IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 4 January 2024, accepted 13 February 2024, date of publication 22 February 2024, date of current version 28 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3368908

RESEARCH ARTICLE

IPO-PEKS: Effective Inner Product Outsourcing Public Key Searchable Encryption From Lattice in the IoT

MIAO WANG[1], LIWANG SUN[1], ZHENFU CAO[1,2], AND XIAOLEI DONG[1,2]

[1]Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai 200062, China
[2]Research Center for Basic Theories of Intelligent Computing, Research Institute of Basic Theories, Zhejiang Laboratory, Hangzhou 311121, China

Corresponding author: Xiaolei Dong (dongxiaolei@sei.ecnu.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFB2701400; and in part by the National Natural Science Foundation of China under Grant 62132005, Grant 62172162, Grant 62172161, Grant U22B2029, and Grant 62272228.

ABSTRACT Lightweight devices in the Internet of Things (IoT) typically need to store massive data on a cloud server with strong processing and storage capabilities for later retrieval and usage. Since these data contain the participant's sensitive information, they cannot be delivered directly to the cloud server. Public-key Encryption with Keyword Search (PEKS) allows customers to search for target encrypted files using keywords. However, the majority of PEKS implementations are unable to repel malicious quantum-capable attackers. And with regard to forward security, they must search for many rounds to obtain the necessary data. To resolve these concerns, we propose a comprehensive Inner Product Outsourcing PEKS system (IPO-PEKS) with forward security based on LWE assumptions, which raises search efficiency by allowing authorized clients to find the information they desire in a single round and achieves more fine-grained searches. Furthermore, we offer an inner product outsourcing calculation technique that allows the server to compute the inner product result without knowing the details of both parties in order to conceal the relevant privacy data of transmitting and decryption states. The paradigm can be utilized for efficient state transition through the use of parallel computing to accomplish the target of one round of iteration.

INDEX TERMS PEKS, forward security, inner product outsourcing, LWE assumptions, quantum-resist.

I. INTRODUCTION

The emergence of the IoT [29] has significantly brought about a global revolution in the information sector. It seeks to seamlessly combine the physical and digital worlds into an unified ecosystem, generating a new intelligent era of the Internet. Due to the expansion of the Internet and wireless access, the development of wearable devices, the decline in embedded computer prices, breakthroughs in storage technology, the IoT is rapidly developing. Smaller and smarter devices are being applied in a variety of IoT environments, including infrastructure monitoring [13], [19], personal healthcare [40], and autonomous cars [41], with each passing day.

The associate editor coordinating the review of this manuscript and approving it for publication was Shuangqing Wei.

However, data collected by IoT sensors may contain sensitive and private information. Leveraging the vulnerabilities in current IoT infrastructures, a number of security threats have surfaced. Therefore, how to use the data safely is a major concern for researchers. Cloud storage [17] is one of the most important IoT services, offering almost unlimited storage capacity and allowing users to remotely upload data to the cloud server. However, once the data is uploaded to the cloud server, the data owner will lose control of it. Since the cloud server may be malicious, outsourced data can be modified or corrupted, compromising the confidentiality, integrity and reliability of the data. The usual solution is to save the data in ciphertext to the cloud server, so effectively searching ciphertext and retrieving target data is a research hotspot in cloud data security.

Public-key Encryption with Keyword Search [9] is one of the IoT technologies that can address the above problems, enabling users to get the required files from an encrypted database. In PEKS, the data owner initially encrypts the data together with its keyword using the public key of the data receiver, then uploads the relevant ciphertexts to the cloud server. The data receiver uses its own private key to create a trapdoor for a certain keyword and transmits it to the server. The cloud server then checks whether the ciphertext of the keyword matches the user's trapdoor. However, most existing PEKS techniques are based on the cyclic group hidden subgroup hypothesis (HSP) [36], [37], [38], [39] and are not resistant to quantum computer attacks. Since Shor published the factorization technique in 1994 [11], they have been adversely affected. Therefore, it is becoming increasingly important to construct post-quantum-safe PEKS schemes.

Recently, breakthrough studies [2], [8], [10] have demonstrated that lattice-based assumptions can be utilized to develop quantum-resistant PEKS systems. Behnia et al. [10] proposed the most essential PEKS models, which were based on the NTRU lattice and LWE assumptions, respectively. Zhang et al. [2] presented an FS-PEKS technique to meet forward security. Since keywords have low entropy, most current PEKS schemes are sensitive to inside keyword guessing attacks (IKGA) from disruptive cloud servers. Zhang et al. [8] employs a dual-server architecture and provides an IKGA-resistant PO-IBEKS solution. However, in each forward security schemes mentioned above, multiple iterations must be used to retrieve the target data. As the number of keys updates increases, search performance will significantly decrease. Hence, how to improve search efficiency is a great challenge.

A. OUR CONTRIBUTIONS AND TECHNOLOGIES

To solve the aforementioned difficulties, we propose an Inner Product Outsourcing Public Key Searchable Encryption (IPO-PEKS) scheme from Lattice with forward security in the IoT, which provides a post-quantum secure promise and one round of iteration search. FIGURE 1 depicts the system model of our IPO-PEKS scheme, which consists of three roles: the sender, receiver and server. Specifically, the contributions of this work are described as follows.

- We propose a PEKS scheme with forward security that is lattice-based. The design is resistant to quantum attack. And all necessary files might well be searched in a single iteration and achieves more fine-grained searches. Since the cost of conversion is fixed, the search performance will be greatly increased as the number of secret key state updates climbs.
- We suggest a strategy for internal product outsourcing scheme. It enables the server to compute the inner product of both data owners without knowing each party's data, preserving both parties' privacy.
- We provide a comprehensive security proof for our scheme's IND-CKA security. Since our method can be successfully reduced to the LWE assumption, we can say that it is quantum resistant.

Technically, to achieve the objective of one iteration, We introduce a ciphertext conversion method, an inner product outsourcing calculation technique, which allows the server to compute the inner product result without knowing the details of both parties. That is, using this technique, all ciphertexts are changed to the ciphertext space that can be decrypted with the current secret key. While meeting high efficiency, it is also necessary to ensure that the ciphertext must be forward compatible in order to meet forward security. In other words, only ciphertexts in states older than the secret key can be effectively converted, but ciphertexts in other states cannot. To accomplish this target, we conduct matrix transformation on the sender's and receiver's state matrices. However, using them directly on the server would expose the participants' sensitive information, we should ensure that the transformation of the state matrix is completed without revealing the data. As depicted in FIGURE 2, Inner Product Function Encryption (IPFE) [1] is a technique for computing inner products that can also be used to compute matrix multiplications, but it is not appropriate for outsourced computing scenarios. Hence, we introduce an inner product outsourcing calculation approach to address the issue of the server achieving a safe switching of the states without disclosing information.

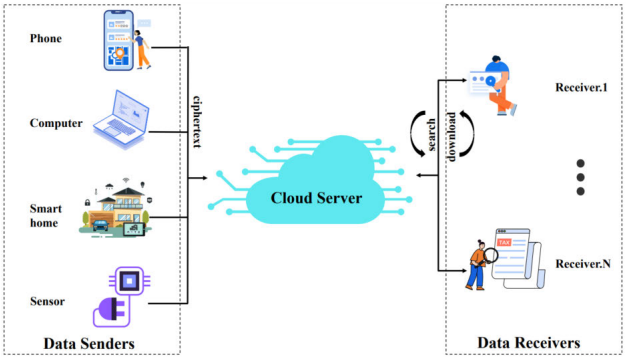

FIGURE 1. System model of IPO-PEKS for IoT.

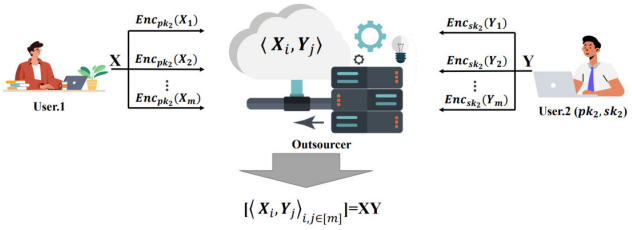

FIGURE 2. Model for matrix multiplication computation.

B. PAPER ORGNIZATION

The remainder of this paper is structured as follow. In Section II, we review recent related work. In Section III, we lay forth the fundamentals, definitions of notations, and security concepts required to construct our scheme. In Section IV, we propose the detailed IPOC and IPO-PEKS constructions as well as all parameter settings. In Section V, we present complete security proof. In Section VI, we give a comprehensive performance evaluation and finally conclude the paper in section 7.

