0 \mathbf{A}_i^* + \delta_i \mathbf{B}$, where $\mathbf{B}$ has been initialized through Game 2
end for
```

We set $pp = (\mathbf{M}_0, \mathbf{M}_1, ..., \mathbf{M}_l, \mathbf{B}, \mathbf{v}_0)$ and give it to Attacker A to end this part.

Hash part: Let p_i be the number of H_i queries made by A. During this process, S maintains a list $L_i := \{\mathbf{v}_{ik}, \boldsymbol{\varepsilon}_{ik}, w_k, a_k, p_{ik}\}$ to record the *k*-th query to H_i and chooses $p_{ik}{}^*$ randomly. If the keyword w_k queried by A is not in the list L_i, S obtains $\boldsymbol{\varepsilon}_{ik}$ which satisfies $\left(\mathbf{M}_0 \middle| \sum_{j \in a_k} \left(\mathbf{B} + \mathbf{M}_j\right)\right) \boldsymbol{\varepsilon}_{ik} = \mathbf{v}_i$, appends $\{\mathbf{v}_{ik} = \mathbf{v}_i, \boldsymbol{\varepsilon}_{ik}, w_k, a_k, p_{ik}\}$ to the list L_i, and sends $\mathbf{v}_{ik}$ to attacker A. Otherwise, S sends $H_i(w_k)$ to attacker A.

Trapdoor part: Firstly, A sends attribute $a_k \in \{0,1\}^l$ and keyword w_k to S for trapdoor query. If $p_{ik} \neq p_{ik}{}^*$, for $1 \leq j \leq d$, S generates $\boldsymbol{\varepsilon}_j \in \mathbb{Z}_q^{2m}$ to obtain the trapdoor $Trap_{w_k} = (\boldsymbol{\varepsilon}_{1k}, \boldsymbol{\varepsilon}_{2k}, ..., \boldsymbol{\varepsilon}_{dk})$. Then, S sends $Trap_{w_k}$ to attacker A. Otherwise, S will halt this process.

Challenge part: A sends keywords w_0 and w_1 without trapdoor query and attribute $a^* \in \{0,1\}^l$ to S, such that $a \not\subset a^*$. If $p_{ik} \neq p_{ik}{}^*$ and $h(a^*) = 0$, S chooses $b = 0$ or $b = 1$ and performs H_i query on the keyword w_b to get corresponding ciphertext $CT^* = (u^*, \eta_1, \eta_2, ..., \eta_m, \eta^*, \boldsymbol{\mu}_1^*, \boldsymbol{\mu}_2^*, ..., \boldsymbol{\mu}_l^*)$. Finally, S sends CT^* to attacker A. Otherwise, S will halt this process.

Guess part: A gives a guess for $b^* \in \{0,1\}$.

```
if $b^* \neq b$
    S obtained $\mathbf{v}_i \leftarrow \mathbb{Z}_q^n$ and $\eta_i \leftarrow \mathbb{Z}_q$ randomly.
else
    $y_i \leftarrow \chi$, $\mathbf{v}_i \leftarrow \mathbb{Z}_q^n$ and $\mathbf{v}_i^T \mathbf{c} + y_i \leftarrow \mathbb{Z}_q$ is sampled by simulator S
end if
```

In summary, cracking IND-CKA of AAQ-PEKS is equal to distinguish $(\mathbf{v}_i, \eta_i)$ from $(\mathbf{v}_i, \mathbf{v}_i^T \mathbf{c} + y_i)$, which is reduced to the LWE hardness assumption.

Corollary 1. In our scheme, the EMR records are encrypted through *Encrypt* algorithm, satisfying the privacy-preserving.

Proof of Corollary 1. The secret key sk_a will be assigned by TA, which is completely trustworthy. Without knowledge of them, our scheme achieves privacy-preserving.

Corollary 2. AAQ-PEKS supports fine-grained access control to EMRs among multiple parties in searching process.

Proof of Corollary 2. Access control authenticates the user's identity and grants the user access to sensitive EMRs within the scope of the license. In order to realize access control in our scheme, we introduce attribute $a \in \{0,1\}^l$ for each multiple party and set an attribute limitation $a' \in \{0,1\}^l$ in *Encrypt*, *Trapdoor* and *Search* algorithm. The data owner can execute these algorithms and access medical data only if the decimal of attribute a is less than or equal to the decimal of attribute limitation a', which is denoted as $a \subset a'$. Access control policy strengthens the trusted control of medical data by authoritative organizations and ensures data security and privacy in AAQ-PEKS.

