Hindawi
Security and Communication Networks
Volume 2022, Article ID 8453228, 13 pages
https://doi.org/10.1155/2022/8453228

*Research Article*

# IsoqurPEKS: An Isogeny-Based Quantum-Resistant Public-Key Encryption Scheme with Keyword Search

**Qing Fan**,[1,2] **Min Luo**,[3] **Cong Peng**,[3] **Lianhai Wang**,[4] **and Debiao He**[3,5]

[1]*School of Mathematics and Statistics, Wuhan University, Wuhan 430072, China*
[2]*School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing 100081, China*
[3]*School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China*
[4]*Shandong Provincial Key Laboratory of Computer Networks, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250014, China*
[5]*Shanghai Key Laboratory of Privacy-Preserving Computation, Matrix Elements Technologies, Shanghai, China*

Correspondence should be addressed to Lianhai Wang; wanglh@keylab.net and Debiao He; hedebiao@163.com

Received 23 May 2022; Accepted 11 October 2022; Published 1 November 2022

Academic Editor: Jie Cui



Since the convenience and advancement of cloud applications, many users (e.g., companies or individuals) adopt remote cloud services to reduce the local storage overload and computing consumption. However, before transferring them to the cloud server, users always encrypt outsourced data for the privacy of important data, which deprives flexible usage of these data. Public key encryption with keyword search (PEKS) undoubtedly offers a precise resolution to this issue. Unfortunately, most PEKS schemes cannot fight against quantum computing attackers, which is increasingly a research hotspot. To achieve postquantum security and privacy-preserving search function, we propose a quantum-resistant PEKS scheme named IsoqurPEKS. Our proposed instantiation satisfies basic semantic security indistinguishable against chosen keyword attack (IND-CKA), and IsoqurPEKS is proved to be secure under the security model. Furthermore, we compare IsoqurPEKS with the other eight current PEKS schemes with respect to security properties, communication, and computation costs. The comparison results indicate that the proposed scheme has the best security and performance among the nine PEKS schemes.

## 1. Introduction

Remote cloud services have advantages of data accessibility, data scalability, data sharing, and consistent backups of enormous data [1]. Cloud applications, such as cloud storage, cloud computing, and cloud retrieval, are becoming more prevalent for data users and enterprises. Data uploaders usually outsource their data to the cloud server, saving their local storage cost and offering easy data access. However, these remote servers are not always trusted since some malicious insiders may have full access to plaintext data. Once critical and sensitive data are exposed to hackers, significant threats to users' property and life safety may happen. Therefore, before uploading data to the cloud server, data providers encrypt these data using encryption algorithms to provide privacy protection while depriving all search capacities of data users.

Many cloud services, such as Baidu Cloud, Google Cloud, Windows Azure, and Amazon simple storage service [2] promote the development of cloud storage and searching technologies. When performing data retrieval, a straightforward approach for cloud servers is to obtain a decryption key and search required items in plaintext. However, this method breaks the initial intention of outsourced data encryption because a corrupted insider (e.g., a compromised cloud storage provider's machine [3]) could access any unauthorized data. Another solution for data users is to download the whole database, decrypt all data locally, and retrieve interesting documents, which require a lot of memory space and computation capacity. This method does not play the role of a cloud server; instead, it puts forward high requirements for users, which are impractical in most applications [4]. To achieve data confidentiality and search

function at the same time, Song et al. [5] first put forward the conception of searchable encryption.

The searchable encryption mechanism enables data providers to upload encrypted data and multiple searchable keywords ciphertexts, while data users produce trapdoors of intended keywords. Utilizing the trapdoor, a cloud server could execute a search to seek matched keywords and corresponding data ciphertext. According to distinct generation types of encrypted keywords and trapdoors [6], searchable encryption is generally classified into public key encryption with keyword search (PEKS) and symmetric searchable encryption (SSE). Although SSE has efficient retrieval efficiency, which has been extensively researched [7], it still has the same key distribution problem as symmetric encryption. Then, Boneh et al. [8] introduced the first PEKS scheme, whose system architecture is shown in Figure 1. In our scheme, the data provider produces searchable ciphertext using users' public keys, and the user generates a keyword search trapdoor by their private key. Then, the trapdoor is transmitted to the cloud server for searching matched ciphertexts, which are finally returned to the user. Furthermore, Boneh et al. have also formalized the notion of indistinguishability against chosen keyword attacks (IND-CKAs) of PEKS, which ensures the privacy of searchable ciphertext [8].

However, most of the current PEKS schemes are designed based on classical hard assumptions such as discrete logarithm (DL) problem and computational/decisional Diffie–Hellman (CDH/DDH) problem. Shor [9] pointed out that there is a quantum algorithm to crack the DL problem in polynomial time, which inspires scholars to explore quantum-resistant PEKS scheme construction [10]. According to the report on postquantum cryptography [11], families of postquantum primitives are designed by the lattice, multivariate polynomial, code, and isogeny. In comparison, code-based and lattice-based cryptographies suffer from large key sizes. In addition, there are no searchable encryption-compatible structures based on multivariate polynomials and hash-based cryptography as far as we know.

Isogeny-based cryptography overcomes the above problem and has the potential for searchable encryption construction. Isogeny is a rational mapping from one elliptic curve to another, which is distinguished by its degree or kernel [12]. The isogeny problem is to seek a mapping path given two specified isogenous elliptic curves. Studies on isogeny-based cryptography have matured gradually, and the fastest known algorithm to find such an isogeny takes subexponential time [13]. Isogeny-basedencryption [14] gives a specific verification equation (i.e. $e(\phi(P), Q) = e'(P, \widehat{\phi}(Q))$ where $\phi$ is an isogeny between two elliptic curves $E \longrightarrow E'$ and $\widehat{\phi}$: $E' \longrightarrow E$ are mutual dual isogeny, $E \longrightarrow GT$ and $E' \longrightarrow GT$ are two bilinear maps, and $P/Q$ are separately the generators of $E/E'$), and the first PEKS scheme enlightens us to design an isogeny-based quantum-resistant PEKS scheme.

This paper puts forward a new quantum-resistant PEKS scheme using isogeny named IsoqurPEKS. Then, we prove its IND-CKA security under the quantum random oracle (QROM) model and analyze communication cost and computation cost by comparing IsoqurPEKS with other eight PEKS schemes. Analysis results demonstrate that IsoqurPEKS has the least communication and computation overload while maintaining the property of withstanding quantum computer attacks.

*1.1. Organizations of This Paper.* Section 2 introduces related works about isogeny quantum-resistant PEKS schemes. Preliminaries containing the elliptic curve and isogeny knowledge are introduced in Section 3. PEKS definitions, consistency, and security definitions of the quantum-resistant PEKS scheme are given in Section 4. We present the system model of the proposed IsoqurPEKS scheme and the threat model of each entity, and the design goals of this paper are presented in Section 5. Then, we introduce the quantum-resistant IsoqurPEKS scheme in Section 6, and we give the formal security proof of IsoqurPEKS in Section 7. Section 8 shows the property, communication cost, and time consumption comparisons with eight PEKS schemes. Eventually, we summarize this paper in Section 9.

## 2. Related Works

Boneh et al. [8] first put forward the notion of public key encryption with keyword search (PEKS). Following this seminal word, some further works on PEKS schemes [15–17] have been proposed in traditional public-key cryptography settings. Scholars have mainly explored two types of research orientations: diverse functionality search and security studies.

Concerning functionality search, Kim et al. [18] proposed the first privacy-preserving algorithm to test whether an encrypted string includes an encrypted pattern. Meanwhile, they designed a novel wildcard search on encrypted databases, which are used to support compound queries. In terms of a multikeyword search, Wang et al. [19] proposed a secure searchable encryption scheme under the standard model supporting multikeyword retrieval. Liu et al. [20] put forward a multiuser and multikeyword search with the hiding search pattern and access pattern. Zhang et al. [21] proposed a fuzzy multikeyword search in the cloud system using Word2vec technology. Liang et al. [22] utilized advanced $k$-nearest neighbor ($k$-NN) technology to enhance search accuracy and achieve an exact multikeyword fine-grained search. Zarezadeh et al. [23] presented a multikeyword rank search scheme that enhances usability and file retrieval accuracy. Asymmetric encryption schemes supporting Boolean queries in different scenarios such as cloud applications and mobile clouds were also studied [24, 25]. However, the above schemes are built on classical intractable assumptions and cannot resist quantum computing attacks.