II. RELATED WORK

The concept of searchable encryption (SE) was originally proposed to allow keyword searches over encrypted content. SE is categorized into two types: symmetric searchable encryption (SSE) and public-key encryption with keyword search (PEKS). In 2000, Song et al. [30] presented the first symmetric searchable encryption system. However, their system and other schemes [31], [32] based on it have the drawback that it is difficult to share one's own data with others, making it only suitable for the data holder to search through the ciphertexts. SSE cannot be widely adopted in a multi-user architecture because to its low scalability. Therefore, the first PEKS systems were proposed by Boneh et al. [9] in 2004, breaking the restriction on data sharing. In PEKS, a data owner leverages a receiver's public key to run the encryption algorithm, and the receiver creates a trapdoor to search a keyword over ciphertexts using its secret key.

The deployment of PEKS must be considered more comprehensively. According to Baek et al. [33], Boneh et al's model must construct a secure channel, which incurs significant communication overhead. They presented a channel-free PEKS technique to address this issue. Park et al. [34] and Golle et al. [35] presented public-key encryption with conjunctive keyword search (PECKS) systems to address the requirements of multi-keyword searches, which enable receivers to search for data which includes all of many keywords in a single task. ABEKS [36] (attribute-based encryption with keyword search) was suggested to enable fine-access control in PEKS. In 2016, Jiguo Li et al. [44] proposed a cryptographic scheme (KSFOABE) with keyword search function based on ABE. In 2020, Yang Lu et al. [45] provided a privacy-preserving and pairing-free multirecipient certificateless encryption with keyword search (CLKS) scheme for cloud-assisted industrial IoT. And in 2022, Yang Lu and Jiguo Li [47] proposed a lightweight PAEKS (public key authenticated encryption with keyword search) scheme which avoided the bilinear pairing operations. Similarly, there are other ABE-based cryptographic schemes [42], [46]. In 2022, Yuyan Guo et al. [43] also designed a revocable blockchain-aided ABE with escrow-free system. Other PEKS variants with novel features have been introduced [37], [38], [39]. They are all functional enhancements of the previous PEKS scheme.

Actually, the previously mentioned PEKS solution is sensitive to keyword guessing attacks initiated by a malicious system insider (IKGA), such as a misbehaving cloud server. And those schemes will be broken by quantum computers. To tackle those issues, Behnia et al. [10] presented PEKS methods based on the NTRU and LWE assumption. To resist IKGA, Zhang et al. [8] presented a PO-IBEKS system based on proxy. Gang Xu et al. [48] designed a post-quantum Public-key Searchable Encryption scheme (PPSEB) for E-healthcare scenarios which introduced blockchain technology to solve the problem of third-party untrustworthiness in the search process. Yang et al. [49] proposed a FS-IBEKS (forward secure identity-based encryption with keyword search) scheme from lattice. In 2023, Fan et al. [50] provided a lattice-based designated-server PAEKS (dPAEKS) scheme, which is quantum-resistant and can resist IKGA. In the same year, two other schemes were presented in the Cryptology ePrint Archive [51], [52]. The reference [52] provided a generic construction of forward-secure PAEKS primitive without trusted authorities, mitigating the secret key exposure while ensuring quantum-safe properties. And the reference [51] proposed lattice-based PEAKS, which permits the authority to authorize users to search different keyword sets while ensuring quantum-safe properties.

III. PRELIMINARIES

A. NOTATION

In the following, we employ some special notation in the paper. For any positive integer n, $[n]$ denotes a collection of $\{1, 2, \ldots, n\}$. We use bold lowercase letters to represent column vectors, e.g. $\mathbf{x}$, and uppercase letters to represent matrices, e.g. X. The *i-th* row and column of a matrix forms the row and column vector that are denoted by X_i^R and X_i^C. And we define S as a collection of column vectors $\{s_1, s_2, \ldots, s_i\}$, and $\|S\|$ as the largest L_2 normal form, i.e. $max_i\|s_i\|$. The Schmidt orthogonalized form of S is denoted by $\widetilde{S}$. $x \xleftarrow{\$} D$ means x is a random sample of D.

B. DISCRETE GAUSSIANS

Definition 1: Define L as a subset of $\mathbb{Z}^m$. We define

$$\rho_{\sigma,c}(x) = exp(-\pi \frac{\|x-c\|^2}{\sigma^2}) \quad and \quad \rho_{\sigma,c}(L) = \sum_{x\in L} \rho_{\sigma,c}(x)$$

for every vector c belonging to $\mathbb{R}^m$ and any integer parameter σ. A discrete Gaussian distribution on L with a mean of c and a standard deviation of σ is called

$$\forall y \in L: \quad \mathcal{D}_{L,\sigma,c}(y) = \frac{\rho_{\sigma,c}(y)}{\rho_{\sigma,c}(L)}$$

We typically ignore $\sigma = 1$ and $c = 0$ out of convenience; for instance, $\mathcal{D}_{L,\sigma,0}$ and $\mathcal{D}_{L,\sigma}$ are equal, as is $\mathcal{D}_{L,1,0}$ and $\mathcal{D}_L$.

C. LATTICE

One way to develop quantum-resistant methods with excellent computing efficiency is using lattice-based encryption. I will next go through some essential basic lattice theory.

Definition 2: For a large prime q, $A \in \mathbb{Z}_q^{n\times m}$ and a $u \in \mathbb{Z}_q^n$, we define:

$$\Lambda_q(A) := \{y \in \mathbb{Z}^m : \exists s \in \mathbb{Z}_q^n, y = A^\top s (mod\ q)\}$$
$$\Lambda_q^\perp(A) := \{x \in \mathbb{Z}^m : Ax = 0 (mod\ q)\}$$
$$\Lambda_q^u(A) := \{x \in \mathbb{Z}^m : Ax = u (mod\ q)\}$$

Definition 3 ([4]): Assign a large prime q, an integer $n > 0$, and a noise distribution χ over $\mathbb{Z}_q$. A $(\mathbb{Z}_q, n, \chi) - LWE$ problem instance consists of access to an unspecified challenge oracle $\mathcal{O}$,which is either a pseudo-randomly sampled oracle $\mathcal{O}_s$ with a fixed key $s \in \mathbb{Z}_q^n$, or a true random-sampling oracle $\mathcal{O}_\$$. The following are the two oracles:

$\mathcal{O}_s$: It outputs samples of the form $(a, b) = (a, a^\top s + e) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, where e is chosen at random from the distribution χ, a and s are both picked from a uniform distribution on $\mathbb{Z}_q$.

$\mathcal{O}_\$$: It delivers truly uniform random samples from $\mathbb{Z}_q^n \times \mathbb{Z}_q$.

The issues with $(\mathbb{Z}_q, n, \chi) - LWE$ can be repeatedly asked to the challenge oracle $\mathcal{O}$ for answers. We claim that an algorithm $\mathcal{A}$ determines the $(\mathbb{Z}_q, n, \chi) - LWE$ issue if $|Pr[\mathcal{A}^{\mathcal{O}_s} = 1] - Pr[\mathcal{A}^{\mathcal{O}_\$} = 1]|$ is non-negligible for a random $s \in \mathbb{Z}_q^n$.

Definition 4: Consider an $\alpha \in (0, 1)$ and a prime q, we define $\overline{\Psi}_\alpha$ as from the distribution of $\lfloor qX + \frac{1}{2} \rfloor mod\ q$ over $\mathbb{Z}_q$, where X is a normal random variable with mean 0 and standard deviation $\alpha/\sqrt{2\pi}$.

Theorem 1 ([4]): If there is an effective, potentially quantum algorithm for solving the $(\mathbb{Z}_q, n, \chi) - LWE$ issue for $q > 2\sqrt{n}/\alpha$, then, in the worst case, there is a quantum algorithm that is effective for approximating the SIVP and GapSVP problems to within $\widetilde{\mathcal{O}}(n/\alpha)$ factors in the l_2 norm.

Theorem 2: Given that $m := \lceil 6n \log q \rceil$ and q is an odd number greater than 3. Using a probabilistic polynomial-time algorithm called TrapGen(q, n, m), it is possible to produce a pair $(A \in \mathbb{Z}_q^{n\times m}, S \in \mathbb{Z}_q^{m\times m})$ where A is statistically near to a uniform matrix in $\mathbb{Z}_q^{n\times m}$ and S is a basis for $\Lambda_q^\perp(A)$ fulfilling

$$\|\widetilde{S}\| \le O(\sqrt{n \log q} \quad and \quad \|S\| \le O(n \log q)) \qquad (1)$$

with all but an extremely tiny risk in n.

After that, I'll discuss several lattice sampling techniques that will aid in the delegation of the lattice basis and the construction of certain one-way functions.

Lemma 1: Assume that $\sigma > \|\widetilde{T_A}\| \cdot \omega(\sqrt{log(m + \overline{m})})$ and $q > 2$, $m > 2n \log q$. Subsequently, the method SampleLeft$(A, \overline{M}, T_A, u, \sigma)$ generates a vector $e \in \mathbb{Z}^{m+\overline{m}}$ statistically distributed near $\mathcal{D}_{\Lambda_q^u(F_1),\sigma}$ where $F_1 := (\overline{M}|A)$, $\overline{M} \in \mathbb{Z}_q^{n\times\overline{m}}$, $u \in \mathbb{Z}_q^n$ and T_A is a short basis of A.