VII. Performance Evaluation

In this sector, we will compare our proposed scheme with other top-tier existing searchable encryption schemes [6, 10, 12, 20, 36-46] in terms of theoretical evaluation and experimental evaluations. More specifically, we will analyze comprehensive performance evaluation through communication efficiency, computation efficiency, security primitives, operational times, encryption times, and search times, respectively.

TABLE III

Nomenclature in Experiments

Acronyms	Descriptions
N	The number of attributes
s	The size of the access control structure
n	The number of the keywords
n_t	The number of attributes associated with a trapdoor
κ	The length of the string $\{0,1\}$
$\lvert T\rvert$	The length of set T
$\lvert T'\rvert$	The length of set T'
T_H	The evaluation of Hash-to-point
T_{BP}	The evaluation of Bilinear Pairing
T_{ME}	The evaluation of Modular Exponentiation
T_{MT}	The evaluation of Matrix Multiplication
T_{SP}	The evaluation of SamplePre algorithm

TABLE IV

Communication efficiency Comparison

Schemes	System parameter	Secret key	Trapdoor	Ciphertext
Boneh et al. [6]	$\lvert\mathbb{G}_1\rvert + \lvert\mathbb{G}_2\rvert$	$\lvert\mathbb{Z}_p^*\rvert$	$\lvert\mathbb{G}_1\rvert$	$\lvert\mathbb{G}_1\rvert + \{0,1\}^\kappa$
Sultan et al. [36]	$\lvert\mathbb{G}_1\rvert + \lvert\mathbb{G}_2\rvert$	$(2N+6)\lvert\mathbb{Z}_p^*\rvert + \lvert\mathbb{G}_2\rvert$	$(n_t+2)(\lvert\mathbb{G}_1\rvert + \lvert\mathbb{G}_2\rvert)$	$n_t\lvert\mathbb{G}_1\rvert + 2\lvert\mathbb{G}_2\rvert + (n_t+1)\lvert\mathbb{Z}_p^*\rvert$
Chen et al. [10]	$\lvert\mathbb{G}_1\rvert$	$4\lvert\mathbb{Z}_p^*\rvert$	$2\lvert\mathbb{G}_1\rvert$	$3\lvert\mathbb{G}_1\rvert$

Zeng et al. [37]	$3\|\mathbb{G}_1\|+3\|\mathbb{Z}_p\|$	N/A	$(3+2\|T\|)\|\mathbb{G}_1\|$	$(4+\|T\|)\|\mathbb{G}_1\|$
Zhang et al. [38]	$8\|\mathbb{Z}_p\|$	$2\|\mathbb{Z}_p\|$	$m\|\mathbb{Z}_p\|$	$(11m+1)\|\mathbb{Z}_p\|$
Gu et al. [39]	$\|\mathbb{G}_1\|+\|\mathbb{G}_2\|+\|\mathbb{Z}^*_p\|+\|\mathbb{Z}_p\|$	$3\|\mathbb{G}_1\|$	$(N+3)\|\mathbb{G}_1\|+N\|\mathbb{Z}_p\|$	$(s+5)\|\mathbb{G}_1\|+(s+1)\|\mathbb{G}_T\|$
Qu et al. [40]	$\|\mathbb{G}_1\|+\|\mathbb{G}_2\|$	$2\|\mathbb{G}_2\|$	$\|\mathbb{G}_2\|$	$2\|\mathbb{G}_1\|+\|\mathbb{G}_2\|+\{0,1\}^\kappa$
Ling et al. [41]	$3\|\mathbb{G}_1\|$	$3\|\mathbb{Z}^*_p\|$	$\|\mathbb{Z}^*_p\|$	$3\|\mathbb{G}_1\|+2\{0,1\}^\kappa$
