Practical Post-quantum Password-Authenticated Key Exchange Based-on Module-Lattice

Peixin Ren and Xiaozhuo Gu*

SKLOIS, Institute of Information Engineering, CAS, Beijing, China
School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
{renpeixin, guxiaozhuo}@iie.ac.cn

Abstract. Password-authenticated key exchange (PAKE) is a neat technology that can establish secure remote communications between the client and the server, especially with the preponderance of amplifying a memorable password into a strong session key. However, the arrival of the quantum computing era has brought new challenges to traditional PAKE protocols. Thus, designing an efficient post-quantum PAKE scheme becomes an open research question. In this paper, we construct a quantum-safe PAKE protocol which is a horizontal extension of the PAK protocol [22] in the field of module lattice. Subsequently, we accompany our proposed protocol with a rigorous security proof in the Bellare-Pointcheval-Rogaway (BPR) model with two adaptions: applying the CDF-Zipf model to characterize the ability of the adversary and using the pairing with errors (PWE) assumption to simplify the proof. Taking the flexibility of the module learning with errors (MLWE) problem, we elaborately select 3 parameter sets to meet different application scenarios (e.g., classical/quantum-safe Transport Layer Security (TLS), resource-constrained Internet of Things (IoT) devices). Specifically, our Recommended implementation achieves 177-bit post-quantum security with a generous margin to cope with later improvement in cryptanalysis. The performance results indicate that our MLWE-PAKE is quite practical: compared with the latest Yang-PAK, our Recommended-PAK reduces the communication cost and the running time by 36.8% and 13.8%, respectively.

Keywords: Password-authenticated key exchange · Module learning with errors · Post-quantum · Lattice-based.

1 Introduction

Passwords have several advantages of being human-memorable, avoiding expensive computation of public key infrastructure (PKI) to distribute client certificates, and preventing dedicated hardware for storing secret keys. Thus, passwords constitute the prevalent and irreplaceable authentication approach to identify human users [31, 26], especially in the proliferation of mobile devices.

* Corresponding author: guxiaozhuo@iie.ac.cn.

PAKE is an important cryptographic primitive that enables two parties (e.g., a client and a server) to utilize a simple password to negotiate a high-entropy session key in an insecure network. In 1992, Bellovin and Merritt [3] proposed a *symmetric*-PAKE protocol, encrypted key exchange (EKE), where two parties hold the same password and establish a shared session-key at the end. However, *symmetric*-PAKE protocols [2, 14, 18] only focus on the part of password-using and omit how to constrain the impact of password leakage.

In reality, *asymmetric*-PAKE protocols [4] are widely deployed and standardized in the domain of existing client-to-server Internet or IoT. In asymmetric-PAKE schemes, the server only gets the knowledge of the hashed password with a random salt, not the actual password. In this case, even if the server is compromised, the adversary cannot obtain the password directly. Therefore, many asymmetric-PAKE protocols have been proposed and analyzed, such as [5, 15, 19, 32]. However, the hardness of these protocols depends on traditional number-theoretic problems (the integer factorization problem, the discrete logarithm problem etc.) that are vulnerable to quantum attacks [16, 27].

With the advent of quantum computing, standards bodies and academia [23, 24] have triggered widespread interest in cryptographic algorithms believed to resist quantum computers. According to [23], lattice is one of the most promising and ideal competitive primitives for the construction of post-quantum schemes. However, the majority of lattice-based schemes focus on key exchange without authentication [1, 7, 11] and key encapsulation mechanisms [8, 12].

Until 2017, Ding et al. [10] constructed a post-quantum asymmetric-PAKE protocol in the ideal lattice area and proved its security in the BPR model. The primary problem is that this protocol emphasizes the theoretical feasibility at the expense of efficient implementation in practice. Subsequently, following the work of [10], many literatures [13, 21, 30] proposed or implemented quantum-safe PAKE protocols. More specifically, Gao et al. [13] utilized the NFLlib library to accelerate the optimization of Ding's scheme [10], and gave a parameter set suitable for the use of the number theoretic transform (NTT) algorithm (for speeding up polynomial multiplication), but the proposed parameter set does not consider the communication burden. Yang et al. [30] further optimized Ding's solution, but only provided one lightweight parameter set without considering multiple security requirements. Moreover, inspired by the two-party, Liu et al. [21] presented a three-party RLWE-based PAKE protocol, where two clients aim to agree on a session key with the help of a trusted server. To our knowledge, as a compromise between learning with errors (LWE) and RLWE, MLWE [9] retains the matrix format, and concurrently introduces the ring polynomials. Therefore, when designing a lattice-based scheme in multiple security scenarios, MLWE is more flexible and straightforward than other primitives [8].

Given the above, we try to solve the following question: *Is it possible to construct an efficient and lightweight MLWE-based asymmetric PAKE protocol while resisting against quantum computer attacks?*

1.1 Contributions

In this work, we answer the above question in the affirmative. We construct a three-flow asymmetric PAKE protocol which is a parallel extension of the class of Random Oracle Model (ROM)-based PAK protocol [22] but in the module lattice setting. We prove its security under the BPR model and implement 3 deployment schemes that are tailored to the security level as well as the potential applications.

To construct the protocol efficiently, the majority of lattice-based schemes [1, 10, 13, 30] are based on the RLWE problem. However, in the light of our observation, the MLWE problem [9] with the advantage of a compromise between LWE and RLWE is more suitable for the construction of practical PAKE. Using the feature of MLWE, by superimposing or reducing the number of ring polynomials, different deployment schemes can be realized. As a result, we propose the practical MLWE-based PAKE protocol in the random oracle model.

By constructing the PAKE as a self-contained system, we demonstrate that our protocol is directly dependent on the hardness of MLWE and PWE, which can be reduced to MLWE. The security of our proposed protocol is proved under the BPR model [2] with two adaptions: first, to simplify the proof, we introduce the PWE assumption; second, we use the CDF-Zipf model [29] to characterize the ability of the adversary to conduct an online dictionary attack. Finally, we establish a complete security proof of the protocol, reduce its security to online dictionary attacks, and demonstrate that it satisfies the forward security.

In terms of concrete implementation, we comprehensively consider indicators such as failure rate, post-quantum security, communication cost, and computational efficiency, and select 3 high-quality parameter sets. To evaluate the performance of our proposals, we summarize the key technologies and the security level of state-of-the-art lattice-based schemes and our schemes in Table 2, and compare the running time, the communication cost and the failure rate of these schemes in Table 3. Particularly, our Recommended-PAK offers 177-bit post-quantum security with a generous margin to cope with later improvement in cryptanalysis. Compared with the latest RLWE-based Yang-PAK, the communication cost and the running time are reduced by 36.8% and 13.8%, respectively. Finally, in conjunction with the performance results, we discuss two potential real-world applications for our MLWE-PAK protocol: resource-constrained IoT devices and classical/post-quantum TLS.

2 Preliminaries

In this section, we provide both the notations of the parameters used in our construction and the description of some basic knowledge.

2.1 Notations

If A is a probabilistic algorithm, $a \leftarrow \mathsf{A}(b)$ represents the output of A assigned to a. If χ is a probability distribution, $a \leftarrow \chi$ denotes sampling a following

χ. We represent sampling a uniformly at random from a set S as $a \leftarrow S$. We denote $R_q = \mathbb{Z}_q[X]/(X^n+1)$ as the ring of integer polynomials modulo (X^n+1) where each coefficient is reduced by modulus q. We define a centered binomial distribution with parameter $\eta \in \mathbb{Z}_+$ as β_η. Throughout this paper, a normal font letter such as p represents a ring element in R_q. For the vector $\mathbf{v}$ including d elements, we denote it as $\mathbf{v} \in R_q^d$ using the bold lower-case letter; for the matrix $\mathbf{A}$ consisting of $m \times n$ entities, we denote it as $\mathbf{A} \in R_q^{m \times n}$ using the bold upper-case letter. By default, all vectors are column vectors. For a vector $\mathbf{v}$ (or a matrix $\mathbf{A}$), $\mathbf{v}^T$ (or $\mathbf{A}^T$) is used as its transpose.

2.2 The Decision Module-LWE Problem

Here, we define the decision version of the MLWE problem as follows.