Lemma 2: Assume that q is prime and $m > (n + 1)\log_2 q + \omega(\log n)$. Assume that A, B are matrices chosen uniformly in $\mathbb{Z}_q^{n\times m}$, and that R is a $m \times m$ matrix selected uniformly in $\{1, 1\}^{m\times m}$ mod q. Consequently, the distribution $(A, AR, R^\top\omega)$ for all vectors ω in $\mathbb{Z}_q^m$ is statistically near to the distribution $(A, B, R^\top\omega)$.

Lemma 3: Basis $T_B \in \mathbb{Z}_q^{m\times m}$ with $B = AR^{-1} \in \mathbb{Z}_q^{n\times m}$ may be produced via the probabilistic polynomial method **BasisDel**(A, R, T_A, σ)*, where T_A is a short basis of A, $R \xleftarrow{R} D_{m\times m}$, $\sigma > \|\widetilde{T_A}\| \cdot \sigma_R\sqrt{m}\omega(\log^{3/2} m)$ and $\sigma_R := \sqrt{n \log q} \cdot \omega(\sqrt{\log m})$.*

D. IPFE

*Definition 5 ([1]): Several PPT algorithms (**Setup**, **KeyGen**, **Encrypt**, **Decrypt**) make up the functional encryption (FE) scheme on a set of functions called $\mathcal{F} := \{f : \mathcal{X} \to \mathcal{Z}\}$.*

Setup$(1^\lambda, \mathcal{F})$: The algorithm produces a master public key mpk and master secret key msk.

KeyGen(msk, f) : The algorithm generates a secret key sk_f when given the master secret key and a functionality $f \in \mathcal{F}$.

Encrypt(mpk, x) : The algorithm produces a ciphertext c after receiving the public key and a message x from the message space $\mathcal{X}$ as inputs.

Decrypt(mpk, sk_f, c) : The method produces z given a ciphertext and a secret key corresponding to some function $f \in \mathcal{F}$.

Correctness: We demand that for $(mpk, msk) \leftarrow Setup(1^\lambda, \mathcal{F})$, for all $x \in \mathcal{X}$, all $f \in \mathcal{F}$, for all state-related identity T_j, $c \leftarrow Encrypt(mpk, x, T_j)$ and $sk_f \leftarrow KeyGen(msk, f)$, with overwhelming probability, we have $Decrypt(mpk, sk_f, c) = f(x)$.

E. IPO-PEKS

I'll describe our solution with a definition in this portion. Our concept is built on a server that fairly enforces the protocol we provide, which is referred to as a server that is semi-honest. The scheme is described in the following style.

*Definition 6: Five polynomial-time algorithms are included in an IPO-PEKS system $\mathcal{E}$: **Setup**, **KeyDelegation**, **Derive**, **PEKS** and **Test**.*

Setup(1^λ) : The security parameter λ is inputted into the algorithm in probabilistic polynomial time (PPT), which produces the public parameter pp and the master secret key msk.

KeyDelegation(msk, pp) : A state-related key that is used to decrypt messages during this state period is built after entering the master private key, public parameters, and state-related identity.

Derive(sk, pp) : To determine whether the keyword search was successful, the algorithm inputs public parameters and secret keys and generates a token $t_{w:j}$ and sk_f.

PEKS(mpk, w, T_j) : The algorithm takes as inputs public parameters and the keyword w, giving a ciphertext Ct that is associated with both the keyword and the state-related identity T_j.

Test(Ct, sk_f, t_w) : The intention of this algorithm is to determine if the keyword matches the ciphertext. If they do, the algorithm returns 1, otherwise it returns 0.

Correctness: We demand that for $(mpk, msk) \leftarrow Setup(1^\lambda)$, for all $w \in \mathcal{W}$, for all T_j, $Ct_{w:j} \leftarrow$

$PEKS(mpk, w, T_j)$ and $sk_f, t_{w;j} \leftarrow Derive(sk, pp)$, with overwhelming probability, we have $Test(Ct_{w;j}, sk_f, t_{w;j}) = 1$.

F. SECURITY MODEL

In this part, we recall how existing IPO-PEKS schemes define security. Based on the security notion known as semantic security against chosen keyword attack (IND-CKA) proposed by Goh [53], we provide the ciphertext indistinguishability of our system. The complete system, including the public and private key pair of the sender and receiver and the public parameters, is established by Challenger $\mathcal{C}$. Adversary $\mathcal{A}$ issues a limited number of queries to Challenger $\mathcal{C}$.

We use a game to demonstrate the indistinguishability of ciphertext and the definition of IPO-PEKS under the selectively CKA, which entails semantic security, state period anonymity and keyword hiding. We consider d to be the longest state period. Here is a more detailed description:

Setup: The adversary $\mathcal{A}$ decides the attacked target's state duration $T_i^* = (t_1^*, \ldots, t_i^*), i < d$. And Challenger $\mathcal{C}$ develops system and gives adversary $\mathcal{A}$ access to public parameters.

Query I: The adversary $\mathcal{A}$ can query about the Trapdoor $t_{w,j}$ and the key SK_j of any state period T_j and any keyword w from Challenger $\mathcal{C}$, where $T_j = (t_1, \ldots, t_j)$ for some $j > i$.

Challenge: $\mathcal{A}$ chooses two unsearched keywords (w_0^*, w_1^*) and sends them to $\mathcal{C}$. In order to generate the ciphertext $CT \Longleftarrow PEKS(w_b^*, PK_i)$ and deliver it to $\mathcal{A}$, $\mathcal{C}$ chooses a random keyword w_b^* for a bit $b \in \{0, 1\}$ to be encrypted at the state T_i^*.

Query II: $\mathcal{A}$ continues *Phase I*'s issue, besides the fact that it is unable to query the related ciphertext that was created by w_0^* and w_1^*.

Guess: $\mathcal{A}$ produces a final guess, b'. Only if $b' = b$ does $\mathcal{A}$ win the game.

We refer to the advantage of $\mathcal{A}$ correctly distinguishing the ciphertexts of w_0^* and w_1^* at the state T_i^* as

$$Adv_{d,\mathcal{A}}^{\mathcal{E}}(\lambda) = |Pr[b' = b] - \frac{1}{2}|$$

Definition 7: A comprehensive IPO-PEKS system $\mathcal{E}$ is IND-CKA secure for depth d and selective-keyword, indistinguishable from random, if the function $Adv_{d,\mathcal{A}}^{\mathcal{E}}(\lambda)$ is negligible for each IND-CKA PPT adversary $\mathcal{A}$.

IV. SCHEMES

A. THE PROPOSED INNER PRODUCT OUTSOURCED COMPUTING

We achieve forward security by modifying the state information's ciphertext to display as follows:

$$F_i = A_0 R_i^{-1}$$
$$F_j = A_0 R_j^{-1}$$
$$\{F_i \to F_j\} = F_i R_i R_j^{-1} = F_j$$

The straight computation of $R_i R_j^{-1}$ will leak the private information of R_i and R_j since R_i and R_j respectively reflect the encryption state of the sender and the key generation state of the receiver. So we created a method to calculate $R_i R_j^{-1}$ securely. We will discuss their work in terms of the sender, recipient, and server responsibilities. Here, we'll suppose that the server determines the value of $\langle x, y \rangle$ when the sender's message is $x \in \mathbb{Z}_q^m$ and the receiver's message is $y \in \mathbb{Z}_q^m$. We need d to be as large as possible for Algorithm 1.

Algorithm 1 Recipient

SETUP(q, m)
- *select randomly two prime numbers* p, r
- $n = p \cdot q \cdot r$
- *select randomly generator* $g \in \mathbb{Z}_q$
- *select randomly vector* $s = (s_1, s_2, \ldots, s_m)$
- *compute* $\beta = (g^{s_1}, g^{s_2}, \ldots, g^{s_m})$
- *select randomly integer* e *st.* $gcd(e, \phi(n)) = 1$
- *compute* $ed \equiv 1 \mod \phi(n)$
- $pp = \{q, g, \beta, e\}$
- $msk = \{d, s, \phi(n)\}$
- *return* pp, msk

DERIVE(mpk, msk, y)
- *compute* $sk_y = \langle s, y \rangle$
- *compute* $sk_d = \{sk_{d,i} = d \cdot y_i \mod \phi(n)\}_{i \in [m]}$
- $sk = \{sk_y, sk_d\}$
- *return sk to sever*

The sender encrypts its own message x and transmits it to the server using the public parameters pp after the recipient establishes the public key system.