Ma et al. [42]	$6\|\mathbb{G}_1\|$	$6\|\mathbb{Z}_p\|$	N/A	$6\|\mathbb{G}_1\|$
Lee et al. [43]	$3\|\mathbb{G}_1\|+2\|\mathbb{G}_2\|$	$\|\mathbb{G}_1\|$	$3\|\mathbb{G}_1\|$	$(2n+15)\|\mathbb{G}_1\|+\|\mathbb{Z}_p\|$
Elhabob et al. [44]	$2\|\mathbb{G}_1\|$	$2\|\mathbb{Z}^*_p\|$	$2\|\mathbb{G}_1\|$	$6\|\mathbb{G}_1\|+2\{0,1\}^\kappa$
Lee et al. [45]	$2\|\mathbb{G}_1\|$	$2\|\mathbb{Z}_p\|$	$\|\mathbb{Z}_p\|$	$2\|\mathbb{G}_1\|+\{0,1\}^\kappa$
Qin et al. [12]	$2\|\mathbb{G}_1\|$	$2\|\mathbb{Z}_p\|$	$\|\mathbb{G}_1\|$	$\|\mathbb{G}_1\|+\{0,1\}^\kappa$
Our scheme	$2\|\mathbb{G}_1\|$	$3\|\mathbb{Z}_p\|$	$(d+l)\|\mathbb{Z}_p\|$	$(2+d)\|\mathbb{Z}_p\|$

TABLE V
COMPUTATION EFFICIENCY COMPARISON

Schemes	Encrypt	Decrypt	Trapdoor	Search
Boneh et al. [6]	$T_{BP}+2T_{ME}+2T_H$	N/A	T_H+T_{ME}	$T_{BP}+T_H$
Sultan et al. [36]	$(n_t+2n+3)T_{ME}+(n_t+1)T_{BP}$	N/A	$(2n_t+7)T_{ME}+2T_{BP}$	$(n_t+5)T_{ME}+6T_{BP}$
Chen et al. [10]	$5T_{ME}$	N/A	$5T_{ME}$	$4T_{ME}+3T_{BP}$
Zeng et al. [37]	$(6+\|T\|)T_{ME}+\|T\|T_H$	N/A	$(5+4\|T'\|)T_{ME}+\|T'\|T_H$	$5T_{BP}+3T_{ME}$
Gu et al. [39]	$(N+4)T_{ME}$	$4T_{BP}+3sT_{ME}$	$(N+3)T_{ME}$	$3T_{BP}$
Qu et al. [12]	$5T_{ME}+2T_{BP}$	$2T_{ME}+2T_{BP}$	T_{ME}	$4T_{BP}$
Ling et al. [41]	$5T_{ME}$	$2T_{ME}$	T_{ME}	$2T_{ME}+2T_{BP}$
Ma et al. [42]	$7T_{ME}$	$6T_{ME}$	N/A	$4T_{ME}$
Lee et al. [43]	$15T_{ME}+T_{BP}$	$11T_{ME}+9T_{BP}$	T_{ME}	$6T_{ME}+6T_{BP}$
Elhabob et al. [44]	$7T_{ME}+4T_{BP}$	$6T_{ME}+6T_{BP}$	$2T_{ME}$	$4T_{BP}$
Qin et al. [12]	$3T_{ME}+T_{BP}$	N/A	$2T_{ME}$	$T_{ME}+T_{BP}$
Wang et al. [20]	$3T_{BP}+9T_{ME}$	$T_{BP}+3T_{ME}$	$5T_{ME}$	$T_{ME}+2T_{BP}$
Our scheme	$(d+l+2)T_{MT}$	$2T_{MT}$	dT_{SP}	$2dT_{MT}$

A. Theoretical evaluation

There are several acronyms utilized in our evaluation analysis, with concrete descriptions elaborated on Table 3 for perusal. Here, we take some significant symbols, such as $|\mathbb{Z}_p|$, $|\mathbb{Z}^*_p|$, $|\mathbb{G}_1|$ and $|\mathbb{G}_2|$ represent the length of the element in $\mathbb{Z}_p$, $\mathbb{Z}^*_p$, $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. Furthermore, we take note that T_H, T_{BP}, T_{ME}, T_{MT} and T_{SP} represent the evaluation time of Hash-to-point, Bilinear Pairing and Modular Exponentiation, Matrix Multiplication, and SamplePre algorithm, respectively.

