QPASE: Quantum-Resistant Password-Authenticated Searchable Encryption for Cloud Storage

Jingwei Jiang and Ding Wang

***Abstract*— Searchable encryption is a powerful tool that enables secure and private searches of encrypted data. It allows users to outsource their data to cloud servers while maintaining the confidentiality and privacy of their data. Password-authenticated symmetric searchable encryption (PASE) can help users avoid the complexity and security risks associated with key management while maintaining the advantages of searchable encryption. To the best of our knowledge, none of the existing PASE schemes can resist security threats in the post-quantum era, and there is an urgent need to design quantum-resistant solutions. However, post-quantum cryptography (e.g., lattice-based cryptography) varies significantly from traditional cryptography, and it is challenging to design a quantum-resistant PASE for cloud storage. In this work, we take the first step towards this challenge by proposing QPASE, a quantum-resistant password-authenticated symmetric searchable encryption for cloud storage. We employ lattice-based threshold oblivious pseudorandom function to achieve password re-randomization and formally prove that QPASE is authentication secure and indistinguishability against chosen keyword attacks secure under quantum computers. QPASE can be extended to multi-keyword search and allows servers to update keys without affecting the users. The comparison results show that QPASE outperforms its foremost counterparts in security and computation overhead.**

***Index Terms*— Lattice, password authentication, searchable encryption, data outsourcing, cloud storage.**

I. Introduction

WITH the continuous development of the Internet of Things, a massive number of devices and objects are being connected to the Internet. Users with limited computing and storage resources can outsource large amounts of data to cloud servers for management [1], which in turn brings about a vast amount of data [2]. The abundance of sensitive information (e.g., personal identification, financial) within outsourced data poses significant challenges to data management [3].

Manuscript received 11 April 2023; revised 20 December 2023 and 17 February 2024; accepted 22 February 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62222208; and in part by the Natural Science Foundation of Tianjin, China, under Grant 21JCZXJC00100. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Alptekin Küpçü. *(Corresponding author: Ding Wang.)*

Jingwei Jiang is with the College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China, and also with State Key Laboratory of Cryptology, Beijing 100878, China (e-mail: jiangjingwei@hrbeu.edu.cn).

Ding Wang is with the Key Laboratory of Data and Intelligent System Security (Ministry of Education), Nankai University, Tianjin 300350, China, and also with the State Key Laboratory of Cryptology, Beijing 100878, China (e-mail: wangding@nankai.edu.cn).

Digital Object Identifier 10.1109/TIFS.2024.3372804

Data encryption is the most direct technique to prevent data leakage [4], however, it complicates user querying [5].

To address this issue, cloud servers (e.g., Google Drive [6]) allow users to control online-key decrypted data via authentication and retrieve items of interest. But such a solution crucially depends on the cloud servers' honesty, as they have the potential to gain access to the plaintext of data [7]. Alternatively, users download all encrypted data and decrypt data locally to search for the required data but incur heavy communication costs for uploading and downloading.

Searchable encryption (SE) [8], [9] provides an elegant solution. SE enables users to perform searches on encrypted outsourced data while maintaining the confidentiality of data from the cloud server. Currently, SE can be classified into two main categories based on the structures: 1) Symmetric searchable encryption (SSE) schemes [10], [11] that employ high entropy shared keys; and 2) Public key encryption with keyword search (PEKS) schemes [12], [13] that require a high entropy private-public key pair. In practice, the high entropy keys are difficult to remember and require additional storage devices. This reduces the flexibility of users to outsource data, or retrieve and recover data using multiple different devices unless the high entropy keys are stored in all devices [14].

Symmetric Searchable Encryption: Song et al. [8] propose the first practical SSE based on symmetric primitives. In SSE, data is organized and indexed in a way that allows efficient search while preserving confidentiality [8]. After that, Goh [15] first proposes indistinguishability against chosen keyword attacks (IND-CKA) to characterize the semantic security of SSE. Such these works spark a series of studies on the security [16], [17], [18], [19], performance [20], [21], [22], and functionality [23], [24], [25] of SSE. Overall, SSE employs a masked index table to achieve ciphertext retrieval [26], [27].

At a high level, the user employs a symmetric encryption scheme to encrypt a set of data and output a ciphertext C. Meanwhile, the user creates a masked index Ct based on the message keyword. Then, the user can upload C and Ct on the cloud server. When users need to access data, they can generate a search token based on the encrypted keyword index and request the cloud server to return a C based on Ct.

Another variant of SSE is dynamic SSE (DSSE) [28], [29], [30], [31]. DSSE supports the addition and deletion of outsourced data. Driven by leakage-abuse attacks [32], a series of studies employ padding [33] and secure multi-party computation [34] to achieve forward security [35] (i.e., the added data cannot be associated with the original data), and backward security [17] (i.e., the deleted data cannot be retrieved). SSE allows users to preprocess data by building an index, and subsequent search queries can be performed efficiently, with only a small amount of computation required by both the user and the server [36]. Hence, SSE is a practical scheme for cloud storage where data needs to be searched and accessed frequently, while also being kept confidential.

Public Key Encryption With Keyword Search: Boneh et al. [37] propose the first PEKS using bilinear maps and trapdoor permutations. PEKS allows users to associate keywords and outsourced data without disclosing any data-related information [38], [39]. Baek et al. [40] propose a secure channel-free PEKS and carry out the proof under the random oracle.

Subsequently, a series of variants of PEKS were studied, such as conjunctive keyword search [41], fuzzy keyword search [42], ranked keyword search [43], attribute-based keyword search [44]. However, malicious clouds can obtain underlying keywords by guessing candidate keywords offline. Thus, a line of work [45], [46], [47] has been done against the keyword guessing attacks by the public-key authenticated encryption. Other variants of PEKS include multi-user settings, deterministic searchable encryption [48], and plaintext-checkable encryption [49], enriching the research of PEKS.

Password-Authenticated Searchable Encryption: In practice, SE relies on high entropy keys for encryption and retrieval. Therefore, users need to employ a storage device to hold high-entropy keys, which increases the burden of key management when outsourcing and retrieving data using different devices. Chen et al. [14] present a password-authenticated symmetric searchable encryption (PASE) scheme to avoid costly key management for users, and achieve device-agnostic. Specifically, a user registers a human-memorable password on the server and reuses the password to outsource and retrieve data. PASE transfers the management overhead of users on high entropy keys to servers with strong storage capabilities.

Additionally, Huang et al. [50] propose a password-authenticated keyword search (PAKS) scheme. PAKS employs asymmetric primitives to encrypt data and retrieve target ciphertext, resulting in slower encryption/decryption speeds. Hence, PAKS is suitable for many-to-one scenarios (e.g., data sharing), where data owners use secret keys to generate trapdoors for the keywords to be retrieved, instead of outsourcing data of a single user. PASE employs symmetric primitives for encryption and trapdoor generation, making it more suitable for single-user data outsourcing scenarios. Our work aims in the scenario where users outsource data to cloud servers, and subsequently retrieve and recover their data from the cloud servers. Therefore, we focus on the construction of PASE.

However, to the best of our knowledge, none of those schemes [14], [50] mentioned above can resist security threats in the post-quantum world. The reason is that all of them are built on the hardness assumptions of traditional cryptography (e.g., large integer decomposition, discrete logarithms, elliptic curves). Hence, existing PASE is vulnerable after the advent of quantum computers, which are capable of efficiently solving traditional hardness problems using Shor's algorithm [51].

In the realm of quantum-resistant schemes, lattice-based schemes are considered the most promising general-purpose algorithms for public-key encryption by NIST [52], [53]. Numerous quantum-resistant password-based schemes [54], [55], [56], [57], [58], [59] have been proposed over lattices. To the best of our knowledge, there is no quantum-resistant password-authenticated symmetric searchable encryption scheme. The main goal of our scheme is to answer the following question:

> *Is it possible to construct a lattice-based password-authenticated symmetric searchable encryption scheme to satisfy that only a user who knows the password can outsource and retrieve data?*

Our answer to the above question is affirmative. Next, we show the design challenges and overview of our technique.

A. Overview of Our Technique

Before elaborating on our results and techniques, we first highlight two crucial observations. On the one hand, PASE is not simply a combination of a password-based authentication scheme and SSE. At a high level, PASE allows users to employ a password to derive strong keys, which are shared in multiple distributed cloud servers, to encrypt data [12]. PASE helps users avoid complex key management while improving the security of outsourced data [14]. Specifically, data encryption on the user side is only related to the correct password. It is independent of the device that stores the key, which increases the usability of the data outsourcing scheme. In addition, the user encrypts data locally which can prevent malicious cloud servers from snooping on outsourced data. Even if the strong key on distributed cloud servers is leaked, the adversary would still need to guess the user's password to retrieve and recover the outsourcing data on cloud server [7], [60], [61], [62].

On the other hand, it is essential to accurately verify the password to prevent the risk of data loss resulting from typographical errors on the part of the user. Specifically, in the recovery phase, PASE [14] allows users to employ the same password used during encryption to retrieve data. However, if the user inputs the wrong password during encryption (e.g., typing error), the "correct" password would lead to decryption failure in the recovery phase. Even requiring users to input the password twice before encryption cannot completely solve this problem [7]. In addition, implicit authentication leads to the server's inability to recognize online password-guessing attacks, which would increase the risk to the system.

Passwords are the most widely used identity authentication mechanism, but their low entropy and vulnerability have raised serious security concerns [63], [64]. Although there is a growing consensus that password-based authentication is likely to retain its status for the foreseeable future [65], [66], how to protect low-entropy passwords remains a challenging problem, especially in the coming post-quantum era [52], [53].

A feasible approach to constructing quantum-resistant PASE (QPASE) is lattice-based cryptography, and the primary issue is the re-randomization of passwords. Jiang et al. [4] proposed a password re-randomization method based on lattice-based fully homomorphic encryption [67], but this method can only provide implicit authentication and is not suitable for our goal. Although the password-authenticated secret sharing [7], [61] has been constructed through an oblivious pseudorandom function (OPRF), we cannot achieve the same goal simply by employing lattice-based threshold OPRF (TOPRF) [68], which can only provide the approximate protocol due to a series of noise interferences. Inspired by Ding et al. [69], we adopt the robust extractor [69] to "rounding" the noise, and propose a variant TOPRF to realize the re-randomization of passwords.

At a high level, QPASE is modeled as an SSE scheme, where the user U can register with the password psw_u on a set of cloud servers $S = \{S_1, \ldots, S_n\}$ and reuse psw_u for multiple sessions of outsourcing and retrieval protocols. In each outsourcing session, users can outsource encrypted keywords and data ciphertexts to S. The retrieval protocol implements the search process based on the keywords input by U and provides U with all data related to that keyword. We define binary security to include authentication security and keyword privacy security. Specifically, we characterize the authentication security of QPASE based on the Bellare-Pointcheval-Rogaway (BPR) model [70] widely used in password-based schemes. Then, we define keyword privacy security based on indistinguishability against chosen keyword attacks (IND-CKA). These two security models are not orthogonal, i.e., authentication security can prevent impersonation attacks and protect for SSE.

Finally, there are two challenges in constructing QPASE. The first challenge pertains to updating server-side keys. To resist the perpetual leakage [71], server-side keys need to be updated in a fixed period. Although the generation of user-specific keys is closely related to server-side keys, the update of server-side keys should not affect the decryption of outsourcing data. Secondly, it concerns the password distribution model. It is commonly assumed in password-based schemes [7], [14], [61], [62], [72], [73] that the selection of passwords is uniformly distributed. Recent research [74] suggests that human-chosen passwords follow the Zipf distribution. Wang et al. [74], [75] show that adversary's advantages are underestimated in the uniform model. The impact of password distribution assumptions should be fully considered.

Contributions: We propose the first quantum-resistant pass- word-authenticated symmetric searchable encryption for cloud storage, named QPASE. Our construction starts with PASE [14] and follows the more general approach to realizing an SSE but employs quantum-secure cryptographic primitives. In summary, our contributions are three-fold:

- **QPASE**. We design a quantum-resistant PASE for cloud storage, called QPASE, to help users avoid costly and security-risky key management when using cloud storage services. Registered users can perform outsourcing, and retrieval of data via human memorable passwords only. By a password re-randomization method based on the lattice-based TOPRF, QPASE is secure against offline password-guessing attacks. Users can retrieve all data under the same keyword without revealing data. Moreover, QPASE allows servers to actively and independently update keys to resist perpetual leakage.
- **Security analysis.** We employ the BPR model [70] to characterize the authentication security of QPASE, and define the privacy of QPASE keywords through IND-CKA. On this basis, we make a rigorous security proof based on the Zipf model [74] and formally prove that QPASE is secure and robust under various attacks from both attacks of classical and quantum computers.
- **Performance comparison.** We evaluate the quantum security level of QPASE under two parameter settings. The experimental analysis shows QPASE is not only more secure (our implementation can provide 128-bit quantum security) but also offers better computation efficiency than the state-of-the-art traditional PASE [14].

TABLE I
NOTATIONS

Symbol	Description	Symbol	Description
U	the user	S	the server
w	the keyword	**sig**	the signature
μ_u	the counter	$\mathcal{A}$	the adversary
Ct	hidden indexes	psw_u	password of user U
$\mathbf{A}$	random matrices	$\mathbf{K}_u$	the user-specific key
ID_u	identity of user U	d	the plaintext of data
$\mathcal{X}$	noise distribution	C	the ciphertext of data
Enc	symmetric encryption	Dec	symmetric decryption

B. Paper Organization

In Section II, we review the related notions and the basic components required for constructing the scheme. In Section III, we formally model the functionality and define the primary security properties of QPASE. In Section IV, we articulate QPASE and provide a correctness analysis. We also provide a server key update protocol and an extended version supporting multiple keywords. In Section V, we formally demonstrate that QPASE satisfies authentication and IND-CKA security, discuss the parameter selection and security level of QPASE, and provide evidence of QPASE's resilience against corruption attacks. In Section VI, we present the results of the experiments and compare the related works with QPASE. Finally, in Section VII, we conclude the paper.

II. PRELIMINARIES

Notations: Let κ be the security parameter. $\mathcal{Z}$ and $\mathcal{R}$ denote the set of all integers and the set of real numbers, respectively. For any integer q, $\mathcal{Z}_q$ is the ring of integer mod q. We write lower-case bold $\mathbf{x}$ letter as vectors and upper-case bold letter $\mathbf{A}$ as matrices. Let $x \leftarrow \mathcal{D}$ to denote the sampling of x according to distribution $\mathcal{D}$ and $x \leftarrow S$ for a finite set S to indicate sample uniformly at random from S. In addition, we employ a series of intuitive notations listed in Table I.

A. Lattices, LWE, and Gaussian Sampling

Definition 1 [76]: Define a lattice is $\Lambda_q(\mathbf{A}) = \{\mathbf{As} \mid \mathbf{s} \in \mathcal{Z}_q^n\}$ *with m-dimensional, where the basis* $\mathbf{A} \in \mathcal{Z}_q^{m\times n}$ *for* $m \geq n\log q$, *and the determinant of* Λ *is* $det(\Lambda) = \sqrt{det(\mathbf{B}^T\mathbf{B})}$.

Definition 2 [76]: If $\chi = \chi(\kappa)$ *over the integers is a distribution ensemble and it satisfies* $\Pr[x \xleftarrow{R} \chi; |x| \geq B] \leq 2^{-\tilde{\Omega}(n)}$, *then we have* $|\chi| \leq B$ *and* χ *is called B-bounded.*

Definition 3 (**Gaussian Distributions**, *[77]): For a standard deviation* $\sigma > 0$ *and* $\mathbf{c} \in \mathcal{R}^n$ *is the center, the discrete Gaussian distribution over* $\Lambda \subseteq \mathcal{Z}^m$ *centred at* $\mathbf{c}$ *with* σ *to be:* $D_{\mathcal{R},\sigma}(\mathbf{z}) = \rho_{\mathbf{c},\sigma}(\mathbf{x})/\rho_{\mathbf{c},\sigma}(\Lambda)$, *where* $\mathbf{x} \in \Lambda$, $\rho_{\mathbf{c},\sigma}(\mathbf{x}) = e^{\pi||\mathbf{x}-\mathbf{c}||^2/\sigma^2}$, *and* $\rho_{\mathbf{c},\sigma}(\Lambda) = \sum_{\mathbf{x}\in\Lambda} \rho_{\mathbf{c},\sigma}\mathbf{x}$.