Concerning security, scholars generally consider forward privacy and backward security of searchable encryption. Forward privacy ensures that inserting new files will not expose previous search information, and backward security means deleting files will not disclose more information in the following search process. Zhang et al. [26] and Ning et al. [27] have discussed threats brought to searchable encryption by file-injection attacks and passive attacks. Then, Bost et al. [28] used constrained pseudorandom functions and puncturable encryption and put forward

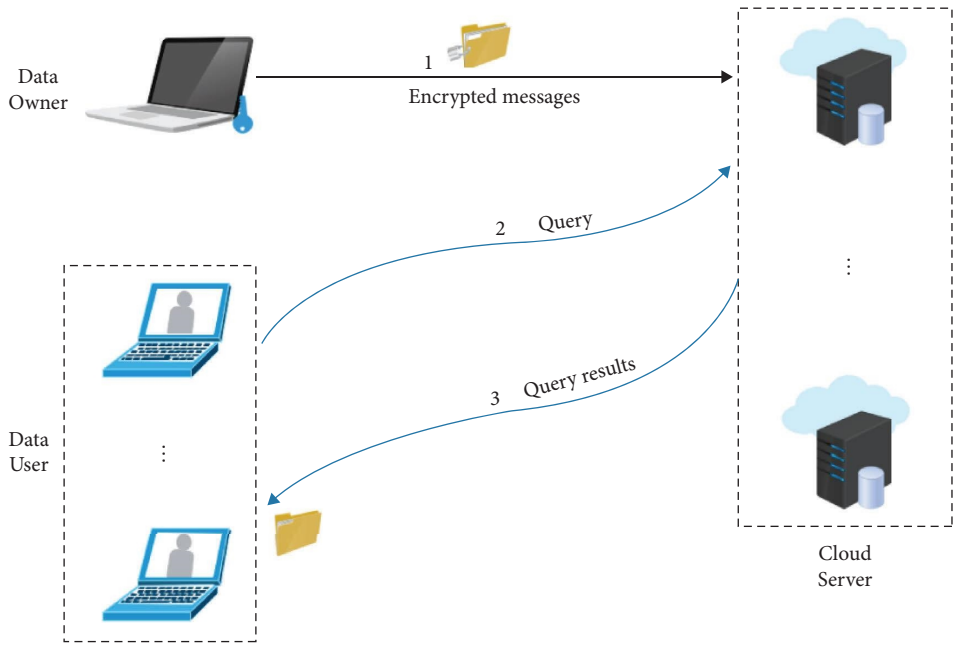



Figure 1: General PEKS architecture.

forward and backward secrecy searchable encryption under the symmetric mechanism. Zeng et al. [29] introduced searchable public key encryption built on attribute-based encryption, which satisfies forward privacy. These schemes still do not take quantum-resistant attacks into account.

Behnia et al. [30] proposed two lattice-based PEKS schemes with well computational efficiency and better security than the current ones. Xu et al. [10] utilized learning with error (LWE) hard problems and also proposed a lattice-based searchable encryption scheme, which satisfies post-quantum security. However, lattice-based PEKS schemes have large-sized keys because they are composed of matrices. Isogeny-based cryptography has small-sized keys which have been studied deeply [12, 31–33].

We put forward a PEKS scheme based on the isogeny hardness assumption to resist quantum-computing attacks. Then, we prove its security under the QROM model. Although there has been one isogeny-based PEKS scheme [34], the proposed IsoqurPEKS scheme has better efficiency. Moreover, we also evaluate this scheme by comparing it with the current eight PEKS schemes to communication cost and computation cost, indicating that our scheme has the best security and performance among these nine PEKS schemes.

## 3. Preliminaries

In this section, we introduce a basic elliptic curve and supersingular isogeny knowledge used in the scheme design. Notations used in this paper are shown in Table 1.

*3.1. Elliptic Curve.* In our scheme, we will take advantage of the following basic knowledge. $F_q$ is a finite field with the order $q$. The equation $y^2 = x^3 + cx + d \bmod q$ defines an elliptic curve $E$ over $F_q$, where $c, d \in F_q$. Points on the elliptic curve $E$: $y^2 = x^3 + cx + d \bmod q$ with the addition of the point at infinity $\mathcal{O} = (0, 0)$ constitute an additive cyclic group $E[N]$ whose order is an integer $N$. The map $e$: $E[N] \times E[N] \longrightarrow G[N]$ is a bilinear map satisfying:

Table 1: Notation description.

| Notations | Descriptions |
|---|---|
| $\lambda$ | Security parameter |
| $N$ | The elliptic curve's order |
| $\mathbb{F}_q$ | A finite field with the order $q$ |
| $E[N]$ | The elliptic curve point group with the order $N$ |
| $P_1, P_2$ | Two base points in $E[N]$ |
| $\#E$ | The cardinality of the elliptic curve point group |
| $j$-Variant | An index for determining elliptic curve isomorphism |
| $\phi$ | Isogeny $\phi$: $E \longrightarrow E'$ with a degree $l$ |
| $\hat{\phi}$ | $\phi$'s dual isogeny $\hat{\phi}$: $E' \longrightarrow E$ |
| $h$ | General hash function $h$: $\{0,1\}^* \longrightarrow \{0,1\}^{\lg N}$ |
| $H'$ | Hash-to-point function $H'$: $\{0,1\}^* \longrightarrow E'[N]$ |
| $G_T, G_T'$ | Two multiplicative groups |
| $e$ | Bilinear map $e$: $E[N] \times E[N] \longrightarrow G_T$ |
| $e'$ | Bilinear map $e'$: $E'[N] \times E'[N] \longrightarrow G_T'$ |
| PK | The public key of the user |
| $SK$ | The private key of the user |
| $\omega$ | Probability of one value appearing in random oracle |

(i) Bilinearity. $e(xP, yP) = e(P, P)^{xy}$ for any $x, y \in Z_q^*$

(ii) Computability. $e(P, Q)$ can be easily calculated in the polynomial time for any $P, Q \in E[N]$

(iii) Nondegeneracy. $e(P, P) \neq 1$

where $G[N]$ is a multiplicative cyclic group with the order $N$ and $P$ is an arbitrary generator of $E[N]$.

*3.2. Isogeny.* Isogeny is defined based on two elliptic curves $E$ and $E'$, a rational and surjective mapping $\phi$: $E \longrightarrow E'$. It keeps the computing law of the point group, i.e., $\phi(P + Q) =$

$\phi(P) + \phi(Q)$ for any points $P, Q$ on $E$. Two elliptic curves $E$ and $E'$ defined over a finite field $F_q$ are isogenous with the necessary and sufficient condition that they have the same cardinality, i.e., $\#E = \#E'$. Since isogeny could be represented by a rational polynomial, a degree, similar to polynomials, could be defined and used to differentiate various isogenies. According to Burdges et al. [14], any isogeny $\phi$: $E \longrightarrow E'$ has one and only one corresponding dual isogeny $\widehat{\phi}$: $E' \longrightarrow E$ which has a specific relationship as follows:

$$e_N(\phi(P), Q) = e'_N(P, \widehat{\phi}(Q)), \tag{1}$$

for any points $P \in E[N], Q \in E'[N]$, where $e$ (resp. $e'$) is any bilinear pairing (e.g., Weil, Tate, and Ate pairing) on $E$ (resp. $E'$).

Next, we consider some preliminary knowledge for difficult problems resisting quantum computers attack. We first give the following proposition.

**Proposition 1.** *Let $E$ be an elliptic curve determined by $y^2 = x^3 + cx + d$, where $c, d$ are from the finite field $F_q$, and then, we give the j-variant definition as follows:*

$$j(E) = 1728\frac{4c^3}{4c^3 + 27d^2}, \tag{2}$$

*and j-variant distinguishes the isomorphism class since the necessary and sufficient condition of two isomorphic curves is that they have the same j-variant.*

The graph structure can embody isogeny-related hard problems. This graph structure is composed of isomorphism classes denoted by nodes and isogenies between curves denoted by edges. The isogeny graphs constructed by different degrees of isogenies are diverse, and the isogeny star is made up of various isogeny graphs while having the same nodes. Literature [31] gives detailed descriptions and visualized depictions of the isogeny star as shown in Figure 2. There are many isogeny paths from one node to another, which may consist of multiple isogenies. When the isogeny star is quite large, finding a path from the initial elliptic curve to the end elliptic curve, respectively, in different isomorphism classes will be rather difficult, which is the isogeny problem.

Childs et al. [13] have pointed out that the most efficient traditional algorithm requires exponential time to seek an isogeny between two isogenous elliptic curves. However, they came up with a quantum algorithm to construct an isogeny between two given elliptic curves with the same cardinality in subexponential time. However, the running time is bounded above by $\exp[(\sqrt{3}/2 + O(1))\sqrt{\ln q \ln\ln q}]$ under the generalized Riemann hypothesis. Most importantly, there exists no faster quantum algorithm than in the study by Childs et al. as far as we know. Assume that $\phi$ is an isogeny mapping from the elliptic curve $E$ to $E'$, we give the following two difficult problems under quantum computers.

*3.2.1. Supersingular Isogeny (SSI) Problem.* Assume that the kernel $\langle [s]P + [t]Q \rangle$ specifies an isogeny $\phi$: $E \longrightarrow E'$, where $s$ and $t$ are chosen randomly from $\mathbb{Z}/l^e\mathbb{Z}$ and are not divisible by $l$. Given the elliptic curve $E'$ and points $\phi(P), \phi(Q)$ on $E'$, it is difficult to find a generator $T$ of $\langle [s]P + [t]Q \rangle$. It should be specified that given a generator $T = [s]P + [t]Q$, it is trivial to resolve for $(s, t)$. In other words, given two elliptic curves $E$ and $E'$ with the identical cardinality, it is hard to calculate an isogeny $\phi$: $E \longrightarrow E'$ utilizing the quantum algorithm in the polynomial time.

*3.2.2. Extensional Computational Isogeny Problem (ECIP).* Given $P$ and $\phi(xP)$ with $x, \phi$ unknown, where the point $P$ is randomly selected on $E[N]$ and $x$ is a random number in $F_q$, it is difficult to calculate $\phi$ and $x$ in the polynomial time for the quantum computer.

## 4. Public Key Encryption with Keyword Search

This section introduces public key encryption with keyword search (PEKS) from three aspects: definitions, consistency, and security.