In Table 4, we introduce the communication efficiency of our scheme compared with [6, 10, 12, 36-45]. To begin with, the security parameter size of our scheme is $2|\mathbb{G}_1|$, which has a significant advantage over [37, 39, 41-43]. The size of the secret key, trapdoor, and ciphertext is $3|\mathbb{Z}_p|$, $(d+l)|\mathbb{Z}_p|$ and $(2+d)|\mathbb{Z}_p|$, respectively. Although the size of these parameters is higher than other schemes, such as [6, 10, 12, 40, 41, 44, 45], our scheme achieves access control in the whole procedure to improve supreme security and privacy of sensitive EMR, which is not available in other schemes.

Table 5 describes the computation efficiency of our four algorithms, including Encrypt, Decrypt, Trapdoor, and Search algorithms. In order to encrypt the keyword and obtain the corresponding ciphertext, our scheme needs to spend $(d+l+2)T_{MT} \approx 17T_{MT}$, which has more advantages than other schemes. Owing to our scheme achieving the access control mechanism in terms of encrypting keywords, it consumes more computation efficiency than other schemes. As for the Trapdoor algorithm, we have a relatively large computation efficiency due to the introduction of the SamplePre algorithm to generate a short vector. However, the execution frequency of the Trapdoor algorithm is prominently less than other algorithms, consequently, it did not have a substantial influence on the efficiency of the whole procedure. With regard to the keyword searching process, our scheme achieves the larger computationally intensive without involving Hash-to-point and Bilinear Pairing operation. Last but not least, our scheme is outstanding in decrypting phase than [12, 20, 39-44], and this advantage will be significantly improved with the increase of keywords.

In Table 6, we conduct a comprehensive comparison with [6, 10, 12, 15, 36-38, 40-46] in terms of the security model, security assumption, security primitives, and infrastructure. In our scheme, we design a searchable encryption scheme using an Attribute-Based Cryptosystem (ABC) and adapt lattice-based cryptography, which makes our scheme resist to IND-CPA, and IND-CKA, and quantum computing attack. Moreover, we reduce the difficulty of cracking our scheme to the LWE assumption and utilize a Random Oracle Model (ROM) to prove the security of our scheme.

B. Comprehensive experimental evaluation

The experiments were based on the C++, with the pairing-friendly elliptic-curve and MATLAB language to finish the simulation performance on Windows 11 with Intel(R) Core(TM) i9-13900H CPU, 128 GB RAM. From Table 7, it is clear to understand the operational time of each evaluation and their relationship are rough $T_{SP} \gg T_H \approx T_{BP} > T_{ME} > T_{MT}$. Nevertheless, our scheme adopts the SamplePre algorithm, enhancing the running time of the Trapdoor procedure, but we keep the utmost security level with anti-quantum computing.

Combining Fig. 2 with Fig. 3 elaborates the computation costs of encryption and search algorithms concerning the number of keywords compared with several schemes. The schemes we compare can divide into two categories, based on discrete logarithm [4, 6, 20, 48] and lattice [27, 47]. Initially, the time costs of both algorithms are linear with the number of keywords. Furthermore, from both pictures, we can see that our proposed scheme is roughly lesser than other existed schemes since we avoid the computation-intensive operations by utilizing the Matrix operations for lattice assumptions, instead of using scalar point multiplication operations. Moreover, with the increment in the number of keywords, our scheme is more practical and scalable compared with others.

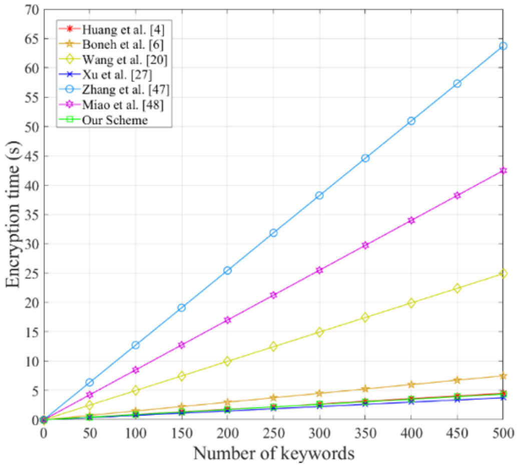

Fig. 2. Computation cost of the Encryption phase.