Definition 4 (Decision-$\mathsf{LWE}_{n,q,\chi,m}$*, [76]): For a prime integer q, integers* $m, n > 0$*, and a noise distribution* $\mathcal{X}$ *over* $\mathcal{Z}_q$*, sample* $\mathbf{A} \leftarrow \mathcal{Z}_q^{m\times n}, \mathbf{s} \leftarrow \mathcal{Z}_q^{n\times 1}, \mathbf{e} \leftarrow \chi_\sigma^{m\times 1}, \mathbf{b} \leftarrow \mathcal{Z}_q^{m}$*. The* $DLWE_{n,q,\chi,m}$ *problem is to distinguish between: 1)* $(\mathbf{A}, \mathbf{A}\cdot\mathbf{s}+\mathbf{e}\ mod\ q) \in \mathcal{Z}_q^{m\times n} \times \mathcal{Z}_q^{m\times 1}$ *and 2)* $(\mathbf{A}, \mathbf{b}) \in \mathcal{Z}_q^{m\times n} \times \mathcal{Z}_q^{m\times 1}$*. For any PPT adversary* $\mathcal{A}$*, the two distributions above-mentioned are computationally indistinguishable. In other words, the adversary's advantage in solving* $DLWE_{n,q,\chi,m}$ *problem is as follow:*

$$Adv_{\mathcal{A}}^{\mathsf{DLWE}}(\kappa) = |Pr[\mathcal{A}(q, m, n, \mathcal{X}_\sigma, \mathbf{A}, \mathbf{s})] - Pr[\mathcal{A}(q, m, n, \mathcal{X}_\sigma, \mathbf{A}, \mathbf{b})]| \le \varepsilon(\kappa).$$

Definition 5 (Rounding [78]): Let $q, m, n \in \mathcal{Z}^+$*. For* $\mathbf{A} \leftarrow \mathcal{R}_q^{m\times n}$*, the rounding algorithm F is deterministic for the "rounding" function* $\lfloor\cdot\rceil : \mathcal{R}_q \to \mathcal{R}_p$ *that enables* $\mathbf{x} \to \mathbf{u}$ *s.t.* $F(\mathbf{x}) = \mathbf{A}\cdot\mathbf{u}$*.*

Definition 6 (LWR [78]): For $p, q, m, n \in \mathcal{Z}^+$*, the LWR problems state that the two distributions* $(\mathbf{A}, \lfloor\mathbf{A}\cdot\mathbf{s}\rfloor_p)$ *and* $(\mathbf{A}, \lfloor\mathbf{u}\rfloor_p)$ *are computationally indistinguishable, where* $\mathbf{A} \leftarrow \mathcal{R}^{m\times n}, \mathbf{s} \leftarrow \mathcal{R}^m$ *and* $\mathbf{u} \leftarrow \mathcal{R}^n$*.It means that for any PPT adversary* $\mathcal{A}$*, there is*

$$Adv^{LWR} = |Pr[\mathcal{A}(\mathbf{A}, \lfloor\mathbf{A}\cdot\mathbf{s}\rfloor_p) \to 1] - Pr\left[\mathcal{A}(\mathbf{A}, \lfloor\mathbf{u}\rfloor_p) \to 1\right]| \le \varepsilon(\kappa).$$

Modulus Switching: As discussed in [78], for any positive integers p, q, the modulus switching function $\lfloor\cdot\rceil\ q \to p$ is denoted as:$\lfloor x\rceil_{p\to q} = \lfloor (p/q)\cdot x\rceil (\text{mod}\, q)$. It is easy to show that for any $x \in \mathcal{Z}_q$ and $p < q \in \mathbb{N}$, $x' = \lfloor\lfloor x\rceil_{q\to p}\rceil_{q\to p}$ is an element near to x, i.e.,$|x' - x(mod\, q)| \le \lfloor q/(2p)\rceil$. When $\lfloor\cdot\rceil\, q \to p$ is used to an element $x \in \mathcal{Z}_q$ or a vector $\mathbf{x} \in \mathcal{Z}_{q'}^k$, the procedure is applied to each coefficient individually.

Definition 7 (1D-SIS [79]): For $q, m, t \in \mathcal{Z}^+$*, given a* $\mathbf{v} \leftarrow \mathcal{Z}_q^m$*, the one-dimensional SIS problem (1D-SIS) is to find a non-zero* $\mathbf{z} \in \mathcal{Z}^m$ *s.t.* $\|\mathbf{z}\|_\infty \le t$ *and* $\langle \boldsymbol{v}, z\rangle \in [-t, t] + q\mathcal{Z}$*.*

B. Threshold Oblivious Pseudorandom Function

Threshold oblivious pseudorandom function (TOPRF) is widely used in various password-based schemes [7], [62], [72] to hide passwords to resist offline dictionary attacks. In Fig. 1, we employ the lattice-based TOPRF [68] (where k_i is the key of server S_i and pk_i is the public key of S_i) to implement password re-randomization, and derive a special key for the user U with servers $S = \{S_1, \ldots, S_n\}$.

Let $\ell = \lceil log_2 q\rceil$. Define $G : R_q^{\ell\times\ell} \to R_q^{1\times\ell}$ to be the linear operation corresponding to left multiplication by $(1, 2, \ldots, 2^{\ell-1})$ and $G^{-1} : R_q^{1\times\ell} \to R_q^{\ell\times\ell}$. It can be regarded as the decomposition of G. Fix an array of $\mathbf{a}_0, \mathbf{a}_1 \leftarrow R_q^{1\times\ell}$. For any $\mathbf{x} = (x_1, \ldots, x_L) \in \{0, 1\}^L$ subject to $\mathbf{ax} := \boldsymbol{a}_{x_1}\cdot G^{-1}\left(\boldsymbol{a}_{x_2}\cdot G^{-1}\left(\ldots\left(\boldsymbol{a}_{x_{L-1}}\cdot G^{-1}\left(\boldsymbol{a}_{x_L}\right)\right)\right)\right) \in R_q^{1\times\ell}$. Based on the Definition 4, a PRF $F_k(x)$ is defined as follows.

Lemma 1 (PRF, [80]): Sample $k \leftarrow \mathcal{Z}_q$ *and recursively as* $\mathbf{a}^F(\mathbf{x}) = \mathbf{a}_\mathbf{x}$*, the function* $F_k(x) = \lfloor\frac{p}{q}\cdot\mathbf{a}^F(\mathbf{x})\cdot k\rceil$ *is a PRF over the decision-*$\mathsf{LWE}_{n,q,\chi,m}$ *if* $q \gg p\cdot\sigma\cdot n\cdot\ell\cdot\sqrt{L}$*.*

According to Albrecht et al. [79], for a PPT algorithm $r \leftarrow \Pi_x(\mathbf{a}_0, \mathbf{a}_1)$ s.t. $\|r\|_\infty \le B$ and $\exists\, c \in (q/p)\cdot\mathcal{Z} + [-T, T]$ with non-negligible probability, where B is distribution bounded, c is the coefficient of $\mathbf{a}_x\cdot r$, and $T = 2\sigma^2 n^2 + \sigma'\sqrt{n}$. Then there is a PPT algorithm that can solve $\text{1D-SIS}_{q/p, n\ell, \max\{n\ell B, T\}}$ with non-negligible probability. We write $adv_{\mathcal{A}}^{1D-SIS}$ as the advantage of the adversary. According to Definition 7, we have

$$Adv_{\mathcal{A}}^{1D\text{-}SIS} \le \varepsilon(\kappa).$$

Oblivious Computation of PRF $F_k(\mathbf{x})$

1. On input $\mathbf{x}$, U chooses $s \leftarrow Z_q^{n\times 1}$ and $e \leftarrow \mathcal{X}_\sigma^{m\times 1}$; Sends $\mathbf{x}^* = \mathbf{A}\cdot s + e + \mathbf{a}^F(\mathbf{x})$ to at least t-many S_i.
2. S_i chooses $e_i' \leftarrow \mathcal{X}_{\sigma'}^{m\times 1}$ responds with $\mathbf{x}_{k_i}^* = \mathbf{x}^*\cdot k_i + e_i'$;
3. After receiving at least t responses, U computes $PK = \sum_{i=1}^{t}\lambda_{i,j}\cdot pk_i$ and output $F_K(\mathbf{x}) = \lfloor\frac{p}{q}\cdot\mathbf{a}^F(x)\cdot msk\rceil$.

Fig. 1. The TOPRF algorithm of Jiang et al. [68].

In addition, the amplified noise still causes the same input to derive different PRF keys in practice. To tackle this challenge, we employ the approach of Ding et al. [69] to eliminate noise.

Definition 8 (Robust Extractors [69]): Let δ *be error tolerance. The robust extractor contains a deterministic algorithm E and a hint algorithm S, which are as follows:*

- *$\sigma \leftarrow S(y)$ is a hint algorithm. When input a $y \in \mathcal{R}_q$ and outputs $\sigma \in \{0, 1\}$. Specifically, for prime $q > 2$, there are two signal $\sigma_0(x), \sigma_1(x)$ as follows.*

$$\sigma_0(x) = \begin{cases} 0, & x \in [-\lfloor\frac{q}{4}\rfloor, \lfloor\frac{q}{4}\rfloor] \\ 1, & \text{otherwise.} \end{cases}$$

$$\sigma_1(x) = \begin{cases} 0, & x \in [-\lfloor\frac{q}{4}\rfloor + 1, \lfloor\frac{q}{4}\rfloor + 1] \\ 1, & \text{otherwise.} \end{cases}$$

- *$k \leftarrow E(x, \sigma)$ is a deterministic algorithm. When input an $x \in \mathcal{R}_q$ and a signal $\sigma \in \{0, 1\}$, outputs $k \in \{0, 1\}$. Specifically, we have $E(x, \sigma) = (x+\sigma)\cdot(q-1)/2 \mod 2$.*
- *For any $x, y \in \mathcal{Z}_q$ such that $x - y$ is even and $|x - y| \le \delta$, then it holds that $E(x, \sigma) = E(y, \sigma)$, where $\sigma \leftarrow S(y)$.*

The variant-TOPRF with robust extractors is shown in Fig. 2. Next, we analyze the correctness of variant-TOPRF.

Theorem 1 (Correctness): Let $q, m, n, \sigma > 0$ *depend on* κ *and* $\ell = \lfloor log\ q\rfloor$*. The secret input* $\mathbf{x}$ *is converted into binary by the user U. The output* $F_K(\mathbf{x})$ *of the variant-TOPRF is indistinguishable from the PRF* $F_k(x)$ *in Definition 1.*

Proof: The explicit expression of pk_i in Fig. 2 is $pk_i = \lfloor\mathbf{a}\cdot k_i\rceil_p$. Let λ_i be the Lagrange coefficient s.t. $K = \sum_{i=1}^{t}\lambda_i\cdot k_i \in \mathcal{R}_q$. According to Fig. 2, we have

$$\begin{aligned} F_K(\mathbf{x}) &= \sum_{i=1}^{t}\lambda_i\cdot\mathbf{b}_{k_i} - \sum_{i=1}^{t}\lambda_i\cdot pk_i\cdot r\ mod\ q \\ &= \mathbf{a}\cdot r\cdot\sum_{i=1}^{t}\lambda_i\cdot k_i + \mathbf{a}_\mathbf{x}\cdot\sum_{i=1}^{t}\lambda_i\cdot k_i \\ &\quad + 2e\sum_{i=1}^{t}\lambda_i\cdot k_i + 2\sum_{i=1}^{t}\lambda_i\cdot e_i' \\ &\quad - \mathbf{a}\cdot r\cdot\sum_{i=1}^{t}\lambda_i\cdot k_i - 2r\sum_{i=1}^{t}\lambda_i\cdot e_i\ mod\ q \\ &= \mathbf{a}_\mathbf{x}\cdot K + 2e''\ mod\ q \end{aligned}$$

where $e'' = e\cdot K + \sum_{i=1}^{t}\lambda_i\cdot e_i' + r\sum_{i=1}^{t}\lambda_i\cdot e_i$. Thus,

$$\begin{aligned} K_u &= \lfloor E(F_K(\mathbf{x}), \sigma)\rceil_p \\ &= \lfloor((\mathbf{a}_\mathbf{x})\cdot K + \sigma((b_{k_i})\cdot\frac{q-1}{2})\ mod\ q + 2e'')\ mod\ 2\rceil_p \end{aligned}$$

Initialization
Set (t, N) as threshold, where $t, N \in Z^{+}$ and $0 < t < N$. A subset $\mathcal{S}$ from $[N]$ of size t. N servers execute the DKG algorithm to generate the key $k_i \in \mathcal{R}_q$, and compute vk_i, recursively as $\mathbf{a}^F(\mathbf{x}) = \mathbf{a}_x$. Concretely, $vk_i = \mathbf{a} \cdot k_i$. *gadgets* $G : R_q^{\ell \times \ell} \rightarrow R_q^{1 \times \ell}$.
TOPRF between the user U and N servers S_i
U picks up $r \leftarrow \mathrm{R}_q$, $e \leftarrow \mathcal{X}$ and then input a secret $\mathbf{x}$. U sends $\mathbf{b} = \lfloor \mathbf{a} \cdot r + \mathbf{a}^F \mathbf{x} + 2e \rceil_p$ to each S_i. S_i responds with $(\mathbf{b}_{k_i}, S(vk_i))$, $\mathbf{b}_{k_i} = \lfloor \mathbf{b} \cdot k_i + 2e_i' \rceil_p$, $vk_i = \lfloor \mathbf{a} \cdot k_i + 2e_i \rceil_p$. Here, U outputs $F_{msk}(\mathbf{x}) = \lfloor ((\mathbf{a_x}) \cdot msk + \sigma((b_{k_i}) \cdot \frac{q-1}{2})\ mod\ q)\ mod\ 2 \rceil_p$.

Fig. 2. The variant-TOPRF algorithm Π_{TOPRF}.

$$= \lfloor ((\mathbf{a_x}) \cdot K + \sigma((b_{k_i}) \cdot \frac{q-1}{2})\ mod\ q)\ mod\ 2 \rceil_p$$

It means that the output K_u of the variant-TOPRF is indistinguishable from $F_K(\mathbf{x})$.

The security analysis of our TOPRF is divided into two parts, i.e., unpredictability and obliviousness [68], [79]. Intuitively, unpredictability refers to the scenario where even the adversary $\mathcal{A}$ can compromise the client, i.e., $\mathcal{A}$ gets $\mathbf{x}$, and corrupts $t' < t$ servers, $\mathcal{A}$ cannot predict the output $F_K(\mathbf{x})$ of TOPRF. Obliviousness indicates that even $\mathcal{A}$ can get the output $F_K(\mathbf{x})$ and corrupt $t' < t$ servers, $\mathcal{A}$ cannot learn anything about $\mathbf{x}$. Together, unpredictability and obliviousness ensure that the output of the TOPRF remains independent of the input. Notably, the use of signals does not undermine the security of underlying intractable problem [69] (e.g. $\mathsf{DLWE}_{n,q,\chi,m}$ and 1D-$\mathrm{SIS}_{q/p,n\ell,\max\{n\ell B,T\}}$). Moreover, robust extractors only reveal the range of noise without affecting the output of TOPRF since the noises are eliminated by rounding operations.

C. Hash Key Derivation Function

As a crucial component of the PASE construction, the hash key derivation function (HKDF) is capable of deriving a single input into multiple distinct secret values, serving as encryption keys. This functionality ensures that different data can be protected by unique symmetric keys, thereby preventing a scenario where the compromise or loss of one key would render all data vulnerable. We directly use the password that has been authenticated to generate the data search key, which prevents the user from permanently losing data due to erroneous keystrokes. We can conveniently introduce lattice-based PRF [80] in the framework proposed by Krawczyk [81] to construct a quantum-secure HKDF.

Definition 9: Let $\mathsf{TOPRF}(1^\kappa, psw, sk)$ *denotes the algorithm* Π_{TOPRF} *in Fig. 1. An HKDF must contain the following four polynomial algorithms:*

- *$pp \leftarrow \mathsf{Setup}(1^\kappa)$ is a probabilistic algorithm that generates the set of parameters pp.*
- *$\{\mathbf{K}_u, pk_i\} \leftarrow \mathsf{TOPRF}(1^\kappa, psw, K)$ is a deterministic algorithm that generates a user-special key $\mathbf{K}_u$ by taking as input user's password psw and server's secret key K.*
- *$\mathsf{w} \leftarrow \mathsf{Keyword}(1^\kappa, M)$ is a probabilistic algorithm that generates a keyword w by inputting 1^κ and message M.*
- *$\mathsf{dsk} \leftarrow \mathsf{HKDF}(\mathbf{K}_u, \mathsf{w})$ is a deterministic algorithm and outputs a data search key dsk by taking as input a user-special key $\mathbf{K}_u$, and a keyword w.*

Definition 10 [81]: Let C denote the ciphertext of retrieved information with dsk. The HKDF is called (T, Q, ε)-secure if for any PPT algorithm $\mathcal{A}$ running in time T with at most Q oracle queries the probability $Adv_{\mathcal{A}}^{\mathsf{HKDF}}(\kappa) \le \varepsilon(\kappa)$ or distinguishing the output of $\mathsf{dsk} \leftarrow \mathsf{HKDF}(\mathbf{K}_u, \mathsf{w})$ from uniformly random strings of the same length.

