Achieving One-Round Password-Based Authenticated Key Exchange over Lattices

Zengpeng Li and Ding Wang, *Member, IEEE*

Abstract—*Password-based authenticated key exchange* (PAKE) protocol, a widely used authentication mechanism to realize secure communication, allows protocol participants to establish a high-entropy session key by pre-sharing a low-entropy password. An open challenge in PAKE is how to design a quantum-resistant round-optimal PAKE. To solve this challenge, lattice-based cryptography is a promising candidate for post-quantum cryptography. In addition, Katz and Vaikuntanathan (ASIACRYPT'09) design the first *three-round* PAKE protocol by leveraging the smooth projective hash function (SPHF) over lattices. Subsequently, Zhang and Yu (AISACRYPT'17) optimized Katz-Vaikuntanathan's approximate SPHF via a splittable public key encryption. They then constructed a *two-round* PAKE by using the simulation-sound non-interactive zero-knowledge (NIZK) proofs, but how to construct a lattice-based simulation-sound NIZK remains an open research question. In other words, how to design a one-round PAKE via an efficient lattice-based SPHF still remains a challenge. In this work, we attempt to fill this gap by proposing a lattice-based SPHF with adaptive smoothness. We then obtain a *one-round* PAKE protocol over lattices with rigorous security analysis by integrating the proposed SPHF into the one-round framework proposed by Katz and Vaikuntananthan (TCC'11). Furthermore, we explore the possibilities of achieving two-round PAKE and universal composable (UC) security from our SPHF, and show the potential application of our PAKE in Internet of Things (IoTs) where communication cost is the main consideration.

Index Terms—Password-based authenticated key exchange, smooth projective hash function, lattice-based cryptography

1 Introduction

Password-based authentication still constitutes the most widespread method of authentication [1], [2], especially on the Internet today, e.g., [3], [4]. Password-based authenticated key exchange (PAKE) protocol is an important cryptographic primitive that enables two players (e.g., a client and a server) to generate a high entropy session key by pre-sharing a common, low-entropy password. Then, the players could utilize the established session key to protect communications over an insecure network (see Fig. 1). Building on earlier literatures such as the EKE (a.k.a., encrypted key exchange) scheme [5] and the Bellare-Pointcheval-Rogaway (BPR) model [6], the classical Kata-Ostrorsky-Yung (in short KOY) framework [7] is proposed under the decisional Diffie-Hellman (DDH) assumption. Subsequently, Gennaro and Lindell (hereafter GL scheme) [8] generalized the KOY scheme by introducing the smooth projective hash function (SPHF) in the BPR security model. Since then, considerable attention [9], [10], [11], [12] has been devoted to the development of round-optimal PAKE protocols via efficient SPHFs.

The concept of SPHF was first denoted by Cramer and Shoup [13]. The authors used SPHF to gain the first encryption scheme that is indistinguishable against (adaptive) chosen ciphertext attacks (or IND-CCA2) under the DDH assumption in the standard model. A number of DDH-based or lattice-based SPHFs have been successively proposed (e.g., [9], [14], [15]). With these SPHFs as the building block, constant-round PAKE protocols (e.g., [9], [10], [12], [15]) can be obtained. However, most of these PAKE protocols are built on DDH-based SPHFs in the group (or pair) setting.

According to our investigation, most of the existing PAKE protocols (e.g., [9], [16], [17], [18]) under the GL framework [8] need at least three rounds, and they leverage IND-CCA2 encryption schemes to establish a high-entropy session key. However, how to reduce communication rounds and relax the security requirement of the underlying primitives remain two research challenges that have yet to be resolved. Notably, Jiang and Gong [16] relaxed the security of GL framework by using the combination of an IND-CPA (indistinguishable against chosen plaintext attacks) scheme at the user-side and an IND-CCA2 scheme at the server-side. However, their PAKE protocol still needs three rounds of communication. In 2015, Abdalla et al. [19] improved the GL framework and obtained a two-round PAKE protocol under the DDH-based SPHF, where the client requires an IND-CPA-secure scheme and the server requires an indistinguishable against plaintext checkable attack (IND-PCA) resistant scheme. Note that an IND-CCA2-secure scheme is equivalent to an IND-PCA-secure scheme. However, most of the aforementioned PAKE protocols (e.g., [16], [17], [18]) under the DDH assumptions are insecure in the coming quantum era.

- *Z. Li is with the College of Computer Sciences and Technology, Qingdao University, Qingdao 266071, P.R. China, and also with the School of Computing and Communications, Lancaster University, Lancaster LA1 4WA, United Kingdom. E-mail: zengpengli@hotmail.com.*
- *D. Wang is with the School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China, and also with State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China. E-mail: wangdingg@pku.edu.cn.*

Manuscript received 24 Oct. 2018; revised 20 June 2019; accepted 23 Aug. 2019. Date of publication 6 Sept. 2019; date of current version 4 Feb. 2022.
(Corresponding author: Ding Wang.)
Digital Object Identifier no. 10.1109/TSC.2019.2939836

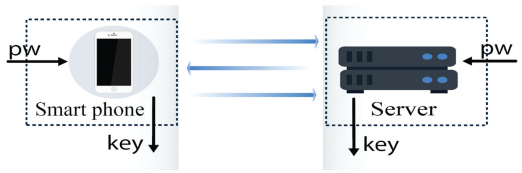

Fig. 1. Password-based Authentication key exchange.