*4.1. PEKS Definitions.* A PEKS scheme consists of four algorithms, namely, setup, PEKS, trapdoor, and test. In the fist PEKS scheme, it only considers the encryption with single keyword search [8]. In practice, a file usually contains many keywords. Therefore, we use the general extended definition of PEKS, which takes a set of keywords as inputs and keeps consistency. The formal constructions are as follows:

(i) $Setup(1^\lambda)$: the setup algorithm is performed to generate the user's keys. With inputting the security parameter $\lambda$, this probabilistic polynomial time (PPT) algorithm returns a pair of public/private key $(pk_{\text{PEKS}}, sk_{\text{PEKS}})$ for the user.

(ii) PKES $(pk_{\text{PEKS}}, \mathscr{W})$: the PKES algorithm is performed by a data provider. Taking the public key $pk_{\text{PEKS}}$ and a set of keywords $\mathscr{W}$ as inputs, this PPT algorithm returns corresponding keywords' searchable ciphertext $CT_{\mathscr{W}}$.

(iii) Trapdoor $(sk_{\text{PEKS}}, kw')$: the trapdoor algorithm is executed by a search user. With the secret key $sk_{\text{PEKS}}$ and a keyword $kw'$ as inputs, the trapdoor algorithm returns a trapdoor $T_{kw'}$ of $kw'$.

(iv) $Test(pk_{\text{PEKS}}, CT_{\mathscr{W}}, T_{kw'})$: the test algorithm is executed by the cloud server. With the searchable ciphertext $CT_{\mathscr{W}}$ and trapdoor $T_{kw'}$ as inputs, this deterministic algorithm outputs 1 if $kw' \in \mathscr{W}$; otherwise, it outputs 0.

It should be noted that the trapdoor algorithm is either deterministic or probabilistic, which is determined by the specific scheme design and security requirements. We only consider the initial form, i.e., the deterministic trapdoor algorithm in this paper.

*4.2. Consistency.* For a PEKS scheme, the most essential and critical requirement is consistency [16]; that is, the returned results from the cloud server should be what the user wants to acquire. Specifically, when the cloud server gets a trapdoor

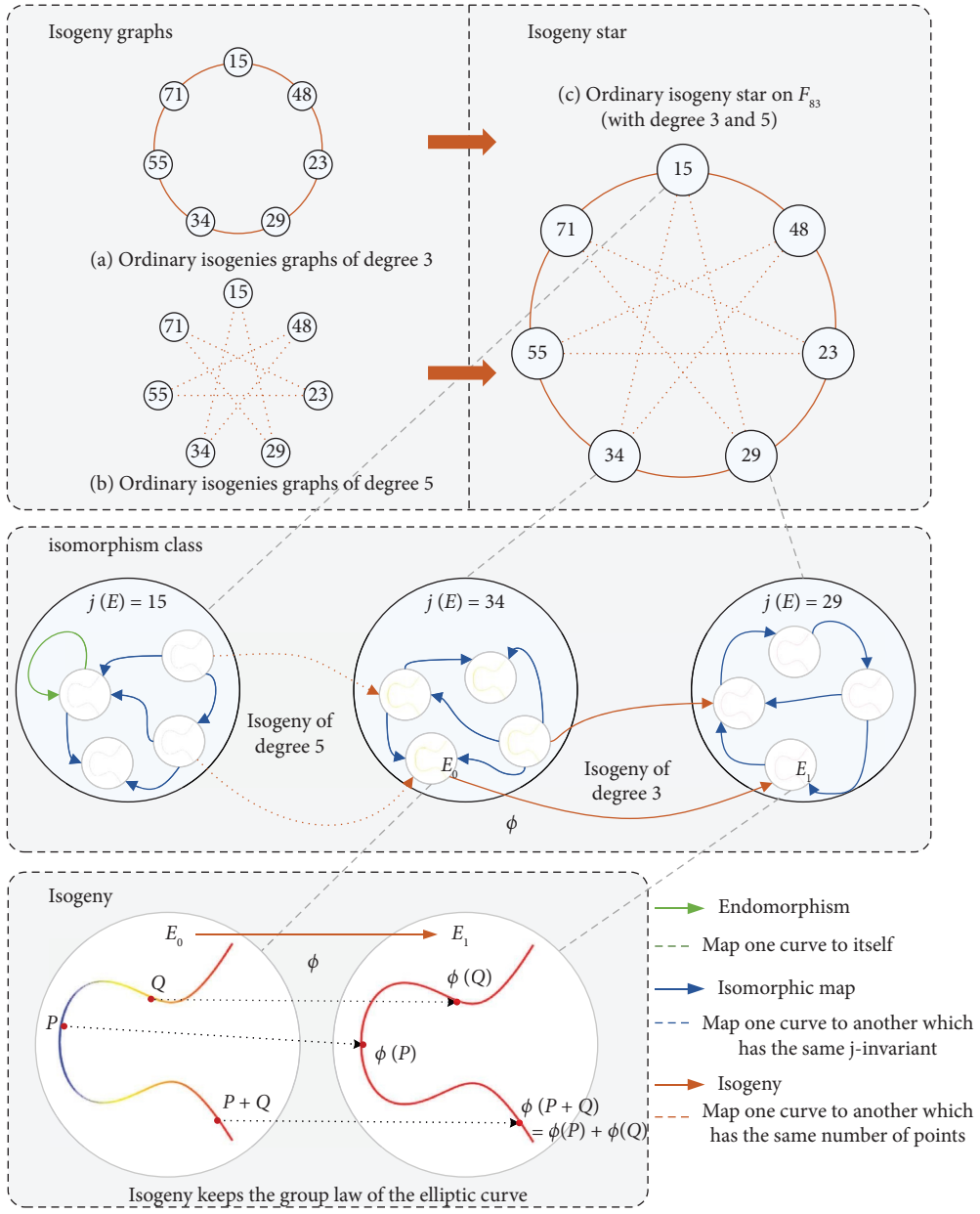



Figure 2: Isogeny graphs and isogeny star [31].

$T_{kw'}$ produced by the trapdoor algorithm and ciphertext $CT_{\mathscr{W}}$ generated by the *PEKS* algorithm, we formulate consistency as follows:

(i) $Test(CT_{\mathscr{W}}, T_{kw'})$ always outputs 1 if $kw' \in \mathscr{W}$

(ii) The probability $\Pr[\text{Test}(CT_{\mathscr{W}}, T_{kw'}) = 1]$ is negligible if $kw' \notin \mathscr{W}$

*4.3. Security Definitions of the Quantum-Resistant PEKS Scheme.* The academic community usually defines the security of PEKS as the indistinguishability of keywords under chosen keyword attacks (IND-CKAs). It means that the PEKS ciphertext has the confidentiality of its contained keywords against an adversary who could not obtain the corresponding keyword search trapdoor. Specifically, IND-CKA security allows a PPT adversary $\mathscr{A}$ to get a public key, query the keyword retrieval trapdoor of some desired keywords, and adaptively select two sets of keywords with the same size to challenge. A PEKS scheme is recognized to be secure if $\mathscr{A}$ cannot distinguish the two PEKS ciphertexts of two challenge keyword sets.

*4.3.1. IND-CKA Security.* In the depiction of IND-CKA security, a challenger $\mathscr{C}$ and an adversary $\mathscr{A}$ will execute interactive games as follows:

(i) Setup phase: On inputting the security parameter $\lambda$, a challenger $\mathscr{C}$ produces public parameters $PP$ and executes the setup algorithm. Then, they produce receivers' public/private key pairs $pk_{\text{PEKS}}, sk_{\text{PEKS}}$ and send $PP, pk_{\text{PEKS}}$ to the adversary $\mathscr{A}$.

(ii) Query phase 1: $\mathcal{A}$ could adaptively release the following keyword search trapdoor query of expected keyword polynomial times in this phase:

Trapdoor query $Q_T(kw)$: for any search trapdoor query of the keyword $kw$, $\mathcal{C}$ produces the corresponding trapdoor $CT_{kw}$ by executing $\text{Trapdoor}(sk_{\text{PEKS}}, kw)$ and gives back $CT_{kw}$ to $\mathcal{A}$.

(iii) Challenge phase: Having terminated query phase 1, $\mathcal{A}$ adaptively selects two challenge keyword sets $\mathcal{W}_1^*, \mathcal{W}_2^*$ with the same number of keywords, i.e., $|\mathcal{W}_1^*| = |\mathcal{W}_2^*|$ and transmits these two sets to $\mathcal{C}$. Then, $\mathcal{C}$ chooses a random 0/1 bit $b$ and calculates the challenge ciphertext $CT^* = CT_{\mathcal{W}_c^*}$ by performing the $\text{PEKS}(sk_{\text{PEKS}}, kw)$ algorithm. Eventually, $\mathcal{C}$ transmits $CT^*$ to $\mathcal{A}$.

(iv) Query phase 2: $\mathcal{A}$ can carry on executing the search trapdoor query of any keywords in this phase as in query phase 1, except for the keywords in challenge sets $\mathcal{W}_1^*, \mathcal{W}_2^*$.

(v) Guess phase: At last, $\mathcal{A}$ returns a guess bit $b\prime$ to challenge ciphertext $CT^*$.

We said that the adversary $\mathcal{A}$ succeeds in the above game if they successfully guess the right bit, i.e., $b' = b$. Assume $\Pr[b' = b]$ denotes the probability of $\mathcal{A}$ successfully guessing the bit, the advantage of $\mathcal{A}$ winning this game is set as

$$\text{Adv}_{\mathcal{A}}^{\text{IND-CKA}}(\lambda) = \left|\Pr[b' = b] - \frac{1}{2}\right|. \tag{3}$$

*Definition 1.* A PEKS scheme is recognized to be indistinguishable against chosen keywords attacks if for any PPT adversary $\mathcal{A}$, and the advantage $\text{Adv}_{\mathcal{A}}^{\text{IND-CKA}}(\lambda)$ of succeeding in the above game is nonnegligible.