D. Distributed Key Generation

Since lattice is an infinite additive group, it can not be directly combined with Shamir's scheme [82]. Fortunately, we can share elements of a finite abelian quotient group G with identity element 0 by (t, N)-threshold secret sharing scheme [83]. Let $e(G)$ denote exponent of G and $s \in G$.

Definition 11: There is the smallest $m \in \mathcal{Z}^+$ such that $ms = s + s + \ldots + s = 0$, i.e. s is a module over the ring $R = \mathcal{Z}_e(s)$. The value s can be share by a formal polynomial $f(X) = \sum_{j=0}^{t} f_j X^j \in S[X]$ of the maximum degree at t, where $f(0) = s$ and the $f(i) \in G$ for $i \in [1, n]$ are uniformly random and independent. At least t + 1 participants can reconstruct the secret s.

In order to use the above secret sharing over lattices, we also need to set relevant parameters. Let $k \ge log_p(n+1)$, where p is the smallest prime divisor of $e(G)$, we can share $s \in G$ among n servers using shares in G^k. By [83], we can use $R = \mathcal{Z}_e(G)[X]/F(X)$ for any monic degree-k polynomial $F(X) = \sum_{i=0}^{k} F_i X^i \in \mathcal{Z}_e(G)$ that is irreducible modulo every prime dividing $e(G)$ that is irreducible modulo every prime dividing $e(G)$. We write $[s]^i$ to denote i-th server's share and the tuple of all shares by $[s]$. By combining the idea of integer sampling and MPC, we can realize distributed server key generation without the trusted center as follows:

Definition 12 [83]: The Distributed Key Generation (DKG) must contain the following two polynomial algorithms:

- *$[s_i] \leftarrow \mathsf{Genshare}(\mathcal{Z}_e(G), \mathcal{Z}_q)$ is a probabilistic algorithm that sample $F_i \leftarrow \mathcal{Z}_e(G)$ and generates $[s_i] \leftarrow \mathcal{Z}_q$.*
- *$\mathbf{k}_j \leftarrow \mathsf{Genkey}(i, j, [s_i]^j)$ is a deterministic algorithm that generates secret key $k_j = \sum_{i=1}^{n} [s_i]^j$ by receiving n tuple of $(i, j, [s_i]^j)$. After receiving n numbers of $[s_i]^j$, S_j computes $k_j = \sum_{i=1}^{n} [s_i]^j$.*

An unknown master secret key $K = \sum_{j=1}^{t} [s]_j^0$ that cannot be recovered unless at least $t + 1$ malicious servers collude.

E. EUF-CMA Signature

In digital signature schemes, existential unforgeability under chosen-message attacks (EUF-CMA) ensures that signatures cannot be forged by public keys. Moreover, there are three security properties of signatures beyond unforgeability: 1) Exclusive ownership [84] guarantees that a public key can only verify one corresponding signature; 2) Message-bound signatures guarantee that a signature is only valid for a unique message; 3) Non re-signability [85] guarantees that no signature can be generated with another key given the signature of a certain unknown message. In the post-quantum signature scheme of NIST Round 3 candidates (i.e., CRYSTALS-Dilithium [86], FALCON [87], and Rainbow [88]), CRYSTALS-Dilithium is the only signature scheme that provides EUF-CMA and all three security properties beyond unforgeability above [89].

CRYSTALS-Dilithium [86] is a lattice-based signature scheme and is designed based on the lattice hardness problem (i.e., LWE and a variant of the shortest integer solution problem). In the absence of a secure channel, we employ an instance of the CRYSTALS-Dilithium to prevent adversaries from tampering with the information.

Definition 13 [86]: The CRYSTALS-Dilithium signature scheme for a message space $\mathcal{M}$ is a tuple of PPT algorithms as follows:

- $(\mathbf{pk}, \mathbf{sk}) \leftarrow \mathbf{Gen}(1^\kappa)$ *outputs a verification key* **pk** *and a signing key* **sk**.
- $\mathbf{sig} \leftarrow \mathbf{Sign}(\mathbf{sk}, m)$ *outputs a signature* $\mathbf{sig} \in \{0,1\}^*$ *by input* **sk** *and a message* $m \in \mathcal{M}$.
- $\{0,1\} \leftarrow \mathbf{Ver}(\mathbf{pk}, m, \mathbf{sig})$ *outputs either 1 (accepts) or 0 (rejects) by input* **pk**, *m, and* **sig**.

Correctness: For any $m \in \mathcal{M}$, the verification algorithm $\mathbf{Ver}(\mathbf{pk}, m, \sigma)$ outputs 1 with overwhelming probability, if $(\mathbf{pk}, sk) \leftarrow \mathbf{Gen}(1^\kappa, pp)$ and $\mathbf{sig} \leftarrow \mathbf{Sign}(sk, m)$.

Security: The CRYSTALS-Dilithium signature scheme is EUF-CMA-secure if the advantage of any PPT adversary $\mathcal{A}$ without knowing sk to forge a signature $\mathbf{sig}^*$ is that $Adv_{\mathcal{A}}^{Sig}(\kappa) \leq \varepsilon(\kappa)$. $\mathcal{A}$ can output a list of query messages $m_1, \ldots, m_Q$, and query $\mathbf{Gen}(1^\kappa, pp)$ and $\mathbf{Sign}(\mathbf{sk}, m)$. In addition, according to Proposition 6.1. in [89], CRYSTALS-Dilithium signature scheme can provide the security properties of signatures beyond unforgeability, i.e., exclusive ownership, message-bound, and non re-signability, if CRYSTALS-Dilithium employ a collision-resistant and non-malleable hash.

In QPASE, both the user and servers employ the user's fixed **pk** for verification. Hence, a signature meeting EUF-CMA-secure provides the necessary security and eliminates the need for encryption together with the message.

F. Symmetric Encryption

In our QPASE, we employ symmetric encryption to protect the outsourced data and retrieval index. The definition of a symmetric encryption algorithm is as follows:

Definition 14: A symmetric encryption for a message space $\mathcal{M}$ and a key space $\mathcal{K}$ is a tuple of PPT algorithms as follows:

- $ct \leftarrow \mathbf{Enc}(k, m)$ *is a encryption algorithm that picks* $k \leftarrow \mathcal{K}$ *and* $m \leftarrow \mathcal{M}$ *and outputs a ciphertext* ct.
- $m \leftarrow \mathbf{Dec}(k, ct)$ *is a decryption algorithm that employs the same key* k *as* $\mathbf{Enc}(k, m)$ *and inputs a* ct. *Then, the decryption algorithm outputs the plaintext* m.

To the best of our knowledge, the Grover algorithm [90] is currently the most effective algorithm for symmetric cryptosystems under the quantum computing model. The Grover algorithm can reduce the exhaustive search practice of 2^n-bit keys to $\sqrt{2^n}$. In other words, AES-256 still has 128-bit security in the exhaustive search of quantum computers.

III. Scheme Architecture and Security Model

A. Scheme Architecture

We first model the functionality and formally define the security of QPASE. QPASE in Fig. 3 comprises two entities.

- **User:** The user U, acting as the data owner, possesses an identity ID_u and a human-memorable password psw_u. To register with a set of servers S, U provides (ID_u, psw_u) and subsequently log in with the correct password to obtain the user-specific key K_u. Then, U can derive the data search key and either outsource or retrieve data using symmetric searchable encryption (SSE).
- **Server:** A set of servers $S = \{S_1, \ldots, S_N\}$ stores the user's registration information. During the outsourcing phase, at least t servers assist U in generating a user-specific key and provide authentication for U.

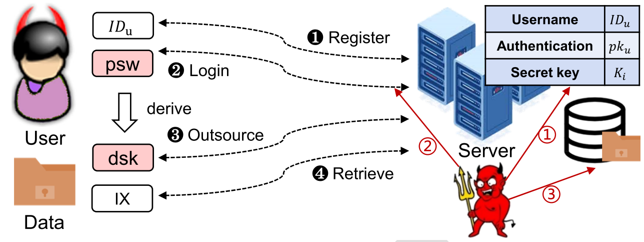

Fig. 3. Exemplary overview of QPASE. For better illustration, the user U selects the (ID_u, psw_u), and ❶ registers with servers and generates characteristic pk_u, which is stored on each server. ❷ U logins to the servers using psw_u, and derives the data search key dsk with the assistance of secret key K_i. Now, U can ❸ outsource or ❹ retrieve data. The adversary can perform ① offline password-guessing attacks, ② online password-guessing attacks, and ③ chosen keyword attacks to breach the security of QPASE.

The scheme proposed in this paper primarily addresses two key issues. The first issue pertains to the interaction between U and S, encompassing the processes of registration and authentication. Given the potential for data loss due to typos, it is crucial to authenticate and verify the correctness of the user's password. In addition, explicit authentication makes S weaken the impact of online password-guessing attacks through the rate limit. The other issue concerns the SSE. After authentication, U derives K_u with the assistance of at least t servers, and executes SSE. To achieve this, we extend the dual-server PASE of Chen et al. [14] to a multi-server PASE with active updates and quantum resistance. Specifically, we formally define the QPASE functionality as follows.

Definition 15: A quantum-resistant password-authenticated symmetric searchable encryption (QPASE) must contain the following five polynomial algorithms:

- $pp \leftarrow \mathbf{Setup}(1^\kappa)$ *generates a set of parameters* pp *by input* κ. S_i *generates the server side key* k_i *via the DKG algorithm in Definition 12.*
- ***Register*** *is executed between* U *(running interactive algorithm* **RegisterU**) *and* S *(running interactive algorithm* **RegisterS**) *as following specification:*
  $(\mathbf{pk}, \mathbf{sk}) \leftarrow \mathbf{RegisterU}(pp, ID_u, psw_u, S_i)$: *The user* U *chooses a password* psw_u *and interacts with* N *servers to obtain a key pair* $(\mathbf{pk}, \mathbf{sk})$. U *sends* **pk** *to all servers and remembers* (ID_u, psw_u).
  $\{0,1\} \leftarrow \mathbf{RegisterS}(pp, ID_u, k_i)$: S_i *assists* U *with registration. If the registration fails, outputs 0. Otherwise, it outputs 1 and stores the authentication information.*
- ***Login*** *is executed between* U *(running interactive algorithm* **LoginU**) *and* S *(running interactive algorithm* **LoginS**) *according to the following specification:*
  $(K_u, pk'_u, sk'_u) \leftarrow \mathbf{LoginU}(pp, ID_u, psw_u, S_i)$: U *interacts with at least* t*-many* S_i *using a registered password* psw_u *to recover* K_u. *Then,* U *derives a key pair* (pk'_u, sk'_u) *with* K_u *for check the correctness of* psw_u. *If confirming that* psw_u *is correct,* U *achieves login and can outsource or retrieve data.*
  $\{0,1\} \leftarrow \mathbf{LoginS}(pp, ID_u, k_i, pk_u)$: S_i *assists* U *to recovers* K_u *and check the valid of user's credentials. If* U *is valid,* S_i *outputs 1 and provides data outsourcing and retrieval services. Otherwise, outputs 0 and aborts.*

- ***Outsource*** *is executed between U (running interactive algorithm* **OutsourceU***) and S (running interactive algorithm* **OutsourceS***) as following specification:* $(Ct, C, i, sig_{ct}) \leftarrow$ **OutsourceU**$(pp, K_u, sk_u, w, d, S_i)$*: U inputs* K_u*,* sk_u*, the outsourced data d, and the keyword w to generate the data search key* dsk*. U output ciphertext Ct of searchable index by dsk, ciphertext C of d by* K_u *and a signature* sig_{ct} *of Ct.* $\{0, 1\} \leftarrow$ **OutsourceS**(pp, pk_u, k_i)*:* S_i *verifies* sig_{ct} *by* pk_u*. If* sig_{ct} *is valid,* S_i *output 1 and receives* (Ct, C, i)*. Otherwise,* S_i *outputs 0 and aborts.*
- ***Retrieve*** *is executed between U (running interactive algorithm* **RetrieveU***) and S (running interactive algorithm* **RetrieveS***) after* ***Login*** *as follows:* $(dsk, sig_{dsk}, d) \leftarrow$ **RetrieveU**$(pp, K_u, sk_u, pk_u, w, S_i)$*: U input* K_u *and w to recover the dsk and retrieve the outsourced data by interacting with S. Then, U decrypts C to get the plaintext d of data.* $\{\mathcal{L}_i, \perp\} \leftarrow$ **RetrieveS**(pp, pk_u, Ct, C)*: S assists certified authentic U to retrieve the outsourced data. S returns the search list* $\mathcal{L}_i$*. Otherwise,* S_i *aborts.*

Correctness: The correctness of the QPASE means that all data under the keyword can be retrieved whenever the registered user inputs the correct password in **Login**.

Definition 16 (Correctness): Let **IX** *denote all data under the keyword w and* $C \in$ **IX***.* $pp \leftarrow \mathsf{Setup}(1^\kappa)$*. The probability* $Pr[C \in \mathbf{IX}] = 1$ *iff U executes the following algorithm:*

$$(\mathbf{pk}, \mathbf{sk}) \leftarrow \mathbf{RegisterU}(pp, ID_u, psw_u, S_i);$$
$$(K_u, pk'_u, sk'_u) \leftarrow \mathbf{LoginU}(pp, ID_u, psw_u, S_i);$$
$$(Ct, C, i, sig_{ct}) \leftarrow \mathbf{OutsourceU}(pp, K_u, sk_u, w, d, S_i);$$
$$(dsk, sig_{dsk}, d) \leftarrow \mathbf{RetrieveU}(pp, K_u, sk_u, pk_u, w, S_i).$$

and all servers output 1 by executing the following algorithm: $1 \leftarrow \langle \mathbf{RegisterS}(pp, ID_u, k_i), \mathbf{LoginS}(pp, ID_u, k_i, pk_u), \mathbf{OutsourceS}(pp, pk_u, k_i), \mathbf{RetrieveS}(pp, pk_u, Ct, C)\rangle$.

B. Security Model

We consider adversaries with quantum computing capabilities to mount various attacks to capture outsourced data. For our QPASE scheme, we consider three security goals: quantum resistance, authentication, and indistinguishability against chosen keyword attacks (IND-CKA). To formally capture the capabilities of an adversary in our QPASE, and specify how the adversary interacts with honest parties, we employ the *Bellare-Pointcheval-Rogaway* (BPR) model [70], where the adversary's capabilities are modeled through queries and define a series of security notions. We briefly recall the BPR model as follows. Recalling Definition 15, each $U \in \mathsf{User}$ holds a password psw, while $S_i \in S$ holds the server side key k_i. Let U^i and S^j denote user instances and key server instances, respectively, where $i, j \in \mathcal{Z}$. We denote any kind of instance by $I \in \mathsf{User} \cup \mathsf{Server}$.

1) Adversarial Model: We consider the adversary $\mathcal{A}$ with quantum computing capabilities can fully control the external network, which implies that $\mathcal{A}$ is free to manipulate messages and adaptive request any session keys. Moreover, for N key servers in the scheme, we define that the adversary can simultaneously corrupt at most $t' < t < N$ servers.

2) Queries: $\mathcal{A}$ interacts with the participants by using oracle queries that simulate the adversary's capabilities in a real attack. The query models available to $\mathcal{A}$ are as follows.

- Execute(U^i, S^j) captures a passive attack, such as eavesdropping. The output of execution consists of the messages exchanged during the honest execution.
- Send(I, m) captures an active attack, in which $\mathcal{A}$ sends a message to instance I and outputs the response of I to handle the message according to the protocol.
- Text(I) is used to define the semantic security of the session key and is only allowed to query once. This query outputs a random bit b in the real-or-random flavor. If $b = 1$, $\mathcal{A}$ gets the actual session key. Otherwise, $\mathcal{A}$ obtains a random key of the same size.
- Reveal(I) allowed $\mathcal{A}$ obtains the session key of I.
- Corrupt(I) captures the corrupt attack. If $I = U$, it outputs the password psw_u. If $I = S^i$, it outputs k_i.