Motivations. In short, our main motivation is to design a quantum-resistant round-optimal PAKE protocol. With advances in quantum computing, quantum attacks against conventional cryptographic primitives (e.g., DDH-based ones) are becoming a reality. Not surprisingly, post-quantum cryptographic primitives are receiving increased attention. Especially, the cryptographic primitive over lattices is one popular line of research. In lattice-based cryptography, the worst-case hardness of lattice assumptions (e.g., the Short Integer Solution (SIS) and the Learning with Errors (LWE) [25], [26]) have been demonstrated to be a great success for attribute-based encryption (ABE), functional encryption (FE) and fully homomorphic encryption (FHE) [27], [28].

Accordingly, many traditional cryptographic primitives have been re-constructed depending on the assumptions over lattices, including lattice-based SPHF schemes and PAKE protocols. In addition, there are only three lattice-based SPHF constructions under the learning with errors (or LWE) assumptions (i.e., [9], [14], [15]). A comparative summary of lattice-based SPHFs is presented in Table 1. Apparently, as shown in Table 1, it still remains an open research question as to:

> *Is there any possibility for us to design a* one-round PAKE *protocol via an efficient* SPHF *over lattices?*

1.1 Our Contributions and Techniques

We answer the above question in the affirmative. More specifically, we optimize the SPHF scheme based on IND-CCA-secure Micciancio and Peikert [22] scheme by leveraging the label and the deterministic rounding function respectively, and then construct a one-round PAKE protocol over our new optimized SPHF scheme by adopting the general one-round PAKE framework of Katz and Vaikuntanathan [23]. In summary, our contributions are four-fold:

1) *Neat LWE-based approximate* SPHF. Inspired by SPHF of Benhamouda et al. [15] for the hash proof system, we construct a word-independent approximate SPHF via the Micciancio-Peikert trapdoor [22], in accordance with the principle of Katz and Vaikuntanathan [9]. As our construction focuses on the practicality and flexibility of one-round lattice-based PAKE protocols, we avoid the underpinning "error correcting code" (or ECC) [14]. Instead, we adopt a simple deterministic rounding function to obtain the established session key. Although the rounding function of Benhamouda et al. [15] is more accurate in rounding the hash value, it is more complex than the deterministic rounding function [9] we use.
2) *One-round PAKE over lattices*. With our new lattice-based approximate SPHF, we design an efficient one-round PAKE protocol over lattices in the standard model, and provide a formal security analysis. Notably, Benhamouda et al. [29] followed Katz-Vaikuntananthan's framework [23] and proposed a one-round PAKE protocol via trapdoor-SPHF, whose security relies on the DDH assumption. Zhang and Yu [14] proposed a two-round PAKE over lattices, and its main drawback is that the IND-CCA-secure encryption scheme depends on simulation-sound non-interactive zero-knowledge (NIZK) proofs [21]. So far, how to design the lattice-based simulation-sound NIZK in standard setting is still an open question. At PKC'18, Benhamouda et al. [15] demonstrated how one can obtain the one-round PAKE. However, how to instantiate the protocol over lattices is an open question. Thus, we explore the potential of SPHF over lattices in this paper to construct a concrete one-round PAKE over lattices, thereby realizing the idea of Benhamouda et al. [15].
3) *New security bound*. We employ Zipf's law in passwords [30], [31] to quantify the advantage Adv of the adversary (in short, Adv is representing the advantage). In other words, we assume an attacker who is with the advantage $\mathsf{Adv} = C' \cdot Q(\lambda)^{s'} + \mathrm{negl}(\lambda)$ can make at most $Q(\lambda)$ on-line guesses for the system security parameter λ and the password space $\mathcal{D}$. This new security bound is $3 \sim 4$ order of magnitude more accurate than the conventional uniformly-random security bound (i.e., $\mathsf{Adv} = Q(\lambda)/|\mathcal{D}| + \mathrm{negl}(\lambda)$) that has been used in the security proofs of most existing protocols (e.g., [14], [15], [32], [33], [34]). We then demonstrate the effectiveness of our approach by evaluating a large-scale real-world dataset, comprising 524.65 million mail.163.com" passwords. Notably, the '163 Mail' is the largest mail provider in China.
4) *Some potential applications*. We show that our SPHF can be used to design two-round and three-round PAKE protocols over lattices in the standard model. Further, one can probably extend our one-round PAKE to achieve universally composable security

TABLE 1
A Comparative Summary of Lattice-Based SPHF Schemes†

Scheme	Building blocks	SPHF models	Adaptive smooth	Error-correction code	Round
Katz-Vaikuntanathan [9]	Peikert [20]	Gennaro-Lindell [8]	×	✓	3
Zhang-Yu [14]	Peikert [20]+NIZK [21]	Gennaro-Lindell [8]	×	✓	2
Benhamouda et al. [15]	Micciancio-Peikert [22]	Katz-Vaikuntanathan [23]	✓	×	?*

– †*The scheme in [9] builds on the work of [20], [24], while the scheme in [14] builds on the scheme of [9].*
– *NIZK: Non-Interactive Zero-Knowledge; SPHF: Smooth Projective Hash Function.*
– *The symbol * implies there is no concrete PAKE construction, but it provides an interesting point to construct one-round PAKE over lattices.*

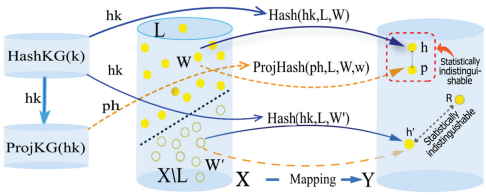

Fig. 2. Smooth projective hash function.