Definition 1 (The decision $\mathsf{MLWE}_{n,d,q,\chi}$ problem). *Let n, d and $q \geq 2$ be the degree of a polynomial, the dimension of a vector, and the modulus, respectively. Let χ be an error distribution and $\mathbf{s} \leftarrow \chi^d$. Define $O_{\chi,\mathbf{s}}$ as the oracle which does the following:*

1. *Sample $\boldsymbol{A} \leftarrow R_q^{d \times d}$, $\boldsymbol{e} \leftarrow \chi^d$;*
2. *Return $(\mathbf{A}, \mathbf{As} + \mathbf{e}) \in R_q^{d \times d} \times R_q^d$.*

The decision MLWE problem for n, d, q, χ is to distinguish between polynomial independent samples from $O_{\chi,\mathbf{s}}$ and the same number of independent samples from an oracle $\mathcal{U}$ that returns uniform random samples from $(R_q^{d \times d}, R_q^d)$.

Remark 1. The secret $\mathbf{s}$ is chosen from the error distribution instead of the uniform distribution since the literature [9] has shown that this problem is as hard as the one in which $\mathbf{s}$ is chosen uniformly at random.

2.3 Bellare-Pointcheval-Rogaway Security Model

Here we review the BPR model [2] that will be used in our security analysis.

Participants, passwords, and execution of the protocol. A client is denoted as $C \in \mathcal{C}$ and a server is denoted as $S \in \mathcal{S}$. Each client C holds a password pw_c, which is independently sampled from the password space $\mathcal{D}$ in accordance with Zipf's law [20], and each server S holds correlated hash value $H(pw_c)$. Moreover, in this model, each participant enables to execute the protocol with different partners multiple times. Thus, we denote instance i of participants $U \in \mathcal{U} = \mathcal{C} \cup \mathcal{S}$ as Π_U^i. Each Π_U^i can be used only once.

Adversarial model. We assume that an adversary $\mathcal{A}$ completely controls the network and provides the input to the instance of principals. Formally, as a probabilistic algorithm with a distinguished query tape, $\mathcal{A}$ launches attacks utilizing random queries in the real world. Thus, we summarize the allowed queries defined in [2] here.

- $\mathsf{Execute}(C, i, S, j)$: causes protocol P between Π_C^i and Π_S^j to be executed and outputs the transcript to $\mathcal{A}$.
- $\mathsf{Send}(U, i, M)$: causes message M to be sent to instance Π_U^i. Π_U^i computes what the protocol says, and sends it back to $\mathcal{A}$.
- $\mathsf{Reveal}(U, i)$: If the instance Π_U^i has accepted and holds its session key sk, this query outputs sk to the adversary $\mathcal{A}$.
- $\mathsf{Test}(U, i)$: A coin b possessed by Π_U^i is tossed, then the following happens. If $b = 0$, Π_U^i returns sk to $\mathcal{A}$; otherwise, it returns a random string drawing from the space of session keys.
- $\mathsf{Corrupt}(U)$: If $U \in Client$, pw_c is output; otherwise, $H(pw_c)$ is output.

Partnering. An instance Π_C^i holding (pid, sid, sk) and an instance Π_S^j holding (pid', sid', sk') are *partnered*, if $pid = S, pid' = C, sid = sid', sk = sk'$, where pid, sid and sk denote the partner-id, the session-id and the session-key, respectively. In addition, no other instance accepts with its session-id equal to sid.

Freshness with forward secrecy. An instance Π_U^i is *fresh-fs* unless either 1) a $\mathsf{Reveal}(U, i)$ query occurs, or 2) a $\mathsf{Reveal}(U', j)$ query occurs, where Π_U^i has partnered with $\Pi_{U'}^j$, or 3) a $\mathsf{Corrupt}(U)$ query occurs before the $\mathsf{Test}(U, i)$ query and the $\mathsf{Send}(U, i, M))$ query.

Advantage of the adversary. We now define the advantage of the adversary against the authenticated key exchange protocol P. Let $\mathrm{Succ}_{\mathcal{A}}^{\mathsf{P}}(\lambda)$ be the event that the adversary $\mathcal{A}$ makes a $\mathsf{Test}(U, i)$ query to some fresh instances Π_U^i, and outputs a single bit b', where $b' = b$ for the bit b which was chosen in the Test query. The advantage of $\mathcal{A}$ is defined as follows

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{P}}(\lambda) = 2\Pr[\mathrm{Succ}_{\mathcal{A}}^{\mathsf{P}}(\lambda)] - 1.$$

Furthermore, if we have two protocols P and P' which satisfy the following relationship

$$\Pr[\mathrm{Succ}_{\mathcal{A}}^{\mathsf{P}}(\lambda)] = \Pr[\mathrm{Succ}_{\mathcal{A}}^{\mathsf{P}'}(\lambda)] + \epsilon,$$

then we have the fact that

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{P}}(\lambda) = \mathsf{Adv}_{\mathcal{A}}^{\mathsf{P}'}(\lambda) + 2\epsilon.$$

2.4 Error Reconciliation Mechanism

In [17], Jin and Zhao formally formulated a universal and convenient error reconciliation mechanism referred to as optimally-balanced key consensus with noise (OKCN). The inherent upper-bound analyzed in Jin's paper guides the parameter selection and balances between the accuracy and the bandwidth. Especially, OKCN is more suitable for incorporating into the existing DH-based protocols like TLS, IKE. Thus, it shows more advantages in choosing OKCN as the error reconciliation mechanism of our scheme.

Before showing the description of the OKCN algorithm, we first give a function $|a-b|_q$ to represent the distance between two elements $a, b \in \mathbb{Z}_q$.

$$|a-b|_q = \min\{(a-b) \bmod q, (b-a) \bmod q\}.$$

Moreover, for two approximate polynomials $w = \sum_{i=0}^{n-1} w_i X^i, v = \sum_{i=0}^{n-1} v_i X^i$, we define the distance between them as

$$|w-v|_q = \max\{|w_1 - v_1|_q, |w_2 - v_2|_q, \cdots, |w_{n-1} - v_{n-1}|_q\}.$$

Algorithm 1 OKCN: Optimally-balanced Key Consensus with Noise

$\mathsf{params} = (q, m, g, l, aux), aux = \{q' = \mathsf{lcm}(q, m), \alpha = q'/q, \beta = q'/m\}$
procedure CON(σ_s, params) ▷ $\sigma_s \in [0, q-1]$
  $e \leftarrow [-\lfloor(\alpha-1)/2\rfloor, \lfloor\alpha/2\rfloor]$
  $\sigma_A = (\alpha\sigma_s + e) \bmod q'$
  $k_s = \lfloor\sigma_A/\beta\rfloor \in \mathbb{Z}_m$
  $v' = \sigma_A \bmod \beta$
  $v = \lfloor v'g/\beta\rfloor$
  return (k_s, v)
end procedure
procedure REC(σ_c, v, params) ▷ $\sigma_c \in [0, q-1]$
  $k_c = \lfloor\alpha\sigma_c/\beta - (v+1/2)/g\rceil \bmod m$
  return k_c
end procedure

Algorithm 1 describes the calculation process of the conciliate function Con and the reconcile function Rec. The error reconciliation can be extended to R_q by applying OKCN to each coefficient of the ring. For any ring polynomial $\sigma = \{\sigma_0, \cdots, \sigma_{n-1}\} \in R_q$, we set $\mathsf{Con}(\sigma) = \{\mathsf{Con}(\sigma_0), \cdots, \mathsf{Con}(\sigma_{n-1})\}$. Rec function does the same way.

Theorem 1 (Efficiency Upper Bound [17]). *$OKCN = (\mathsf{params}, Con, Rec)$ is a secure and correct mechanism, then*

$$(2l+1)m < q(1 - \frac{1}{g})$$

with $\mathsf{params} = (q, m, g, l, aux)$, where q is the modulus, m is the length of each negotiated value, g is the length of the hint signal, l represents the distance between two approximate polynomials σ_c and σ_s in R_q.

3 MLWE-based PWE Assumption & Security Reduction

To expediently prove the security of our construction, we proposed the PWE assumption with the different version of the MLWE problem. This assumption with the version of the RLWE problem first appeared in the literature [10] so as to provide the security proof of its RLWE-PAKE.