Algorithm 2 Sender

Encrypt(pp, x)
- *select randomly* $r \in \mathbb{Z}_q$
- *compute* $C_1 = g^r \mod q$
- *compute* $C_2 = \{C_{2,i} = (g^{x_i} \beta_i^r)^e \mod q\}_{i \in [m]}$
- $C = (C_1, C_2)$
- *return* C to **sever**

The server begins calculating the plaintext of $\langle x, y \rangle$ after receiving the sender's and receiver's ciphertexts.

Algorithm 3 Server

Evaluation(C, sk, pp)
- *compute* $D_1 = \prod_{i=1}^m (C_{2,i}^{sk_{d,i}} \mod q) \mod q$
- *compute* $D_2 = C_1^{sk_y} \mod q$
- $\langle x, y \rangle = \log_g \frac{D_1}{D_2}$
- *return* $\langle x, y \rangle$

Correctness: Note that $D_1 = (\prod_{i=1}^m C_{2,i}^{sk_{d,i}} \mod q) = (\prod_{i=1}^m (g^{x_i y_i} g^{r s_i y_i})^{ed} \mod q) = (\prod_{i=1}^m g^{x_i y_i} g^{r s_i y_i} \mod q) = (g^{\langle x,y \rangle} g^{r \langle s,y \rangle} \mod q)$ and $D_2 = (C_1^{sk_y} \mod q) = (g^{r \langle s,y \rangle}$

mod q), which implies $\frac{D_1}{D_2} = g^{\langle x,y\rangle}$. When $\langle x, y\rangle$ is small, we can determine its value by resolving the discrete logarithm issue. In order to recover the result of $\langle x, y\rangle$ by utilizing [5], we contract $|\langle x, y\rangle| \leq L$, where $L = poly(\lambda)$ is a bounded polynomial. It is feasible to constrain the sampled values by using the fact that the components of R_i are samples from the distribution $\mathcal{D}_{m\times m}$, which is a sample close to the origin.

Theorem 3: Our IPO system is secure if RSA is an effective method of encryption.

Proof: There are only two possible ways to disrupt our scheme. The value of $\phi(n) = \phi(p \cdot q \cdot r) = \phi(q) \cdot \phi(p \cdot r)$ is determined in one step, and the receiver's y and private key d are determined in the other step using the output of sk_d. Solving for the value of $\phi(p \cdot r)$ is still an RSA instance even though the value of $\phi(q)$ is known. This implies that $\phi(n)$ is secure even if the parameter n is exposed. Since the server doesn't know either y_i or d, they combine to creates a difficult factoring challenge for large integers over a group from the standpoint of $sk_{d,i} = y_i \cdot d \mod \phi(n)$. The most critical component is that the adversary cannot solve $\phi(n)$ since our variable n is private in the scheme. As a result, the security of our IPO scheme is dependent upon the security of the RSA scheme.

B. THE PROPOSED IPO-PEKS SCHEME

1) CONSTRUCTION

Let's first comprehend certain rules before we design the system. Two mapping functions, designated as H and H_1, are required. An FRD [3] code is H_1. H is a particular Hash function that works as follows:

$$H : (\{0, 1\}^*)^{\leq d} \to \mathbb{Z}_q^{m\times m} : t \mapsto H(t) \sim \mathcal{D}_{m\times m} \tag{2}$$

where the sampling on $\mathcal{D}_{m\times m}$ [6] is $\mathbb{Z}_q$-invertible, and d is the greatest temporal depth. A group of state series is used to represent each unique state identity $T_i = (t_1, t_2, \ldots, t_i)$ where $i < d$. The keyword $w \in \mathcal{W}$ and state T_i are represented by the following encoding:

$$\begin{aligned} F_{w;i} &:= (A_1 + H_1(w) \cdot B | A_0 R_i^{-1}) \\ F_i &:= A_0 R_i^{-1} \\ F_w &:= A_1 + H_1(w) \cdot B \end{aligned} \tag{3}$$

where $R_i = H(t_i)H(t_{i-1}) \cdots H(t_1)$. Our current IPO-PEKS scheme is operated as follows:

Setup: The method takes the security parameter λ as an input, sets the parameters q, n, m, σ, τ and α to predetermined values, and then outputs the master secret key msk and the public parameters pp. Algorithm 4 provides the specific operation.

The receiver runs Algorithm 4 and uses the public parameters to represent their own identity. The sender transmits data using these public parameters.

KeyDelegation: We link the key created by the master key to the state identity to accomplish the goal of forward security. As a result, each key can only be used to decrypt the ciphertext of a certain state identity period. As demonstrated in Algorithm 5, input the public parameters pp, the secret key sk_i under the state series $T_i = (t_1, t_2, \ldots, t_i)$ and the present state series $T_j = (t_1, t_2, \ldots, t_j)$, and output the secret key sk_j under the state series T_j.

Algorithm 4 Setup

SETUP(n, m, q, λ)
- *select randomly two prime numbers* p, r
- $\widehat{n} = p \cdot q \cdot r$
- *select randomly integer* e *st.* $gcd(e, \phi(\widehat{n})) = 1$
- *compute* $ed \equiv 1 \mod \phi(\widehat{n})$
- $(A_0 \in \mathbb{Z}_q^{n\times m}, T_{A_0} \in \mathbb{Z}_q^{m\times m}) \leftarrow TrapGen(n, q)$
- $\mathbf{u} \xleftarrow{\$} \mathbb{Z}_q^n$, $(A_1, B) \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$
- *select randomly generator* $g \in \mathbb{Z}_q$
- $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_q^m$, *compute* $\beta := g^{\mathbf{s}} = (g^{s_1}, g^{s_2}, \ldots, g^{s_m})$
- $msk = \{T_{A_0}, \mathbf{s}, d, \phi(\widehat{n})\}$
- $pp = \{A_0, A_1, B, \mathbf{u}, q, g, \beta, e\}$
- *return* $\{pp; msk\}$

Algorithm 5 KeyDelegation

KEYDELEGATION(pp, sk_i, T_i, T_j)
- $R_i = H(t_i)H(t_{i-1}) \cdots H(t_1) \in \mathbb{Z}^{m\times m}$
- $F_i = A_0 R_i^{-1}$
- $R_{j-i} = H(t_j)H(t_{j-1}) \cdots H(t_{i+1})$, $F_j = F_i R_{j-i}^{-1}$
- $sk_j \leftarrow BasisDel(F_i, R, sk_i, \tau_j)$
- *if* R_i *is None* :
  - $F_0 = A_0$, $sk_0 = T_{A_0}$
  - KEYDELEGATION(pp, sk_0, T_0, T_j)
- *return* sk_j

The required ciphertext sent to the receiver within the state T_j may be decrypted using the secret key sk_j associated with T_j that is generated by the receiver using Algorithm 5. The following public and private key pairs will be generated through delegation, thus the secret key cannot be shared and can only be accessed by the holder of the master private key.

Derive. The receiver needs to produce a special trapdoor rather than the original key in order to determine if the ciphertext with the keyword on the server includes the desired keyword. Input the public parameters pp, secret key sk_j, keywords $w \in \mathcal{W}$, and state series T_j as indicated in Algorithm 6, then output the trapdoor needed for the **Test** phase.

PEKS: By using the recipient's public key to encrypt his own data, the sender uploads it to the server so that the recipient may access it through a trapdoor and recover the data he need. The public parameters, keywords, and state series are fed into algorithm 7; it then outputs the ciphertext of the keyword with the sending state. Here, we employ a common hashing algorithm, designated as H_2.

Algorithm 6 Derive

DERIVE(pp, sk_j, T_j, w)
$R_j = H(t_j)H(t_{j-1})\cdots H(t_1) \in \mathbb{Z}_q^{m\times m}$
$F_j = A_0 R_j^{-1} \in \mathbb{Z}_q^{n\times m}$
$F_w = A_1 + H_1(w)\cdot B \in \mathbb{Z}_q^{n\times m}$
$F_{w;j} = (F_w|F_j) \in \mathbb{Z}_q^{n\times 2m}$
$\mathbf{e} \in \mathbb{Z}_q^{2m} \leftarrow SampleLeft(F_j, F_w, sk_j, \mathbf{u}, \sigma_j)$
$st.\ F_{w;j}\cdot \mathbf{e} = \mathbf{u}\ mod\ q$
$sk_1 = ENCTIME(pp, msk, R_j^{-1})$
$sk_2 = ENCTIME(pp, msk, R_j)$
$sk_3 = ENCTIME(pp, msk, R_jR_jR_j)$
$sk = \{sk_1, sk_2, sk_3, \mathbf{e}\}$
return sk

ENCTIME(pp, msk, R)
for i *in* $[m]$:
sk_i=Recipient.DERIVE(pp, msk, R_i^{row})
return $\{sk_i\}_{i\in[m]}$