(UC) according to [17], [23], but it is beyond the scope of this paper. We also explore the applications of our PAKE in Internet of Things (IoTs) inspired by various applications [35].

2 Related Work

Before describing our constructions, we will briefly review the literatures of PAKE and SPHF.

2.1 PAKE

Three-Round PAKE. Katz, Ostrovsky, and Yung [7] proposed the first three-round PAKE under DDH assumption without SPHF. Two years later in 2003, Gennaro and Lindell [8] extended the Katz-Ostrovsky-Yung scheme using SPHF on a labeled IND-CCA-secure PKE scheme. In the following year, Jiang and Gong [16] relaxed the security of Gennaro-Lindell framework using random oracles. In this setting, an IND-CPA-secure scheme can satisfy the requirements of the client, but the server still requires an IND-CCA2-secure scheme to preserve the privacy. However, these schemes were only achieved in the standard model (under the stand-alone setting). Groce and Katz [18] extended the Jiang-Gong scheme [16] in the universal composability (UC) framework [17], [36] and proved it secure. Along similar line, Katz and Vaikuntanathan [9] proposed the first three-round PAKE over lattices in ASIACRYPT'09. However, the proposed PAKE requires three rounds.

Two-Round PAKE. Reducing the number of communication rounds and relaxing the security assumptions are two ongoing research focus in the PAKE literature. Over the years, there were only two successful attempts to reduce the communication rounds (or flows) from three to two. The first scheme, proposed by Abdalla et al. [19], introduced a new cryptographic primitive IND-PCA-secure PKE scheme with an associated SPHF to satisfy the client requirements of the two-round PAKE. They also adopted the IND-CPA-secure scheme with an associated SPHF to meet the requirements of the server. In this setting, the stronger assumption (e.g., IND-CCA2-secure scheme along with an associated SPHF) is no longer required by the server. The second scheme, introduced by Zhang and Yu [14], instantiated the first two-round PAKE over lattices using the splittable public key encryption (or PKE) scheme with the associated non-adaptive approximate SPHF. Unfortunately, their PAKE depends on simulation-sound non-interactive zero-knowledge (NIZK) proofs and how to construct a lattice-based simulation-sound NIZK remains an open question.

One-Round PAKE. The first one-round PAKE framework based on conventional DDH assumption was designed by Katz and Vaikuntanathan [23], where the client and the server are required to send the message to each other simultaneously. The authors then proved the security of the protocol in the standard model and in the UC framework separately. Benhamouda et al. [29] followed the framework of [23] and designed an efficient *one-round* PAKE via trapdoor-SPHF on a Cramer-Shoup ciphertext, but their scheme was still based on conventional DDH assumption. Currently, there is no known concrete one-round PAKE protocol construction over lattices in the literature.

2.2 SPHF

As illustrated in Fig. 2, SPHF is defined on the NP language L over a domain X. At the time of research, there is no efficient adversary that can distinguish between an element (or a point) x in NP language L (i.e., $x \in L$) and an element (or a point) $x \in X \backslash L$ for domain X. Furthermore, SPHFs contain two keyed functions (i.e., $\mathsf{Hash}(\cdot)$ and $\mathsf{ProjHash}(\cdot)$). Concretely, function $\mathsf{Hash}(\cdot)$ can be computed by taking as input the hashing key hk, and the function $\mathsf{ProjHash}(\cdot)$ can be computed by taking as input the projective hashing key ph. The output values of both functions are the same (i.e., statistically indistinguishable) for the word W over the language L (i.e., $W \in L$). In other words, the output value satisfies $\mathsf{Hash}(hk, W) = \mathsf{ProjHash}(ph, W, w)$, where w is the witness and the word W contains the labeled IND-CCA ciphertext $\mathbf{c}$ and the message msg. Thus, even when given ph, the adversary cannot guess $\mathsf{Hash}(hk, W \in L)$. The formal definition is presented in Section 3.