Definition 2 (Pairing with Errors). *We define a probabilistic, polynomial-time adversary $\mathcal{A}$ taking $(\mathbf{A}, \mathbf{x}, \mathbf{y}, v)$ as its input, where $\mathbf{A} \leftarrow R_q^{d\times d}$, $\mathbf{x} \leftarrow R_q^d$, $\mathbf{s}, \mathbf{e} \leftarrow \beta_\eta^d$, $e_\sigma \leftarrow \beta_\eta$, $\mathbf{y} = \mathbf{A}^T\mathbf{s} + \mathbf{e}$, $(v, k) = \mathsf{Con}(\mathbf{x}^T\mathbf{s} + e_\sigma)$. The goal of $\mathcal{A}$ is to obtain the value of string k from its output. Therefore, we define the advantage of $\mathcal{A}$ as follows:*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{PWE}}(\lambda) = \Pr[k \in \mathcal{A}(\mathbf{A}, \mathbf{x}, \mathbf{y}, v)].$$

Let $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{PWE}}(t, N) = \max_{\mathcal{A}}\{\mathsf{Adv}_{\mathcal{A}}^{\mathsf{PWE}}(\lambda)\}$, where the maximum is take over all adversaries in running time t that output a list containing at most N elements of $\{0,1\}^k$. The PWE assumption denotes that $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{PWE}}(t, N)$ is negligible for t and N which are polynomial in security parameter λ.

Now, we describe a reduction from the PWE assumption to the decision MLWE problem. We consider the following sequence of reductions:

$$\text{PWE} \longrightarrow \text{DPWE} \xrightarrow{\text{Lemma 1}} \text{MLWE-DH} \xrightarrow{\text{Lemma 2}} \text{D-MLWE}$$

The decision version of the PWE problem can be defined as follows. Obviously, if DPWE is hard so is PWE.

Definition 3. (DPWE). *Given $(\mathbf{A}, \mathbf{x}, \mathbf{y}, v, k) \in R_q^{d\times d} \times R_q^d \times R_q^d \times \{0,1\}^n \times \{0,1\}^k$ where $(v, k) = \mathsf{Con}(\sigma)$ for some $\sigma \in R_q$. The decision Pairing with Errors (DPWE) problem is to decide whether $\sigma = \mathbf{x}^T\mathbf{s} + e_\sigma$ and $\mathbf{y} = \mathbf{A}^T\mathbf{s} + \mathbf{e}$ for some $\mathbf{s}, \mathbf{e} \leftarrow \beta_\eta^d, e_\sigma \leftarrow \beta_\eta$, or $(\sigma, \mathbf{y})$ is uniformly random in $R_q \times R_q^d$.*

Definition 4. (MLWE-DH). *Given $(\mathbf{A}, \mathbf{x}, \mathbf{y}, \sigma)$, where $(\mathbf{A}, \mathbf{x})$ is uniformly random in $R_q^{d\times d} \times R_q^d$, the MLWE-DH problem is to figure out whether $\sigma = \mathbf{x}^T\mathbf{s} + e_\sigma$ and $\mathbf{y} = \mathbf{A}^T\mathbf{s} + \mathbf{e}$ for some $e_\sigma \leftarrow \beta_\eta$ and $\mathbf{e} \leftarrow \beta_\eta^d$, or $(\sigma, \mathbf{y})$ is uniformly random in $R_q \times R_q^{d\times d}$.*

Lemma 1. *The DPWE problem is hard if the MLWE-DH problem is hard.*

Proof. Suppose that the MLWE-DH problem is hard to solve, and there exists a polynomial-time algorithm D can solve the DPWE problem with non-negligible probability. Using algorithm D as a subroutine call, we build a distinguisher D' to solve the MLWE-DH problem.

1. Input $(\mathbf{A}, \mathbf{x}, \mathbf{y}, \sigma)$
2. Compute $(v, k) = \mathsf{Con}(\sigma)$
3. Call D to solve DPWE problem using the input $(\mathbf{A}, \mathbf{x}, \mathbf{y}, v, k)$
4. If D outputs 1 then D' outputs 1, means that $\sigma = \mathbf{x}^T\mathbf{s} + e_\sigma$ and $\mathbf{y} = \mathbf{A}^T\mathbf{s} + \mathbf{e}$; otherwise, $(\sigma, \mathbf{y})$ is uniformly random in $R_q \times R_q^d$.

Since D solves the DPWE problem with a non-negligible advantage, D' solves the MLWE-DH problem with a non-negligible advantage as well, which contradicts the hardness assumption of the MLWE-DH problem. □

Lemma 2. *If the D-MLWE problem is hard, the MLWE-DH problem is hard as well.*

Proof. Suppose that there exists a probabilistic polynomial-time algorithm D can solve the MLWE-DH problem with non-negligible advantage. Given two samples of an MLWE challenge $(\mathbf{A}_1, \mathbf{b}_1)$ and $(\mathbf{a}_2, b_2)$, both share the same $\mathbf{s} \leftarrow \beta_\eta^d$. We call algorithm D to build a distinguisher D' to solve the MLWE problem as follows:

1. Set $(\mathbf{A}, \mathbf{x}, \mathbf{y}, \sigma) = (\mathbf{A}_1, \mathbf{a}_2, \mathbf{b}_1, b_2)$
2. Input $(\mathbf{A}, \mathbf{x}, \mathbf{y}, \sigma)$ to D
3. If D outputs 1, then D' outputs 1 which D' determines $b_2 = \mathbf{a}_2^T\mathbf{s} + e_\sigma$ and $\mathbf{b}_1 = \mathbf{A}_1^T\mathbf{s} + \mathbf{e}$ for some $e_\sigma \leftarrow \beta_\eta, \mathbf{e} \leftarrow \beta_\eta^d$; otherwise, D outputs 0, then D' outputs 0, means that $b_2, \mathbf{b}_1$ is uniformly random in $R_q \times R_q^d$.

Obviously, D' can solve the MLWE problem with non-negligible advantage as well, which contradicts the hardness of the MLWE problem. Hence, if the MLWE problem is hard to solve then the MLWE-DH problem is also hard to solve. □

4 Our MLWE-based PAKE Scheme

Here we describe our MLWE-PAKE protocol in detail and show its correctness.

4.1 Protocol Description

Client C		**Server** S
Input S, pw_c		$\mathbf{\Gamma}' = -H_1(pw_c)$
$\rho \sim \{0,1\}^{256}$		
$\mathbf{A} \sim R_q^{d\times d} := Sam(\rho)$		Abort if $\mathbf{m} \notin R_q^d$
$(\mathbf{s}_c, \mathbf{e}_c) \leftarrow \beta_\eta^d \times \beta_\eta^d$	$\xrightarrow{C,\mathbf{m},\rho}$	$\mathbf{A} \sim R_q^{d\times d} := Sam(\rho)$
$\mathbf{y}_c = \mathbf{A}\mathbf{s}_c + \mathbf{e}_c$		$(\mathbf{s}_s, \mathbf{e}_s) \leftarrow \beta_\eta^d \times \beta_\eta^d$
$\mathbf{\Gamma} = H_1(pw_c)$		$\mathbf{y}_s = \mathbf{A}^T\mathbf{s}_s + \mathbf{e}_s$
$\mathbf{m} = \mathbf{y}_c + \mathbf{\Gamma}$		$\mathbf{y}_c = \mathbf{m} + \mathbf{\Gamma}'$
		$e_\sigma \leftarrow \beta_\eta$
		$\sigma_s = \mathbf{y}_c{}^T\mathbf{s}_s + e_\sigma$
Abort if $\mathbf{y}_s \notin R_q^d$	$\xleftarrow{\mathbf{y}_s,v,k}$	$(k_\sigma, v) = \mathsf{Con}(\sigma_s)$
$\sigma_c = \mathbf{s}_c^T\mathbf{y}_s$		$k = H_2(C, S, \mathbf{m}, \mathbf{y}_s, k_\sigma, \mathbf{\Gamma}')$
$k_\sigma = \mathsf{Rec}(\sigma_c, v)$		$k'' = H_3(C, S, \mathbf{m}, \mathbf{y}_s, k_\sigma, \mathbf{\Gamma}')$
Abort if $k \neq H_2(C, S, \mathbf{m}, \mathbf{y}_s, k_\sigma, \mathbf{\Gamma}')$		
$k' = H_3(C, S, \mathbf{m}, \mathbf{y}_s, k_\sigma, \mathbf{\Gamma}')$	$\xrightarrow{k'}$	Abort if $k' \neq k''$
$sk_c = H_4(C, S, \mathbf{m}, \mathbf{y}_s, k_\sigma, \mathbf{\Gamma}')$		$sk_s = H_4(C, S, \mathbf{m}, \mathbf{y}_s, k_\sigma, \mathbf{\Gamma}')$