3) Partnering: Let sid denotes the session identifier and pid denotes the partner identifier. For the U^i and S^j in an instance I, we said they are partnered if the following conditions are satisfied: 1) Both of them have accepted; 2) $sid_{U^i} = sid_{S^j} = sid$; 3) $pid_{U^i} = S$ and $pid_{S^j} = U$.

4) Freshness: I is fresh if the following conditions are true: 1) I has accepted and computed a session key.; 2) Neither I nor its partner has been asked for a query Reveal(I).

5) Semantic Security: In the sequences of games, $\mathcal{A}$ can ask a polynomial number of query Execute(U^i, S^j), Send(I, m), and Reveal(I). Finally, $\mathcal{A}$ asks a query Text(I) to get a guess bit b' for the bit b involved. For any PPT $\mathcal{A}$, the advantage holds that $Adv_{\mathcal{A}}^{\mathsf{QPASE}} = 2Pr[b' = b] - 1$. In the BPR model, each entity can execute the PASE with all of the servers multiple times. Furthermore, the BPR model permits any entity to instantiate unlimited instances but each instance is used only once. $\mathcal{A}$ is capable of accessing different instances of entities.

Let L denote a list maintained by the experiment. We define that the adversary $\mathcal{A}$ can access the following oracles.

- **Challenge**(b, sid_i, w_i): The oracle aborts if $(sid_i^* \geq 0) \vee (sid_i \geq sid_j) \vee ((sid_i, w_i) \in L)$. Otherwise, it set $sid_{i^*} \leftarrow sid_i$ and access **OutU**(sid_{i^*}, w_b, C^*).
- **Reg**(i): The experiment first initializes D_{i,sid_j} as a database. Then, it randomly picks psw satisfy $(sid_i, psw, i) \notin L$. $\mathcal{A}$ interacts with the honest user and server (oracle) as the corrupted server. After access, the experiment records $L[sid_i] \leftarrow (i, psw.auth_i, K_u)$, delivers j to $\mathcal{A}$ and set $j \leftarrow j + 1$.
- **LoginU**(i): The experiment initializes D_{i,sid_j} as a database. Then, it randomly picks psw satisfy $(sid_i, psw, i) \notin L$. $\mathcal{A}$ interacts with the honest user and server (oracle) as the corrupted server. After access, the experiment records $L[sid_i] \leftarrow (i, psw.auth_i, K_u)$, delivers j to $\mathcal{A}$ and set $j \leftarrow j + 1$.
- **LoginS**(sid_i): The oracle aborts if $sid_i \geq sid_j$. Otherwise, it gets $(i, psw.auth_i) \leftarrow L[sid_i]$. $\mathcal{A}$ interacts with the honest server as the corrupted server.
- **OutU**(sid_i, w, C): The oracle aborts if $sid_i \geq sid_j$. Otherwise, it gets $(i, psw.auth_i, K_u) \leftarrow L[sid_i]$. $\mathcal{A}$ interacts with the honest user and server (oracle) as the corrupted server. In $Exp_{PASE,\mathcal{A}}^{Auth}(\kappa)$, the oracle additionally computes $L \leftarrow L \cup (sid_i, w, C)$.

$Exp^{Auth}_{PASE,\mathcal{A}}(\kappa)$
$Auth_i \leftarrow \emptyset; sid_j \leftarrow 0; params \leftarrow Setup(1^\kappa);$
$(sid_{i^*}, w^*, ix^*) \leftarrow \mathcal{A}^{oracle(\cdot)}(\mathbf{param})$
$IX \leftarrow \mathbf{Retrieve}(sid_{i^*}, w^*);$
$(sid_{i^*}, w^*, ix^*) \notin L \wedge (ix^* \in IX)$ return 1
else rutrun 0
$Exp^{IND-CKA-b}_{PASE,\mathcal{A}}(\kappa)$
$Auth_i \leftarrow \emptyset; sid_i \leftarrow (-1); sid_j \leftarrow 0;$
$L \leftarrow \emptyset; \mathbf{param} \leftarrow Setup(1^\kappa);$
$b' \leftarrow \mathcal{A}^{oracle(\cdot)}(\mathbf{param});$
return b'

Fig. 4. PASE security experiments.

- **OutS**(sid_i): The oracle aborts if $sid_i \geq sid_j$. Otherwise, it gets $(i, psw.auth_i, K_u) \leftarrow L[sid_i]$. $\mathcal{A}$ interacts with the honest server as the corrupted server.
- **RetU**(sid_i, w): The oracle aborts if $(sid_i \geq sid_j) \vee (sid_i = sid_{i^*}) \vee (w \in \{w_i\})$. Otherwise, it gets $(i, psw.auth_i, K_u) \leftarrow L[sid_i]$. $\mathcal{A}$ interacts with the honest user and server (oracle) as the corrupted server. In the IND-CKA experiment, if $(sid_{i^*} = -1)$ the oracle additionally computes $L \leftarrow L \cup (sid_i, w)$.
- **RetS**(sid_i): The oracle aborts if $sid_i \geq sid_j$. Otherwise, it gets $(i, psw.auth_i, K_u) \leftarrow L[sid_i]$. $\mathcal{A}$ interacts with the honest server as the malicious entities.

6) Quantum Resistance: To mitigate the potential impact of Shor's [51] influential quantum attack algorithms, QPASE is designed based on the learning with errors problem. Shor's algorithm can efficiently solve large integer factorization and discrete logarithm problems on quantum computers. By leveraging the computational hardness of DLWE, the QPASE scheme aims to provide a secure cryptographic solution that is resistant to quantum attacks. Concretely, we construct the QPASE scheme based on Decision-LWE$_{n,q,\chi,m}$ in Definition 4, i.e., for any PPT $\mathcal{A}$, the advantage holds that $Adv^{\mathbf{DLWE}}_{QPASE}(\kappa) = |Pr[1 \leftarrow \mathcal{A}(\mathcal{Z}_q^{m\times n}, q, n, \chi, A, b)] - Pr[1 \leftarrow \mathcal{A}(\mathcal{Z}_q^{m\times n}, q, n, \chi, r_1, r_2)]| \leq \varepsilon(\kappa)$.

7) Authentication: In the **Login** of QPASE, we follow part of the experiment $Exp^{auth}_{PASE,\mathcal{A}}(k)$ outlined by Chen et al. [14], as depicted in Fig. 4, except that we employ more servers in our scenario. Notably, $\mathcal{A}$ making at most q_s online attacks, the adversary's advantage Adv is denoted as $q_s(\kappa)/|\mathcal{D}| + \varepsilon(\kappa)$ for all dictionary sizes $|\mathcal{D}|$ in the existing uniform-model. Recent research [74], [75] provided a rigorous analysis to constrain the adversary's advantage as $C' \cdot q^{s'}_{send}(\kappa) + \varepsilon(\kappa)$ for the Zipf parameters C' and s', with considering the password distribution follows the Zipf-distribution. We show that the advantages of the adversary $\mathcal{A}$ are underestimated in the uniform model in Section VI. For any PPT $\mathcal{A}$ making at most q_{send} online attacks, the advantage of $\mathcal{A}$ holds that

$$Adv^{\mathbf{Auth}}_{QPASE}(\kappa) = Pr[1 \leftarrow Exp^{Auth}_{\mathcal{A}}(k)] \leq C' \cdot q^{s'}_s(\kappa) + \varepsilon(\kappa).$$

8) IND-CKA: In the IND-CKA property of QPASE, we follow the part of the experiment $Exp^{IND\text{-}CKA\text{-}b}_{PASE,\mathcal{A}}(k)$ in [14] as shown in Fig. 4 except that we prefer the CDF-Zipf distribution [74], [75], and the attacker's advantage can be formulated as $Adv^{\mathbf{IND\text{-}CKA}}_{QPASE,\mathcal{A}}(\kappa) = Pr[b' = b : b' \leftarrow Exp^{IND\text{-}CKA\text{-}b}_{PASE,\mathcal{A}}(k)] - \frac{1}{2}$. A QPASE scheme is called IND-CKA-secure if the probability $Adv^{\mathbf{IND\text{-}CKA}}_{QPASE,\mathcal{A}}(\kappa) \leq C' \cdot q^{s'}_s(\kappa) + \varepsilon(\kappa)$.

IV. QPASE: Our New Scheme

In this section, we present a detailed description of our QPASE including five phases: **Setup**, **Register**, **Login**, **Outsource**, and **Retrieve**. Our scheme is secure in a semi-honest setting with a secure channel. Specifically, Fig. 5 illustrates the phases **Register** and **Login**, and Fig.6 illustrates the phases **Outsource** and **Retrieve**. Moreover, we provide a server key update phase to resist perpetual leakage.

A. Setup

During the setup phase, execute the algorithm $pp \leftarrow$ **Setup**(1^κ). Specifically, with the security parameter κ, generate public parameters $\mathbf{a} \in R_q^{1\times\ell}$. For the parameter $\sigma > 0$ and any $\mathbf{c} \in \mathcal{R}^m$ are defined as Gaussian distributions defined in Definition 3. Choose parameters params $m > cklog(q)$ and $q \geq poly(k)(\sqrt{logk})$. Let μ be an upper limit that a user fails to pass S_i authentication. We set an upper limit μ as the number of login requests issued by a user in an era. *Enc* is a symmetric encryption algorithm and *Dec* denote corresponding decryption algorithm. Set $N \leq \frac{1}{4}log2\frac{L\cdot\ell\cdot n - \sigma\sqrt{n}}{\sigma\sqrt{n}-1}$ as the a total number of servers and t is the threshold number. $H : \{0,1\}^* \rightarrow \mathcal{Z}_q^n$ is a collision-resistance hash function. Let $\mathcal{K}_u$ and $\mathcal{K}_s$ denote the user and server key spaces, respectively. Each S_i generates the server-side key k_i via the DKG algorithm in Definition 12. $\mathsf{HKDF} : R_q \times \mathcal{W} \rightarrow \mathcal{K}_u$. $\mathsf{PRF} : \mathcal{K}_u \times \{0,1\}^{\mathcal{K}} \rightarrow \{0,1\}^{\mathcal{K}}$. We employ the signature in Definition 13 to prevent $\mathcal{A}$ from tampering with the information. For conciseness, we do not explicitly show the signature and verification.

B. Register

The **Register** phase allows the unregistered user U to register with a set of servers $S = \{S_1, \ldots, S_N\}$ using the user's ID_u and a human-memorable password psw_u. U need to convert psw_u to binary $\mathbf{x} = (x_1, \ldots, x_L) \in \{0,1\}^L$. The registration phase needs a secure channel.

- 1. U inputs $(ID_u, \mathbf{x})$ and interacts with each S_i to execute algorithm Π_{TOPRF} in Fig. 1 to get $K_u = F_K(\mathbf{x})$. Each S_i checks whether ID_u is a duplicate. If yes, S_i notifies U. Otherwise, S_i stores ID_u.
- 2. U executes $(pk_u, sk_u) \leftarrow Gen(1^\kappa, K_u)$ in Definition 13 and sends pk_u to S_i, where $i \in [1, N]$.
- 3. S_i initiates $\mu_u = 0$ and securely storage ID_u, pk_u, k_i, μ_u for a subsequently authenticating.
- 4. U only needs to secure storage ID_u and psw_u.

C. Login

In the **Login** phase, the user U executes **LoginU** to recover K_u with at least t-many servers and achieve authentication. U inputs ID_u and psw_u, and computes $\mathbf{x} = (x_1, \ldots, x_L) \in \{0,1\}^L$ from psw_u.

- L1. For $i \in [1, t]$, U uses $(ID_u, \mathbf{x})$ to execute algorithm Π_{TOPRF} with S_i to get $K_u = F_K(\mathbf{x})$. S_i checks whether $\mu_u < \mu$. If no, S_i aborts. Otherwise, S_i set $\mu_u := \mu_u + 1$ and assists U to recover K_u. Then, S_i returns pk_u to U.
- L2. U computes $(pk'_u, sk'_u) \leftarrow Gen(1^\kappa, K_u)$ checks whether $pk'_u = pk_u$. If $pk'_u \neq pk_u$, U re-inputs the

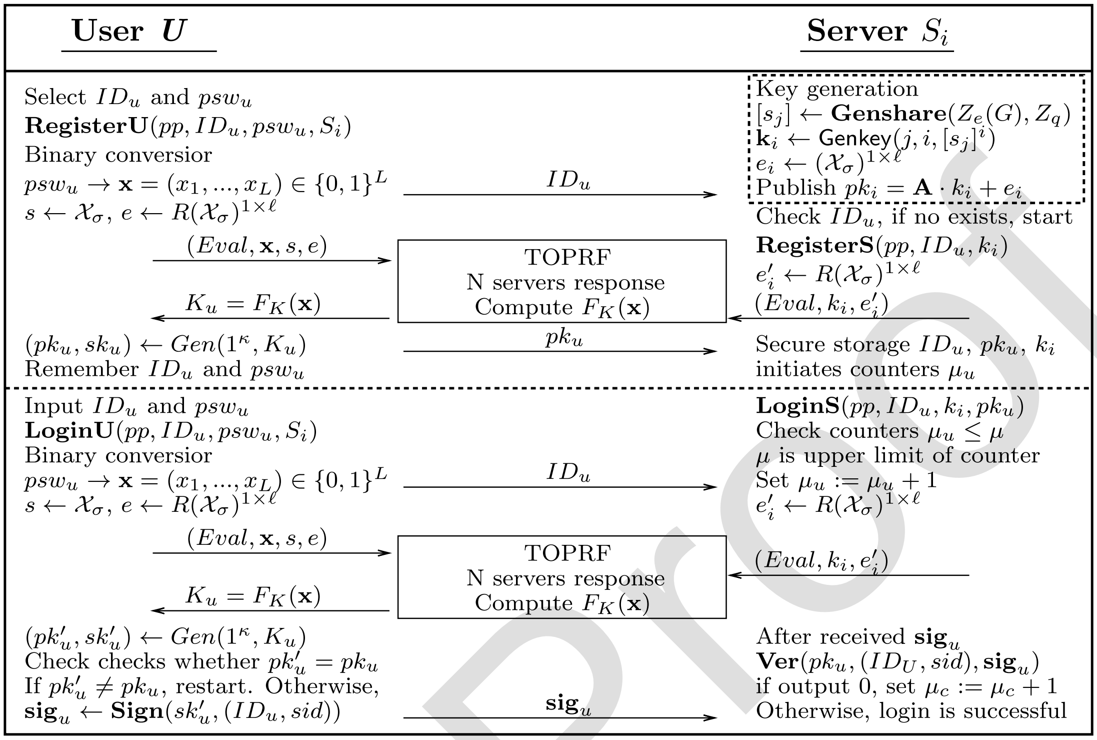

Fig. 5. The **Register** and **Login** of the QPASE.

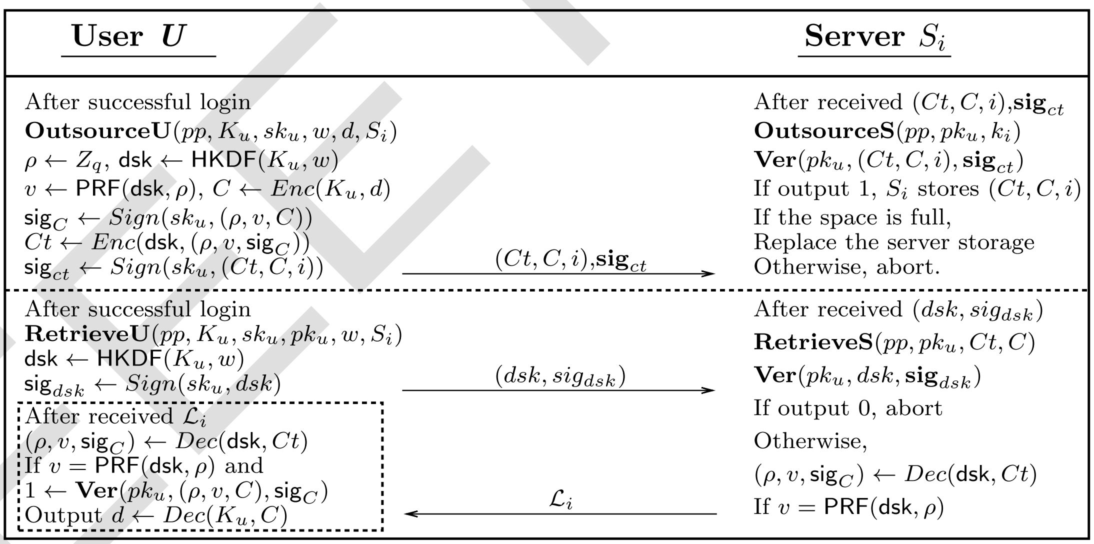

Fig. 6. The **Outsource** and **Retrieve** of the QPASE.

password and executes L1. Otherwise, U executes $\mathbf{sig}_u \leftarrow \mathbf{Sign}(sk'_u, (ID_u, sid))$ and sends $\mathbf{sig}_u$ to S_i.