Word-Independent CS-SPHF *with Non-Adaptive Soundness.* SPHF was first proposed by Cramer and Shoup [13] (CS-SPHF) to facilitate the design of PAKE protocol. The syntax of CS-SPHF is as follows:

$$\begin{aligned}\{&hk \leftarrow \mathsf{HashKG}(L); ph \leftarrow \mathsf{ProjKG}(hk, pk, \perp);\\ &h \leftarrow \mathsf{Hash}(hk, W); p \leftarrow \mathsf{ProjHash}(ph, W, w)\}.\end{aligned} \tag{2.1}$$

Word-Dependent GL-SPHF *with Non-Adaptive Soundness.* Gennaro and Lindell [8] optimized CS-SPHF, but made the projective key ph dependent on the word W. We abbreviate it to GL-SPHF, whose syntax is as follows:

$$\begin{aligned}\{&hk \leftarrow \mathsf{HashKG}(L); ph \leftarrow \mathsf{ProjKG}(hk, pk, W);\\ &h \leftarrow \mathsf{Hash}(hk, W); p \leftarrow \mathsf{ProjHash}(ph, W, w)\}.\end{aligned} \tag{2.2}$$

Both CS-SPHF and GL-SPHF achieve non-adaptive smoothness, namely that W is independent of the project key ph and the adversary cannot see the projective key ph before choosing the word W. Namely that even if we can see ph, we cannot change W.

Word-Independent KV-SPHF *with Adaptive Soundness.* Katz and Vaikuntanathan [9] achieved adaptive smoothness and made the projective key ph independent on W, where adaptive smoothness implies that we can choose W after having seen the projective key ph. For convenience, we abbreviate it to KV-SPHF, whose syntax is as follows:

$$\begin{aligned}\{&hk \leftarrow \mathsf{HashKG}(L); ph \leftarrow \mathsf{ProjKG}(hk, pk, \perp);\\ &h \leftarrow \mathsf{Hash}(hk, f(W)); p \leftarrow \mathsf{ProjHash}(ph, W, w)\}.\end{aligned} \tag{2.3}$$

We say f is adaptive if $\mathcal{A}$ sees ph before choosing W.

TABLE 2
Parameters and Abbreviations

Notations	Descriptions
λ	Security parameter
$Q(\lambda)$	$Q(\lambda)$ on-line guesses
$\mathcal{D}$	The password space
$\mathrm{negl}(\lambda)$	A negligible function in λ
s' and C'	Parameters of the Zipf distribution
$\mathcal{A}$	The adversary
Adv	The advantage of the adversary
Π	The proposed one-round PAKE protocol
PPT	Probabilistic polynomial time
IND-CPA	Indistinguishable against chosen plaintext attacks
IND-CCA1	Indistinguishable against (Non-adaptive) chosen ciphertext attacks
IND-CCA2	Indistinguishable against (Adaptive) chosen ciphertext attacks
IND-PCA	Indistinguishable against plaintext checkable attack

– For convenience, in the following discussion, unless stated otherwise, all the IND-CCA *are represented* IND-CCA2.

2.3 IND-CCA-Secure Scheme over Lattices

There are two paradigms of IND-CCA2-secure encryption that are built on IND-CPA-secure encryption:

- Dolev-Dwork-Naro paradigm [37]: utilizing one-time signature and one-time NIZK (OT-NIZK).
- Naro-Yung/Sahai paradigm ([21], [38]): utilizing one-time simulation-sound NIZK (OT-SS-NIZK).

Furthermore, there are a number of variants of the Cramer-Shoup scheme [13] designed to achieve IND-CCA2-secure encryption, by using the hash proof system, such as the schemes of Kiltz et al. [39] and of Kurosawa-Desmedt [40].

However, only three straight constructions of the IND-CCA-secure PKE over lattices have been proposed in the literature. Concretely, Peikert and Waters [41] proposed the first IND-CCA1-secure PKE scheme under the (worst-case) lattice assumption, along with some optimization [20]. However, but the schemes are based on lossy trapdoor functions. In a separate work, Katz and Vaikuntanathan [9] (KV) used Regev scheme [25] as the building block and designed the IND-CCA1-secure PKE scheme. However, the decryption algorithm of Katz-Vaikuntanathan scheme needs to invoke the $\mathsf{BBDSolve}(\cdot)$ procedure multiple times. Subsequently, Micciancio and Peikert [22] proposed a new efficient IND-$CCA1$-secure PKE scheme, by introducing the **G**-trapdoor function and tweaking the decryption algorithm. In other words, plaintexts are recovered via querying the trapdoor inversion procedure $\mathsf{Invert}(\cdot)$ multiple times.

3 Preliminaries

Below, we first list parameters in Table 2 that will be used in our construction and security analysis.

Furthermore, vectors and matrices are denoted as bold lower-case letter (e.g., $\mathbf{x}$) and bold upper-case letter (e.g., $\mathbf{A}$), respectively. An m-dimension lattice can be denoted as $\Lambda = \{\mathbf{Bs} \,|\, \mathbf{s} \in \mathbb{Z}^n\}$, where $\mathbf{B} \in \mathbb{Z}^{m \times n}$ is referred to as the basis of Λ for the parameter $m \geq n\lceil \log q \rceil$. The determinant of Λ is denoted as $det(\Lambda) = \sqrt{det(\mathbf{B}^T\mathbf{B})}$, q-ary lattices is defined as follows,

$$\Lambda(\mathbf{A}) = \{\mathbf{A} \cdot \mathbf{s} \,|\, \mathbf{s} \in \mathbb{Z}_q^n\} + q\mathbb{Z}^m, \tag{3.1}$$

$$\Lambda^{\perp}(\mathbf{A}) = \{\mathbf{z} \in \mathbb{Z}^m \,|\, \mathbf{z}^T \cdot \mathbf{A} = \mathbf{0} \,(\mathrm{mod}\ q)\}, \tag{3.2}$$

$$\Lambda^{\perp}_{\mathbf{u}}(\mathbf{A}) = \{\mathbf{z} \in \mathbb{Z}^m \,|\, \mathbf{z}^T \cdot \mathbf{A} = \mathbf{u} \,(\mathrm{mod}\ q)\}. \tag{3.3}$$

Note that, $\Lambda(\mathbf{A})$ and $\Lambda^{\perp}(\mathbf{A})$ are dual of each other. $\Lambda^{\perp}_{\mathbf{u}}(\mathbf{A})$ is the coset of $\Lambda^{\perp}(\mathbf{A})$ for a syndrome $\mathbf{u} \in \mathbb{Z}_q^n$.