Algorithm 7 PEKS

PEKS(pp, w, T_i)
$R_i = H(t_i)H(t_{i-1})\cdots H(t_1) \in \mathbb{Z}_q^{m\times m}$
$F_i = A_0 R_i^{-1} \in \mathbb{Z}_q^{n\times m}$
$F_w = A_1 + H_1(w)\cdot B \in \mathbb{Z}_q^{n\times m}$
select randomly vector $\xi \leftarrow \{0,1\}^l$ (l *is large enough*)
select randomly vectors $\mathbf{x} \xleftarrow{\Psi_\alpha} \mathbb{Z}_q^l$ and $\mathbf{y} \xleftarrow{\Psi_\alpha} \mathbb{Z}_q^m$
select randomly $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_q^n$
comput $C_0 = \{C_{0,k} = \mathbf{u}^\top\mathbf{r} + x_k + \xi_k\lfloor\frac{q}{2}\rfloor\}_{k\in[l]}$
select randomly $R \leftarrow \{-1,1\}^{m\times m}$
compute $\mathbf{z} = R^\top\mathbf{y} \in \mathbb{Z}_q^m$
$C_{1,1} = F_w^\top\mathbf{r} + \mathbf{z}$
$C_{1,2} = F_i^\top\mathbf{r} + \mathbf{y}$
$C_2 = ENCRYPT(pp, R_i) \quad C_3 = H_2(\xi)$
return $C = \{C_0, C_1, C_2, C_3\}$

ENCRYPT(pp, $\mathbf{R}$)
for i *in* $[m]$
$ct_{r_i} = Sender.ENCRYPT(pp, \mathbf{R}_i^{row})$
return $\{ct_{r_i}\}_{i\in[m]}$

Please take note that the sending state T_i in this case may also utilize future state, allowing for greater flexibility than the original function by delivering the message to the recipient at a future state as opposed to immediately.

Test: After receiving the sender and receiver's ciphertexts, the server begins looking for keywords. To compute all matching results in a single iteration, we transform the sender's ciphertext into a state series that the receiver can decrypt. But because of the limitations of forward security, we must enforce some rules on the transformation, as shown in Algorithm 8. The matched outcome of the keyword search is produced after entering the public parameters, the receiver's trapdoor, and the sender's ciphertext.

Algorithm 8 Test

TEST(pp, sk, C)
$M_1 = R_iR_j^{-1} = MATRIXEVAL(C_2, sk_1, pp)$
$M_2 = R_iR_j = MATRIXEVAL(C_2, sk_2, pp)$
$M_3 = R_iR_j^3 = MATRIXEVAL(C_2, sk_3, pp)$
$C_{1,2} = \lceil\frac{|M_2|^2}{|M_3|}\rceil M_1^\top C_{1,2}$
$C_1 = (C_{1,1}|C_{1,2})$
select vector $\xi = \{0\}^l$
for k *in* $[l]$:
compute $\gamma \leftarrow C_{0,k} - e^\top C_1$
if $|\gamma - \lfloor\frac{q}{2}\rfloor| < \lfloor\frac{q}{4}\rfloor$ *then* $\xi_k = 1$
else $\xi_k = 0$
if $C_3 == H_2(\xi)$ *then return* 1
else return 0

MATRIXEVAL(Ct_{R_i}, sk_{R_j}, pp)
for x *in* $[m]$:
for y *in* $[m]$:
$\langle R_{i,x}, R_{j,y}\rangle$=Server.EVALUATION($Ct_{R_{i,x}}$, $sk_{R_{j,y}}$, pp)
return R_iR_j ($R_iR_j = [\langle R_{i,x}, R_{j,y}\rangle_{x,y}]$)

2) CORRECTNESS

We configure the proper global parameters q, m, σ_j, α_j and τ_j for $j \le d$ to perform the LWE decryption correctly. We must first determine the error term:

$$\gamma = C_{0,k} - e^\top C_1 = \xi_k\lfloor\frac{q}{2}\rfloor + \underbrace{x_k - e^\top\begin{bmatrix}M_1^\top z\\ y\end{bmatrix}}_{error\ term}$$

As a result, we must determine how the parameters relate to one another in order to constrain the error term to be inside $q/4$. Here, e is divided into (e_2, e_1). e_1 and e_2 are equal, as you can see.

$$\begin{aligned}|error\ term| &= |x_k - e^\top\begin{bmatrix}M_1^\top z\\ y\end{bmatrix}|\\ &\le |x_k| + |e_1^\top y| + |(RM_1e_2)^\top y|\end{aligned}$$

([6]Lemma 3.) and ([3] Lemma 5.) show that we have

$$\begin{aligned}|x_k| &\le q\alpha_j\omega(\sqrt{\log m}) + 1/2\\ |e_1^\top y| &\le \|e_1\|(q\alpha_j\omega(\sqrt{\log m}) + \sqrt{m}/2)\\ |(RM_1e_2)^\top y| &\le \|RM_1e_2\|(q\alpha_j\omega(\sqrt{\log m}) + \sqrt{m}/2)\end{aligned}$$

We determine $\|e_1\|$ and $\|RM_1e_2\|$'s values individually.

$$\|e_1\| < \sigma_j\cdot\omega(\sqrt{2m})$$

The Leftover Hash Lemma's corollary([7], Lemma 2.10) gives us the following conclusion:

$$\begin{aligned}\|RM_1e_2\| &\le \|M_1e2\|\sqrt{m}\omega(\sqrt{\log m})\\ &\le \sigma_j(m\cdot\omega\sqrt{\log m})^{j+1}\end{aligned}$$

the error term is less than $q/5$ w.h.p.

$$|e_1^\top y| \le q/5$$
$$|(RM_1e_2)^\top y| \le q/5$$

and

$$q > 2\sqrt{n}/\alpha$$

Thus, in conclusion

$$\begin{aligned} m &= 6n\log q \\ \alpha_j &= [\sigma_j m^{j+1}\omega(\sqrt{\log m})^{j+2}\ ]^{-1} \\ \sigma_j &= m^{j+3/2}\ \omega(\log m)^{j+1} \\ q &= 2\sqrt{n}\ m^{2d+5/2}\omega(\log m)^{3d/2+2} \\ \tau_j &= m^{3j/2+1/2}\omega(\log n)^{2j} \end{aligned} \tag{4}$$

All parameters have now been correctly configured. With these settings, the error has been made to fall inside the range smaller than $q/5$, allowing for successful decryption.

3) SECURITY

We demonstrate that under a selective keyword attack identical to Definition 7, the essential IPO-PEKS structure cannot be distinguished from random. Remember that the phrase ''cannot be distinguished from random'' refers to how difficult it would be to recognize the challenge ciphertext apart from a random element in the ciphertext space. This property implies receiver anonymity as well as semantic security.

Theorem 4: If the $(\mathbb{Z}_q, n, \overline{\Psi}_\alpha) - LWE$ *assumption is hard, the IPO-PEKS system with parameters* $(q, m, \alpha, \sigma, \tau)$ *as in (4) is IND-CKA secure.*

Proof: To demonstrate that the original IPO-PEKS scheme in Definition 7 is IND-CKA secure, we will play a sequenceÂ of games. The original solution is discovered in the first game, and an impossibility is found in the last game. By utilizing the LWE assumption and a sequence of identical games to reduce the original scheme to the final game, we demonstrate the security of our system. Here, we'll suppose that the challenge keyword is w^* and the state depth of the generated ciphertext is j.

*Game*0 : This is the initial IND-CKA game from Definition 7, which puts an attacker $\mathcal{A}$ opposing our scheme against an challengerÂ $\mathcal{C}$ with IND-CKA.

*Game*1 : In Game 0, the challenger $\mathcal{C}$ produces the public parameter pp by choosing three random matrices, A_0, A_1*and* B, as well as a trapdoor, sk_j is the basis of $\Lambda_q^\perp(\overline{A})$, where $\overline{A} = A_0R_j^{-1}$. The challenger $\mathcal{C}$ chooses a random matrix $R^* \xleftarrow{\$} \{-1, 1\}^{m\times m}$ to create the challenge ciphertext CT^* during the challenge phase.

In Game 1, we perform a little adjustment to the public parameters pp that create A_1. The challenger $\mathcal{C}$ in Game 1 will construct A_1 in the manner described below using the aforementioned challenge keyword w^* and random matrix R^*.

$$A_1 := \overline{A}R^* - H_1(w^*)\cdot B \tag{5}$$

The remaining components are unchanged.

We will use the Leftover Hash Lemma([3], Lemma 4.) to demonstrate that Game 0 and Game 1 are identical. The Leftover Hash Lemma enables us to determine that $\overline{A}$ and $\overline{A}R^*$ cannot be distinguished in $\mathbb{Z}_q^{n\times m}$ by statistical distance. A_1 in Games 0 and 1 are therefore indistinguishable from the perspective of the adversary $\mathcal{A}$.