We use a quantum random oracle to simulate hash functions in the formal security proof of the proposed IsoqurPEKS scheme. However, an obstacle to security proofs is how to produce random values for exponential queries, that is, how to simulate hash function under the quantum random oracle model (QROM). In the next part, we give several preliminary definitions used in the QROM.

*4.3.2. Specific Techniques Used in QROM.* For a hash function $H: \mathcal{D} \longrightarrow \mathcal{R}$ simulation, an adversary tosses a superposition $|\varphi\rangle = \sum \lambda_x |x\rangle$ and random oracle outputs $\sum \lambda_x |H(x)\rangle$. If $\mathcal{R}$ is tremendous for a quantum simulator, it is hard to give back all random responses of $H$ through computing $\sum \lambda_x |H(x)\rangle$. Zhandry [35] put forward a measure by introducing the concept of $k$-wise independent functions.

In the following, we introduce the concept of marginal weight distribution. A weight distribution on a set $\mathcal{D}$ is defined by a probability distribution function $D: \mathcal{D} \longrightarrow \mathbb{R}$ that has $\sum_{x \in \mathcal{D}} D(x) = 1$, where $D(x) \geq 0$ for all $x \in \mathcal{D}$ is an assignment on $\mathcal{D}$. We consider a family of functions $H: \mathcal{D} \longrightarrow \mathcal{R}$ for a domain $\mathcal{D}$ and range $\mathcal{R}$, denoted by $\mathcal{H}_{\mathcal{D},\mathcal{R}}$. We give the definition of marginal weight distribution $D_{\mathcal{G}}$ of $D$ on $H_{\mathcal{G},\mathcal{R}}$ where the weight of a function $H_{\mathcal{G}}: \mathcal{G} \longrightarrow \mathcal{R}$ equals to the sum of the weights of all $H \in H_{\mathcal{D},\mathcal{R}}$ that is consistent with $H_{\mathcal{G}}$ on $\mathcal{G}$. In other words,

$$D_{\mathcal{G}}(H_{\mathcal{G}}) = \Pr[H(x) = H_{\mathcal{G}}(x), \quad \forall x \in \mathcal{G}]. \tag{4}$$

*Definition 2.* Two weight distributions $D_1$ and $D_2$ on $H_{\mathcal{D},\mathcal{R}}$ are called $t$-wise equivalents if for all $\mathcal{G} \subset \mathcal{D}$ with size $t$, and the marginal weight distributions $D_{1,\mathcal{G}}$ and $D_{2,\mathcal{G}}$ over $H_{\mathcal{G},\mathcal{R}}$ are the same.

*Definition 3.* A function $g$ is called $t$-wise independent function if $g$ is equal to a random function for all $\mathcal{G} \subset \mathcal{D}$ with size $t$.

Next, we give the definition of semiconstant distribution, which is used to support inserting a random value into a small but essential part of oracle inputs.

*Definition 4.* The semiconstant distribution $SC_\omega$ over $H_{\mathcal{D},\mathcal{R}}$ is defined as follows:

(i) First, a random value $y$ is selected from $\mathcal{R}$.

(ii) Then, for each $x \in \mathcal{D}$,

Assign $y$ to $H(x)$ with probability $\omega$. $x$ is said to be a distinct input to $H$.
Otherwise, assign a random element in $\mathcal{R}$ to $H(x)$.

## 5. Problem Formulation

In this section, we describe the system architecture of IsoqurPEKS, the threat model of each entity, and the design goals of this paper.

*5.1. System Model.* The system includes the following parties: cloud server (CS), data providers (DPs), and request users (RUs) as depicted in Figure 3. The characteristics and function of each party are depicted as follows:

(i) Data providers (DPs): Each DP produces his or her public key and private key upon inputting the security parameter. Moreover, the DP extracts keywords from files, encrypts files using symmetric encryption, and computes the searchable keyword ciphertexts associated with corresponding files. Finally, the DP stores encrypted files and searchable ciphertexts on the CS.

(ii) RUs: Request users utilize targeted keywords to generate search trapdoors and send them to the CS for information retrieval operation. Then, RUs decrypt desired files when receiving matched ciphertexts from CS.

(iii) CS: The cloud server has almost unlimited storage and computing power in the PEKS system. The CS is in charge of storing encrypted files and searchable ciphertexts received from DPs. Then, the CS addresses search queries and returns corresponding searching results ciphertexts to RUs.

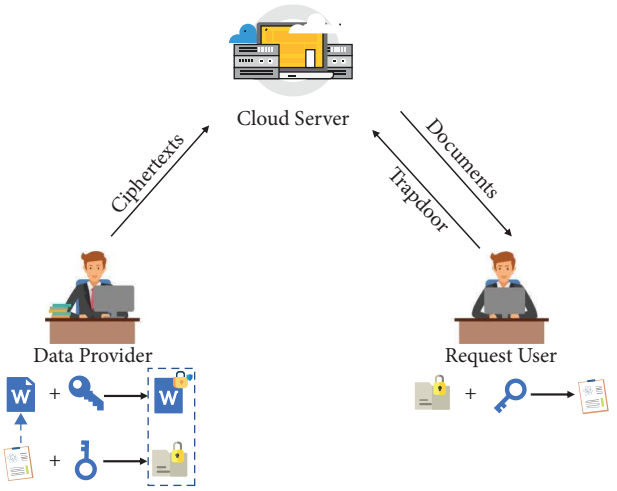



Figure 3: System model of the proposed IsoqurPEKS scheme.

In our proposed IsoqurPEKS scheme, the data provider first extracts keywords (e.g., using such as Porter temming algorithm [36]) from documents to be uploaded. Then, they use the targeted user's public key and a symmetric key to generate a searchable keyword ciphertext and corresponding encrypted document, which will be transferred to the cloud server (CS). When a request user (RU) searches some documents, including a specific keyword, the RU utilizes their secret key to produce a keyword search trapdoor and transfers it to the CS. Finally, the CS returns matched encrypted documents by a verifying equation through inputting a trapdoor and searchable ciphertexts.

*5.2. Threat Model.* In our scheme, we suppose that DPs honestly follow the PEKS algorithm to produce searchable ciphertexts for authorized users and transmits these ciphertexts to the CS.

RUs are assumed to be semihonest adversaries. They honestly execute the scheme when conducting a search query while may attempt to know some sensitive information associated with ciphertexts and queries, respectively, produced by other DPs and RUs.

The CS is supposed to be honest but curious which will honestly perform the test algorithm and has an interest in obtaining desired information of other parties through either intermediate values or computation results.

*5.3. Design Goals.* Our goal is to propose an isogeny-based quantum-resistant PEKS scheme equipped with functions and security requirements.

(i) Data privacy: nobody could get information about searchable ciphertexts uploaded and encrypted by DPs. In other words, the searchable ciphertext remains confidential.

(ii) Access pattern hiding: Retrieval results of a query such as encrypted files matching the user's query are concealed from the CS.

(iii) Quantum attack resilience: There does not exist a polynomial-time quantum algorithm that could acquire RUs' private information such as private secrets and uploaded plaintexts by DPs.

## 6. Proposed IsoqurPEKS

Our proposed scheme consists of four algorithms: setup, PEKS, trapdoor, and test. The setup algorithm is executed by a user and generates the user's public and private key pair using an isogeny and a random number by inputting a security parameter. ThePEKS algorithm is performed by data providers and used to produce the searchable ciphertext against quantum computer attacks. To obtain some files containing specific keywords from the cloud server, a user utilizes their secret to perform the trapdoor algorithm to output a searchable trapdoor. Finally, the cloud server inputs the trapdoor, the user's public key, and searchable ciphertext and then returns correct ciphertexts to the user by the test algorithm

(i) Setup $(\lambda)$: the setup algorithm is executed by a user to produce a pair of private key and public key.

First, the user selects an elliptic curve $E$ with an order $N$ where $P_1, P_2$ are two base points on the additive cyclic group $E[N]$.
Then, they randomly choose $\alpha \in \mathbb{Z}_N^*$ and generate an $l$-isogeny $\phi: E \longrightarrow E'$ with its dual $l$-isogeny $\widehat{\phi}: E' \longrightarrow E$ and two bilinear mappings $e, e'$, respectively, on $E[N], E'[N]$.
They also produce two secure cryptographic hash functions $H': \{0,1\}^* \longrightarrow E'[N]$ and $h: \{0,1\}^* \longrightarrow \{0,1\}^{lgN}$.
Finally, they compute $\phi(\alpha P_1), \phi(\alpha P_2)$ and set $PK = \{PK_1 := \phi(\alpha P_1), PK_2 := \phi(\alpha P_2), P_1, P_2, e, e', H', h\}, SK = \{\alpha, \phi, \widehat{\phi}\}$.