Definition 3.1 (Hamming Metric). Hamming *distance is a classic representative of several string metrics and can be used to calculate the edit distance between two strings, then we can write* $\mathsf{HD}(x, y)$. *To facilitate, we write* $\mathsf{HD}(x, y)$ *any two strings of equal length* $x, y \in \{0,1\}^v$.

3.1 Lattice Background

Definition 3.2 (Decision-LWE$_{n,q,\chi,m}$, [25]). *We first assume that there exist two different distributions*

- *(1)* $\mathcal{A}_{\mathbf{s},\chi} := \{(\mathbf{A}, \mathbf{b}) : \mathbf{A} \leftarrow \mathbb{Z}_q^{m \times n}, \mathbf{s} \leftarrow \mathbb{Z}_q^{n \times 1}, \mathbf{e} \leftarrow \chi^{m \times 1}, \mathbf{b} = \mathbf{A} \cdot \mathbf{s} + \mathbf{e} \,(\mathrm{mod}\ q)\}$;
- *(2) the uniform distribution (i.e.,* $\{(\mathbf{A}, \mathbf{b}) : \mathbf{A} \leftarrow \mathbb{Z}_q^{m \times n}, \mathbf{b} \leftarrow \mathbb{Z}_q^{m \times 1}\}$*).*

If no one can distinguish an independent sample $(\mathbf{A}, \mathbf{b}) \in \mathbb{Z}_q^{m \times n} \times \mathbb{Z}_q^{m \times 1}$ *with overwhelming probability, where the specific sample is distributed according to either the distribution (1) or the distribution (2), we then can say that the above two distributions are computationally indistinguishable.*

Remark 3.3. Reductions between approximating the shortest vector problem (SVP) in lattices (for appropriate parameters) and the LWE problem have been discussed by Regev and others [20], [25], [41], [42]. The reduction is out of scope of this work, thus the related corollaries of the reductions are omitted at this stage. More details are available in [20], [25], [41], [42].

Below, we state the Lemma 3.4 which is a key result used to show the correctness of our construction. Before describing the Lemma 3.4, we will first introduce the computable gadget matrix $\mathbf{G} \in \mathbb{Z}_q^{m \times N}$ for the parameters $N \geq m\lceil \log q \rceil$ [22]. $\mathbf{G}$ is a fixed structure matrix which is composed of a set of the gadget vector $\mathbf{g} = (2^0, 2^1, 2^2, \cdots, 2^{\ell-1}) \in \mathbb{Z}_q^{\ell}$. Along with the matrix $\mathbf{G}$, there is an efficiently computable "short preimage" function $\mathbf{G}^{-1}(\cdot)$ which is also a deterministic inverse function.

Lemma 3.4 ([22]). *Regarding the parameters* N, q, m *and* m', *there exists an equation* $\mathbf{G}\mathbf{G}^{-1}(\mathbf{M}) = \mathbf{M}$, *if the inverse function* $\mathbf{G}^{-1}(\cdot)$ *inputs takes as input on a matrix* $\mathbf{M} \in \mathbb{Z}_q^{m \times m'}$ *and the output result satisfies* $\mathbf{G}^{-1}(\mathbf{M}) \in \{0,1\}^{N \times m'}$.

Corollary 3.5. *In order to invert the injective trapdoor function* $g_{\mathbf{A}}(\mathbf{s}, \mathbf{e}) = \mathbf{s}^T \cdot \mathbf{A} + \mathbf{e} \,(\mathrm{mod}\ q)$, *a* PPT *algorithm* $\mathsf{Invert}(\cdot)$ *should fulfill the following requirements:*

- *The algorithm takes the following parameters as input:* ***1).*** *a parity-check matrix* $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$; ***2).****. a* **G***-trapdoor* $\mathbf{R} \in \mathbb{Z}^{\bar{m} \times n\ell_q}$, *where* $\mathbf{A} \cdot \begin{pmatrix} \mathbf{R} \\ \mathbf{I} \end{pmatrix} = \mathbf{H} \cdot \mathbf{G}$ *for the invertible tag* $\mathbf{H} \in \mathbb{Z}_q^{n \times n}$ *of* $\mathbf{R}$.*; and* ***3).*** *an* LWE *instance* $\mathbf{b}$ *satisfying* $\mathbf{b} = \mathbf{s}^T \cdot \mathbf{A} + \mathbf{e} \,(\mathrm{mod}\ q)$.
- *The algorithm outputs the secret vector* $\mathbf{s}$ *(with reference to the value of* $\mathbf{b}^T \cdot \begin{pmatrix} \mathbf{R} \\ \mathbf{I} \end{pmatrix}$*.) and the noise vector* $\mathbf{e} = \mathbf{b} - \mathbf{A}^T\mathbf{s}$.