- L3. S_i verify **sig** with $\mathbf{Ver}(pk_u, m, \mathbf{sig}_u)$. If output 1, S_i allows U to upload outsourcing data or retrieval data. Otherwise, S_i aborts and set $\mu_u := \mu_u + 1$.

D. Outsource

In the **Outsource** phase, the authenticated user executes **OutsourceU** to upload the outsourcing data d.

- O1. U selects $\rho \leftarrow \mathcal{Z}_q$ and computes the data search key $\mathsf{dsk} \leftarrow \mathsf{HKDF}(K_u, w)$, $v \leftarrow \mathsf{PRF}$ (dsk, ρ), $C \leftarrow Enc(K_u, d)$ $\mathbf{sig}_C \leftarrow \mathbf{Sign}(sk_u, (\rho, v, C))$, and $Ct \leftarrow Enc(\mathsf{dsk}, (\rho, v, \mathbf{sig}_C))$, $\mathbf{sig}_{Ct} \leftarrow \mathbf{Sign}(sk_u, (Ct, C, i))$. U sends (Ct, C, i) and $\mathbf{sig}_{Ct}$ to arbitrary S_i.
- O2. S_i executes $\mathbf{Ver}(pk_u, (Ct, C, i), \mathbf{sig}_{Ct})$. If output 1, S_i stores (Ct, C, i) in its database. Notably, each S_i provides M storage space for users.

E. Retrieve

In the **Retrieve** phase, the authenticated user executes **RetrieveU** to retrieve and recover d.

- R1. After **Login**, U has $K_u = F_K(\mathbf{x})$ and (pk_u, sk_u). U computes $\mathsf{dsk} \leftarrow \mathsf{HKDF}(K_u, w)$, $\mathbf{sig}_{dsk} \leftarrow \mathbf{Sign}(sk_u, dsk)$ and sends them to S_i, where $i \in [1, N]$.
- R2. S_i executes $\mathbf{Ver}(pk_u, dsk, \mathbf{sig}_{dsk})$. If outputs 1, S_i computes $(\rho, \upsilon, \mathbf{sig}_C) \leftarrow Dec(\mathsf{dsk}, Ct)$. S_i initializes a set $\mathcal{L}_i \leftarrow \emptyset$. If $1 \leftarrow Ver(pk_u, (\rho, \upsilon, C), \mathbf{sig}_C)$ and $\upsilon = PRF(\mathsf{dsk}, \rho)$, S_i adds (Ct, C, i) to $\mathcal{L}_i$. When **Retrieve** is complete, S_i sends $\mathcal{L}_i$ to U.
- R3. U receives all $\mathcal{L}_i$ and executes $(\rho, \upsilon, \mathbf{sig}_C) \leftarrow Dec(\mathsf{dsk}, Ct)$. If $1 \leftarrow Ver(pk_u, (\rho, \upsilon, C), \mathbf{sig}_C)$ and $\upsilon = PRF(\mathsf{dsk}, \rho)$, U decrypts the corresponding C, i.e., $d \leftarrow Dec(K_u, C)$, to obtain data.

F. Server Key Update

It is necessary to update the server key to resist perpetual leakage [71]. The server key update phase is performed internally by the server without user participation, and the update does not affect the server's authentication and searchable encryption. Specifically, each server S_i updates the server-sid key k_i within a fixed period called an epoch. According to the Definition 12 and the server key update scheme of Jiang et al. [68], the specific operations during the server key update phase are as follows:

- 1. Let $q = e(S)$, S_j randomly chooses a polynomial $[F] = \sum_{k=1}^{t-1} \alpha_k X^k$, where $[F]^0 = 0$.
- 2. At least t-many S_j computes $h = \{H_j(\alpha_k)\}$, $F_j^i = [F]_j^i \ mod \ q$, where $k \in [1, t-1]$, $i \in [1, N]$, $j \in [1, t]$.
- 3. S_j broadcast the message $F_\upsilon^{(\omega)} = \{j, \omega, h, \ E(i, F_j^i)\}$ and $\mathbf{sig}_j \leftarrow \mathbf{Sign}(sk_j, (id, F_\upsilon^{(\omega)}))$. S_j sends F_j^i and $\mathbf{sig}_j$ to S_i, where $i \in [1, N]$, $j \in [1, t]$.
- 4. S_i decrypts the shares intended $\{F_j^i\}_{j \in [1,t]}$ for S_i and verifies the correctness of the share by checking the equivalent $H(F_j^i) = \sum_{k=1}^{t-1} H(\alpha_k)^{i^k}$ and executing $\mathbf{Ver}(pk_i, (id, [F]_j^i), \mathbf{sig}_j)$. If **Ver** output 1 and the equation holds, each S_i computes a new $k_i' = k_i + \sum_{j=1}^{t} \lambda_{i,j} [F]_j^i \ mod \ q$. After receiving $[F]_j^i \ mod \ q$ from S_j, each S_i executes $\mathbf{Ver}(pk_i, (id, [F]_j^i), \mathbf{sig}_j)$. If **Ver** output 1, each S_i computes a new $k_i' = k_i + \sum_{j=1}^{t} \lambda_{i,j} [U]_j^i \ mod \ q$.
- 5. S_i recalculates $pk_i' = \mathbf{A} \cdot k_i' + e_i$, where $\mathbf{A} \in R_q^{1 \times \ell}$ is a public matrix and $e_i \in R_q^{1 \times \ell}$, and resets μ_u to begin $(\omega + 1)$-th epoch.

Lemma 2: Let **IX** *denote all data under the keyword w and $C \in$* **IX**. *$pp \leftarrow \mathsf{Setup}(1^\kappa)$. The probability $Pr[C \in \mathbf{IX}] = 1$ iff the quantum-secure* HKDF*, the EUF-CMA signature, and the symmetric encryption Enc is correctness.*

Proof: In **OutsourceU**, there are $\mathsf{dsk} \leftarrow \mathsf{HKDF}(K_u, w)$, $\upsilon \leftarrow \mathsf{HKDF}(\mathsf{dsk}, \rho)$, $\mathbf{sig}_C \leftarrow \mathbf{Sign}(sk_u, (\rho, \upsilon, C))$, and $Ct \leftarrow Enc(\mathsf{dsk}, (\rho, \upsilon, \mathbf{sig}_C))$. In **Retrieve**, iff the symmetric encryption Enc is correctness, there is $(\rho, \upsilon, \mathbf{sig}_C) \leftarrow Dec(\mathsf{dsk}, Ct)$. Similarly, iff the EUF-CMA signature is correctness, there are $1 \leftarrow Ver(pk_u, (\rho, \upsilon, C), \mathbf{sig}_C)$. Each S_i adds (Ct, C, i) to $\mathcal{L}_i$ and $Pr[C \in \mathbf{IX}] = 1$. Vice versa.

Lemma 3: Let $[F]$ and $[G]$ denote the master key polynomial and the update polynomial, respectively. At the end of the era, each S_i executes the server key update protocol to renew its secret key k_i without changing the master secret key K.

Proof: According to definition 12, we know that $K = \sum_{i=1}^{n} [F]_i^0 = \sum_{i=1}^{n} f_i(0)$. Suppose that $K' = \sum_{i=1}^{n} f_i'(0)$. Since $f_i'(x) = f_i(x) + G_i(x)$, we have

$$K' = \sum_{i=1}^{n} f_i'(0) = \sum_{i=1}^{n} f_i(0) + U_i(0) = \sum_{i=1}^{n} [S]_i^0 + [U]_i^0$$

$$= \sum_{i=1}^{n} [s]_i^0 + 0 = \sum_{i=1}^{n} [s]_i^0 = K.$$

G. Multiple Keywords

Notice that the QPASE construction we gave uses only one keyword in the **Outsource** and **Retrieve** phases. Our scheme can be extended to multiple keywords to construct associative data. Let $\mathbf{w} = (w_1, \ldots, w_k)$ be a set of keywords for a series of C. In the **Outsource** phase, O1. U selects $\rho \leftarrow \mathcal{Z}_q$ and computes $\mathsf{dsk}_j \leftarrow \mathsf{HKDF}(K_u, w_j)$, $\upsilon_j \leftarrow (\mathsf{dsk}_j, \rho)$, $\mathbf{sig}_x \leftarrow \mathbf{Sign}(sk_u, (\rho, \mathbf{v}, C))$, and $Ct \leftarrow Enc(\mathsf{dsk}, (\rho, \mathbf{v}, \mathbf{sig}_C))$, where $\mathbf{v} = (\upsilon_1, \ldots, \upsilon_k)$. U sends (Ct, C, i) to S_i. Inspired by Chen et al. [14], we use a similar method to construct the query. Let $\mathbf{w}' = (w_1', \ldots, w_p')$, $p \leq k$.

In **Retrieve**, R1. U sends $\mathsf{dsk}_j \leftarrow \mathsf{HKDF}(K_u, w_j')$ to S_i, where $i \in [1, N]$, $j \in [1, p]$. R2. S_i computes $(\rho, \mathbf{v}, \mathbf{sig}_C) \leftarrow Dec(Ct)$. Initialize a set $\mathcal{L}_i \leftarrow \emptyset$. If $Ver(pk_u, (\rho, \mathbf{v}, C), \mathbf{sig}_C)$ and $\mathbf{v} = PRF(\mathsf{dsk}_j, \rho)$, S_i adds (Ct, C, i) from the database to $\mathcal{L}_i$. The search query includes three cases:

- Conjunctive queries $w_1' \wedge \ldots \wedge w_p'$ if $\mathbf{v} = \mathbf{v}'$.
- Disjunctive queries $w_1' \vee \ldots \vee w_p'$ if $|\mathbf{v} \cap \mathbf{v}'| > 0$.
- Subset queries $(w_1', \ldots, w_p') \subseteq \mathbf{w}$ if $\mathbf{v}' \subseteq \mathbf{v}$.

V. Security Analysis

In the following, we prove the security of our scheme within the formal model defined in Section III, assuming the Decision-LWE$_{n,q,\chi,m}$ problem is intractable.

A. Formal Security Analysis of QPASE

We prove the security of our lattice-based PASE scheme based on subsection III-B with the standard game-based proof.

Theorem 2 (Authentication): In QPASE, let $\mathcal{A}$ get pp and access q_s times query. The frequency distribution of password dictionary $\mathcal{D}$ follows Zipf's law [74], [75]. For any PPT $\mathcal{A}$, the advantage of disrupting authentication that

$$Adv_{QPASE,\mathcal{A}}^{\mathbf{Auth}}(\kappa) \leq 2C' \cdot q_s^{s'}(\kappa) + Adv_{\mathcal{A}}^{\mathsf{DLWE}}(\kappa)$$
$$+ Adv_{\mathcal{A}}^{1D-SIS} + Adv_{\mathcal{A}}^{Sig}(\kappa) + \varepsilon(\kappa).$$

We employ the Zipf model of the Taobao password distribution in Fig. 7, where $|\mathcal{D}| = 15{,}072{,}667$, $C' = 0.0166957$, and $s' = 0.194179$.

Proof: **Game** G_0^{Auth}. This game simulates the real environment between the protocol challenger and the passive adversary $\mathcal{A}$. $\mathcal{A}$ obtains $\mathbf{A}, \mathbf{x}^*, \mathbf{x}_{k_i}^*, \mathbf{pk}_u, S(\mathbf{pk}_i)$. The simulator initializes $Auth_i$, sid_j, L, and pp as defined in the real security game $G_{QPASE,\mathcal{A}}^{Auth}(\kappa)$. $\mathcal{A}$ access oracle including $\mathbf{Reg}(i)$, $\mathbf{LoginU}(i)$, and $\mathbf{LoginS}(sid_i)$, which is defined in Section III-B. Specifically, the simulator $\mathcal{S}$ initializes $\mathcal{L}_{sid_j}^i \leftarrow \emptyset$ and plays U and S_i. $\mathcal{A}$ interacts with the honest user and server (oracle) as the corrupted server. After access, $\mathcal{S}$

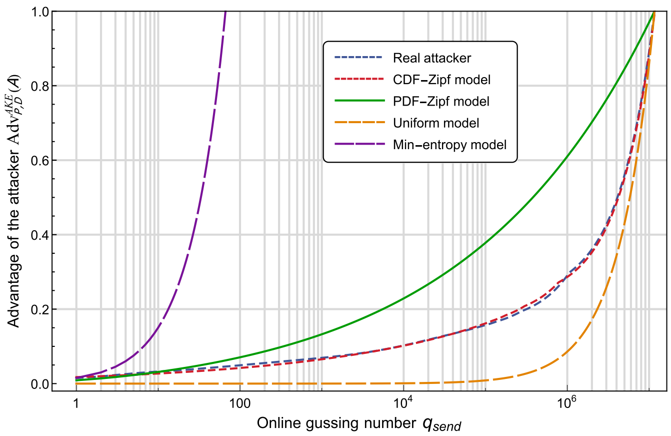

Fig. 7. Online guessing advantages of the real attacker and attackers modeled by the CDF-Zipf, PDF-Zipf, uniform and min-entropy distributions, respectively (using the 15,072,667 passwords leaked from Taobao). The overlap of the CDF-Zipf attacker with the real one indicates well prediction.

records $L[sid_j] \leftarrow (i, psw.auth_i)$, delivers sid_j to $\mathcal{A}$ and set $j \leftarrow j+1$. We have

$$Adv^{\mathbf{Auth}}_{QPASE,\mathcal{A}}(\kappa) = Pr[succ_0^{Auth}].$$

Game G_1^{Auth}*: This game is similar to* G_0^{Auth} except that $\mathcal{S}$ executes the oracles **Login**(i) and **LoginS**(sid_i) and s is fresh in every session. Thus, we have

$$Pr[succ_1^{Auth}] = Pr[succ_0^{Auth}].$$

Game G_2^{Auth}*: This game is similar to* G_1^{Auth} except that $\mathcal{A}$ access **Reg**(i), **Login**(i), **LoginS**(sid_i). $\mathcal{A}$ sets $\mathbf{b} = \lfloor \mathbf{a}\cdot r + \mathbf{a}^F\mathbf{x} + 2e\rceil_p$ and $\mathbf{b}_{k_i} = \lfloor \mathbf{b}\cdot k_i + 2e_i'\rceil_p$. According to Lemma 4, the advantage of $\mathcal{A}$ is $Adv^{\mathsf{DLWE}}_{\mathcal{A}}(\kappa)$. Thus, the views in G_0^{Auth} and G_1^{Auth} are computationally indistinguishable for any PPT $\mathcal{A}$, and there is

$$Pr[succ_2^{Auth}] - Pr[succ_1^{Auth}] = Adv^{\mathsf{DLWE}}_{\mathcal{A}}(\kappa).$$