*Game*2 : We altered the method A_0 is generated in Game 2. As a variant to the Leftover Hash Lemma ([7], Lemma 2.10), we build A_0 as follows.

$$\overline{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times m} \quad and \quad A_0 = \overline{A}R_j$$

By statistical distance, [7] demonstrated that $U = \overline{A}R(R \leftarrow \mathcal{D}^{m\times m})$ and $U' \xleftarrow{\$} \mathbb{Z}_q^{n\times m}$ cannot be distinguished. Consequently, A_0 cannot be distinguished from Game 1 or Game 2 from the adversary's perspective.

*Game*3 : To address the adversary's Token query in Game 3, we will modify how B is generated. We build matrix B using the **TrapGen** technique, which is identical to random sampling on $\mathbb{Z}_q^{n\times m}$, and the challenger also has a lattice base T_B on $\Lambda_q^\perp(B)$ lattice space. To respond to an adversary's keyword query, the basis can create a token. The remainder is unchanged from Game 2 and is the same.

We produce the token e with the keyword $w \ne w^*$ using the **SampleLeft** ([3], Section IV-B) method when the adversary $\mathcal{A}$ begins the keyword token query.

$$\begin{aligned} F_{w;j} &= (A_1 + H_1(w)\cdot B|\overline{A}) \\ &= (\overline{A}R^* + (H_1(w) - H_1(w^*))\cdot B|\overline{A}) \end{aligned}$$

Because H_1 is an **FRD** code, T_B is also the lattice base of $\Lambda_q^\perp((H_1(w) - H_1(w^*))\cdot B)$. We can obtain the token of the search keyword by running the SampleLeft algorithm.

$$e \leftarrow SampleLeft(\overline{A}, ((H_1(w) - H_1(w^*))\cdot B, R^*, T_B, u, \sigma)$$

To the adversary $\mathcal{A}$, we transmit $sk_{w;j} = e$. Game3 and Game2 are identical because the distribution of e is near to $\mathcal{D}_{\Lambda_q^\perp(F_id),\sigma}$.

*Game*4 : With the exception of how the ciphertext is produced, Game 4 and Game 3 are identical. The $\mathbb{Z}_q^l \times \mathbb{Z}_q^{2m}$ space is randomly sampled to create the ciphertext $CT = (CT_0, CT_1)$ that Game 4 produces. As a result, the adversary's advantage in this game is zero, and the challenge ciphertext is always a random element in the ciphertext space.

Finally, to demonstrate that the scheme is IND-CKA secure, it just has to demonstrate that Games 3 and 4 are indistinguishable to PPT adversaries. Next, we need reduce this challenge to the LWE assumption.

a: REDUCTION FROM LWE

The indistinguishability of 1 bit of encryption only has to be demonstrated. We can next apply the generalization to the situation of encrypting l bits. We can construct Games 3 and 4 using these instances in accordance with the definition of LWE instances, which reduces to the LWE problem we require.

b: SETUP

1) Adversary $\mathcal{A}$ decides to attack the keyword w^*. Simulator $\mathcal{B}$ samples $m+1$ LWE instances $(u_i, v_i)_{i\in[m+1]} \in \mathbb{Z}_q^n \times \mathbb{Z}_q$.
2) m LWE instances are chosen by $\mathcal{B}$ to build matrix $\overline{A} = (u_1, u_2, \ldots, u_m)$.
3) $\mathcal{B}$ chooses public parameter $u = u_0$.
4) The other public parameters are identical to those in Game 3.

c: QUERIES

$\mathcal{B}$ responds to $\mathcal{A}$'s token query in a similar way as Game 3.

d: CHALLENGE

$\mathcal{B}$ chooses a random bit $\xi_i \in \{0, 1\}$. The generating process of ciphertext of the target keyword w^* is as follows

1) Compute $C^*_{0;i} = v_0 + \xi_i \lfloor \frac{q}{2} \rfloor = u_0^\top s + x_i + \xi_i \lfloor \frac{q}{2} \rfloor$.
2) Set $v^* = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_m \end{bmatrix} \in \mathbb{Z}_q^m$
3) Set $C_1^* = \begin{bmatrix} (R^*)^\top v^* \\ v^* \end{bmatrix}$
4) $\mathcal{B}$ picks a random $r \xleftarrow{\$} \{0, 1\}$. Depending on the value of r, it sends the corresponding ciphertext CT_{w^*} to the adversary $\mathcal{A}$, where (C_0, C_1) is random element of the ciphertext space $\mathbb{Z}_q \times \mathbb{Z}_q^{2m}$.

$$CT_{w^*} = \begin{cases} (C^*_{0;i}, C_1^*), & \text{if } r = 0 \\ (C_0, C_1), & \text{otherwise.} \end{cases}$$

We properly simplify C_1^*. We get

$$\begin{aligned} F_{w^*;j} &= (\overline{A}R^* + (H_1(w^*) - H_1(w^*)) \cdot B|\overline{A}) \\ &= (\overline{A}R^*|\overline{A}) \end{aligned}$$

for $w = w^*$. Then

$$\begin{aligned} C_1^* &= \begin{bmatrix} (R^*)^\top v^* \\ v^* \end{bmatrix} = \begin{bmatrix} (R^*)^\top (\overline{A}^\top s + y) \\ \overline{A}^\top s + y \end{bmatrix} \\ &= \begin{bmatrix} (\overline{A}R^*)^\top s + (R^*)^\top y \\ \overline{A}^\top s + y \end{bmatrix} \\ &= F_{w^*;j}^\top \cdot s + \begin{bmatrix} (R^*)^\top y \\ y \end{bmatrix} \end{aligned}$$

Accordingly, $(C^*_{0,i}, C_1^*)$ and (C_0, C_1) are the valid ciphertexts for Games 3 and 4, respectively.

e: GUESS

$\mathcal{B}$ uses $\mathcal{A}$'s estimate as the answer after receiving the challenge ciphertext CT_{w^*} to distinguish LWE instances from random elements.

We already showed that the adversary's perspective is identical to Game 3 when $\mathcal{O} = \mathcal{O}_s$. The adversary's perspective is the same as in Game 4 when $\mathcal{O} = \mathcal{O}_\$$. As a result, $\mathcal{B}$'s advantage in solving LWE is equivalent to $\mathcal{A}$'s advantage in correctly differentiating Games 3 and 4. This completes our proof for the IND-CKA security of the scheme.

V. PERFORMANCE

In this section, we first evaluate the performance of IPO-PEKS against that of existing PEKS schemes [2], [8], [9], [10], including communication and computational overhead. All trials are performed out on a laptop running Windows 10 with an Intel Core i7 CPU and 16 GB DDR 4 RAM. We are using Python 3.8 and the gmpy2 library. All parameters have been set based on the previous analysis. Then, the IPO-PEKS scheme's performance is comprehensively evaluated.

Communication and computational overhead are two critical factors that we must take into account in order to deploy IPO-PEKS into the IoT efficiently. Given that we think the client portion has little processing power and storage, we need reduced communication costs as well as lower computational costs, or we should shift as many computing jobs as we can to the server. Our strategy is to assign as many computing work to servers as we can. I will now contrast the performance with a few already existing PEKS implementations [2], [8], [9], [10]. Let $|\mathbb{G}_1|, B_q, B_n$ represent the size of an element on groups $\mathbb{G}_1, \mathbb{Z}_q, \mathbb{Z}_n$. Let $T_{mu}, T_{Pa}, T_{Ex}, T_{sp}$ *and* T_{ha} indicate respectively multiplication time, bilinear pairing time, exponential computation time, SamplePre computation time and hash computation time. The detailed runtime overhead of these fundamental operations is presented in TABLE 3.

TABLE 1 displays the detailed communication overhead for the five systems mentioned above. The two systems in [2], [8] achieve forward security, as can be shown, but a single test corresponds to all ciphertext related to a certain keyword, and cannot achieve more fine-grained searches. [9], [10] do not meet the requirements for forward security, and they also did not consider fine-grained searches. In fact, the state related to a keyword ciphertext is not just limited to the user's ID, in order to achieve fine-grained search, we can add more features to distinguish the same keyword in different states. Although the proposed IPO-PEKS has a large communication overhead, it corresponds to the communication overhead of a keyword with j features, while other schemes all correspond to the overhead of one feature. It simultaneously meets fine-grained search and forward security. For a fair comparison, we assume that each keyword has j feature states. We roughly assume that the computational overhead associated with the schemes compared is j times that of the

TABLE 1. Communication overhead.

Schemes	PEKS size	Trapdoor size	States	Forward Security
FS-PEKS [2]	$(l+ml)B_q$	mB_q	1	TRUE
PO-IBEKS [8]	$(ml+l+m)B_q$	mB_q	1	TRUE
BCOP-PEKS [9]	$(\lvert\mathbb{G}_1\rvert+l)$	$\lvert\mathbb{G}_1\rvert$	1	FALSE
BOOY-PEKS [10]	$3mB_q$	mB_q	1	FALSE
Ours	$(2m+l+nm+n)B_q$	$(3m+1)nB_n+2mB_n$	j	TRUE

TABLE 2. Computational overhead.