TABLE VI
SECURITY PRIMITIVES COMPARISON

Schemes Properties	Boneh et al. [6]	Ma et al. [42]	Ling et al. [41]	Qu et al. [40]	Lee et al. [43]	Lee et al. [45]	Lee et al. [46]	Elhabob et al. [44]	Qin et al. [12]	Sultan et al. [36]	Zeng et al. [37]	Zhang et al. [38]	Our scheme
Security model	ROM	Standard	ROM	ROM	Standard	ROM	ROM	ROM	ROM	Standard	ROM	ROM	ROM
Security assumption	BDH	DDH	CDH	BDH	CDH DBDH	Hash	CDH	BDH	CDH	BDH DDH	BDH Hash	LWE ISIS	LWE
Security primitives	Hash function IND-CKA	Hash function S-PRIV1-CCA	OW-CCA IND-CCA	OW-CCA IND-CCA	OW-CCA2 IND-CCA2	OW-CCA2 IND-CCA2	IND-CCA2 Hash function	OW-CCA OW-ID-CCA	MCI S-KGA	O-CKA IND-CKA	IND-CKA Hash function	IND-KGA Quantum computing	IND-CPA IND-CKA Quantum computing
Infrastructure	IBC	PKI	PKI	CL-PKC	PKI	PKI	IBC	IBC CL-PKC	PKI	ABC	ABC	PKI IBC	ABC

For example, the computation cost of encryption in [47], [48], and [20] compared with our scheme is nearly thirteen times, eight times, and five times, respectively when the number of keywords is 500. Not only this, but the same as for the computation cost of the search phrase, as Huang et al. [4] and Boneh et al. [6] are approximately two and a half of our schemes. Lastly, the search time of our scheme is a little bit higher than [48], nevertheless, our scheme is significantly better than the remaining protocols.

VIII. CONCLUSION

With the emergence of quantum computers, they have threatened cryptography based on discrete logarithm. In addition, the access control of EMR needs to be guaranteed. Therefore, we have introduced an attribute-based anti-quantum public-key encryption scheme with keyword search for the E-health scenarios in this paper. Our proposed scheme can resist quantum computing attack. Moreover, the proposed AAQ-PEKS achieves fine-grained access control for patients. In addition, we illustrate the computational security proofs and corollaries, specifying our scheme is secure under the LWE assumptions. Lastly, compared with other schemes, our protocol realized higher computation and communication efficiency, which is practical for the E-health applications.

TABLE VII

OPERATIONAL TIMES

Operation	Running time (s)
T_H	6.58×10^{-3}
T_{BP}	6.47×10^{-3}
T_{ME}	3.10×10^{-5}
T_{MT}	1.62×10^{-6}
T_{SP}	4.91

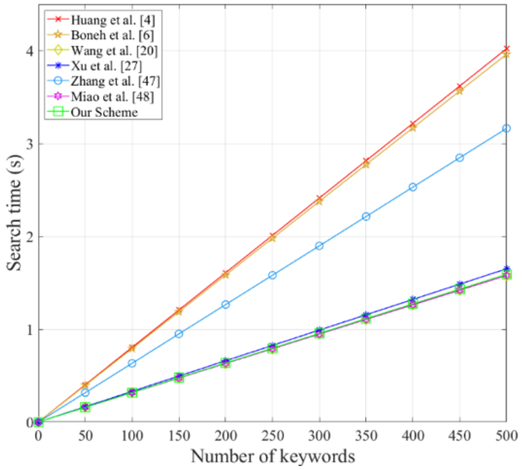

Fig. 3. Computation cost of the Search phase.

REFERENCES

[1] H. Ma, R. Zhang, G. Yang, Z. Song, K. He and Y. Xiao, "Efficient Fine-Grained Data Sharing Mechanism for Electronic Medical Record Systems with Mobile Devices," *IEEE Trans. Dependable Secur. Comput.*, vol. 17, no. 5, pp. 1026-1038, 2020.

[2] J. Liu, T. Liang, R. Sun, X. Du and M. Guizani, "A Privacy-Preserving Medical Data Sharing Scheme Based on Consortium Blockchain," in *Proc. of GLOBECOM*, Taipei, Taiwan, 2020, pp. 1-6.