3.2 Smooth Projective Hash Functions

The projective hash function families were first proposed by Cramer and Shoup [13] at EUROCRYPT'02. SPHFs act as an important type of projective hash function that can be used to design various cryptographic protocols. In a nutshell, in the projective hash function families, the adversary is computationally hard to distinguish a random element in language L from a random element in the domain $X \setminus L$ for the existence of a domain X along with $L \subseteq X$, where L is an underlying NP language.

As shown in Fig. 2 that is inspired by [43], an approximate SPHF acts as a special proof system and contains the required elements: a witness w and a word W. W contains the labeled IND-CCA ciphertext $\mathbf{c}$ and message msg. Further, SPHF contains both *approximate correctness* and *smoothness* properties. Approximate SPHF over $L \subseteq X$ is defined by four PPT algorithms.

- $hk \leftarrow \mathsf{HashKG}(L)$. It takes an NP language L as input, and gains a "private" hash key hk as outputs.
- $ph \leftarrow \mathsf{ProjKG}(hk, L, W)$. The algorithm takes L, a hash key hk, and a word $W \in L$ as input and generates a "public" projection hash key ph as output.
- $h \leftarrow \mathsf{Hash}(hk, L, W)$. The algorithm takes an NP language L, a hash key hk, and a word $W \in L$ as input and outputs a hash value h over $\{0,1\}^v$ for some positive integer $v = \Omega(\lambda)$.
- $\mathsf{ProjHash}(ph, L, W, w)$. The algorithm takes an NP language L, a projective hash key ph, a word $W \in L$, and a witness w as input and outputs a projective hash value p over $\{0,1\}^v$.

Meanwhile, approximate SPHF fulfills the following two properties of (approximate) correctness and smoothness:

- *Approximate Correctness:* If a word W is in language L, i.e., $W \in L$ there exists

$$\Pr[\mathsf{HD}\big(\mathsf{ProjHash}(hk, L, W), \mathsf{Hash}(hk, L, W)\big) < \varepsilon \cdot v] \geq 1 - \mathrm{negl}(\lambda), \tag{3.4}$$

  then the approximate correctness (i.e., ε-correct) property holds.
- *Smoothness:* If any word $W \in X \setminus L$, the statistical distance of the following two distributions is in a negligible λ:

$$1).\{(ph, h) \mid hk \leftarrow \mathsf{HashKG}(L), ph = \mathsf{ProjKG}(hk, L, W), \underline{h \leftarrow \mathsf{Hash}(hk, L, W)}\}. \tag{3.5}$$

$$2).\{(ph, h) \mid hk \leftarrow \mathsf{HashKG}(L), ph = \mathsf{ProjKG}(hk, L, W), \underline{h \leftarrow \{0,1\}^v}\}. \tag{3.6}$$

  then the smoothness property holds.

If the approximate SPHF is 0-correct, then we call it SPHF. However, in the lattice setting, we cannot obtain the 0-correct feature.

3.3 PAKE Security Model

Below, we review the PAKE security model by following the definitions of Bellare, Pointcheval, and Rogaway [6].

Participants, Passwords, and Initialization. A fixed set of protocol participants (also known as users) is denoted as U. For every distinct $U_1, U_2 \in \mathsf{U}$, we assume U_1 and U_2 share a password pw_{U_1,U_2}, (i.e., pw). It is assumed that each pw_{U_1,U_2} is independently sampled from the password space $D(\lambda)$ according to Zipf's law [30].

Execution of the Protocol. The protocol in reality is to describe how users behave after receiving commands (or inputs) from the environment. On the contrary, the adversary provides these inputs in the formal model. In this model, the protocol can be executed by each party with different partners multiple times (possibly concurrently). Meanwhile, in this model, each party is allowed to instantiate an unlimited number of instances, and the adversary is allocated oracle access to these different instances. For convenience, the ith instance of the user U is denoted as Π_U^i. In particular, each instance can be used in this model only once. Furthermore, the (local) state that is updated during the course of the experiment is maintained by corresponding each instance. Below, we describe the concrete variables of local state maintained by each user instance Π_U^i:

- sid_U^i, session id.
- pid_U^i, partner id.
- skey_U^i, session key id.
- acc_U^i, a boolean variable implies the final result is accepted at the end of the execution.
- term_U^i, a boolean variable implies the procedure terminates at the end of the execution.