(ii) PEKS $(PK, \mathscr{W})$: when a data provider transmits encrypted files to the cloud server for secure storage and retrieval, they extract keywords $\mathscr{W} = \{kw_1, \ldots, kw_n\}$ from the file to upload and perform the following steps:

For each $kw_i \in \mathscr{A}$, the DP randomly selects $r, s \in \mathbb{Z}_N^*$ and uses the random numbers and the authorized user's public key to compute searchable ciphertexts $C_0 = rP_1 + sP_2$, $t_i = e'(PK_1, rH'(kw_i)) \cdot e'(PK_2, sH'(kw_i))$, $C_i = h(t_i)$.
Then, they initialize a history-independent array $\mathscr{L}$ to store ciphertexts $C_i, i \in \{1, \ldots, n\}$.
Finally, this algorithm outputs PEKS ciphertexts $AE = (C_0, \mathscr{L})$ and sends them together with the corresponding encrypted file to the cloud server.

(iii) Trapdoor $(SK, kw)$: if a user desires to request files including the keyword $kw'$, they utilize their private key $\alpha, \widehat{\phi}$ to compute a trapdoor $T_{kw'} = \widehat{\phi}(\alpha H'(kw'))$ and transfer the trapdoor $T_{kw'}$ to the cloud server.

(iv) Test$(PK, \mathrm{AE}, T_{kw'})$: given public key $PK$, a PEKS ciphertext AE = $(C_0, \mathscr{L})$, and a trapdoor $T_{kw'}$, the cloud server performs steps as follows:

It initializes an empty set $S$ and verifies whether $h(e(C_0, T_{kw'})) \in \mathscr{L}$.
If $h(e(C_0, T_{kw'})) \in \mathscr{L}$, the CS adds the corresponding encrypted file to $S$; otherwise, it searches the next ciphertext.
Finally, the CS returns $S$ to the user.

What should be specified is that our main work focuses on security against quantum attacks, and we suppose keywords are uncertain and unlimited. Thus, we do not take into account the keyword guessing attack.

*6.1. Consistency.* According to the introduction in Section B, we analyze the correctness of isoqurPEKS. Given $T_{kw'}$ produced by the trapdoor algorithm and $(C_0, \mathscr{L})$ produced by the PEKS algorithm, we have

$$\begin{aligned} e(C_0, T_{kw'}) &= e\left(rP_1 + sP_2, \widehat{\phi}(\alpha H'(kw'))\right), \\ &= e'(\phi(rP_1 + sP_2), \alpha H'(kw')), \\ &= e'(\phi(rP_1), \alpha H'(kw')) \cdot e'(\phi(sP_2), \alpha H'(kw')), \\ &= e'(PK_1, rH'(kw')) \cdot e'(PK_2, sH'(kw')). \end{aligned} \tag{5}$$

On the other hand, $C_i = h(e'(PK_1, rH'(kw_i)) \cdot e'(PK_2, sH'(kw_i)))$. Thus, the equation $h(e(C_0, T_{kw'})) \in \mathscr{L}$ if there exists a $C_t, t \in \{1, 2, \ldots, n\}$ which satisfies $kw_t = kw'$.

## 7. Security Proof

In this section, we will prove IND-CKA security of the IsoqurPEKS instantiation under the QROM, and the proof method of which has been used in lecture [37].

**Theorem 1.** *For the advantage $Adv^{SSI}(\mathscr{R})$ of the computational isogeny problem and the advantage $Adv^{IND-CKA}_{IsoqurPEKS}(A)$ of $\mathscr{A}$ breaking IsoqurPEKS's security, we have the following equation under the quantum random oracle model:*

$$\mathrm{Adv}^{\mathrm{IND-CKA}}_{\mathrm{IsoqurPEKS}}(A) \le \mathrm{Adv}^{\mathrm{SSI}}(R)^{1/2}\left(\frac{3}{4}(q_h + 1)^4 + 2q_h\right)^{1/2}, \tag{6}$$

*where $q_h$ is the maximum of hash function queries.*

*Proof.* Game $G_0$: this game is executed by the adversary $\mathscr{A}$ who tries to break the real scheme as 1 and the challenger $\mathscr{C}$. Specifically, $\mathscr{C}$ responds the trapdoor query according to the trapdoor algorithm:

(i) Trapdoor query $Q_T(kw')$: Given a query keyword $kw'$, $\mathscr{C}$ computes $T_{kw'} \leftarrow \mathrm{Trapdoor}(SK, kw')$ and gives back $T_{kw'}$ to $\mathscr{A}$.

Assume that the adversary $\mathscr{A}$'s advantage in this game is $\mathrm{Adv}(\mathscr{A}, G_0) = \epsilon$ and the challenger $\mathscr{C}$ knows related secret values. We have the advantage of $\mathscr{A}$ breaking IND-CKA in $G_0$ is the same as that in the real word:

$$\mathrm{Adv}(\mathscr{A}, G_0) = \mathrm{Adv}(\mathscr{A}, \mathrm{Real}). \tag{7}$$

Game $G_1$: The game $G_1$ is identical to $G_0$ except for the challenge ciphertext $\mathscr{CT}^*$ generation in the challenge phase. What should be specified is that the public keys are changed into $(\widehat{\phi}(xQ_1), \widehat{\phi}(xQ_2), Q_1, Q_2)$ and that private keys are $(x, \widehat{\phi}, \phi)$, where $x \in \mathbb{F}_q$, $Q_1, Q_2 \in E'[N]$ are two points and $\widehat{\phi}: E'[N] \longrightarrow E[N]$ is an isogeny. These settings correspond to $PK$ and $SK$ of the proposed scheme in Section 6. Therefore, the challenge ciphertext is accordingly transformed into $C_0^* = r\widehat{\phi}(xQ_1) + s\widehat{\phi}(xQ_2)$, $t_i^* = h(e'(Q_1, rH'(kw_i))e'(Q_2, sH'(kw_i)))$, $C_i^* = h(t_i^*)$, where $r, s$ are two random numbers in $\mathbb{F}_q$, $H': \{0,1\}^* \longrightarrow E'[N]$ is a hash function, and $h: \{0,1\}^* \longrightarrow \{0,1\}^{\lg N}$ is a general hash function. This transformation does not change the searchable ciphertext computing rule, and no more information has been leaked; thus, we have

$$\mathrm{Adv}(\mathscr{A}, G_1) = \mathrm{Adv}(\mathscr{A}, G_0). \tag{8}$$

Game $G_2$: In this game, we introduce the rule of aborting. Let $\omega$ be selected from (0, 1), and $\mathscr{W}$ is a subset of $\mathscr{D}$ where $kw$ is randomly chosen from $\mathscr{D}$ and placed in $\mathscr{W}$ with an independent probability $\omega$. $\mathscr{W}_0^*, \mathscr{W}_1^*$ are two challenge keyword sets chosen by $\mathscr{A}$. $G_2$ aborts if two challenge keyword sets $\mathscr{W}_0^*, \mathscr{W}_1^* \subsetneq \subsetneq$, $\mathscr{A}$ queries $H'(kw)$ where $kw \in \mathscr{W}$, and we have

$$\begin{aligned} \mathrm{Adv}(\mathscr{A}, G_2) &\ge \omega(1 - \omega q_{H'})\mathrm{Adv}(\mathscr{A}, G_2), \\ &\ge \omega \mathrm{Adv}(\mathscr{A}, G_2) - \omega^2 q_{H'}. \end{aligned} \tag{9}$$

Before continuing to the next simulation, we give the following lemmas [35] to depict QROM.

**Lemma 1.** *Let $\mathscr{A}$ be an adversary with the capacity of quantum computing and makes $q$ queries to an random oracle $H: \mathscr{D} \longrightarrow \mathscr{R}$. We depict $H$ using some weight distribution $D$; that is, for each $z$, the probability value $Pr_{H \leftarrow D}[H = z]$ is a linear combination of $Pr_{H \leftarrow D}[H(r_i) = d_i, i \in \{1, \ldots, 2s\}]$ for all possible $r_i$ and $d_i$.*

**Lemma 2.** *Suppose there is a $2q_i$-wise independent function. In that case, it can be successfully simulated by a quantum algorithm $\mathscr{R}$ when any quantum adversary $\mathscr{A}$ makes $q_i$ queries to random oracles $\mathscr{B}_i$, which could have the same output values while making no queries.*

According to the above lemmas, we can see that quantum random oracles are simulated by a quantum algorithm $\mathscr{R}$ in the polynomial time. This technique can simulate hash function queries and responses of the $H'$-query and $h$-query in the IsoqurPEKS's security proof.

In addition, how to insert some randomly selected values to the intended quantum oracle queries is another problem of security proofs under QROM. Then, we have Lemma 3 as follows.

**Lemma 3.** *The The distribution of outputs of a quantum algorithm making $q_H$ queries to an oracle drawn from the semi-constant distribution $\omega$ is at most a distance $3/8q_H^4\omega^2$ away from the case when the oracle is drawn from the uniform distribution.*

We assume that if an adversary $\mathscr{A}$ queries the inserted value of corresponding oracle outputs, then the simulation is successful with the probability $\epsilon$. Furthermore, the probability of successful simulations is $\epsilon\ \omega - 3/8q_H^4\omega^2$ if $\mathscr{A}$ utilizes one of the values with the probability $\omega$, where the choice of $\omega$ could decide the success probability. We employ this solution to insert a hard-to-be-resolved SSI problem into a hash function $h$ output in the IsoqurPEKS's security proof.