[3] J. Wei, X. Chen, X. Huang, X. Hu and W. Susilo, "RS-HABE: Revocable-Storage and Hierarchical Attribute-Based Access Scheme for Secure Sharing of e-Health Records in Public Cloud," *IEEE Trans. Dependable Secur. Comput.*, vol. 18, no. 5, pp. 2301-2315, 2021.

[4] Q. Huang, G. Yan and Y. Yang, "Privacy-Preserving Traceable Attribute-Based Keyword Search in Multi-Authority Medical Cloud," *IEEE Trans. Cloud Comput.*, early access, 2022, doi: 10.1109/TCC.2021.3109282.

[5] X. Chen, S. Xu, T. Qin, Y. Cui, S. Gao and W. Kong, "AQ-ABS: Anti-Quantum Attribute-based Signature for EMRs Sharing with Blockchain," in *Proc. of WCNC*, Austin, TX, USA, 2022, pp. 1176-1181.

[6] D. Boneh, G. Di Crescenzo, R. Ostrovsky and G. Persiano, "Public Key Encryption with Keyword Search," in *Advances in Cryptology – EUROCRYPT*, Interlaken, Switzerland, 2004, pp. 506-522.

[7] M. Zhandry, "Redeeming Reset Indifferentiability and Applications to Post-quantum Security," in 27th *Advances in Cryptology – ASIACRYPT*, Singapore, 2021, pp. 518-548.

[8] P. Jiang, Y. Mu, F. Guo and Q. Wen, "Public Key Encryption with Authorized Keyword Search," in *Proc. of ACISP*, Melbourne, VIC, Australia, 2016, pp. 170-186.

[9] R. Chen, Y. Mu, G. Yang, F. Guo and X. Wang, "Dual-Server Public-Key Encryption With Keyword Search for Secure Cloud Storage," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 4, pp. 789-798, 2016.

[10] B. Chen, L. Wu, S. Zeadally and D. He, "Dual-Server Public-Key Authenticated Encryption with Keyword Search," *IEEE Trans. Cloud Comput.*, vol. 10, no. 1, pp. 322-333, 2022.

[11] X. Yuan, X. Wang, C. Wang, C. Yu and S. Nutanong, "Privacy-Preserving Similarity Joins Over Encrypted Data," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 11, pp. 2763-2775, 2017.

[12] B. Qin, Y. Chen, Q. Huang, X. Liu and D. Zheng, "Public-key authenticated encryption with keyword search revisited: Security model and constructions," *Inform. Sci.*, vol. 516, pp. 515-528, 2020.

[13] X. Song, C. Dong, D. Yuan, Q. Xu and M. Zhao, "Forward Private Searchable Symmetric Encryption with Optimized I/O Efficiency," *IEEE Trans. Dependable Secur. Comput.*, vol. 17, no. 5, pp. 912-927, 2020.

[14] J. Cui, J. Lu, H. Zhong, Q. Zhang, C. Gu and L. Liu, "Parallel Key-Insulated Multiuser Searchable Encryption for Industrial Internet of Things," *IEEE Trans. Ind. Informatics* vol. 18, no. 7, pp. 4875-4883, 2022.

[15] T. Hoang, A. A. Yavuz and J. Guajardo, "A Secure Searchable Encryption Framework for Privacy-Critical Cloud Storage Services," *IEEE Trans. Serv. Comput.*, vol. 14, no. 6, pp. 1675-1689, 2021.

[16] G. Xu, Y. Cao, S. Xu, X. Liu, X-B. Chen, Y. Yu, et al. "A Searchable Encryption Scheme Based on Lattice for Log Systems in Blockchain," *CMC-Comput. Mater. Con.*, vol. 72, no. 3, pp. 5429-5441.

[17] X. Chen, S. Xu, Y. He, Y. Cui, J. He and S. Gao, "LFS-AS: Lightweight Forward Secure Aggregate Signature for e-Health Scenarios," in *Proc. of ICC*, Seoul, South Korea, 2022, early access.

[18] R. Zhang, R. Xue and L. Liu, "Searchable Encryption for Healthcare Clouds: A Survey," *IEEE Trans. Serv. Comput*, vol. 11, no. 6, pp. 978-996, 2018.