Fig. 1. The MLWE-based PAKE protocol

Figure 1 describes the complete protocol. Let the rank n of a ring be a power of 2. Let q be an odd prime such that $q \equiv 1 \bmod 2n$. Function $H_1 : \{0,1\}^* \rightarrow R_q^d$

hashes passwords into a vector in R_q^d; $H_i : \{0,1\}^* \to \{0,1\}^{\mathsf{k}}$ $(i = 2,3)$ be hash functions for verification of communications; $H_4 : \{0,1\}^* \to \{0,1\}^{\mathsf{k}}$, the Key Derivation Function (KDF), generates the session-key of length k-bit. The comprehensive protocol is described as follows:

- **Client Initiation.** A client C takes on the role of initiating the protocol by inputting its password and the server-id S. Then, C produces its ephemeral key pair $(\mathbf{y}_c, \mathbf{s}_c)$, and uses the password to encapsulate $\mathbf{y}_c$ as the message $\mathbf{m}$. Finally, C finishes the initiation process by sending initialized message $\langle C, \mathbf{m}, \rho \rangle$ to the server S, where ρ is a seed that generates the public matrix.
- **Server Response.** Upon receiving $\langle C, \mathbf{m}, \rho \rangle$, the server S checks the rationality of $\mathbf{m}$. If $\mathbf{m} \in R_q^d$, S first generates ephemeral public-key/secret-key pair $(\mathbf{y}_s, \mathbf{s}_s)$. Then, S recovers $\mathbf{y}_c$ from $\mathbf{m}$. In the next moment, S uses $\mathbf{s}_s$ and $\mathbf{y}_c$ to compute the coordination polynomial σ_s and the shared-key $(k_\sigma, v) = \mathsf{Con}(\sigma_s)$. Subsequently, S generates two hash values k, k'' for indicating its identity and verifying the identity of C, respectively. Finally, S sends its public-key $\mathbf{y}_s$, the signal hint v and the hash value k to C.
- **Client Finish.** After receiving $\langle \mathbf{y}_s, v, k \rangle$, C utilizes its secret-key $\mathbf{s}_c$ and the public-key $\mathbf{y}_c$ to generate its coordination polynomial σ_c (which is appropriately equal to σ_s) so that it can obtain the shared key through $k_\sigma = \mathsf{Rec}(\sigma_c, v)$. C verifies the identity of S and generates the hash value k' to indicate its identity. Finally, C sends out k' and derives the session-key sk_c.
- **Server Finish.** Server S verifies k' and k'' in the same way. If $k' = k''$, S computes its session-key sk_s; otherwise, S rejects to compute a session-key.

Remark 2 (Mutual authentication). The client hashes its password, appends it to the message, and sends it out. At this time, only the server storing the hash value of the password can correctly recover the message and use the hash function H_2 to generate its verification key k. When the client receives the verification key k from the server, the client uses reconcile function Rec to negotiate its shared key k_σ and verifies whether H_2 is equal to k, thereby achieving authentication of the server. Meanwhile, it uses H_3 to generate its verification key k' provided verification information to the server. Finally, mutual authentication finishes.

4.2 Correctness

Obviously, the protocol's correctness is revealed by the equality of the values negotiated by the participants. Therefore, the correctness depends on the distance between the two approximate elements used to derive the shared-key. If this distance is within the allowable range of OKCN, correctness can be guaranteed. We compare the two approximate polynomials in R_q:

$$\begin{aligned} |\sigma_s - \sigma_c|_q &= |\mathbf{y}_c^T \mathbf{s}_s + e_\sigma - \mathbf{s}_c^T \mathbf{y}_s|_q \\ &= |\mathbf{s}_c^T \mathbf{A}^T \mathbf{s}_s + \mathbf{e}_c^T \mathbf{s}_s + e_\sigma - \mathbf{s}_c^T \mathbf{A}^T \mathbf{s}_s + \mathbf{s}_c^T \mathbf{e}_s|_q \\ &= |\mathbf{e}_c^T \mathbf{s}_s + e_\sigma - \mathbf{s}_c^T \mathbf{e}_s|_q \\ &\leq l < \frac{q(1-\frac{1}{g}) - m}{2m}. \end{aligned}$$

P_0	The original protocol P.
P_1	If the value of $\mathbf{m}$ or $\mathbf{y}_s$ randomly chosen by the honest participants has already appeared in the previous protocols, then the protocol halts and $\mathcal{A}$ fails.
P_2	The protocol answers Send and Execute queries without making any random oracle queries. Subsequently, random oracle queries are backpatched to be consistent with the responses to Send and Execute queries as much as possible.
P_3	If an $H_l(\cdot), l \in \{2,3,4\}$ query is made, it is not checked for consistency against Execute queries. The protocol responds with a random output instead of maintaining consistency with an Execute query.
P_4	If a correct password guess is made against an instance Π_C^i or Π_S^j before a Corrupt query, the protocol halts and the adversary automatically succeeds.
P_5	Once the adversary $\mathcal{A}$ makes a password guess against the client and the server instances that have partnered, the protocol halts and the adversary fails.
P_6	There is an internal password oracle that knows all passwords and tests the correctness of a given password for a specific client/server pair.

Fig. 2. Description of the protocol P_0 through P_6

By Theorem 1, if $|\sigma_s - \sigma_c| \leq l$ where $l < \frac{q(1-\frac{1}{g})-m}{2m}$, both sides can reconcile the same value in our MLWE-PAK scheme. We represent the failure rate δ of our scheme as

$$\delta = \Pr\Big[|\sigma_s - \sigma_c|_q \geq \frac{q(1-\frac{1}{g})-m}{2m}\Big].$$

In section 6, we will select the parameter sets to make the failure rate δ negligible in the practical implementations.

5 Proof of Security

The security proof is to show that $\mathcal{A}$ attacking the protocol is unable to determine the fresh sk with greater advantage than that of an *online* dictionary attack.

Theorem 2 (Advantage of the adversary). *The PAKE protocol P is secure, if the advantage of the adversary $\mathcal{A}$ is*

$$Adv_{\mathcal{A}}^{\mathsf{P}}(\lambda) \leq C' N^{s'}(\lambda) + O\big(n_{se} Adv_D^{PWE}(\lambda) + Adv_D^{PWE}(\lambda)\big) + negl(\lambda).$$

where $N(\lambda)$ is the number of online attacks, and the password dictionary follows the Zipf-like distribution with parameter $C' = 0.062239$ and $s' = 0.155478$ [20].

Remark 3. The majority of existing PAKE schemes (e.g. [10, 19]) assumed that passwords adhere to a uniformly random distribution. Thus, the advantage of the adversary $\mathcal{A}$ was formulated as $N(\lambda)/\mathcal{D} + negl(\lambda)$. However, according to [20], the traditional uniform-model based expression significantly underestimates

the power of real adversary $\mathcal{A}$. Instead, we prefer to characterize the adversary $\mathcal{A}$ using the CDF-Zipf model [29].

Proof. Our proof will proceed by introducing a series of protocols $\mathsf{P}_0, \mathsf{P}_1, \cdots, \mathsf{P}_6$ (can be seen in Figure 2) related to P, with $\mathsf{P}_0 = \mathsf{P}$. In P_6, $\mathcal{A}$ will be reduced to an *online* guessing attack which will provide a straightforward analysis. The detail description of each P_i can be seen in Figure 2. A fixed adversary $\mathcal{A}$ makes n_{se}, n_{ex}, n_{ro} queries of Send, $\mathsf{Execute}$ and random oracles, respectively.