Game G_3^{Auth}*: This game is identical to* G_2^{Auth} except that $\mathcal{A}$ sets $\mathbf{b}_{k_i} = \lfloor \mathbf{b}\cdot k_i + 2e_i'\rceil_p$ and $K_u = \lfloor(\sum_{i=1}^{t}\lambda_i\cdot \mathbf{b}_{\mathbf{k}_i} - \sum_{i=1}^{t}\lambda_i\cdot pk_i\cdot r\rceil_p$. In the setup phase, $\mathcal{A}$ plays malicious server S^*. S^* computes pk^* from k^* and publishes it, where $k_i \leq \sigma\cdot\sqrt{n}$. In the query phase, the simulator randomly selected $r \xleftarrow{R} R_q^{1\times\ell}$ and send to S^*. Waiting for a response of $\mathbf{x}^*_{k_i}$ from S^*. Finally, the honest user U send $F_K(\mathbf{x})$ to $\mathcal{A}$. In real protocol, $\mathbf{x}^*$ generated by the honest user U. The secret value $\mathbf{x}$ is hidden by the encryption algorithm based on Decision-LWE$_{n,q,\chi,m}$. Therefore, $\mathcal{A}$ cannot distinguish a real $\mathbf{x}^*$ from r. Let $\mathbf{x} \xleftarrow{R} R(\chi_\sigma)$ and $e \xleftarrow{R} R(\chi_\sigma)^{1\times\ell}$ are sampled by U. For U executes Π_{TOPRF}, and computes

$$\begin{aligned}F_K(\mathbf{x}) &= \lfloor\sum_{i=1}^{t}\lambda_i\cdot\mathbf{b}_{k_i} - \sum_{i=1}^{t}\lambda_i\cdot pk_i\cdot r \ mod\ q\rceil_p\\ &= \lfloor \mathbf{a}\cdot r\cdot\sum_{i=1}^{t}\lambda_i\cdot k_i + \mathbf{a}_{\mathbf{x}}\cdot\sum_{i=1}^{t}\lambda_i\cdot k_i + 2e\sum_{i=1}^{t}\lambda_i\cdot k_i\\ &\quad + 2\sum_{i=1}^{t}\lambda_i\cdot e_i' - \mathbf{a}\cdot r\cdot\sum_{i=1}^{t}\lambda_i\cdot k_i\\ &\quad - 2r\sum_{i=1}^{t}\lambda_i\cdot e_i \ mod\ q\rceil_p\\ &= \lfloor \mathbf{a}_{\mathbf{x}}\cdot K + 2e'' \ mod\ q\rceil_p\end{aligned}$$

where $e'' = e\cdot K + \sum_{i=1}^{t}\lambda_i\cdot e_i' - r\sum_{i=1}^{t}\lambda_i\cdot e_i$. According to Definition 1, set $\sigma' \gg \max\{L\cdot\ell\cdot\sigma n^{3/2}, \sigma^2 n^2\}$. The coefficient of $\frac{p}{q}\cdot(\mathbf{a}^F(\mathbf{x})\cdot K + 2e'')$ is further than T away from $\mathcal{Z}+\frac{1}{2}$. According to Definition 1, set $T = \frac{p}{q}\left(\sigma'\cdot\sqrt{n} + L\cdot\ell\cdot\sigma n^{3/2}\right) \ll 1$ such that $T \leq \frac{p}{q}\cdot|e''|_\infty$. Therefore, G_2^{Auth} and G_1^{Auth} are computationally indistinguishable. According to Definition 7, we have

$$Pr[succ_2^{Auth}] - Pr[succ_1^{Auth}] = Adv^{1D-SIS}_{\mathcal{A}}.$$

Game G_4^{Auth}*: This game is similar to* G_3^{Auth} except that $\mathcal{A}$ sets $\mathbf{b} = \lfloor \mathbf{a}\cdot r + \mathbf{a}^F\mathbf{x} + 2e\rceil_p$. Concretely, in the setup phase, $\mathcal{A}$ and uniform $pk_{\mathcal{A}} \leftarrow \mathcal{Z}_q^{1\times\ell}$ are generated. Send pk_i to $\mathcal{A}$. Initialize an empty list $\mathbf{Q}$. During the query stage, for each message pk_i, $\mathcal{A}$ extracts $\mathbf{x}_{\mathcal{A}}, e_{\mathcal{A}}$, and queries $\mathbf{x}$. If returns $F_K(\mathbf{x}) \in R_p^{1\times\ell}$ and $F_K(\mathbf{x}) \notin \mathbf{Q}$, sample $\boldsymbol{F}_q \leftarrow R_q^{1\times\ell} \cap \left(\frac{q}{p}\mathbf{y} + R^{1\times\ell}_{\leq\frac{q}{2p}}\right)$ and add $(\mathbf{x}, \boldsymbol{F}_q)$ into $\mathbf{Q}$. Return $\boldsymbol{F}_q$ to $\mathcal{A}$. If returns $F_K(\mathbf{x}) \in R_p^{1\times\ell}$ and $F_K(\mathbf{x}) \in \mathbf{Q}$, set $\boldsymbol{F}_q = F_K(\mathbf{x}) \in R_p^{1\times\ell}$. Choose $e_i^* \xleftarrow{R} \chi_{\sigma'}$ and send $\mathbf{x}^*_{k_i} = pk_u\cdot k_i + e_i^* + \boldsymbol{F}_q$ to $\mathcal{A}$. Each round of queries uses different errors sampled from $R(\chi_{\sigma'}^{1\times\ell})$. In a real protocol, if $\mathcal{A}$ can calculate the correct $\boldsymbol{F}_q$, it can perform the same operation on the message received from the simulator. $\boldsymbol{F}_q$ is sampled by the simulator and the corresponding value $\mathbf{x}^*_{k_i}$. Let $e_{\lfloor\rceil} := \mathbf{y}_q - (q/p)\cdot\mathbf{y} \in R^{1\times\ell}_{\leq\frac{q}{2p}}$, we have $F_K(\mathbf{x}) = \lfloor\frac{p}{q}(\mathbf{a}^F(\mathbf{x})\cdot K + e_{\lfloor\rceil} + e'')\rceil$, where $e'' \leq L\cdot\ell\cdot\sigma\cdot n^{3/2}$. Let $T = L\cdot\ell\cdot\sigma\cdot n^{3/2}$, there is $\|e_{\lfloor\rceil}\| < q/(2p) - T$. Thus, $\|K\cdot e + \sum_{i=1}^{N}\lambda_i e_i' - \sum_{i=1}^{N}e_i\| \leq \frac{1}{2}$. G_3^{Auth} and G_2^{Auth} are computationally indistinguishable except guessing the $\mathbf{x}$ (i.e. psw_u). Hence, there is

$$Pr[succ_3^{Auth}] - Pr[succ_2^{Auth}] = C'\cdot q_s^{s'}(\kappa).$$

Game G_5^{Auth}*: This game is similar to* G_4^{Auth} except that $\mathcal{A}$ gets $(pk_u, \mathbf{sig}_u)$ and accesses the oracle **LoginS**(sid_i), **OutS**(sid_i), and **RetS**(sid_i). According to Section II-E, $\mathcal{A}$ cannot forge a signature $\mathbf{sig}^*$ with $(pk_u, \mathbf{sig}_u)$ since EUF-CMA signature is secure. Therefore, we have

$$Pr[succ_3^{Auth}] - Pr[succ_2^{Auth}] = Adv^{Sig}_{\mathcal{A}}(\kappa).$$

Game G_6^{Auth}*: This game is similar to* G_5^{Auth} except that $\mathcal{A}$ queries q_s times and guesses psw_u. According to Section III-B, the $\mathcal{A}$'s advantage is $C'\cdot q_s^{s'}(\kappa)$. Therefore

$$Pr[succ_5^{Auth}] - Pr[succ_4^{Auth}] = C'\cdot q_s^{s'}(\kappa).$$

In summary, the advantage of disrupting authentication is that: $Adv^{\mathbf{Auth}}_{QPASE,\mathcal{A}}(\kappa) \leq 2C'\cdot q_s^{s'}(\kappa) + Adv^{\mathsf{DLWE}}_{\mathcal{A}}(\kappa) + Adv^{1D-SIS}_{\mathcal{A}} + Adv^{Sig}_{\mathcal{A}}(\kappa) + \varepsilon(\kappa)$. □

Theorem 3 (IND-CKA): QPASE construction provides authentication based on the hardness of the decision Decision-LWE$_{n,q,\chi,m}$ and 1D-SIS problem and security of

HKDF and EUF-MCA. For any PPT $\mathcal{A}$, the advantage of disrupting IND-CKA security that

$$Adv^{\textbf{IND}}_{QPASE,\mathcal{A}}(\kappa) \le 2C' \cdot q_s^{s'}(\kappa) + Adv^{\mathsf{DLWE}}_{\mathcal{A}}(\kappa) + Adv^{1D-SIS}_{\mathcal{A}} + Adv^{Sig}_{\mathcal{A}}(\kappa) + Adv^{HKDF}_{\mathcal{A}}(\kappa) + \varepsilon(\kappa).$$

We employ the Zipf model of the Taobao password distribution in Fig. 7, where $|\mathcal{D}| = 15,072,667$, $C' = 0.0166957$, and $s' = 0.194179$.

Proof: Game G_0^{IND}: The game initializes sid_i^*, sid_j, $\mathcal{L}$, and pp as defined in the real security experiment $G^{IND\text{-}CKA}_{QPASE,\mathcal{A}}(\kappa)$. $\mathcal{A}$ accesses oracle **Challenge**(b, sid_i, w_i), **OutU**(sid_i, w, C), and **RetU**(sid_i, w), which are defined in subsection III-B. Specifically, the simulator $\mathcal{S}$ initializes $\mathcal{L}^i_{sid_j} \leftarrow \emptyset$ and plays U and S_i. $\mathcal{A}$ interacts with the honest user and server (oracle) as the corrupted server. After access, $\mathcal{S}$ records $L[sid_j] \leftarrow (i, K_u)$, delivers sid_j to $\mathcal{A}$ and sets $j \leftarrow j+1$. We have $Adv^{\textbf{IND-CKA}}_{QPASE,\mathcal{A}}(\kappa) = Pr[succ_0^{IND}] - \frac{1}{2}$.

Game G_1^{IND}: This game is similar to G_0^{IND} except that $\mathcal{A}$ executes **Login** to pass the authentication and obtain K_u with the $psw_{\mathcal{A}}$. By Theorem 2, we have

$$Pr[succ_1^{IND}] - Pr[succ_0^{IND}] = Adv^{\textbf{Auth}}_{QPASE,\mathcal{A}}(\kappa).$$

Game G_2^{IND}: This game is similar to G_1^{IND} except that in each session sid_i of the oracle **OutU**(sid_i, w, C) and oracle **RetU**(sid_i, w), $\mathcal{A}$ cannot distinguish the search key $\mathsf{dsk} \leftarrow$ **HKDF**(K_u, W) and dsk', which is a uniform-random value. By the uniform distribution of K_u and the security of HKDF in Definition 10. We have

$$Pr[succ_2^{IND}] - Pr[succ_1^{IND}] \le Adv^{HKDF}_{\mathcal{A}}(\kappa).$$

Game G_3^{IND}: This game is similar to G_2^{IND} except that $\mathcal{A}$ can forge EUF-CMA signature to be verified by **OutU**(sid_i, w, C), **RetU**(sid_i, w) and **RetS**(sid_i). According to Section II-E and CRYSTALS-Dilithium [86], we have

$$Pr[succ_3^{IND}] - Pr[succ_2^{IND}] \le Adv^{Sig}_{\mathcal{A}}(\kappa).$$

Game G_4^{IND}: This game is similar to G_3^{IND} except that $\mathcal{A}$ cannot distinguish the key $v \leftarrow PRF(\mathsf{dsk}, \rho)$ and $v' \leftarrow PRF(r_1, r_2)$ where r_1 and r_2 are random picked. According to G_3^{Auth} and Fig 2, the coefficient of $\frac{p}{q} \cdot \mathbf{a}^F(\mathbf{x}) \cdot K$ is further than e'' away from $\mathcal{Z} + \frac{1}{2}$. Therefore, we have

$$Pr[succ_4^{IND}] - Pr[succ_3^{IND}] \le Adv^{\mathsf{DLWE}}_{\mathcal{A}}(\kappa) + \frac{1}{2}.$$

In summary, for any PPT $\mathcal{A}$, the advantage of disrupting IND-CKA security that: $Adv^{\textbf{IND}}_{QPASE,\mathcal{A}}(\kappa) \le 2C' \cdot q_s^{s'}(\kappa) + Adv^{\mathsf{DLWE}}_{\mathcal{A}}(\kappa) + Adv^{Sig}_{\mathcal{A}}(\kappa) + Adv^{HKDF}_{\mathcal{A}}(\kappa) + \varepsilon(\kappa)$. □

B. Further Security Discussion

The main goal of the **Login** phase of QPASE is to allow a user to recover high entropy keys with a correct password. **Login** calls include a lattice-based TOPRF (see Section II-B) and a EUF-CMA signature scheme (see Section II-E). The construction of lattice-based TOPRF starts from the lattice-based OPRF of Albrecht et al. [79], which is reduced to $\mathsf{DLWE}_{n,q,\chi,m}$ and 1D-$\mathrm{SIS}_{q/p,n\ell,\max\{n\ell B,T\}}$. On this basis, Jiang et al. [68] extend the lattice-based OPRF to the lattice-based TOPRF and provide a threshold constraint.

In this work, we employ the robust extractor [69] to achieve a deterministic user-specific key generation for QPASE. The use of a robust extractor does not undermine the security of underlying intractable problem [69] (e.g. $\mathsf{DLWE}_{n,q,\chi,m}$ and 1D-$\mathrm{SIS}_{q/p,n\ell,\max\{n\ell B,T\}}$). Moreover, robust extractors only reveal the range of noise without affecting the output of TOPRF since the noises are eliminated by rounding operations.

TABLE II

SECURITY LEVEL OF OUR SCHEME

Parameter	Description	PARAM I	PARAM II
n	dimension	512	595
q	original modulus	$2^{28}-57$	$2^{28}-57$
β	the BKZ block	342	478
δ_0	Hermite factot	1.0044	1.0035
Classical cost	$2^{0.292\times\beta}$	100-bit	140-bit
Quantum cost	$2^{0.268\times\beta}$	92-bit	128-bit

We follow the security parameter settings of Albrecht et al. [79] to ensure the quantum security of our lattice-based TOPRF, and also carefully consider parameter settings for other components. For the instantiation of QPASE, we employ CRYSTALS-Dilithium [86] to achieve authentication and thwart adversaries from tampering with the information. In both **Outsource** and **Retrieve**, we employ an HKDF [81] and PRF [80] for deriving the search key.

Let the lattice dimension of DLWE $n = \kappa^c$, where $c > 2$ is a constant, and the lattice dimension of 1D-SIS $n' = \kappa$. Set the bit-length of $\mathbf{x}$ $L = \kappa$, the secret and error distribution $\sigma = \mathsf{poly}(n)$, and $\sigma' = \sigma^2 n^2 \cdot \kappa^{\omega(1)}$. let $q = p \cdot \prod_{i=1}^{n'} p_i$, where $p_i = \sigma' \cdot \omega\left(\sqrt{nn' \log q \log n'}\right)$. There is $q = p \cdot \sigma' \cdot \kappa^{\omega(1)}$ [79]. According to CRYSTALS-Dilithium [86], there is $q = 56(n\sqrt{n\kappa}/log\ n)^2/n\sqrt{\kappa n/log\ n} = 56(n\kappa/log\ n)^{3/2} \ge 2^{23}$.

We employ the "lwe-estimator"[1] with the quantum cost model [91] to achieve the security estimates of QPASE. In order to acquire more conservative parameters, we utilize the core-SVP methodology following [4], employing the classical cost $2^{0.292\times\beta}$ and quantum cost $2^{0.268\times\beta}$. Specifically, we set $q = 2^{28} - 57$, $n = 512$ following Bai et al. [86]. Our QPASE can provide 100-bit classical security and 92-bit quantum security with the BKZ block $\beta = 342$. To provide 128-bit quantum security for QPASE, we adjust the parameters to $q = 2^{28} - 57$, $n = 595$ with $\beta = 478$. Table II shows more details on two sets of parameters.

Next, we consider the impact of $\mathcal{A}$ executing corruption attacks on servers. The form of the user-specific key $K_u = \lfloor \frac{p}{q} \mathbf{a}^F(\mathbf{x}) \cdot K \rceil$ shows that the key is only related to the user's password psw ($\mathbf{x}$ is the binary form of the password) and K. Even if $\mathcal{A}$ has corrupted t' ($t' < t$) servers can not launch disclose attacks and impersonation attacks since the lattice-based TOPRF has observability and unpredictability [68]. If $\mathcal{A}$ can corrupt more than t servers, $\mathcal{A}$ can generate the user-specific key by collecting data from the first phase of the TOPRF interaction. Therefore, we assume that $\mathcal{A}$ cannot corrupt more than t servers in the same epoch again (which is consistent with the idea of the (t, N) threshold scheme).

Lemma 4: Let $[F]$ and $[G]$ denote the master secret key polynomial and the update polynomial, respectively. $\lambda_{i,j}$ is the Lagrangian coefficient. $\mathcal{A}$ cannot obtain the master secret key K of the servers, if $\mathcal{A}$ cannot corrupt more than $t^{prime} < t$ servers in an era.