[19] C. Xu, N. Wang, L. Zhu, K. Sharif and C. Zhang, "Achieving Searchable and Privacy-Preserving Data Sharing for Cloud-Assisted E-Healthcare System," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8345-8356, 2019.

[20] H. Wang, J. Ning, X. Huang, G. Wei, G. S. Poh and X. Liu, "Secure Fine-Grained Encrypted Keyword Search for E-Healthcare Cloud," *IEEE Trans. Dependable Secur. Comput.*, vol. 18, no. 3, pp. 1307-1319, 2021.

[21] Y. Bao, W. Qiu and X. Cheng, "Secure and Lightweight Fine-Grained Searchable Data Sharing for IoT-Oriented and Cloud-Assisted Smart Healthcare System," *IEEE Internet Things J.*, vol. 9, no. 4, pp. 2513-2526, 2022.

[22] S. Xu, X. Chen, C. Wang, Y. He, K. Xiao and Y. Cao, "A Lattice-Based Ring Signature Scheme to Secure Automated Valet Parking," in *Proc. of WASA 2021*, vol. 12938, Nanjing, China, 2021, pp. 70-83.

[23] M. Ajtai and C. Dwork, "A Public-Key Cryptosystem with Worst-Case/Average-Case Equivalence," in *Proc. of STOC*, El Paso, Texas, USA, 1997, pp. 284-293.

[24] C. Hou, F. Liu, H. Bai and L. Ren, "Public-key searchable encryption from lattices," *Int. J. High Perform. Syst. Archit.*, vol. 5, no. 1, pp. 25-32, 2014.

[25] P. Wang, T. Xiang, X. Li and H. Xiang, "Public key encryption with conjunctive keyword search on lattice," *J. Inf. Secur. Appl.vol.*, vol. 51, pp. 102433, 2020.

[26] R. Behnia, M. O. Ozmen and A. A. Yavuz, "Lattice-Based Public Key Searchable Encryption from Experimental Perspectives," *IEEE Trans. Dependable Secur. Comput.*, vol. 17, no. 6, pp. 1269-1282, 2020.

[27] G. Xu, S. Xu, Y. Cao, F. Yun, Y. Cui, Y. Yu, et al., "PPSEB: A Postquantum Public-Key Searchable Encryption Scheme on Blockchain for E-Healthcare Scenarios", *Secur. Commun. Netw.*, vol. 2022, pp. 3368819, 2022.

[28] A. Sahai and B. Waters, "Fuzzy Identity-Based Encryption," in *Advances in Cryptology – EUROCRYPT*, Aarhus, Denmark, vol. 3494, 2005, pp. 457-473,.

[29] C. Yin, H. Wang, L. Zhou and L. Fang, "Ciphertext-Policy Attribute-Based Encryption with Multi-keyword Search over Medical Cloud Data," in *Proc. of TrustCom*, 2020, pp. 277-284.

[30] J. Li, Y. Li, R. Sun, Q. Pei, N. Zhang, M. Dong, et al., "EMK-ABSE: Efficient Multi-Keyword Attribute-Based Searchable Encryption Scheme Through Cloud-Edge Coordination," *IEEE Internet Things J.*, early access, 2022, doi: 10.1109/JIOT.2022.3163340.

[31] S. Zhao, R. Jiang and B. Bhargava, "RL-ABE: A Revocable Lattice Attribute Based Encryption Scheme Based on R-LWE Problem in Cloud Storage," *IEEE Trans. Serv. Comput.*, vol. 15, no. 2, pp. 1026-1035, 2022.

[32] M. Ajtai, "Generating hard instances of the short basis problem," in *Proc. of ICALP*, Prague, Czech, 1999, pp. 1-9.

[33] C. Gentry, C. Peikert and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *Proc. of STOC*, Victoria, British Columbia, vol. 14, 2008, pp. 197-206.

[34] D. Cash, D. Hofheinz, E. Kiltz and C. Peikert, "Bonsai Trees, or How to Delegate a Lattice Basis," *J. Cryptol.*, vol. 25, pp. 601-639, 2012.

[35] S. Agrawal, D. Boneh and X. Boyen, "Efficient lattice (H)IBE in the standard model," in *Advances in Cryptology – EUROCRYPT*, Riviera, France, 2010, pp. 553-572.