Corollary 1. *For any adversary $\mathcal{A}$, we have that*

$$|\mathsf{Adv}_{\mathcal{A}}^{\mathsf{P}_1}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{\mathsf{P}_0}(\lambda)| \leq negl(\lambda).$$

Proof. By inspection, the probability that the $\mathbf{m}$ or $\mathbf{y}_s$ has appeared in previous Send, $\mathsf{Execute}$, or random oracle query is $\frac{n_{se}+n_{ex}+n_{ro}}{q^{nd}}$. Consider the newly generated value $\mathbf{m}$ or $\mathbf{y}_s$, there are $(n_{se} + n_{ex})$ values to obtain from the Send and $\mathsf{Execute}$ query uniquely. Therefore, the probability of the $\mathbf{m}$ or $\mathbf{y}_s$ which has been generated previously is $\frac{O((n_{se}+n_{ex})(n_{se}+n_{ex}+n_{ro}))}{q^{nd}}$, where the space of R_q^d is q^{nd}. However, this probability is negligible. □

Corollary 2. *For any adversary $\mathcal{A}$, we have that*

$$|\mathsf{Adv}_{\mathcal{A}}^{\mathsf{P}_2}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{\mathsf{P}_1}(\lambda)| \leq negl(\lambda).$$

Proof. This design of P_2 is a standard technique for security analysis of protocols involving random oracles. P_1 and P_2 are indistinguishable unless the adversary makes an $H_l(\cdot)$ query, for $l \in \{2, 3, 4\}$, but the adversary has not actually made the $H_1(pw_c)$ query, the total of probability for this case can be bounded by $O(\frac{n_{ro}}{q^{nd}})$. Or the adversary makes a $\mathsf{Send}(C, i, k)$ (resp. $\mathsf{Send}(S, j, k')$) query that is not the output of an $H_2(\cdot)$ (resp. $H_3(\cdot)$) query which would be a correct password guess. It is easy to show that the probability of this case can be bound by $O(\frac{n_{se}}{2^{\mathsf{k}}})$. The corollary follows. □

Corollary 3. *For any adversary $\mathcal{A}$, we have that*

$$|\mathsf{Adv}_{\mathcal{A}}^{\mathsf{P}_3}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{\mathsf{P}_2}(\lambda)| \leq negl(\lambda).$$

Proof. This can be shown using a reduction from PWE. Given $(\mathbf{A}, \mathbf{x}, \mathbf{y}, v)$, we construct an algorithm D that attempts to solve PWE assumption by running $\mathcal{A}$ on a simulation of the protocol P_2 with these changes:

(1) In an $\mathsf{Execute}$ query, D set $\mathbf{m} = \mathbf{x} + (\mathbf{A}\mathbf{s}_f + \mathbf{e}_f)$, $\mathbf{y}_s = \mathbf{y} + (\mathbf{A}^T\mathbf{s}_{ff} + \mathbf{e}_{ff})$ where $\mathbf{s}_f, \mathbf{e}_f, \mathbf{s}_{ff}, \mathbf{e}_{ff} \leftarrow \beta_\eta^d$, and select $v \leftarrow \{0, 1\}^n$.
(2) When $\mathcal{A}$ finishes, for every $H_l(\cdot)$ query where $l \in \{2, 3, 4\}$, $\mathbf{m}$ and $\mathbf{y}_s$ were generated in an $\mathsf{Execute}$ query, and an $H_1(pw_c)$ query returned $-\mathbf{\Gamma}' = \mathbf{A}\mathbf{s}_h + \mathbf{e}_h \in R_q^d$, then the simulator can compute,

$$\begin{aligned}\sigma_s &= \mathbf{y}_c{}^T \cdot (\mathbf{s}_s + \mathbf{s}_{ff}) + e_\sigma = (\mathbf{x} + \mathbf{A}(\mathbf{s}_f - \mathbf{s}_h) + (\mathbf{e}_f - \mathbf{e}_h))^T \cdot (\mathbf{s}_s + \mathbf{s}_{ff}) + e_\sigma \\ &\approx \mathbf{x}^T \cdot \mathbf{s}_s + (\mathbf{s}_f - \mathbf{s}_h)^T \cdot \mathbf{y} + (\mathbf{x} + \mathbf{A}(\mathbf{s}_f - \mathbf{s}_h) + (\mathbf{e}_f - \mathbf{e}_h))^T \cdot \mathbf{s}_{ff} \\ &= \mathbf{x}^T \cdot \mathbf{s}_s + (\mathbf{s}_f - \mathbf{s}_h)^T \cdot \mathbf{y} + (\mathbf{x} + \mathbf{\Gamma}' + (\mathbf{A}\mathbf{s}_f + \mathbf{e}_f))^T \cdot \mathbf{s}_{ff},\end{aligned}$$

$$k_{\sigma'} = \mathsf{Rec}(\mathbf{x}^T \cdot \mathbf{s}_s, v) = \mathsf{Rec}(\sigma_s - (\mathbf{s}_f - \mathbf{s}_h)^T \cdot \mathbf{y} - (\mathbf{x} + \mathbf{\Gamma}' + (\mathbf{A}\mathbf{s}_f + \mathbf{e}_f))^T \cdot \mathbf{s}_{ff}, v).$$

Finally, add the value of $k_{\sigma'}$ to the list of possible values. Thus, $|\mathsf{Adv}^{\mathsf{P}_2}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{\mathsf{P}_3}_{\mathcal{A}}(\lambda)| \leq 2\mathsf{Adv}^{\mathsf{PWE}}_{D}(\lambda)$ which is negligible. □

Corollary 4. *For any adversary $\mathcal{A}$, we have that*

$$\mathsf{Adv}^{\mathsf{P}_3}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{P}_4}_{\mathcal{A}}(\lambda).$$

Proof. This change will only increase the probability of $\mathcal{A}$ winning the game. □

Corollary 5. *For any adversary $\mathcal{A}$, we have that*

$$|\mathsf{Adv}^{\mathsf{P}_5}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{\mathsf{P}_4}_{\mathcal{A}}(\lambda)| \leq negl(\lambda).$$

Proof. We define the event pairedpwguess that $\mathcal{A}$ makes a password guess against partnered client and server. Obviously, if the pairedpwguess event does not occur, P_4 and P_5 are indistinguishable, thus, we define the event as E with probability ϵ. If $\mathcal{A}$ attacking P_4, we have that $\Pr\left[\mathrm{Succ}^{\mathsf{P}_4}_{\mathcal{A}}(\lambda)\right] \leq \Pr\left[\mathrm{Succ}^{\mathsf{P}_5}_{\mathcal{A}}(\lambda)\right] + \epsilon$, and go further, $\mathsf{Adv}^{\mathsf{P}_4}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{P}_5}_{\mathcal{A}}(\lambda) + 2\epsilon$.

Now we build an algorithm D by calling $\mathcal{A}$ on a simulation of the protocol to solve the PWE assumption. Give the sample $(\mathbf{A}, \mathbf{x}, \mathbf{y}, v)$, D choose a random $m \in \{1, 2, \cdots, n_{se}\}$ and simulates P_4 for $\mathcal{A}$ as follows:

(1) In the mth $\mathsf{Send}(C, i', S)$ query to the instance $\Pi^{i'}_C$, set $\mathbf{m} = \mathbf{x}$.
(2) In a $\mathsf{Send}(S, j, \langle C, \mathbf{m}, \rho\rangle)$ query, set $\mathbf{y}_s = \mathbf{y} + (\mathbf{A}\mathbf{s}_f + \mathbf{e}_f)$ where $\mathbf{s}_f, \mathbf{e}_f \leftarrow \beta^d_\eta$.
(3) In a $\mathsf{Send}(C, i', \langle \mathbf{y}_s, v, k\rangle)$ query, if $\Pi^{i'}_C$ is unpaired, D outputs 0 and halts.
(4) In a $\mathsf{Send}(S, j, k)$ query to Π^j_S, if Π^j_S has paired with $\Pi^{i'}_C$ after its $\mathsf{Send}(S, j, \langle C, \mathbf{m}, \rho\rangle)$ query, but is not now paired with $\Pi^{i'}_C$, no test for correctpw is made, and Π^j_S aborts.
(5) When $\mathcal{A}$ finishes, for every $H_l(\cdot)$ query for $l \in \{2, 3, 4\}$, where $\mathbf{m}$ and $\mathbf{y}_s$ were generated in an $\Pi^{i'}_C$ query, and an $H_1(pw_c)$ query return $-\mathbf{\Gamma}' = \mathbf{A}\mathbf{s}_h + \mathbf{e}_h \in R^d_q$, then the simulator computes:

$$\begin{aligned}\sigma_s &= \mathbf{y}_c^T \cdot (\mathbf{s}_s + \mathbf{s}_f) + e_\sigma = (\mathbf{x} - (\mathbf{A}\mathbf{s}_h + \mathbf{e}_h))^T \cdot (\mathbf{s}_s + \mathbf{s}_f) + e_\sigma \\ &\approx \mathbf{x}^T \cdot \mathbf{s}_s - \mathbf{s}_h{}^T \cdot \mathbf{y} + (\mathbf{x} - (\mathbf{A}\mathbf{s}_h + \mathbf{e}_h))^T \cdot \mathbf{s}_f \\ &= \mathbf{x}^T \cdot \mathbf{s}_s - \mathbf{s}_h{}^T \cdot \mathbf{y} + (\mathbf{x} + \mathbf{\Gamma}')^T \cdot \mathbf{s}_f,\end{aligned}$$

$$k_{\sigma'} = \mathsf{Rec}(\mathbf{x}^T \cdot \mathbf{s}_s, v) = \mathsf{Rec}(\sigma_s + \mathbf{s}_h{}^T \cdot \mathbf{y} - (\mathbf{x} + \mathbf{\Gamma}')^T \cdot \mathbf{s}_f, v).$$