Schemes	PEKS computing time	Test computing time
FS-PEKS [2]	$j[T_{ha}+(nl+nm^2+nml)T_{mu}]$	$jmlT_{mu}$
PO-IBEKS [8]	$j[(nm^2+nml+nl)T_{mu}+2T_{ha}+T_{sp}]$	$j[(nm+ml)T_{mu}+T_{ha}]$
BCOP-PEKS [9]	$j[T_{pa}+2T_{Ex}+2T_{ha}]$	$j[T_{pa}+T_{ha}]$
BOOY-PEKS [10]	$j[2m^2T_{mu}+T_{ha}]$	$j[m^2T_{mu}+T_{ha}]$
Ours	$n(l+3m)T_{mu}+n(2m+1)T_{Ex}+T_{ha}$	$3m^2(m+1)T_{Ex}+(m^3+m^2+2ml)T_{mu}+T_{ha}$

TABLE 3. Runtime of some operations.

T_{mu}	T_{pa}	T_{Ex}	T_{sp}	T_{ha}
0.0004 ms	1.8526 ms	0.0005 ms	35.1360 s	0.0018 ms

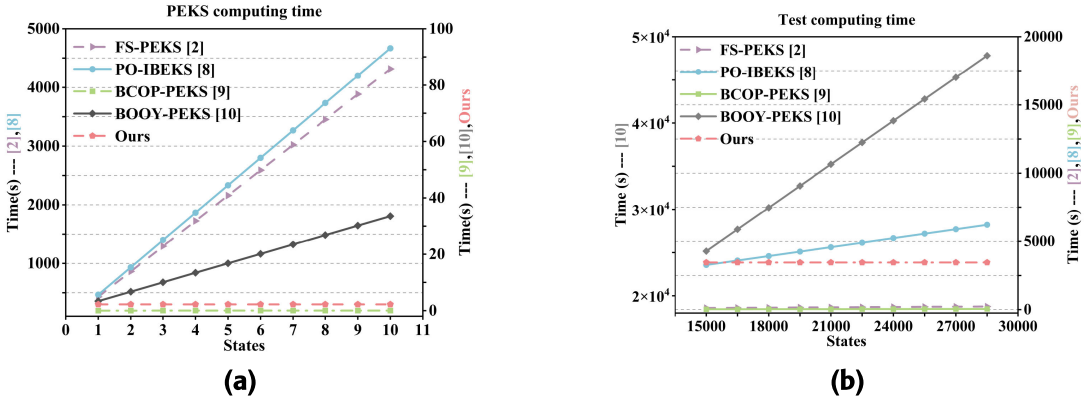

FIGURE 3. The correlation between the quantity of feature states j and the consumption time in two different steps of various schemes.

original algorithms, as their original schemes can only handle one feature state at a time. The results are shown in TABLE 2.

To demonstrate a more intuitive comparison, experimentally utilised security parameter $\lambda = 256$, $l = 10$, FIGURE 3 shows the changes in computational overhead of the PEKS algorithm and the Test algorithm as the number of feature states j increases. In FIGURE 3 (a), compared to other schemes with parameters at the same level, our PEKS algorithm has the lower computational overhead, and it is independent of the number of feature states j. From the perspective of the Internet of Things, the clients require lower computational and communication overhead. Obviously, the proposed PEKS algorithm has a significant advantage in computational cost compared to other schemes. In FIGURE 3 (b), compared to other schemes with parameters at the same level, our Test algorithm also has the lower computational overhead when the number of feature states j increases to a larger value, and it is independent of the number of feature states j. Although it has a slight disadvantage in computational overhead when the number of feature states j is relatively small, we have achieved lower overhead on the client side. Considering the whole search procedure, our IPO-PEKS is therefore suitable for the IoT context.

VI. CONCLUSION

In this paper, we provide a practical and efficient public key searchable encryption mechanism (IPO-PEKS) based on lattice, which has forward security and is resistant to quantum attacks. The presented IPO-PEKS is provable secure and has a low client-side resource requirement. At the same time, we introduce an inner product outsourcing calculation approach, which allows the server to determine the inner product value without knowing the details of both sides. With the use of parallel computing, the paradigm can very efficiently complete the transformation of the ciphertext state, achieving the goal of one round of iterative search for j feature states. When compared to existing PEKS methods, our suggested technique is more computationally efficient and

achieves more fine-grained searches. To further strengthen the security of the system, we will also take into account backward security in the future work.