Adversarial Model. The adversary (or malicious party) is assumed to have the ability to fully control all communication in the external network, namely the adversary is able to do whatever (s)he wants, in the sense of the capability to *1).* modify, block, inject, and delete messages; and *2).* request any session keys adaptively. Formally, the adversary launches attacks using oracle queries model in the real world. Thus, to define security, we introduce a set of oracles and use these oracles to explain how the adversary can interact with various instances. The following oracle queries are the case of one-round challenge-response protocol, the adversary first sends the query to the assigned oracle, then the assigned oracle answers the query to the adversary. More formally description are as follows:

- $\mathsf{Send}(U_C, i, M)$. The oracle is first activated by the adversary, then it sends message M to the instance $\Pi_{U_C}^i$. Upon receiving M from Send, $\Pi_{U_C}^i$ then runs in accordance with the protocol specification and updates the state. The output of $\Pi_{U_C}^i$ is sent to the adversary.
- $\mathsf{Execute}(U_C, i, U_S, j)$. The oracle represents the protocol execution between $\Pi_{U_C}^i$ and $\Pi_{U_S}^j$ without any outside interference from the adversary. To answer this query, the oracle outputs the protocol transcript to the adversary, where the transcript is the complete ordered messages, which can be exchanged between the instances.
- $\mathsf{Reveal}(U_C, i)$. The oracle models known key attacks, and the adversary is permitted to learn session keys of the specified instance from previous and concurrent executions, and outputs the session key skey_U^i. In addition, improper session keys are erased.

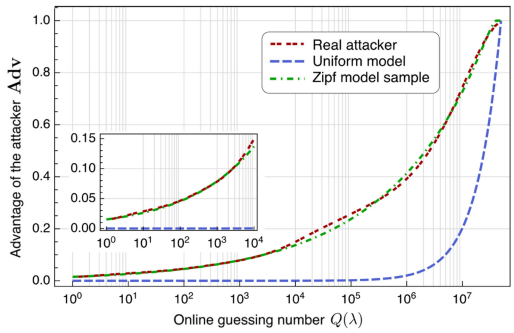

Fig. 3. Guessing advantages of the real attacker, the uniform attacker (e.g., [14], [15], [23], [32]) and our Zipf attacker (using 524.65 million mail.163.com passwords).

– $\mathsf{Test}(U_C, i)$. The adversary is allowed to query the oracle, once and only once, and gains a random bit b. If $b = 1$, then the adversary obtains the session key skey^i_U; otherwise, the adversary is given a uniform session key. Finally, the adversary will guess b' that is a random bit. If the adversary is successful then implies $b = b'$.

Partnering. Let U_C, $U_S \in U$. Instances $\Pi^i_{U_C}$ and $\Pi^j_{U_S}$ are partnered if: (1). $\mathsf{sid}^i_U = \mathsf{sid}^i_{U_C} \neq \mathrm{NULL}$; and (2) $\mathsf{pid}^i_{U_C} = U_C$ and $\mathsf{pid}^j_{U_S} = U_S$.

Correctness. If $\Pi^i_{U_C}$ and $\Pi^j_{U_S}$ are partnered, then there exist $\mathsf{acc}^i_{U_C} = \mathsf{acc}^j_{U_S} = \mathrm{TRUE}$ and $\mathsf{skey}^i_{U_C} = \mathsf{skey}^j_{U_S}$ and both instances can obtain the common session key.

Definition 3.6 (Advantage of the adversary). *We assume that all* PPT *adversaries $\mathcal{A}$ have the ability to make at most $Q(\lambda)$ on-line guessing attacks, and the password dictionary D follows the Zipf-like distribution with parameters $C' = 0.062239$ and $s' = 0.155478$. If the advantage of the adversary holds*

$$\mathsf{Adv}_{\mathcal{A},\Pi}(\lambda) \leq C' \cdot Q^{s'}(\lambda) + \mathrm{negl}(\lambda), \tag{3.7}$$

then the PAKE *protocol Π is said to be secure.*

Remark 3.7. In most existing PAKE literatures (e.g., [9], [14], [15], [32], [35]), it is generally assumed that passwords follow a uniformly random distribution, and these early works formulate the attacker's advantage via $Q(\lambda)/D + \mathrm{negl}(\lambda)$ for the password dictionary with size D. However, recent research [30], [44], [45] has revealed that passwords chosen by users from various languages follow the Zipf's law (but not a uniformly random distribution), and CDF-Zipf model [30] is better than the traditional uniform model to characterize password distributions. Thus, in this paper, we formulate the attacker's adversary via $C' \cdot Q^{s'}(\lambda) + \mathrm{negl}(\lambda)$ with C' and s' being the Zipf parameters. Notably, Fig. 3 shows that the formulation $Q(\lambda)/D + \mathrm{negl}(\lambda)$ ($\forall Q(\lambda) \in [1, |D|]$) in the traditional uniform password distribution model always significantly underestimates the real attacker's Adv. On the contrary, the Zipf based formulation $C' \cdot Q^{s'}(\lambda) + \mathrm{negl}(\lambda)$ well approximates the real attacker's advantage Adv: $\forall Q(\lambda) \in [1, |D|]$. Importantly, we observe that the *largest* deviation between $C' \cdot Q^{s'}(\lambda) + \mathrm{negl}(\lambda)$ and Adv is only 0.491 percent .