Finally, add the value of $k_{\sigma'}$ to the list of possible values. Thus, $|\mathsf{Adv}^{\mathsf{P}_4}_{\mathcal{A}}(\lambda) - \mathsf{Adv}^{\mathsf{P}_5}_{\mathcal{A}}(\lambda)| \leq 2n_{se}\mathsf{Adv}^{PWE}_{D}(\lambda)$ which is negligible. □

Corollary 6. *For any adversary $\mathcal{A}$, we have that*

$$\mathsf{Adv}^{\mathsf{P}_5}_{\mathcal{A}}(\lambda) = \mathsf{Adv}^{\mathsf{P}_6}_{\mathcal{A}}(\lambda).$$

Proof. It is visible that P_5 and P_6 are perfectly indistinguishable. □

Now, we analyze the advantage of $\mathcal{A}$ against the protocol P_6. In the P_6, the adversary $\mathcal{A}$ has only two ways to succeed, making online guessing attacks or making a Test query, namely. We define the event $\mathsf{correctpw}$ as $\mathcal{A}$ successfully obtaining the password through online dictionary attacks. Thus, the probability of the event $\mathsf{correctpw}$ can be bounded by $C' n_{se}^{s'}$ following the CDF-Zipf model [29].

For the second case, $\mathcal{A}$ makes a Test query to a fresh instance Π_U^i, since the view of $\mathcal{A}$ is independent of sk_U^i, the probability of success is exactly $\frac{1}{2}$. Thus,

$$\begin{aligned}\Pr[\mathrm{Succ}^{\mathsf{P}_6}_{\mathcal{A}}(\lambda)] &\leq \Pr[\mathsf{correctpw}] + \Pr[\mathrm{Succ}^{\mathsf{P}_6}_{\mathcal{A}}(\lambda)|\neg\mathsf{correctpw}]\Pr[\neg\mathsf{correctpw}] \\ &\leq C' n_{se}^{s'} + \frac{1}{2}(1 - C' n_{se}^{s'}) \\ &\leq \frac{1}{2}(1 + C' n_{se}^{s'}).\end{aligned}$$

Then, we conclude that $\mathsf{Adv}^{\mathsf{P}_6}_{\mathcal{A}}(\lambda) = 2\Pr[\mathrm{Succ}^{\mathsf{P}_6}_{\mathcal{A}}(\lambda)] - 1 \leq C' n_{se}^{s'}$. Finally, the Theorem 2 follows from the above conclusion and corollaries from 1 to 6. □

6 Implementation

This section gives all details of our implementation of MLWE-PAK written in C and describes the encoding of messages.

6.1 Parameter Selection

Table 1 lists the concrete parameter sets. In the MLWE-based protocol, the most time-consuming operation is polynomial multiplication. An efficient parameter instantiation under the MLWE problem is such that the degree n of a polynomial is a power of 2 and the modulus q is a prime satisfying the congruence condition $q \equiv 1 \bmod 2n$. At this time, the underlying finite field exists primitive $2n$-th root of unity, so that the NTT algorithm can be used efficiently to perform polynomial multiplications. Consider that public-key protocols only need to transmit 256 bits of information, we decide to fix the degree $n = 256$. Increasing or decreasing the dimension d of the vector can change the security of the scheme. We choose the modulus $q = 7681$, as 7681 is the smallest prime that satisfies the aforementioned condition so that polynomials can be transferred in NTT encoding.

For the parameters of OKCN algorithm, we choose $(m, g, l) = (2, 2^6, 1895)$. $m = 2$ means that one-bit shared-key is negotiated by one coefficient of a polynomial. The size g of the hint value per coefficient is 6, which is a comprehensive consideration of the communication cost and the error tolerance distance.

Finally, we provide 3 parameter sets to deal with distinct security scenarios by changing the dimension d of the vector and the parameter η of the error distribution. The Lightweight-PAK can resist classical attacks and be deployed in the lightweight applications. The Recommended-PAK and the Paranoid-PAK with higher security are used to resist quantum attacks.

Table 1. Proposed parameter sets

Scheme	Parameters (n, q, d, η)	Failure rate	Com. cost $\mathcal{C} \to \mathcal{S}$	Com. cost $\mathcal{S} \to \mathcal{C}$
Lightweight-PAK	$(256, 7681, 2, 13)$	$2^{-53.4}$	928	1056
Recommended-PAK	$(256, 7681, 3, 8)$	$2^{-97.4}$	1,344	1,472
Paranoid-PAK	$(256, 7681, 4, 6)$	$2^{-131.6}$	1,760	1,888

6.2 NTT Domain

A polynomial multiplication can be performed by computing

$$c = \mathsf{NTT}^{-1}(\mathsf{NTT}(a) \circ \mathsf{NTT}(b))$$

where $\circ$ denotes the point-wise multiplication.

For a polynomial $p = \sum_{i=0}^{n} p_i X^i \in R_q$, we define the polynomial $\hat{p}$ in NTT domain as

$$\hat{p} = \mathsf{NTT}(p) = \sum_{i=0}^{n} \hat{p}_i X^i, \text{ where } \hat{p}_i = \sum_{j=0}^{n} \psi^j p_j \omega^{ij},$$

where we fix the primitive n-th root of unity $\omega = 3844 \in \mathbb{Z}_q$ and $\psi = \sqrt{\omega} = 62$.

As the inverse of function NTT, the computation of function NTT^{-1} uses $\omega^{-1} \bmod q = 6584$, after the summation, multiplies by powers of $\psi^{-1} \bmod q = 1115$ and multiplies each coefficient by the scalar $n^{-1} \bmod q = 7651$, so that

$$p = \mathsf{NTT}^{-1}(\hat{p}) = \sum_{i=0}^{n} p_i X^i, \text{ where } p_i = n^{-1}\psi^{-i} \sum_{j=0}^{n} \hat{p}_j \omega^{-ij}.$$

6.3 Hash Functions

In our protocol, we use 4 hash functions. Let $H_1(\cdot)$ be an extendable-output function (XOF), it extends a password pw into a polynomial vector $\mathbf{\Gamma}$. Let $H_l(\cdot),\ l \in \{2, 3\}$ be hash functions for verification of communications. And let $H_4(\cdot)$ be a key derivation function (KDF). We choose SHAKE-128 and SHA3-256 as our hash functions, both of them are provided by FIPS-202 [6]. We have $H_1(\cdot) =$ SHAKE-128, and $H_l(\cdot) =$ SHA3-256 where $l \in \{2, 3, 4\}$.

6.4 Generation of A

We use a 256-bit random seed ρ as the input for the generation of $\mathbf{A} = (a_{i,j}) \in R_q^{d \times d}$. For each entry $a_{i,j} \in R_q$, we expand ρ through SHAKE-128 into a stream of 16-bit little-endian integers. On the sequence of 16-bit integers, we adopt the following strategy: If an integer belongs to $\{0, 1, \ldots, q-1\}$, we accept it as the coefficient of $a_{i,j}$; otherwise, reject it. Finally, $\mathbf{A}$ including $d \times d$ polynomials is considered to be in NTT domain, since NTT transforms uniform noise to uniform noise.

6.5 Generation of Noise Polynomials

For each noise polynomial e, we sample it from β_η. First, we extend a uniformly random seed into an array of $n = 256$ with each element of 2η-bit, and then generate the coefficient of the noise polynomial by subtracting the Hamming weight of the most significant η-bit of r_i from the Hamming weight of the least significant η-bit of r_i. Finally, the aforementioned procedure is iterated d times to generate the polynomial vector $\mathbf{e} = \{e_1, \cdots, e_d\}^T \in \beta_\eta^d$.