Game $G_3$: This game introduces a quantum random oracle. In other words, the computing method of the hash function $H'(\cdot)$ is changed. $\eta$ is set as $H'(kw^*)$ for all $kw^* \in \mathscr{W}$, and hash outputs are randomly selected for other queries. In this case, $H'$ is distributed according to $SC_\omega$. According to Lemma 3, the distance of output distribution in $G_3$ is $3/8(q_h+1)^4\omega^2$ from that in $G_2$. Therefore, we have

$$\mathrm{Adv}(\mathscr{A}, G_2) - \mathrm{Adv}(\mathscr{A}, G_3) \geq 3/8(q_h+1)^4\omega^2. \tag{10}$$

Game $G_4$: In game $G_4$, the rule of producing challenge ciphertexts is changed and $C_i^*$, $(i = 1, \ldots, n)$ are randomly selected instead of computing by $h(C_i^*)$. The final challenge ciphertext is independent of the challenge keyword sets. Therefore, we have

$$\mathrm{Adv}(\mathscr{A}, G_4) = 0. \tag{11}$$

We construct an algorithm $\mathscr{R}$ of the isogeny problem with the advantage $\mathrm{Adv}_{\mathscr{W}}^{\mathrm{CSSI}}(\mathscr{R})$. We suppose that $\mathscr{R}$ has quantum access to random oracles $\widehat{H'}: \mathscr{W} \longrightarrow E'[N], \widehat{H''}: \mathscr{W} \longrightarrow \{0,1\}$ and $\hat{h}: G_T \longrightarrow F_q^*$ where the probability of $\widehat{H''}$ outputting 1 is $\omega$. Let $\mathscr{W}$ be the set of $kw^*$ such that $H'(kw^*) = 1$. We can infer that the above conditions are equivalent to $G_4$. According to Lemma 2, $\mathscr{R}$ can simulate $(\widehat{H'}, \widehat{H''})$ and $\hat{h}$ by separately using a $(q_{H'}+1)$-wise and a $(q_h+1)$-wise independent functions without oracle queries. $\mathscr{L}$ is an initially empty list held by $\mathscr{R}$.

(i) Setting of public parameters. Assume that an adversary $\mathscr{A}$ transmits two challenge sets $\mathscr{W}_0^* = \{w_{0,1}^*, w_{0,2}^*, \ldots, w_{0,n}^*\}$ and $\mathscr{W}_1^* = \{w_{1,1}^*, w_{1,2}^*, \ldots, w_{1,n}^*\}$ to the challenger $\mathscr{C}$, where $n$ is a positive integer. Then, given $(P, \phi(x^{-1}P))$ and $(Q_1, Q_2, \hat{\phi}(xQ_1), \hat{\phi}(xQ_2))$, $\mathscr{C}$ sets $pk = (\hat{\phi}(xQ_1), \hat{\phi}(xQ_2), Q_1, Q_2)$ and $sk$ are correspondingly $(x, \phi, \phi)$, where the isogeny $\phi$ is mapping from the elliptic curve $E$ to the elliptic curve $E'$, $\hat{\phi}$ is its dual isogeny, $(P, \hat{\phi}(xQ_1), \hat{\phi}(xQ_2))$ are points on $E$, and $Q_1, Q_2, \phi(x^{-1}P)$ are points on $E'$. $\mathscr{R}$ receives the hard problem instance $(\hat{\phi}(xQ_1), \hat{\phi}(xQ_2), Q_1, Q_2)$. Then, $\mathscr{R}$ transfers the public parameter $E, E'$ and $PK = (\hat{\phi}(xQ_1), \hat{\phi}(xQ_2), Q_1, Q_2)$ to $\mathscr{A}$.

(ii) Challenge ciphertext simulation. $\mathscr{R}$ chooses random $r^*, s^*$ and computes $C_0^* = r^*\hat{\phi}(xQ_1) + s^*\hat{\phi}(xQ_2), \eta$ as the challenge ciphertext.

(iii) $H'$-query. Upon receiving $kw$, $\mathscr{R}$ sets the outputs of $H'$ as follows:

$$H'(kw) = \begin{cases} \zeta, & \text{if } \widehat{H'}(kw) = 1, \\ \widehat{H'}(kw), & \text{otherwise.} \end{cases} \tag{12}$$

(iv) $h$-query. Upon receiving $C_i$, $\mathscr{R}$ simulates $h$ by setting $h(C_i) = \hat{h}(C_i)$.

(v) Query simulation.

Hash query $Q_{H'}(kw)$: It uniformly selects a random $\alpha_i$ and computes $\alpha_i\phi(x^{-1}P_1)$. Then, $\mathscr{C}$ reserves $(kw, \alpha, \alpha_i\phi(x^{-1}P_1))$ in the $H'$-list, and the $\alpha_i\phi(x^{-1}P_1)$ is transferred to $\mathscr{A}$.

Trapdoor query $Q_T(kw')$: Given a query keyword $kw$, $\mathscr{C}$ retrieves $kw'$ in the $H'$-list and uses $\alpha$ in the $H'$-list to compute $\alpha_i P_1$ if $kw'$ is in the $H'$-list; otherwise, it uniformly chooses an $\alpha$ and computes $\alpha P_1$. Finally, $\mathscr{C}$ returns $\alpha_i P_1$ to $\mathscr{A}$.

*7.1. Success Probability Analysis.* If $\mathscr{A}$ performs queries contained in $\mathscr{W}$ to $H'$, $\mathscr{A}$ could distinguish the simulation environment from the real environment. Nevertheless, these events will not appear due to the game hopping in $G_2$. Then, $\mathscr{A}$ succeeds in the game with the advantage $\mathrm{Adv}(\mathscr{A}, G_5)$. Hence, we have

$$\left|\mathrm{Adv}(\mathscr{A}, G_4) - \mathrm{Adv}(\mathscr{A}, G_3)\right| \leq \mathrm{Adv}^{\mathrm{SSI}}(\mathscr{R}). \tag{13}$$

Then, by combining advantages, we have

$$\mathrm{Adv}^{\mathrm{IsqurPEKS}}(A) \leq \frac{1}{\omega}\mathrm{Adv}^{\mathrm{SSI}}(R) + \omega\left(\frac{3}{8}(q_{H'}+1)^4 + q_{H'}\right). \tag{14}$$

Because right side of the equation is minimized when $\omega = \sqrt{\mathrm{Adv}^{\mathrm{SSI}}(\mathscr{R})/3/8(q_{H'}+1)^4 + q_{H'}}$, we have

$$\mathrm{Adv}_{\mathrm{IsoqurPEKS}}^{\mathrm{IND-CKA}}(A) \leq \mathrm{Adv}^{\mathrm{SSI}}(R)^{1/2}\left(\frac{3}{4}(q_{H'}+1)^4 + 2q_{H'}\right)^{1/2}. \tag{15}$$

## 8. Comparison and Analysis

To the best of our knowledge, there is no isogeny-based quantum-resistant PEKS scheme currently. There are many public encryption schemes with keyword search [38–45], while they cannot withstand quantum attacks since these schemes are based on classical DL assumption, DBDH assumption, or CBDH assumption. In this section, we first compare IsoqurPEKS with existing PEKS schemes regarding security properties. Then, we compare IsoqurPEKS with the other eight PEKS schemes from aspects of computation and communication costs.

Table 2: Comparison of security properties with other PEKS schemes.

| Schemes | QR | NTA | Security |
|---|---|---|---|
| XLZCT20 [38] | × | × | SM |
| WXLYY20 [39] | × | ✓ | ROM |
| ZQDT21 [40] | × | ✓ | ROM |
| ZLW19 [41] | × | ✓ | ROM |
| NT21 [42] | × | ✓ | ROM |
| MHFF20 [43] | × | × | SM |
| SBSL21 [44] | × | × | ROM |
| WDC20 [45] | × | × | SM |
| IsoqurPEKS | ✓ | ✓ | QROM |

QR: quantum resistance, NTA: no trusted model, ✓: the scheme supports corresponding features, ×: the scheme fails in supporting the corresponding feature, SM: standard model, ROM: random oracle model, QROM: quantum random oracle model.

Table 3: Operations comparison of PEKS schemes.

| Schemes | Ciphertext generation | Trapdoor generation | Testing |
|---|---|---|---|
| XLZCT20 [38] | $4T_{EX}$ | $52T_{EX} + 5T_{GM}$ | $49T_{EX}$ |
| WXLYY20 [39] | $4T_{EX}$ | $1T_{EX}$ | $2T_{EX}$ |
| ZQDT21 [40] | $T_{BP} + 2T_{EX} + T_H$ | $T_{BP} + T_{EX} + T_H$ | $T_{BP} + T_{EX} + T_H$ |
| ZLW19 [41] | $7T_{EX}$ | $6T_{EX}$ | $6T_{BP}$ |
| NT21 [42] | $2T_{BP} + 5T_{GM} + T_h$ | $2T_{BP} + 5T_{GM} + T_h$ | $2T_{BP} + 2T_{EX} + 2T_{GM}$ |
| MHFF20 [43] | $3T_{BP} + 13T_{EX}$ | $T_{EX} + 4T_h$ | $4T_{BP} + 4T_{EX}$ |
| SBSL21 [44] | $T_{BP} + 3T_H + 3T_{GM}$ | $2T_H + 2T_{GM}$ | $T_{BP} + 4T_H$ |
| WDC20 [45] | $4T_{EX}$ | $T_{BP} + 2T_{EX}$ | $2T_{BP}$ |
| IsoqurPEKS | $2T_{BP} + 2T_{GM}$ | $T_{GM} + T_{Iso}$ | $T_{BP}$ |

$T_H$: running time of a hash-to-point operation, $T_{BP}$: running time of a bilinear pairing operation, $T_{GM}$: running time of a general multiplication over point, $T_{EX}$: running time of a modular exponentiation operation, $T_{\text{Iso}}$: running time of an isogeny operation.