[1]https://bitbucket.org/malb/lwe-estimator/raw/HEAD/estimator.py

TABLE III
RUNNING TIMES OF RELATED OPERATIONS (IN MS)

Operations	T_G	T_{ou}	T_S	T_E	T_P
Time	0.354	0.641	0.241	15	0.554
Operations	T_H	T_{os}	T_V	T_D	T_{exe}
Time	0.02	0.049	0.133	15	3.8

Proof: Suppose that $\mathcal{A}$ has corrupted t servers and obtained more than t private keys in two consecutive eras, which is denoted by $k_1, \ldots, k_{t'}, k^*_{t'+1}, \ldots, k^*_t$, $(a < t' < t < n)$. At this point, the adversary calculates: $K^* = \sum_{i=1}^{t'} \lambda_{i,j} \cdot k_i + \sum_{i=t'+1}^{t} \lambda_{i,j} \cdot k^*_i = \sum_{i=1}^{t'} \lambda_{i,j} \sum_{j=1}^{n} [S]^i_j + \sum_{i=1}^{t} \lambda_{i,j} \sum_{j=1}^{n} [F]^i_j = K + \sum_{i=t'+1}^{t} \lambda_{i,j} \sum_{j=1}^{n} [G]^i_j$.

Since the update key generated by 0-sharing still satisfied the threshold security requirements. That is, the adversary can obtain at most $t' < t$ update keys. In other words, there are $t - t'$ update keys here that the adversary cannot obtain. Therefore, the adversary can not compute: $K = \sum_{i=1}^{t'} \lambda_{ij} sk_i + \sum_{i=t'+1}^{t} \lambda_{ij} sk^*_i - \sum_{i=t'+1}^{t} \lambda_{ij} \sum_{j=1}^{n} [G]^i_j$.

VI. EXPERIMENTS

In this section, we evaluate the overheads and functions of our QPASE and compare our work with related works.

A. Overheads

We calculate the computation cost in terms of basic cryptographic operations. Specifically, T_G, T_H, and T_p denote the key generation, HKDF, and PRF, respectively. T_{ou} and T_{os} denote the execution of the TOPRF algorithm by the user and the server, respectively. T_s and T_v denote the signature and verification, respectively. T_E and T_D denote the symmetric encryption and decryption of 100KB files, respectively. In addition, we use T_{exe} to denote the exponential operation, which is the main overhead of the PASE scheme of Chen et al. [14]. Our implementation is in C++ language and complies with the NTL version 11.5.1, and the measurement is obtained on a LAPTOP with an AMD Ryzen 7 5800H with Radeon Graphics running at 3.20 GHz. The computation cost of basic cryptographic operations is shown in Table III.

Let the dimension $m = n$, an odd prime $q \approx n^c$, where c is constant, and the noise rate $\alpha \approx n^{1/2-c}$, we can get an LWE instance by n, q, α, m. To ensure a 128-bit quantum security level, we employ the PARAM II in Table II as the parameter set. Specifically, set $n = 595$ and $q = 2^{28} - 57 = 268,435,399$. The practical parameters for implementing our QPASE can be found in the scripts of LWE-Frodo[2] and Dilithium.[3] To facilitate comparison with Chen et al.'s scheme [14], we set $N = t = 2$ and employ the same evaluation setup in [92]. The test object of the outsourcing and recovery operation is a 100 KB file, just like PASE [14].

In Table IV, we compare each phase of QPASE with its foremost counterpart i.e., Chen et al.'s PASE [14]. Set security parameters $\kappa = 128$ for both schemes. Table IV illustrates that our QPASE incurs lower computational costs than PASE [14]. On the one hand, our lattice-based scheme eliminates the need for exponentiation operations and only uses relatively lightweight operations like matrix multiplication and addition, which reduces computational overhead. On the other hand, we employ TOPRF to re-randomize passwords, which allows the user to achieve authentication simultaneously with the reconstruction of user-specific keys. This avoids the additional computational cost of commitments and further improves the computational efficiency of our QPASE.

We note that our QPASE has more communication costs than PASE [14], and this is a limitation of our scheme. Therefore, it may be not suitable for scenarios with low network bandwidth. We emphasize that the communication cost during the login phase is fixed. Thus, The communication cost gap between our scheme and PASE [14] decreases as the size of outsourced files increases. Specifically, the instance of LWE in Π_{TOPRF} takes about 0.5 MB [79]. The communication cost increases linearly as the number of servers grows. Besides, the public key transmitted during the registration phase imposes a communication overhead of at least 2.2 MB [86].

Still, compared to PASE [14], our scheme offers more robust security attributes and higher threshold settings. On the one hand, to the best of our knowledge, our scheme is the first password-authenticated symmetric searchable encryption with *quantum-resistance*. On the other hand, in our scheme, the server-side stores users' outsourced data *in a distributed manner*. In the **Retrieve** phase, multiple servers can retrieve data in parallel to further improve efficiency. *In summary, our QPASE outperforms its foremost counterparts (i.e., Chen et al.'s scheme [14]) in security and computation overhead.*

B. Function

Although the password-authenticated secret sharing (PASS) scheme and QPASE scheme have different design ideas and components, both schemes share the same goal of enabling users to recover high-entropy encryption keys through passwords. Therefore, we have compared the functionality and computational overhead of the QPASE scheme with various PASS schemes [7], [60], [61], [62], [72], [73], [93], [94] and PASE schemes [14] in the phase of recovering the high entropy encryption key, as shown in Table V. We measure the computation overhead of schemes in terms of the number of exponential power operations. Randomization of the password through the lattice-based TOPRF [68] can avoid the high computational overhead of the power exponential operation. Moreover, users do not need to perform complex encryption and secret sharing at the registration phase.

It can be seen that Roy et al. [73], Jiang et al. [4], and our QPASE has a more obvious advantage in terms of efficiency, which is based on $\mathsf{DLWE}_{n,q,\chi,m}$. It does not require exponential power operations to hide secrets. In terms of security, the universally composable (UC) model is widely used [62], [72], [94] and can ignore the distribution of passwords. However, it is difficult to measure resistance to quantum attacks within the UC model [95]. Thus, we employ the ROM model to characterize security. Notably, the impact of password distribution on security analysis is crucial in the ROM model.

Fig. 7 shows that in the ROM model, assuming that the password follows a uniform random distribution leads to a "relaxation" of the security reduction. More specifically, the adversary's advantages are drastically underestimated in the uniform random password distribution model. The CDF-Zipf based formulation [74], [75] $C' \cdot q_{send}^{s'}(\kappa) + \varepsilon(\kappa)$ well approximates the real attacker's $Adv : q_{send} \in [1, |\mathcal{D}|]$ (Here

[2] https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8309.pdf
[3] https://github.com/GMUCERG/Dilithium

TABLE IV
COMPARISON OF THE PERFORMANCE EVALUATION BETWEEN CHEN ET AL. [14] AND OUR WORK AT EACH PHASE

Scheme		User		Server		Communication cost	Rounds
		Computation cost	Total Time	Computation cost	Total Time		
Register	Chen et al. [14]	$6T_{exe}$	$\approx$ 22.81ms	0	0	$\approx$ 1280 bit	1
	Our QPASE	$T_{ou}+T_G$	$\approx$ 0.99 ms	T_{os}	$\approx$ 0.05 ms	$\approx$ 4,404,019 bit	1
Recover K_u	Chen et al. [14]	$3T_{exe}$	$\approx$ 11.41ms	$8T_{exe}$	$\approx$ 30.41ms	$\approx$ 1792 bit	1
	Our QPASE	T_{ou}	$\approx$ 0.64 ms	T_{os}	$\approx$ 0.05 ms	$\approx$ 2,097,152 bit	1
Outsource	Chen et al. [14]	$3T_{exe}$+T_E	$\approx$ 26.41 ms	$8T_{exe}$	$\approx$ 30.41 ms	$\approx$ 102,656 bit	2
	Our QPASE	$T_H+T_P+T_E+2T_S$	$\approx$ 16.05 ms	T_V	$\approx$ 0.13 ms	$\approx$ 111,332 bit	1
Retrieve	Chen et al. [14]	$3T_{exe}$+$2T_D$	$\approx$ 41.41 ms	$8T_{exe}$+T_D	$\approx$ 45.52 ms	$\approx$ 102,912 bit	2
	Our QPASE	$T_H+T_S+T_P+T_V+2T_D$	$\approx$ 30.95 ms	$T_P+T_V+T_D$	$\approx$ 15.69 ms	$\approx$ 119,050 bit	1

TABLE V
COMPARISON AMONG RECENT PASS [4], [7], [60], [61], [62], [72], [73], [93], [94] AND PASE [14] WITH OUR WORK

	Threshold	Password distribution	Security model	Technology	**Quantum security**	**Data retrieval**	Round	Computation overhead	
								Server	User
Bagherzandi et al. (CCS'11) [60]	(t,N)	UR	ROM	HE	×	×	2	$16T_{exe}$	$33T_{exe}$
Camenisch et al. (CCS'12) [93]	(2,2)	-	UC	HE	×	×	1	$26T_{exe}$	$19T_{exe}$
Jarecki et al. (ASIACRYPT'14) [61]	(t,N)	UR	ROM	OPRF	×	×	1	$4T_{exe}$	$11T_{exe}$
Yi et al. (ESORICS'15) [94]	(t,N)	UR	UC	HE	×	×	1	$12T_{exe}$	$7T_{exe}$
Jarecki et al. (EuroS&P'16) [62]	(t,N)	UR	UC	OPRF	×	×	1	$1T_{exe}$	$4T_{exe}$
Jarecki et al. (ACNS'17) [72]	(t,N)	UR	UC	OPRF	×	×	1	$1T_{exe}$	$2T_{exe}$
Das et al. (ASIACCS'20) [7]	(N,N)	UR	UC	OPRF	×	×	2	$4T_{exe}$	$10T_{exe}$
Chen et al.(IJIS'21) [14]	$(2,2)$	UR	ROM	HE	×	✓	1	$8T_{exe}$	$3T_{exe}$
Roy et al. (ACNS'21) [73]	(t,N)	UR	ROM	FHE	✓	×	1	0	0
Jiang et al. (TSC'23) [4]	(t,N)	Zipf	ROM	FHE	✓	×	1	0	0
Our work	(t,N)	Zipf	ROM	OPRF	✓	✓	1	0	0

† UR=Uniform random; - means not to consider; Zipf=Zipf distribution; HE=Homomorphic encryption; OPRF=Oblivious pseudorandom function; FHE=Fully homomorphic encryption; For efficiency. we count the most expensive operations, i.e., exponentiations (denoted by T_{exe}).

we use the Zipf model of Taobao, where $|\mathcal{D}| = 15,072,667$, $C' = 0.0166957$ and $s' = 0.194179$, the maximum deviation is less than 0.491%). This CDF-Zipf-based formulation is more accurate than previously used formulations such as the Min-entropy model [96]. Thus, we use the CDF-Zipf based formulation for our QPASE to achieve tighter security than other PASS [61], [62], [72] and PASE schemes [14].

VII. CONCLUSION

The major goal of this paper is to construct a quantum-resistant password-authenticated symmetric searchable encryption scheme based on the lattice to satisfy that only a user who knows the correct password can outsource, search, and retrieve data. To achieve this goal, we employ a lattice-based TOPRF to re-randomize the password that enables the user to generate a user-specific key via a human-memorable password and can resist offline guessing attacks. Then, we propose the first quantum-resistant password-authenticated symmetric searchable encryption for cloud storage, called QPASE.

QPASE offers users a solution to circumvent costly and error-prone key management practices when utilizing cloud storage services. Passwords not only serve as an authentication mechanism but also grant legitimate users access to powerful cloud server keys, enabling the derivation of user-specific keys. This liberates users from device constraints, significantly enhancing data outsourcing flexibility. Our scheme is extendable to support multi-keyword search and enables cloud servers to update keys without disrupting user data retrieval. We show that authentication and searchable encryption are not orthogonal, i.e., authentication security can prevent impersonation attacks and protect searchable encryption. Searchable encryption can also extend the functions and security of password-based authentication schemes. The security analysis confirms that QPASE achieves authentication security and IND-CKA security. Comparative evaluations against related schemes highlight the practicality of our QPASE.

ACKNOWLEDGMENT

The authors thank the anonymous reviewers for their invaluable comments.

REFERENCES

[1] Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun, and K. Ren, "A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 11, pp. 2594–2608, Nov. 2016.

[2] D. Rydning, J. Reinsel, and J. Gantz, "The digitization of the world from edge to core," *Framingham, Int. Data Corp.*, vol 16, pp. 1–28, Nov. 2018.

[3] K. Xue, N. Gai, J. Hong, D. S. L. Wei, P. Hong, and N. Yu, "Efficient and secure attribute-based access control with identical sub-policies frequently used in cloud storage," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 1, pp. 635–646, Jan. 2022.

[4] J. Jiang, D. Wang, and G. Zhang, "QPause: Quantum-resistant password-protected data outsourcing for cloud storage," *IEEE Trans. Services Comput.*, early access, Nov. 8, 2024, doi: 10.1109/TSC.2023.3331000.

[5] G. S. Poh, J.-J. Chin, W.-C. Yau, K.-K.-R. Choo, and M. S. Mohamad, "Searchable symmetric encryption: Designs and challenges," *ACM Comput. Surveys*, vol. 50, no. 3, pp. 1–37, May 2018.

[6] G-Cloud. (2018). *Google Cloud Key Management Service*. [Online]. Available: https://cloud.google.com/kms/.pdf

[7] P. Das, J. Hesse, and A. Lehmann, "DPaSE: Distributed password-authenticated symmetric-key encryption, or how to get many keys from one password," in *Proc. AsiaCCS*, 2022, pp. 682–696.

[8] D. Xiaoding Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proc. IEEE Symp. Secur. Privacy*, May 2000, pp. 44–55.

[9] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: Improved definitions and efficient constructions," *J. Comput. Sec.*, vol. 19, no. 5, pp. 895–934, 2011.

[10] L. Ballard, S. Kamara, and F. Monrose, "Achieving efficient conjunctive keyword searches over encrypted data," in *Proc. ICICS*, vol. 5, 2005, pp. 414–426.

[11] C. Orencik, A. Selcuk, E. Savas, and M. Kantarcioglu, "Multi-keyword search over encrypted data with scoring and search pattern obfuscation," *Int. J. Inf. Secur.*, vol. 15, no. 3, pp. 251–269, Jun. 2016.

[12] R. Chen, Y. Mu, G. Yang, F. Guo, and X. Wang, "Dual-server public-key encryption with keyword search for secure cloud storage," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 4, pp. 789–798, Apr. 2016.

[13] D. Boneh and B. Waters, "Conjunctive, subset, and range queries on encrypted data," in *Proc. TCC*, 2007, pp. 535–554.

[14] L. Chen, K. Huang, M. Manulis, and V. Sekar, "Password-authenticated searchable encryption," *Int. J. Inf. Secur.*, vol. 20, no. 5, pp. 675–693, 2021.

[15] E. Goh. (2003). *Secure Indexes*. [Online]. Available: http://eprint.iacr.org/2003/216

[16] M. Chase and S. Kamara, "Structured encryption and controlled disclosure," in *Proc. 16th Int. Conf. Theory Appl. Cryptol. Inf. Secur. (ASIACRYPT)*, Singapore. Berlin, Germany: Springer, Dec. 2010, pp. 577–594.

[17] R. Bost, B. Minaud, and O. Ohrimenko, "Forward and backward private searchable encryption from constrained cryptographic primitives," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 1465–1482.

[18] B. A. Fisch et al., "Malicious-client security in blind seer: A scalable private DBMS," in *Proc. IEEE Symp. Secur. Privacy*, May 2015, pp. 395–410.

[19] E. Stefanov, C. Papamanthou, and E. Shi, "Practical dynamic searchable encryption with small leakage," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2014, pp. 72–75.

[20] D. Cash et al., "Dynamic searchable encryption in very-large databases: Data structures and implementation," in *Proc. NDSS*, 2014, pp. 1–16.

[21] D. Cash and S. Tessaro, "The locality of searchable symmetric encryption," in *Proc. 33rd Annu. Int. Conf. Theory Appl. Cryptograph. Techn. (EUROCRYPT)*, Copenhagen, Denmark. Berlin, Germany: Springer, May 2014, pp. 351–368.

[22] I. Miers and P. Mohassel, "IO-DSSE: Scaling dynamic searchable encryption to millions of indexes by improving locality," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2017, pp. 1–16.

[23] D. Cash, S. Jarecki, C. Jutla, H. Krawczyk, M.-C. Rosu, and M. Steiner, "Highly-scalable searchable symmetric encryption with support for Boolean queries," in *Proc. 33rd Annu. Cryptol. Conf. (CRYPTO)*, Santa Barbara, CA, USA. Berlin, Germany: Springer, Aug. 2013, pp. 353–373.