6.6 Encoding of Messages.

An initialized message is a 3-tuple $\langle \mathcal{C}, \mathbf{m}, \rho \rangle$, where $\mathcal{C}$ denotes the client ID with $256/8 = 32$ bytes, $\mathbf{m}$ is a vector of d polynomials with 256 13-bit coefficients each, and ρ is a 32-byte seed. We compress the polynomial by the little-endian format to $(256 \times 13)/8 = 416$ bytes. Eventually, we obtain the initialized message of $(416d + 64)$ bytes by concatenating $\mathcal{C}$, compressed polynomials and the seed ρ. The response message is also a 3-tuple $\langle \mathbf{y}_s, v, k \rangle$, where $\mathbf{y}_s$ is a vector of d polynomials with 256 13-bit coefficients as well, the hint signal v is a polynomial with 256 6-bit coefficients each and the verification key k is 32-byte. We compress $\mathbf{y}_s$ by the little-endian format to $((13 \times 256)/8) \times d = 416d$ bytes, then concatenate the encoded hint signal $\mathbf{v}$ of $(256 \times 6)/8 = 192$ bytes followed the little-endian format and the verification key k, in a total of $(416d + 224)$ bytes.

7 Performance and Potential Applications

The benchmarked implementations are written in C. We compiled them with `gcc-9.3.0` with optimization flags `-O3 -formit-frame-pointer-march = native -fPIC`. Our implementations are executed on a 3.60GHz Intel(R) Core(TM) i7-4790 CPU and 4GB RAM computer with 64-bit system.

7.1 Comparison

Here we compare the proposed MLWE-PAKE scheme with several other competitive schemes. The comparative summary is presented in Table 2 and the performance difference is shown in Table 3.

Table 2. Properties comparison of lattice-based candidate schemes.

Scheme	Assumption	Auth. Material[a]	PQ. Security[b]	Mul.[c]	RO[d]	Flows	Error Rec.[e]
NewHope512 [1]	RLWE	×	101	NTT	-	2	D_4 [1]
NewHope1024 [1]	RLWE	×	233	NTT	-	2	D_4 [1]
Frodo [7]	LWE	×	78	FFT	-	2	Peikert [25]
Kyber-AKE-2 [8]	MLWE	Static keys	102	NTT	✓	2	-
Kyber-AKE-3 [8]	MLWE	Static keys	161	NTT	✓	2	-
Ding-PAK [10]	RLWE	Passwords	76	FFT	✓	3	Ding [11]
Ding-PPK [10]	RLWE	Passwords	76	FFT	✓	2	Ding [11]
Gao-PAK [13]	RLWE	Passwords	82	NTT	✓	3	Ding [11]
Gao-PPK [13]	RLWE	Passwords	82	NTT	✓	2	Ding [11]
Yang-PAK [30]	RLWE	Passwords	206	NTT	✓	3	AKCN [17]
Liu-3PAK [21]	RLWE	Passwords	84	NTT	✓	-	Peikert [25]
Our-LightWeight-PAK	MLWE	Passwords	116	NTT	✓	3	OKCN [17]
Our-Recommended-PAK	MLWE	Passwords	177	NTT	✓	3	OKCN [17]
Our-Paranoid-PAK	MLWE	Passwords	239	NTT	✓	3	OKCN [17]

[a] Auth. Material denotes authentication materials.
[b] PQ. Security denotes the post-quantum security level.
[c] Mul. denotes the algorithm of polynomial multiplication.
[d] RO denotes whether this protocol is constructed in the random oracle model.
[e] Error Rec. denotes the error reconciliation mechanism.

Table 2 summarizes protocols [1, 7, 8, 10, 13, 21, 30] and our proposals from different aspects. The majority of these lattice-based protocols are built on RLWE or MLWE, as LWE holds the unattractive feature of low performance and expensive communication cost due to the large matrix. Furthermore, to illustrate the post-quantum security level attained by various schemes, we include an estimation of core-SVP hardness using the approach described in [1]. The results show that NewHope1024, Kyber-AKE-3, Yang-PAK, Our-Recommended-PAK and Our-Paranoid-PAK achieve 128-bit post-quantum security.

Table 3 provides a comparison of our scheme with state-of-the-art lattice-based protocols [1, 7, 8, 10, 30] in the running time, the communication cost, and the failure rate. It is visible that the overall performance of our implementation outperforms other PAKE protocols. Compared with the latest Yang-PAK [30], Our-Recommended-PAK reduces the communication overhead and running time by 36.8% and 13.8%, respectively. The performance of Our-Lightweight-PAK is close to that of NewHope512, but NewHope512 does not achieve authentication. As a MLWE-based AKE scheme, the running time of Our-Recommended-PAK is 43.8% of Kyber-AKE-3's, and the communication cost is 60.7% of Kyber-AKE-3's.

7.2 Potential Applications

Resource-constrained IoT Mobile Devices. At present, a variety of multi-factor authentication schemes [28, 20] augment passwords with fingerprint or iris recognition in order to ensure the security of IoT devices. However, with the advent of quantum computing, conventional schemes based on the intractability of the integer factorization problem and the discrete logarithm problem will be insecure. As shown in Table 3, our Lightweight-PAK scheme with the advantage of lower communication cost and higher performance is more effectively that can be used to connect resource-constrained IoT devices to resource-rich servers.

Table 3. Performance comparison of lattice-based candidate schemes (us)

Scheme	Running Time (us)				Message Size ($bytes$)			Failure
	C_{init}	$S_{resp} + S_{fin}$	C_{fin}	Total	*Client*	*Server*	Total	Rate
NewHope512 [1]	46.089	55.630	60.639	162.358	928	960	1 888	$2^{-55.0}$
NewHope1024 [1]	77.778	108.325	122.772	308.875	1 824	2 048	3 872	$2^{-61.4}$
Frodo [7]	790.013	870.856	107.419	1 768.288	11 296	11 288	22 584	$2^{-38.9}$
Kyber-AKE-2 [8]	81.920	150.490	115.200	347.610	1 568	1 664	3 232	2^{-145}
Kyber-AKE-3 [8]	134.362	268.716	173.876	576.954	2 272	2 368	4 640	2^{-142}
Ding-PAK [10]	2 643.838	2 884.243	337.413	6 702.656	4 136	4 256	8 392	2^{-1023}
Gao-PAK [13]	-	-	-	-	3 904	4 000	7 904	2^{-1023}
Yang-PAK [30]	84.172	144.290	64.859	293.321	1 864	2 592	4 456	2^{-41}
Our-Lightweight-PAK	55.202	93.353	34.558	183.113	928	1 056	1 984	$2^{-53.4}$
Our-Recommended-PAK	79.141	126.048	47.565	252.754	1 344	1 472	2 816	$2^{-97.4}$
Our-Paranoid-PAK	113.891	169.524	61.082	344.497	1 760	1 888	3 648	$2^{-131.6}$

[a] C_{init} denotes the running time of the client initiation.
[b] $S_{resp} + S_{fin}$ denotes the running time of the server response and the server finish.
[c] C_{fin} denotes the running time of the client finish.

Classical/quantum-safe TLS. We consider that TLS should be combined with post-quantum cryptographic primitives since TLS has taken over more than half of web traffic and has been widely deployed with applications such as HTTPS, IMAPS, SMTPS in the real world. As a self-contained system, our MLWE-PAK is designed without additional primitives, which are typically prohibitively expensive in specific applications such as public-key encryption, signatures, or message authentication code. In Table 3, although our Lightweight-PAK has 116-bit post-quantum security, it has 128-bit security against the classical attacks according to the approach from [1]. Therefore, our Lightweight-PAK can be deployed in current TLS to resist classical attacks; our Recommended-PAK and Paranoid-PAK with higher security can be deployed in TLS to resist quantum attacks.

8 Conclusion

The advancement of quantum computing has sparked widely interest in the research of post-quantum cryptography. This paper designed an efficient post-quantum MLWE-based PAKE protocol whose security is demonstrated under the BPR framework. Moreover, benefiting from the flexibility of MLWE, we provided 3 parameter sets and discussed corresponding potential applications (e.g., classic/post-quantum TLS, resource-constrained IoT devices) in real life. Compared with the latest Yang-PAK [30], Our-Recommended-PAK reduces the communication overhead and running time by 36.8% and 13.8%, respectively.