REFERENCES

[1] S. Agrawal, B. Libert, M. Maitra, and R. Titiu, "Adaptive simulation security for inner product functional encryption," in *Proc. IACR Int. Conf. Public-Key Cryptogr.*, Springer, 2020, pp. 34–64.
[2] 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 Secure Comput.*, vol. 18, no. 3, pp. 1019–1032, May 2021.
[3] S. Agrawal, D. Boneh, and X. Boyen, "Efficient lattice(H)IBE in the standard model," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* Springer, 2010, pp. 553–572.
[4] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *J. ACM*, vol. 56, no. 6, pp. 1–40, Sep. 2009.
[5] J. M. Pollard, "Kangaroos, monopoly and discrete logarithms," *J. Cryptol.*, vol. 13, no. 4, pp. 437–447, Sep. 2000.
[6] S. Agrawal, D. Boneh, and X. Boyen, "Lattice basis delegation in fixed dimension and shorter-ciphertext hierarchical IBE," in *Proc. Annu. Cryptol. Conf.* Berlin, Germany: Springer, 2010, pp. 98–115.
[7] P. Ananth, X. Fan, and E. Shi, "Towards attribute-based encryption for rams from LWE: Sub-linear decryption, and more," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, Cham, Switzerland: Springer, 2019, pp. 112–141.
[8] 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. 193–207, Aug. 2019.
[9] D. Boneh, G. Di. Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.* Cham, Switzerland: Springer, 2004, pp. 506–522.
[10] 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, Nov. 2020.
[11] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proc. 35th Annu. Symp. Found. Comput. Sci.*, 1994, pp. 124–134.
[12] J. T. J. Penttinen, "Internet of Things," in *Wireless Communications Security: Solutions for the Internet of Things*. Cham, Switzerland: Springer, 2015, pp. 112–149.
[13] F. Karagulian, M. Barbiere, A. Kotsev, L. Spinelle, M. Gerboles, F. Lagler, N. Redon, S. Crunaire, and A. Borowiak, "Review of the performance of low-cost sensors for air quality monitoring," *Atmosphere*, vol. 10, no. 9, p. 506, Aug. 2019.
[14] B. O'Flynn, R. Martinez-Catala, S. Harte, C. O'Mathuna, J. Cleary, C. Slater, F. Regan, D. Diamond, and H. Murphy, "SmartCoast: A wireless sensor network for water quality monitoring," in *Proc. 32nd IEEE Conf. Local Comput. Netw. (LCN)*, Oct. 2007, pp. 815–816.
[15] A. Adapa, F. F.-H. Nah, R. H. Hall, K. Siau, and S. N. Smith, "Factors influencing the adoption of smart wearable devices," *Int. J. Hum.–Comput. Interact.*, vol. 34, no. 5, pp. 399–409, May 2018.
[16] J. Fan, F. Han, and H. Liu, "Challenges of big data analysis," in *National Science Review*, vol. 1. London, U.K.: Oxford Univ. Press, 2014, pp. 293–314.
[17] T. Dillon, C. Wu, and E. Chang, "Cloud computing: Issues and challenges," in *Proc. 24th IEEE Int. Conf. Adv. Inf. Netw. Appl.*, Apr. 2010, pp. 27–33.
[18] F. Yang, S. Wang, J. Li, Z. Liu, and Q. Sun, "An overview of Internet of Vehicles," *China Commun.*, vol. 11, no. 10, pp. 1–15, Oct. 2014.
[19] G. Chugh, D. Bansal, and S. Sofat, "Road condition detection using smartphone sensors: A survey," *Int. J. Electron. Electr. Eng.*, vol. 7, no. 6, pp. 595–602, 2014.
[20] D. Valtchev and I. Frankov, "Service gateway architecture for a smart home," *IEEE Commun. Mag.*, vol. 40, no. 4, pp. 126–132, Apr. 2002.
[21] M. Ghamari, B. Janko, R. Sherratt, W. Harwin, R. Piechockic, and C. Soltanpur, "A survey on wireless body area networks for eHealthcare systems in residential environments," *Sensors*, vol. 16, no. 6, p. 831, Jun. 2016.
[22] I.-R. Chen, F. Bao, and J. Guo, "Trust-based service management for social Internet of Things systems," *IEEE Trans. Dependable Secure Comput.*, vol. 13, no. 6, pp. 684–696, Nov. 2016.
[23] U. Chatterjee, V. Govindan, R. Sadhukhan, D. Mukhopadhyay, R. S. Chakraborty, D. Mahata, and M. M. Prabhu, "Building PUF based authentication and key exchange protocol for IoT without explicit CRPs in verifier database," *IEEE Trans. Dependable Secur. Comput.*, vol. 16, no. 3, pp. 424–437, May 2019.
[24] Y. Li, W. Susilo, G. Yang, Y. Yu, D. Liu, X. Du, and M. Guizani, "A blockchain-based self-tallying voting protocol in decentralized IoT," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 1, pp. 119–130, Jan. 2022.
[25] M. Wazid, A. K. Das, V. Odelu, N. Kumar, and W. Susilo, "Secure remote user authenticated key establishment protocol for smart home environment," *IEEE Trans. Dependable Secur. Comput.*, vol. 17, no. 2, pp. 391–406, Mar. 2020.
[26] G. S. Poh, P. Gope, and J. Ning, "PrivHome: Privacy-preserving authenticated communication in smart home environment," *IEEE Trans. Dependable Secur. Comput.*, vol. 18, no. 3, pp. 1095–1107, May 2021.
[27] Y. Zhang, J. Yu, R. Hao, C. Wang, and K. Ren, "Enabling efficient user revocation in identity-based cloud storage auditing for shared big data," *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 3, pp. 608–619, May 2020.
[28] S. Aditham and N. Ranganathan, "A system architecture for the detection of insider attacks in big data systems," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 6, pp. 974–987, Nov. 2018.
[29] K. Ashton, "That 'Internet of Things' thing," *RFID J.*, vol. 22, pp. 97–114, Jun. 2009.
[30] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proc. IEEE Symp. Secur. Privacy*, May 2000, pp. 44–55.
[31] C. Wang, N. Cao, K. Ren, and W. Lou, "Enabling secure and efficient ranked keyword search over outsourced cloud data," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 8, pp. 1467–1479, Aug. 2012.
[32] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 1, pp. 222–233, Jan. 2014.
[33] J. Baek, R. Safavi-Naini, and W. Susilo, "Public key encryption with keyword search revisited," in *Proc. Int. Conf. Comput. Sci. Its Appl.* Springer, 2008, pp. 1249–1259.
[34] D. J. Park, K. Kim, and P. J. Lee, "Public key encryption with conjunctive field keyword search," in *Proc. Int. Workshop Inf. Secur. Appl.* Springer, 2004, pp. 73–86.
[35] P. Golle, J. Staddon, and B. Waters, "Secure conjunctive keyword search over encrypted data," in *Proc. Int. Conf. Appl. Cryptogr. Netw. Secur.*, Cham, Switzerland: Springer, 2004, pp. 31–45.
[36] M. Hattori, T. Hirano, T. Ito, N. Matsuda, and T. Mori, "Ciphertext-policy delegatable hidden vector encryption and its application to searchable encryption in multi-user setting," in *Proc. IMA Int. Conf. Cryptogr. Coding*. Springer, 2011, pp. 190–209.
[37] Y. Miao, J. Ma, X. Liu, X. Li, Q. Jiang, and J. Zhang, "Attribute-based keyword search over hierarchical data in cloud computing," *IEEE Trans. Services Comput.*, vol. 13, no. 6, pp. 985–998, Nov. 2020.
[38] K. Liang and W. Susilo, "Searchable attribute-based mechanism with efficient data sharing for secure cloud storage," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 9, pp. 1981–1992, Sep. 2015.
[39] Y. Miao, J. Weng, X. Liu, K.-K. Raymond Choo, Z. Liu, and H. Li, "Enabling verifiable multiple keywords search over encrypted cloud data," *Inf. Sci.*, vol. 465, pp. 21–37, Oct. 2018.
[40] W. Wu, J. Cao, Y. Zheng, and Y. P. Zheng, "WAITER: A wearable personal healthcare and emergency aid system," in *Proc. 6th Annu. IEEE Int. Conf. Pervasive Comput. Commun. (PerCom)*, pp. 680–685, 2008.
[41] R. Hussain and S. Zeadally, "Autonomous cars: Research results, issues, and future challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1275–1313, 2nd Quart., 2019.
[42] Z. Kang, J. Li, J. Shen, J. Han, Y. Zuo, and Y. Zhang, "TFS-ABS: Traceable and forward-secure attribute-based signature scheme with constant-size," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 9, pp. 1–16, Feb. 2023.
[43] Y. Guo, Z. Lu, H. Ge, and J. Li, "Revocable blockchain-aided attribute-based encryption with escrow-free in cloud storage," *IEEE Trans. Comput.*, vol. 72, no. 7, pp. 1–12, Jan. 2023.
[44] J. Li, X. Lin, Y. Zhang, and J. Han, "KSF-OABE: Outsourced attribute-based encryption with keyword search function for cloud storage," *IEEE Trans. Services Comput.*, vol. 10, no. 5, pp. 715–725, Sep. 2017.

[45] Y. Lu, J. Li, and Y. Zhang, "Privacy-preserving and pairing-free multirecipient certificateless encryption with keyword search for cloud-assisted IIoT," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 2553–2562, Apr. 2020.
[46] N. Chen, J. Li, Y. Zhang, and Y. Guo, "Efficient CP-ABE scheme with shared decryption in cloud storage," *IEEE Trans. Comput.*, vol. 71, no. 1, pp. 175–184, Jan. 2022.
[47] Y. Lu and J. Li, "Lightweight public key authenticated encryption with keyword search against adaptively-chosen-targets adversaries for mobile devices," *IEEE Trans. Mobile Comput.*, vol. 21, no. 12, pp. 4397–4409, Dec. 2022.
[48] G. Xu, S. Xu, Y. Cao, F. Yun, Y. Cui, Y. Yu, and K. Xiao, "PPSEB: A postquantum public-key searchable encryption scheme on blockchain for E-healthcare scenarios," *Secur. Commun. Netw.*, vol. 2022, pp. 1–13, Mar. 2022.
[49] X. Yang, X. Chen, J. Huang, H. Li, and Q. Huang, "FS-IBEKS: Forward secure identity-based encryption with keyword search from lattice," *Comput. Standards Interface*, vol. 86, Aug. 2023, Art. no. 103732.
[50] Y. Fan, B. Qin, and D. Zheng, "A lattice-based designated-server public-key authenticated encryption with keyword search," *J. Syst. Archit.*, vol. 145, Dec. 2023, Art. no. 103031.
[51] S. Xu, Y. Cao, X. Chen, Y. Yang, and S. M. Yiu, "Lattice-based public key encryption with authorized keyword search: Construction, implementation, and applications," *Cryptol. ePrint Arch.*, vol. 1, no. 1715, pp. 1–21, Nov. 2023. [Online]. Available: https://eprint.iacr.org/2023/1715
[52] S. Xu, Y. Cao, X. Chen, S. M. Yiu, and Y. Zhao, "Post-quantum public-key authenticated searchable encryption with forward security: General construction, and applications," *Cryptol. ePrint Arch.*, vol. 1, no. 591, pp. 1–20, Oct. 2023. [Online]. Available: https://eprint.iacr.org/2023/591
[53] E. J. Goh, "Secure indexes," *Cryptol. ePrint Arch.*, vol. 1, no. 216, pp. 1–18, Mar. 2003. [Online]. Available: https://eprint.iacr.org/2003/216

MIAO WANG received the master's degree from the School of Mathematics, South China University of Technology, Guangzhou, China, in 2020. She is currently pursuing the Ph.D. degree with the Department of Cryptography and Cyber Security, School of Software Engineering, East China Normal University, Shanghai, China. Her research interests include number theory and applied cryptography theory, quantum computing, and information security.

LIWANG SUN received the master's degree from the Computer Department, Wenzhou University, Wenzhou, China, in 2019. He is currently pursuing the Ph.D. degree with the Department of Cryptography and Cyber Security, School of Software Engineering, East China Normal University, Shanghai, China. His research interests include PEKS, outsourced computing, and lattice encryption.

ZHENFU CAO is currently a Distinguished Professor with East China Normal University, China. Since 1981, he has published over 400 academic papers in journals or conferences. His research interests include cryptography, number theory, and information security. He has received a number of awards, including the Ying-Tung Fok Young Teacher Award, in 1989, the National Outstanding Youth Fund of China, in 2002, and the Special Allowance by the State Council, in 2005. He was a co-recipient of the 2007 IEEE International Conference on Communications Computer Award, in 2007.

XIAOLEI DONG is currently a Distinguished Professor with East China Normal University. She hosts a lot of research projects supported by the National Basic Research Program of China (973 Program), the National Natural Science Foundation of China, and the Special Funds on Information Security of the National Development and Reform Commission. Her research interests include cryptography, number theory, and trusted computing.