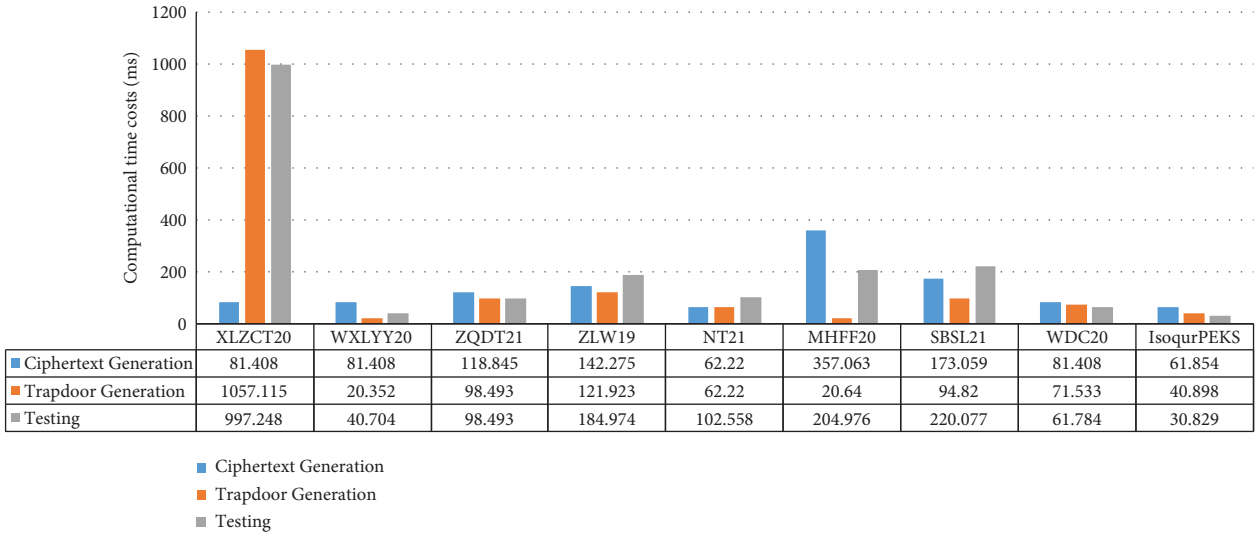
Table 2 indicates the comparison results among the proposed IsoqurPEKS scheme and its counterpart PEKS schemes concerning security properties. The proposed IsoqurPEKS scheme is based on isogeny hard assumption, which has been proved in Section 7 under the quantum random oracle model. Thus, our scheme can resist quantum attacks. Moreover, our construction does not require a trusted authority to generate secret keys, which some PEKS schemes require it.

Subsequently, we analyze the computational complexity with respect to searchable ciphertext generation, trapdoor generation, and testing. We only consider the time-consuming operations, e.g., hash-to-point ($T_H$), bilinear pairing operation ($T_{BP}$), general multiplication over point ($T_{GM}$), modular exponentiation operation ($T_{EX}$), and isogeny operation ($T_{\text{Iso}}$). According to [46], we get the running time of different operations implemented on a Raspbian GNU/Linux 8 system with ARMv7 Processor rev 4 1.2 GHz. Because the isogeny operation happens in the trapdoor generation process, which is performed by the cloud server, we use the isogeny (i.e., group action) computing time as described in [47]. Above all, we have $T_H = 47.312\,ms, T_{BP} = 30.829\,ms, T_{GM} = 0.098\,ms, T_{EX} = 20.352\,ms, T_{\text{Iso}} = 40.8\,ms$. Table 3 shows the different operation comparisons of nine PEKS schemes. In the PEKS phase, the data provider generates one searchable ciphertext for each keyword by computing

$C_0 = rP_1 + sP_2, t_i = e'(PK_1, rH'(kw_i)) \cdot e'(PK_2, sH'(kw_i))$

which requires two bilinear pairing operations and two scalar multiplication operations. In the trapdoor generation phase, the user computes $T_{kw'} = \widehat{\phi}(\alpha H'(kw'))$, which requires one scalar multiplication operation and one isogeny operation. When the server searches matched ciphertexts, it computes $e(C_0, T_{kw'})$ and $h(e(C_0, T_{kw'}))$ which requires one bilinear pairing operation and one general hash function operation. The comparison results in Figure 4 indicate that IsoqurPEKS consumes the least time in ciphertext generation, trapdoor generation, and testing processes among these nine PEKS schemes.

In addition, we also perform a comparison with respect to the communication complexity of single document/keyword encryption and search. Since the elliptic curve point group is defined over a finite field $F_q$, we set $p$ as a 512-bit element. For pairing-based schemes, the pairing operation is $e$: $\mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_T$ where points in $\mathbb{G}_1$ and $\mathbb{G}_T$ are 1024-bit elements. The general hash function is SHA256 in implementation. Thus, the output of $h$ is a 256-bit string. The comparison results are depicted in detail in Table 4. For the searchable ciphertext and trapdoor production of a single keyword in IsoqurPEKS, it outputs one point $C_0$, one hash value $h(t_i)$, and one point $\widehat{\phi}(\alpha H'(kw'))$. Thus, the communication cost of the proposed scheme is 160 bytes and 128 bytes, respectively, which requires the least communication width for keywords trapdoor search and giving back matched ciphertexts.




| | XLZCT20 | WXLYY20 | ZQDT21 | ZLW19 | NT21 | MHFF20 | SBSL21 | WDC20 | IsoqurPEKS |
|---|---|---|---|---|---|---|---|---|---|
| Ciphertext Generation | 81.408 | 81.408 | 118.845 | 142.275 | 62.22 | 357.063 | 173.059 | 81.408 | 61.854 |
| Trapdoor Generation | 1057.115 | 20.352 | 98.493 | 121.923 | 62.22 | 20.64 | 94.82 | 71.533 | 40.898 |
| Testing | 997.248 | 40.704 | 98.493 | 184.974 | 102.558 | 204.976 | 220.077 | 61.784 | 30.829 |

Figure 4: Running time comparison.

Table 4: Communication costs comparison of PEKS schemes (bytes).

| Schemes | Ciphertext | Trapdoor |
|---|---|---|
| XLZCT20 [38] | $5\|\mathbb{G}_1\| + \|\mathbb{G}_T\| = 768$ | $5\|\mathbb{G}_1\| = 640$ |
| WXLYY20 [39] | $\|\mathbb{G}_T\| = 128$ | $\|\mathbb{G}_1\| = 128$ |
| ZQDT21 [40] | $\|q\| + \|\mathbb{G}_1\| = 192$ | $\|\mathbb{G}_1\| = 128$ |
| ZLW19 [41] | $\|q\| + 5\|\mathbb{G}_1\| = 704$ | $\|\mathbb{G}_1\| = 128$ |
| NT21 [42] | $2\|\mathbb{G}_1\| + \|\mathbb{G}_T\| = 384$ | $\|\mathbb{G}_1\| + \|\mathbb{G}_T\| = 256$ |
| MHFF20 [43] | $\|q\| + 4\|\mathbb{G}_1\| + 2\|\mathbb{G}_T\| = 832$ | $\|q\| + \|\mathbb{G}_1\| = 192$ |
| SBSL21 [44] | $\|q\| + \|\mathbb{G}_1\| = 704$ | $\|\mathbb{G}_1\| = 128$ |
| WDC20 [45] | $2\|\mathbb{G}_1\| = 256$ | $\|\mathbb{G}_T\| = 128$ |
| IsoqurPEKS | $\|h\| + \|\mathbb{G}_1\| = 160$ | $\|\mathbb{G}_T\| = 128$ |

$|\mathbb{G}_1|$: the byte size of a point in $\mathbb{G}_1$. $|\mathbb{G}_T|$: the byte size of a point in $\mathbb{G}_T$. $|q|$: the byte size of an element in $q$. $|h|$: the output byte length of the general hash function $h(\cdot)$.

## 9. Conclusion

This paper introduces a new method for the quantum-resistant public encryption scheme with keyword search construction and establishes the hard assumption of elliptic curve isogeny computation. Our proposed scheme, IsoqurPEKS, could fight against attacks of quantum adversaries and is provably secure under the quantum random oracle model. We give formal security proof of IsoqurPEKS and analyze its security properties by comparing it with the other eight PEKS schemes. As far as we know, IsoqurPEKS is not only the first isogeny-based quantum-resistant PEKS scheme but also the most efficient scheme in terms of computation and communication costs compared with the listed current eight PEKS schemes. Since IsoqurPEKS is designed under the assumption that keywords cannot be enumerated, our subsequent work is putting forward an isogeny-based and quantum-resistant PEKS scheme against keyword guessing attack under the assumption that keywords could be listed in the polynomial time.

## Data Availability

The data used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare no conflicts of interest.

## Acknowledgments

The work was supported by the Shandong Provincial Key Research and Development Program (No. 2020CXGC010107, 2021CXGC010107), the National Natural Science Foundation of China (Nos. 62172307, U21A20466), the Special Project on Science and Technology Program of Hubei Province (No. 2020AEA013), the Natural Science Foundation of Hubei Province (No. 2020CFA052), and the Wuhan Municipal Science and Technology Project (No. 2020010601012187).

## References

[1] Y. Zhou, N. Li, Y. Tian, D. An, and L. Wang, "Public key encryption with keyword search in cloud: a survey," *Entropy*, vol. 22, no. 4, p. 421, 2020.