[24] S. Kamara and T. Moataz, "Boolean searchable symmetric encryption with worst-case sub-linear complexity," in *Proc. 36th Annu. Int. Conf. Theory Appl. Cryptograph. Techn. (EUROCRYPT)*. Paris, France: Springer, Apr./May 2017, pp. 94–124.

[25] X. Meng, S. Kamara, K. Nissim, and G. Kollios, "GRECS: Graph encryption for approximate shortest distance queries," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2015, pp. 504–517.

[26] Y. Chang and M. Mitzenmacher, "Privacy preserving keyword searches on remote encrypted data," in *Proc. ACNS*, 2005, pp. 442–455.

[27] M. Naveed, M. Prabhakaran, and C. A. Gunter, "Dynamic searchable encryption via blind storage," in *Proc. IEEE Symp. Secur. Privacy*, May 2014, pp. 639–654.

[28] S. Kamara and C. Papamanthou, "Parallel and dynamic searchable symmetric encryption," in *Proc. FC*, 2013, pp. 258–274.

[29] S. Kamara, C. Papamanthou, and T. Roeder, "Dynamic searchable symmetric encryption," in *Proc. ACM Conf. Comput. Commun. Secur.*, Oct. 2012, pp. 965–976.

[30] F. Hahn and F. Kerschbaum, "Searchable encryption with secure and efficient updates," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2014, pp. 310–320.

[31] P. Xu et al., "Rose: Robust searchable encryption with forward and backward security," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 1115–1130, 2022.

[32] D. Cash, P. Grubbs, J. Perry, and T. Ristenpart, "Leakage-abuse attacks against searchable encryption," in *Proc. ACM CCS*, 2015, pp. 668–679.

[33] Y. Zhang, J. Katz, and C. Papamanthou, "All your queries are belong to us: The power of file-injection attacks on searchable encryption," in *Proc. 25th USENIX Secur. Symp.*, 2016, pp. 707–720.

[34] S. Hohenberger, V. Koppula, and B. Waters, "Adaptively secure puncturable pseudorandom functions in the standard model," in *Proc. ICICS*, 2015, pp. 79–102.

[35] K. S. Kim, M. Kim, D. Lee, J. H. Park, and W.-H. Kim, "Forward secure dynamic searchable symmetric encryption with efficient updates," in *Proc. ACM CCS*, 2017, pp. 1449–1463.

[36] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: Improved definitions and efficient constructions," in *Proc. ACM CCS*, 2006, pp. 79–88.

[37] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn. (EUROCRYPT)*, Interlaken, Switzerland. Berlin, Germany: Springer, May 2004, pp. 506–522.

[38] 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/Jun. 2021.

[39] T. Suzuki, K. Emura, and T. Ohigashi, "A generic construction of integrated secure-channel free PEKS and PKE and its application to EMRs in cloud storage," *J. Med. Syst.*, vol. 43, no. 5, pp. 1–15, May 2019.

[40] J. Baek, R. Safavi-Naini, and W. Susilo, "Public key encryption with keyword search revisited," in *Proc. ICCSA*, 2008, pp. 1249–1259.

[41] P. Golle, J. Staddon, and B. Waters, "Secure conjunctive keyword search over encrypted data," in *Proc. ACNS*, 2004, pp. 31–45.

[42] J. Chen et al., "EliMFS: Achieving efficient, leakage-resilient, and multi-keyword fuzzy search on encrypted cloud data," *IEEE Trans. Services Comput.*, vol. 13, no. 6, pp. 1072–1085, Nov. 2020.

[43] H. Li, Y. Yang, T. H. Luan, X. Liang, L. Zhou, and X. S. Shen, "Enabling fine-grained multi-keyword search supporting classified sub-dictionaries over encrypted cloud data," *IEEE Trans. Dependable Secure Comput.*, vol. 13, no. 3, pp. 312–325, May 2016.

[44] H. Cui, Z. Wan, R. H. Deng, G. Wang, and Y. Li, "Efficient and expressive keyword search over encrypted data in cloud," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 3, pp. 409–422, May 2018.

[45] R. Chen et al., "Server-aided public key encryption with keyword search," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 12, pp. 2833–2842, Dec. 2016.

[46] Q. Huang and H. Li, "An efficient public-key searchable encryption scheme secure against inside keyword guessing attacks," *Inf. Sci.*, vols. 403–404, pp. 1–14, Sep. 2017.

[47] D. He, M. Ma, S. Zeadally, N. Kumar, and K. Liang, "Certificateless public key authenticated encryption with keyword search for industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3618–3627, Aug. 2018.

[48] M. Bellare, A. Boldyreva, and A. O'Neill, "Deterministic and efficiently searchable encryption," in *Proc. 27th Annu. Int. Cryptol. Conf. (CRYPTO)*, Santa Barbara, CA, USA. Berlin, Germany: Springer, Aug. 2007, pp. 535–552.

[49] S. Canard, G. Fuchsbauer, A. Gouget, and F. Laguillaumie, "Plaintext-checkable encryption," in *Proc. CT-RSA*, 2012, pp. 332–348.

[50] K. Huang, M. Manulis, and L. Chen, "Password authenticated keyword search," in *Proc. IEEE Symp. Privacy-Aware Comput. (PAC)*, Aug. 2017, pp. 129–140.

[51] P. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proc. FOCS*, 1994, pp. 124–134.

[52] G. Alagic et al. (2019). *Status Report on the First Round of the NIST Post-Quantum Cryptography Standardization Process*. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/ir/2019/NIST.IR.8240.pdf

[53] G. Alagic et al. (2020). *Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization Process*. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8309.pdf

[54] J. Bos, C. Costello, M. Naehrig, and D. Stebila, "Post-quantum key exchange for the TLS protocol from the ring learning with errors problem," in *Proc. IEEE Symp. Secur. Privacy*, May 2015, pp. 553–570.

[55] Z. Li and D. Wang, "Two-round PAKE protocol over lattices without NIZK," in *Proc. INSCRYPT*, 2018, pp. 138–159.

[56] J. Zhang and Y. Yu, "Two-round PAKE from approximate SPH and instantiations from lattices," in *Proc. 23rd Int. Conf. Theory Appl. Cryptol. Inf. Secur. (ASIACRYPT)*. Hong Kong, China: Springer, Dec. 2017, pp. 37–67.

[57] J. Zhang, Z. Zhang, J. Ding, M. Snook, and O. Dagdelen, "Authenticated key exchange from ideal lattices," in *Proc. 34th Annu. Int. Conf. Theory Appl. Cryptograph. Techn. (EUROCRYPT)*, Sofia, Bulgaria. Berlin, Germany: Springer, Apr. 2015, pp. 719–751.

[58] J. Ding, S. Alsayigh, J. Lancrenon, R. Saraswathy, and M. Snook, "Provably secure password authenticated key exchange based on RLWE for the post-quantum world," in *Proc. CT-RSA*, 2017, pp. 183–204.

[59] Q. Wang, D. Wang, C. Cheng, and D. He, "Quantum2FA: Efficient quantum-resistant two-factor authentication scheme for mobile devices," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 1, pp. 193–208, Jan. 2023.

[60] A. Bagherzandi, S. Jarecki, N. Saxena, and Y. Lu, "Password-protected secret sharing," in *Proc. 18th ACM Conf. Comput. Commun. Secur.*, Oct. 2011, pp. 433–444.

[61] S. Jarecki, A. Kiayias, and H. Krawczyk, "Round-optimal password-protected secret sharing and T-PAKE in the password-only model," in *Proc. 20th Int. Conf. Theory Appl. Cryptol. Inf. Secur. (ASIACRYPT)*. Kaoshiung, Taiwan. Berlin, Germany: Springer, Dec. 2014, pp. 233–253.

[62] S. Jarecki, A. Kiayias, H. Krawczyk, and J. Xu, "Highly-efficient and composable password-protected secret sharing (or: How to protect your Bitcoin wallet online)," in *Proc. IEEE Eur. Symp. Secur. Privacy*, Mar. 2016, pp. 276–291.

[63] J. Bonneau, C. Herley, P. Oorschot, and F. Stajano, "The quest to replace passwords: A framework for comparative evaluation of web authentication schemes," in *Proc. IEEE Symp. Secur. Privacy*, May 2012, pp. 553–567.

[64] X. Gao, Y. Yang, C. Liu, C. Mitropoulos, J. Lindqvist, and A. Oulasvirta, "Forgetting of passwords: Ecological theory and data," in *Proc. 27th USENIX Secur. Symp.*, 2018, pp. 221–238.

[65] J. Bonneau, C. Herley, P. C. van Oorschot, and F. Stajano, "Passwords and the evolution of imperfect authentication," *Commun. ACM*, vol. 58, no. 7, pp. 78–87, Jun. 2015.

[66] D. Wang, Y. Zou, Q. Dong, Y. Song, and X. Huang, "How to attack and generate honeywords," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2022, pp. 966–983.

[67] C. Gentry, A. Sahai, and B. Waters, "Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based," in *Proc. 33rd Annu. Cryptol. Conf. (CRYPTO)*, Santa Barbara, CA, USA. Berlin, Germany: Springer, Aug. 2013, pp. 75–92.

[68] J. Jiang, D.Wang, G. Zhang, and Z. Chen, "Quantum-resistant password-based threshold single-sign-on authentication with updatable server private key," in *Computer Security—ESORICS*, Copenhagen, Denmark. Cham, Switzerland: Springer, Sep. 2022, pp. 295–316.

[69] J. Ding, X. Xie, and X. Lin. (2012). *A Simple Provably Secure Key Exchange Scheme Based on the Learning With Errors Problem*. [Online]. Available: https://eprint.iacr.org/2012/688

[70] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 2000, pp. 139–155.

[71] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung, "Proactive secret sharing or: how to cope with perpetual leakage," in *Proc. 15th Annu. Int. Cryptol. Conf. (CRYPTO)*, Santa Barbara, CA, USA. Berlin, Germany: Springer, Aug. 1995, pp. 339–352.

[72] S. Jarecki, A. Kiayias, H. Krawczyk, and J. Xu, "TOPPSS: Cost-minimal password-protected secret sharing based on threshold OPRF," in *Proc. ACNS*, 2017, pp. 39–58.

[73] P. Roy, S. Dutta, W. Susilo, and R. Safavi-Naini, "Password protected secret sharing from lattices," in *Proc. ACNS*, 2021, pp. 442–459.

[74] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, "Zipf's law in passwords," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 11, pp. 2776–2791, Nov. 2017.

[75] D. Wang and P. Wang, "Two birds with one stone: Two-factor authentication with security beyond conventional bound," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 4, pp. 708–722, Jul./Aug. 2018.

[76] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *J. ACM*, vol. 56, no. 6, pp. 1–40, Sep. 2009.

[77] C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *Proc. 40th Annu. ACM Symp. Theory Comput.*, May 2008, pp. 197–206.

[78] J. Alwen, S. Krenn, K. Pietrzak, and D. Wichs, "Learning with rounding, revisited: New reduction, properties and applications," in *Proc. Annu. Cryptol. Conf.* Berlin, Germany: Springer, 2013, pp. 57–74.

[79] M. Albrecht, A. Davidson, A. Deo, and N. P. Smart, "Round-optimal verifiable oblivious pseudorandom functions from ideal lattices," in *Proc. PKC*, 2021, pp. 261–289.

[80] A. Banerjee and C. Peikert, "New and improved key-homomorphic pseudorandom functions," in *Proc. 34th Annu. Cryptol. Conf. (CRYPTO)*, Santa Barbara, CA, USA. Berlin, Germany: Springer, Aug. 2014, pp. 353–370.

[81] H. Krawczyk, "Cryptographic extraction and key derivation: The HKDF scheme," in *Proc. Annu. Cryptol. Conf.* Berlin, Germany: Springer, 2010, pp. 631–648.

[82] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979.

[83] R. Bendlin, S. Krehbiel, and C. Peikert, "How to share a lattice trapdoor: Threshold protocols for signatures and (H)IBE," in *Proc. ACNS*, 2013, pp. 218–236.

[84] T. Pornin and J. P. Stern, "Digital signatures do not guarantee exclusive ownership," in *Proc. ACNS*, 2005, pp. 138–150.

[85] D. Jackson, C. Cremers, K. Cohn-Gordon, and R. Sasse, "Seems legit: Automated analysis of subtle attacks on protocols that use signatures," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2019, pp. 2165–2180.

[86] S. Bai, L. Ducas, and E. Kiltz. (2021). *Crystals-Dilithium: Algorithm Specifications and Supporting Documentation (Version 3.1)*. [Online]. Available: https://pq-crystals.org/dilithium/index.shtml

[87] P.-A. Fouque, J. Hoffstein, and P. Kirchner. (2020). *FALCON: Fast-Fourier Lattice-Based Compact Signatures Over NTRU Specifications V1.2*. [Online]. Available: https://falcon-sign.info/

[88] J. Ding, M.-S. Chen, and M. Kannwischer. (2020). *Rainbow: Algorithm Specification and Documentation the 3rd Round Proposal*. [Online]. Available: https://www.pqcrainbow.org/

[89] C. Cremers, S. Düzlü, R. Fiedler, M. Fischlin, and C. Janson, "BUFFing signature schemes beyond unforgeability and the case of post-quantum signatures," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2021, pp. 1696–1714.

[90] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proc. 28th Annu. ACM Symp. Theory Comput.*, 1996, pp. 212–219.

[91] M. R. Albrecht, R. Player, and S. Scott, "On the concrete hardness of learning with errors," *J. Math. Cryptol.*, vol. 9, no. 3, pp. 169–203, Oct. 2015.

[92] Z. Li, D. Wang, and E. Morais, "Quantum-safe round-optimal password authentication for mobile devices," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 3, pp. 1885–1899, May/Jun. 2020.

[93] J. Camenisch, A. Lysyanskaya, and G. Neven, "Practical yet universally composable two-server password-authenticated secret sharing," in *Proc. ACM CCS*, 2012, pp. 525–536.

[94] X. Yi, F. Hao, L. Chen, and J. K. Liu, "Practical threshold password-authenticated secret sharing protocol," in *Proc. Eur. Symp. Res. Comput. Secur.*, Vienna, Austria. Cham, Switzerland: Springer,Sep. 2015, pp. 347–365.

[95] D. Boneh, Ö. Dagdelen, M. Fischlin, A. Lehmann, C. Schaffner, and M. Zhandry, "Random oracles in a quantum world," in *Proc. 17th Int. Conf. Theory Appl. Cryptol. Inf. Secur. (ASIACRYPT)*, Seoul, South Korea. Berlin, Germany: Springer, Dec. 2011, pp. 41–69.

[96] F. Benhamouda and D. Pointcheval. (2022). *Verifier-Based Password-Authenticated Key Exchange: New Models and Constructions*. [Online]. Available: https://eprint.iacr.org/2013/833

Jingwei Jiang is currently pursuing the Ph.D. degree with the College of Computer Science and Technology, Harbin Engineering University, China. As the first author, he has published papers at ESORICS 2022, TSC 2023, and the *Chinese Journal of Computers*. His research interests include lattice-based cryptography and authentication.

Ding Wang received the Ph.D. degree in information security from Peking University in 2017, and was supported by the "Boya Postdoctoral Fellowship" from Peking University, from 2017 to 2019. He is currently a Full Professor with Nankai University. As the first author (or corresponding author), he has published more than 90 papers at venues, such as IEEE S&P, ACM CCS, NDSS, Usenix Security, IEEE Transactions on Dependable and Secure Computing, and IEEE Transactions on Information Forensics and Security. His main research interests include passwords, authentication, and provable security. His research has been reported by over 200 media, such as Daily Mail, Forbes, IEEE Spectrum, and Communications of ACM, appeared in the Elsevier 2017 "Article Selection Celebrating Computer Science Research in China," and resulted in the revision of the authentication guideline NIST SP800-63-2. He has been involved in the community as the PC Chair/a TPC Member for over 60 international conferences, such as NDSS 2024/2023, ACM CCS 2022, PETS 2022–2024, ACSAC 2020–2024, RAID 2024/2023, ACM AsiaCCS 2022/2021, IFIP SEC 2018–2021, ICICS 2018–2024, and SPNCE 2020–2022. He received the ACM China Outstanding Doctoral Dissertation Award, the Best Paper Award at INSCRYPT 2018, the Outstanding Youth Award of the China Association for Cryptologic Research, the Young Scientist Nomination Award for Powerful Nation, and the First Prize of Natural Science Award of Ministry of Education.