One of our follow-up works will be aimed at integrating our MLWE-PAKE protocol into TLS. In addition, optimizing the implementation of our MLWE-PAKE protocol on embedded microcontrollers is especially useful for IoT applications once the quantum era comes.

References

1. Alkim, E., Ducas, L., Pöppelmann, T., Schwabe, P.: Post-quantum key exchange – a new hope. In: 25th USENIX Security Symposium (USENIX Security 16). pp. 327–343 (2016), https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/alkim
2. Bellare, M., Pointcheval, D., Rogaway, P.: Authenticated key exchange secure against dictionary attacks. In: International conference on the theory and applications of cryptographic techniques. pp. 139–155. Springer (2000), https://doi.org/10.1007/3-540-45539-6_11
3. Bellovin, S.M., Merritt, M.: Encrypted key exchange: Password-based protocols secure against dictionary attacks. In: Proceedings of the 1992 IEEE Symposium on Security and Privacy. p. 72 (1992), https://doi.org/10.1109/RISP.1992.213269
4. Bellovin, S.M., Merritt, M.: Augmented encrypted key exchange: A password-based protocol secure against dictionary attacks and password file compromise. In: Proceedings of the 1st ACM Conference on Computer and Communications Security. pp. 244–250 (1993), https://doi.org/10.1145/168588.168618
5. Benhamouda, F., Pointcheval, D.: Verifier-based password-authenticated key exchange: New models and constructions. IACR Cryptol. ePrint Arch. **2013**, 833 (2013), https://eprint.iacr.org/2013/833.pdf
6. Bernstein, D.J., Schwabe, P., Assche, G.: Tweetable fips 202, 2015 (2015), http://keccak.noekeon.org/tweetfips202.html
7. Bos, J., Costello, C., Ducas, L., Mironov, I., Naehrig, M., Nikolaenko, V., Raghunathan, A., Stebila, D.: Frodo: Take off the ring! practical, quantum-secure key exchange from LWE. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. pp. 1006–1018 (2016), https://dl.acm.org/doi/abs/10.1145/2976749.2978425
8. Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Schwabe, P., Seiler, G., Stehlé, D.: CRYSTALS-Kyber: a CCA-secure module-lattice-based KEM. In: 2018 IEEE European Symposium on Security and Privacy (EuroS&P). pp. 353–367. IEEE (2018), https://ieeexplore.ieee.org/abstract/document/8406610
9. Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (Leveled) fully homomorphic encryption without bootstrapping. ACM Transactions on Computation Theory (TOCT) **6**(3), 1–36 (2014), https://dl.acm.org/doi/abs/10.1145/2633600
10. Ding, J., Alsayigh, S., Lancrenon, J., Saraswathy, R., Snook, M.: Provably secure password authenticated key exchange based on rlwe for the post-quantum world. In: Cryptographers Track at the RSA Conference. pp. 183–204. Springer (2017), https://link.springer.com/content/pdf/10.1007/978-3-319-52153-4_11.pdf
11. Ding, J., Xie, X., Lin, X.: A simple provably secure key exchange scheme based on the learning with errors problem. IACR Cryptol. ePrint Arch. **2012**, 688 (2012), https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.638.6793&rep=rep1&type=pdf
12. DAnvers, J.P., Karmakar, A., Roy, S.S., Vercauteren, F.: Saber: Module-LWR based key exchange, CPA-secure encryption and CCA-secure kem. In: International Conference on Cryptology in Africa. pp. 282–305. Springer (2018), https://doi.org/10.1007/978-3-319-89339-6_16
13. Gao, X., Ding, J., Li, L., Saraswathy, R., Liu, J.: Efficient implementation of password-based authenticated key exchange from rlwe and post-quantum tls. IACR

Cryptol. ePrint Arch. **2017**, 1192 (2017), https://eprint.iacr.org/2017/1192.pdf
14. Gennaro, R., Lindell, Y.: A framework for password-based authenticated key exchange. In: International Conference on the Theory and Applications of Cryptographic Techniques. pp. 524–543. Springer (2003), https://doi.org/10.1007/3-540-39200-9_33
15. Groce, A., Katz, J.: A new framework for efficient password-based authenticated key exchange. In: Proceedings of the 17th ACM conference on Computer and communications security. pp. 516–525 (2010), https://dl.acm.org/doi/abs/10.1145/1866307.1866365
16. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing. pp. 212–219 (1996), https://dl.acm.org/doi/pdf/10.1145/237814.237866
17. Jin, Z., Zhao, Y.: Optimal key consensus in presence of noise. arXiv preprint arXiv:1611.06150 (2016), https://arxiv.org/abs/1611.06150
18. Katz, J., Ostrovsky, R., Yung, M.: Efficient password-authenticated key exchange using human-memorable passwords. In: International Conference on the Theory and Applications of Cryptographic Techniques. pp. 475–494. Springer (2001), https://doi.org/10.1007/3-540-44987-6_29
19. Katz, J., Vaikuntanathan, V.: Round-optimal password-based authenticated key exchange. In: Theory of Cryptography Conference. pp. 293–310. Springer (2011), https://doi.org/10.1007/978-3-642-19571-6_18
20. Li, Z., Wang, D.: Achieving one-round password-based authenticated key exchange over lattices. IEEE Transactions on Services Computing (2019), https://ieeexplore.ieee.org/abstract/document/8826379
21. Liu, C., Zheng, Z., Jia, K., You, Q.: Provably secure three-party password-based authenticated key exchange from rlwe. In: International Conference on Information Security Practice and Experience. pp. 56–72. Springer (2019), https://doi.org/10.1007/978-3-030-34339-2_4
22. MacKenzie, P.: The PAK suite: Protocols for password-authenticated key exchange. In: IEEE P1363. 2. Citeseer (2002), https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.20.5299
23. NIST post-quantum cryptography round 3 submissions.: https://csrc.nist.gov/projects/post-quantum-cryptography/round-3-submissions, updated on December 23, 2020
24. NSA: NSA suite B cryptography, https://www.nsa.gov/ia/programs/suiteb\textunderscorecryptography/, updated on August 19, 2015.
25. Peikert, C.: Lattice cryptography for the internet. In: international workshop on post-quantum cryptography. pp. 197–219. Springer (2014), https://doi.org/10.1007/978-3-319-11659-4_12
26. Shirvanian, M., Saxena, N., Jarecki, S., Krawczyk, H.: Building and studying a password store that perfectly hides passwords from itself. IEEE Transactions on Dependable and Secure Computing **16**(5), 770–782 (2019), https://ieeexplore.ieee.org/abstract/document/8667308
27. Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings 35th Annual Symposium on Foundations of Computer Science. pp. 124–134 (1994), https://ieeexplore.ieee.org/abstract/document/365700
28. Srinivas, J., Das, A.K., Wazid, M., Kumar, N.: Anonymous lightweight chaotic map-based authenticated key agreement protocol for industrial internet of things. IEEE Transactions on Dependable and Secure Computing **17**(6), 1133–1146 (2018), https://ieeexplore.ieee.org/abstract/document/8413130

29. Wang, D., Cheng, H., Wang, P., Huang, X., Jian, G.: Zipfs law in passwords. IEEE Transactions on Information Forensics and Security **12**(11), 2776–2791 (2017), https://ieeexplore.ieee.org/abstract/document/7961213
30. Yang, Y., Gu, X., Wang, B., Xu, T.: Efficient password-authenticated key exchange from rlwe based on asymmetric key consensus. In: International Conference on Information Security and Cryptology. pp. 31–49. Springer (2019), https://doi.org/10.1007/978-3-030-42921-8_2
31. Zhang, Y., Xu, C., Li, H., Yang, K., Cheng, N., Shen, X.S.: Protect: efficient password-based threshold single-sign-on authentication for mobile users against perpetual leakage. IEEE Transactions on Mobile Computing (2020), https://ieeexplore.ieee.org/abstract/document/9007394
32. Zhang, Z., Yang, K., Hu, X., Wang, Y.: Practical anonymous password authentication and tls with anonymous client authentication. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. pp. 1179–1191 (2016), https://dl.acm.org/doi/abs/10.1145/2976749.2978354