[36]N. H. Sultan, N. Kaaniche, M. Laurent and F. A. Barbhuiya, "Authorized Keyword Search over Outsourced Encrypted Data in Cloud Environment," *IEEE Trans. Cloud Comput.*, vol. 10, no. 1, pp. 216-233, 2022.

[37]M. Zeng, H. Qian, J. Chen and K. Zhang, "Forward Secure Public Key Encryption with Keyword Search for Outsourced Cloud Storage," *IEEE Trans. Cloud Comput.*, vol. 10, no. 1, pp. 426-438, 2022.

[38]X. Zhang, Y. Tang, H. Wang, C. Xu, Y. Miao and H. Cheng, "Lattice-based proxy-oriented identity-based encryption with keyword search for cloud storage," *Inf. Sci.*, vol. 494, pp. 183-207, 2019.

[39]K. Gu, W. Zhang, X. Li and W. Jia, "Self-Verifiable Attribute-Based Keyword Search Scheme for Distributed Data Storage in Fog Computing With Fast Decryption," *IEEE Trans. Netw. Serv. Manag.*, vol. 19, no. 1, pp. 271-288, 2022.

[40]H. Qu, Z. Yan, X. Lin, Q. Zhang and L. Sun, "Certificateless public key encryption with equality test," *Inf. Sci.*, vol. 462, pp. 76-92, 2018.

[41]Y. Ling, S. Ma, Q. Huang, X. Li and Y. Ling, "Group public key encryption with equality test against offline message recovery attack," *Inf. Sci.*, vol. 510, pp. 16-32, 2020.

[42]S. Ma, Y. Mu and W. Susilo, "A Generic Scheme of Plaintext-Checkable Database Encryption," *Inf. Sci.*, vol. 429, pp. 88-101, 2018.

[43]H. T. Lee, S. Ling, J. H. S, H. Wang and T.-Y. Youn, "Public key encryption with equality test in the standard model," *Inf. Sci.*, vol. 516, pp. 89-108, 2020.

[44]R. Elhabob, Y. Zhao, I. Sellla and H. Xiong, "Public key encryption with equality test for heterogeneous systems in cloud computing," *KSII Trans. Internet Inf. Syst.*, vol. 13, no. 9, pp. 4742-4770, 2019.

[45]H. T. Lee, S, Ling, J. H. Seoet and H. Wang, "Public key encryption with equality test from generic assumptions in the random oracle model," *Inf. Sci.*, vol. 500, pp. 15-33, 2019.

[46]H. T. Lee, S. Ling, J. H. Seo, and H. Wang, "Semi-generic construction of public key encryption and identity-based encryption with equality test," *Inf. Sci.*, vol. 373, pp. 419-440, 2016.

[47]X. Zhang, C. Xu, H. Wang, Y. Zhang and S. Wang, "FS-PEKS: Lattice-Based Forward Secure Public-Key Encryption with Keyword Search for Cloud-Assisted Industrial Internet of Things," *IEEE Trans. Dependable Secur. Comput.*, vol. 18, no. 3, pp. 1019-1032, 2021.

[48]Y. Miao, X. Liu, K.-K.R. Choo, R.H. Deng, J. Li, H. Li, et al., "Privacy-Preserving Attribute-Based Keyword Search in Shared Multi-owner Setting," *IEEE Trans. Dependable Secur. Comput.*, vol. 18, no. 3, pp. 1080-1094, 2021.

[49]X. Chen, S. Xu, T. Qin, Y. Cui, S. Gao and W. Kong, "AQ–ABS: Anti-Quantum Attribute-based Signature for EMRs Sharing with Blockchain," in *Proc. of WCNC*, Austin, TX, USA (WCNC), 2022, pp. 1176-1181.

[50]Y. Cao, S. Xu, X. Chen, Y. He, S. Jiang, "A forward-secure and efficient authentication protocol through lattice-based group signature in VANETs scenarios," *Comput. Networks*, vol. 214, no. 109149, 2022.

[51]G. Xu, Y. Cao, S. Xu, X. Liu, X.B. Chen, Y. Yu, et al., "A Searchable Encryption Scheme Based on Lattice for Log Systems in Blockchain. *CMC- Comput. Mater. Contin.*, vol. 72, no. 3, pp.5429-5441, 2022.