[2] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging it platforms: vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, 2009.

[3] G. S. Poh, J.-J. Chin, W.-C. Yau, K. K. R. Choo, and M. S. Mohamad, "Searchable symmetric encryption: designs and challenges," *ACM Computing Surveys*, vol. 50, no. 3, pp. 1–37, May 2017.

[4] C. Bösch, P. Hartel, W. Jonker, and A. Peter, "A survey of provably secure searchable encryption," *ACM Computing Surveys*, vol. 47, no. 2, pp. 1–51, 2015.

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

[6] Z. Y. Liu, Y. F. Tseng, R. Tso, M. Mambo, and Y. C. Chen, *Public-key Authenticated Encryption with Keyword Search: Cryptanalysis, Enhanced Security, and Quantum-Resistant Instantiation*, Cryptology ePrint Archive, Nagasaki, Japan, 2021.

[7] S. F. Sun, X. Yuan, J. K. Liu et al., "Practical backward-secure searchable encryption from symmetric puncturable encryption," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, ser. CCS '18*, October 2018.

[8] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Proceedings of the Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 506–522, Interlaken, Switzerland, April 2004.

[9] P. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pp. 124–134, Santa Fe, NM, USA, November 1994.

[10] L. Xu, X. Yuan, R. Steinfeld, C. Wang, and C. Xu, "Multi-writer searchable encryption: an lwe-based realization and implementation," in *Proceedings of the 2019 ACM Asia Conference On Computer And Communications Security, Ser. Asia CCS '19*, pp. 122–133, Association for Computing Machinery, New York, NY, USA, July 2019.

[11] L. Chen, S. Jordan, Y.-K. Liu et al., "Report on post-quantum Cryptography," US Department of Commerce, Gaithersburg, MD, USA, NISTIR 8105, 2016.

[12] L. DeFeo, "Mathematics of isogeny based cryptography," 2017, https://arxiv.org/abs/1711.04062.

[13] A. Childs, D. Jao, and V. Soukharev, "Constructing elliptic curve isogenies in quantum subexponential time," *Journal of Mathematical Cryptology*, vol. 8, no. 1, pp. 1–29, 2014.

[14] J. Burdges and L. De Feo, "Delay encryption," in *Advances in Cryptology–EUROCRYPT 2021*, A. Canteaut and F.-X. Standaert, Eds., pp. 302–326, Springer, Cham, 2021.

[15] M. Bellare, A. Boldyreva, and A. O'Neill, "Deterministic and efficiently searchable encryption," in *Advances in Cryptology - CRYPTO 2007*, A. Menezes, Ed., pp. 535–552, Springer, Berlin, Heidelberg, 2007.

[16] M. Abdalla, M. Bellare, D. Catalano et al., "Searchable encryption revisited: consistency properties, relation to anonymous ibe, and extensions," in *Advances in Cryptology–CRYPTO 2005*, V. Shoup, Ed., pp. 205–222, Springer, Berlin, Heidelberg, 2005.

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

[18] M. Kim, H. T. Lee, S. Ling, B. H. M. Tan, and H. Wang, "Private compound wildcard queries using fully homomorphic encryption," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 5, pp. 743–756, 2019.

[19] D. Wang, P. Wu, B. Li, H. Du, and M. Luo, "Multi-keyword searchable encryption for smart grid edge computing," *Electric Power Systems Research*, vol. 212, Article ID 108223, 2022.

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

[21] M. Zhang, Y. Chen, and J. Huang, "Se-ppfm: a searchable encryption scheme supporting privacy-preserving fuzzy multikeyword in cloud systems," *IEEE Systems Journal*, vol. 15, no. 2, pp. 2980–2988, 2021.

[22] Y. Liang, Y. Li, Q. Cao, and F. Ren, "Vpams: verifiable and practical attribute-based multi-keyword search over encrypted cloud data," *Journal of Systems Architecture*, vol. 108, Article ID 101741, 2020.

[23] M. Zarezadeh, H. Mala, and M. Ashouri-Talouki, "Multi-keyword ranked searchable encryption scheme with access control for cloud storage," *Peer-to-Peer Networking and Applications*, vol. 13, no. 1, pp. 207–218, 2020.

[24] M. Zeng, K. Zhang, H. Qian, X. Chen, and J. Chen, "A searchable asymmetric encryption scheme with support for boolean queries for cloud applications," *The Computer Journal*, vol. 62, no. 4, pp. 563–578, 2019.

[25] Z. Chen, F. Zhang, P. Zhang, and H. Zhao, "Multi-user boolean searchable encryption supporting fast ranking in mobile clouds," *Computer Communications*, vol. 164, pp. 100–113, 2020.

[26] Y. Zhang, J. Katz, and C. Papamanthou, "All your queries are belong to us: the power of file-injection attacks on searchable encryption," in *Proceedings of the 25th {USENIX} Security Symposium ({USENIX} Security 16)*, pp. 707–720, Austin, TX, USA, August 2016.

[27] J. Ning, J. Xu, K. Liang, F. Zhang, and E.-C. Chang, "Passive attacks against searchable encryption," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 3, pp. 789–802, 2019.

[28] R. Bost, B. Minaud, and O. Ohrimenko, "Forward and backward private searchable encryption from constrained cryptographic primitives," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, ser. CCS '17*, pp. 1465–1482, Association for Computing Machinery, New York, NY, USA, October 2017.

[29] M. Zeng, H.-F. Qian, J. Chen, and K. Zhang, "Forward secure public key encryption with keyword search for outsourced cloud storage," *IEEE Transactions on Cloud Computing*, vol. 10, no. 1, 2019.

[30] R. Behnia, M. O. Ozmen, and A. A. Yavuz, "Lattice-based public key searchable encryption from experimental perspectives," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 6, pp. 1269–1282, 2020.

[31] C. Peng, J. Chen, S. Zeadally, and D. He, "Isogeny-based cryptography: a promising post-quantum technique," *IT Professional*, vol. 21, no. 6, pp. 27–32, 2019.

[32] L. De Feo, D. Jao, and J. Plût, "Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies," *Journal of Mathematical Cryptology*, vol. 8, no. 3, pp. 209–247, 2014.

[33] T. Moriya, H. Onuki, and T. Takagi, "Sigamal: a supersingular isogeny-based pke and its application to a prf," in *Advances in Cryptology–ASIACRYPT 2020*, S. Moriai and H. Wang, Eds., Springer, Cham, pp. 551–580, 2020.

[34] Q. Fan, D. He, J. Chen, C. Peng, and L. Wang, "Isoga: an isogeny-based quantum-resist searchable encryption scheme against keyword guessing attacks," *IEEE Systems Journal*, no. 1–12, pp. 1–12, 2022.

[35] M. Zhandry, "Secure identity-based encryption in the quantum random oracle model," *International Journal of Quantum Information*, vol. 13, no. 04, Article ID 1550014, 2015.

[36] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 211–218, 2006.

[37] K. Yoneyama, "Post-quantum variants of ISO/IEC standards: compact chosen ciphertext secure key encapsulation mechanism from isogenies," *IEICE - Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E104.A, no. 1, pp. 69–78, 2021.

[38] L. Xu, W. Li, F. Zhang, R. Cheng, and S. Tang, "Authorized keyword searches on public key encrypted data with time controlled keyword privacy," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2096–2109, 2020.

[39] W. Wang, P. Xu, D. Liu, L. T. Yang, and Z. Yan, "Lightweighted secure searching over public-key ciphertexts for edge-cloud-assisted industrial iot devices," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4221–4230, 2020.

[40] W. Zhang, B. Qin, X. Dong, and A. Tian, "Public-key encryption with bidirectional keyword search and its application to encrypted emails," *Computer Standards & Interfaces*, vol. 78, Article ID 103542, 2021.

[41] Y. Zhang, Y. Li, Y. Wang, and S. Cimato, "Secure and efficient searchable public key encryption for resource constrained environment based on pairings under prime order group," *Security and Communication Networks*, vol. 2019, pp. 1–14, 2019.

[42] S. K. Nayak and S. Tripathy, "Seps: efficient public-key based secure search over outsourced data," *Journal of Information Security and Applications*, vol. 61, Article ID 102932, 2021.

[43] M. Ma, D. He, S. Fan, and D. Feng, "Certificateless searchable public key encryption scheme secure against keyword guessing attacks for smart healthcare," *Journal of Information Security and Applications*, vol. 50, Article ID 102429, 2020.

[44] M. R. Senouci, I. Benkhaddra, A. Senouci, and F. Li, "An efficient and secure certificateless searchable encryption scheme against keyword guessing attacks," *Journal of Systems Architecture*, vol. 119, Article ID 102271, 2021.

[45] H. Wang, X. Dong, and Z. Cao, "Multi-value-independent ciphertext-policy attribute based encryption with fast keyword search," *IEEE Transactions on Services Computing*, vol. 13, no. 6, pp. 1142–1151, 2020.

[46] Z. Y. Liu, Y. F. Tseng, R. Tso, M. Mambo, and Y. C. Chen, "Public-key authenticated encryption with keyword search: a generic construction and its quantum-resistant instantiation," *The Computer Journal*, vol. 65, 2020.

[47] W. Castryck, T. Lange, C. Martindale, L. Panny, and J. Renes, "Csidh: an efficient post-quantum commutative group action," in *Advances in Cryptology ASIACRYPT 2018*, T. Peyrin and S. Galbraith, Eds., pp. 395–427, Springer, Cham, 2018.