4 Our SPHF via Micciancio-Peikert Scheme

Below, we detail the labeled IND-CCA1-secure Micciancio-Peikert scheme [22]. At present, there are only a small number of IND-CCA1-secure PKE schemes [9], [20], [22], [26], [46], [47] over lattices. Compared with the Katz-Vaikuntanathan scheme [9], the main advantage of the Micciancio-Peikert scheme is that it does not require the invoking of the $\mathsf{Invert}(\cdot)$ algorithm multiple times to recover the plaintext during decryption. Furthermore, the IND-CCA1-secure Micciancio-Peikert scheme can be converted to the level of IND-CCA2 security via relatively generic transformations using either a message authentication code (MAC) along with a weak form of commitment [48] or strongly unforgeable one-time signature [37]. For simplicity, we only present the labeled IND-CCA1-secure Micciancio-Peikert scheme in this section, and omit the phase of transformation to obtain the IND-CCA2-secure scheme. The detail is as follows:

4.1 Labeled CCA1 Micciancio-Peikert (MP) Scheme

- $\mathsf{params} \leftarrow \mathsf{MP.Setup}(\lambda, m, \bar{m}, n, q, \ell_q)$: Takes the security parameter λ, the integers m, $\bar{m}$, n, and the model q as input, where $m = \bar{m} + n\ell_q$ and $\ell_q = \lceil \log q \rceil = O(\log q)$; then outputs the parameters $\mathsf{params} := (\lambda, m, \bar{m}, n, q, \ell_q)$.
- $(sk, pk) \leftarrow \mathsf{MP.KeyGen}(\mathsf{params})$:
  1) Takes the params as input and samples a public matrix $\bar{\mathbf{A}} \leftarrow \mathbb{Z}_q^{n \times \bar{m}}$ along with the trapdoor matrix $\mathbf{R} \leftarrow \mathcal{D}^{\bar{m} \times n\ell_q}_{\mathbb{Z}, \omega(\sqrt{\log n})}$ by invoking the trapdoor generation algorithm (i.e., $(\mathbf{P}, \mathbf{T}) \leftarrow \mathsf{TrapGen}(\mathsf{params}, \bar{\mathbf{A}}, \mathbf{R})$). The dot production between the public key $\mathbf{P}$ and the trapdoor $\mathbf{T}$ is 0 (i.e., $\mathbf{P} \cdot \mathbf{T} = 0 \pmod q$).
  2) Let the matrix $\mathbf{A}_1 := -\bar{\mathbf{A}} \cdot \mathbf{R} \pmod q \in \mathbb{Z}_q^{n \times n\ell_q}$ and the public key $\mathbf{P} := [\bar{\mathbf{A}} \,|\, \mathbf{A}_1] = [\bar{\mathbf{A}} \,|\, -\bar{\mathbf{A}} \cdot \mathbf{R}] \in \mathbb{Z}_q^{n \times m}$ is generated. Then, we denote $(n\ell_q \times n\ell_q)$-dimension identity matrix by $\mathbf{I}_{n\ell_q \times n\ell_q}$ and generate the trapdoor (i.e., secret key) $\mathbf{T} := [\mathbf{R} \,|\, \mathbf{I}_{(n\ell_q \times n\ell_q)}]^T$.
  3) Outputs both secret key $sk := \mathbf{T} \in \mathbb{Z}_q^{m \times n\ell_q}$ and public key $pk := \mathbf{P} \in \mathbb{Z}_q^{n \times m}$.
- $c \leftarrow \mathsf{MP.Enc}(pk = \mathbf{P}, \mathbf{m} \in \{0,1\}^{n\ell_q}, \mathsf{label} = u)$:
  1) In order to encrypt the message $\mathbf{m} \in \{0,1\}^{n\ell_q}$, the algorithm first maps $\mathbf{m} \in \{0,1\}^{n\ell_q}$ to $\mathsf{encode}(\mathbf{m}) = \mathbf{S} \cdot \mathbf{m} \in \mathbb{Z}^{n\ell_q}$ for any $\mathbf{S} \in \mathbb{Z}^{n\ell_q \times n\ell_q}$ of lattice Λ, and takes the public key $\mathbf{P}$ as input.
  2) Then, sample a nonzero label $u \leftarrow \mathcal{U}$ and let

$$\begin{aligned} \mathbf{A}_u &= [\bar{\mathbf{A}} \,|\, \mathbf{A}_1 + h(u)\mathbf{G}] \\ &= [\bar{\mathbf{A}} \,|\, h(u)\mathbf{G} - \bar{\mathbf{A}}\mathbf{R}] \in \mathbb{Z}_q^{n \times m}. \end{aligned} \tag{4.1}$$

  We note that $\mathbf{A}_u \cdot \mathbf{T} = h(u)\mathbf{G}$ for the gadget matrix $\mathbf{G} \in \mathbb{Z}_q^{n \times n\ell_q}$.

Remark 4.1. In the ring $\mathcal{U} = \mathbb{Z}_q[x] \setminus (f(x))$ for $q = p^e$, we denote the label set $\mathcal{U} = \{u_1, \cdots, u_\ell\} \subset \mathcal{U}$ with the unit difference property. More concretely, if the difference $u_i - u_j \in \mathcal{U}$ for any $i \neq j$, then the label matrix $h(u_i - u_j) = h(u_i) - h(u_i) \in \mathbb{Z}_q^{n \times n}$ is invertible.

  3) Sample a random vector $\mathbf{s} \leftarrow \mathbb{Z}_q^{n \times 1}$, a noise vector $\bar{\mathbf{e}} \leftarrow \mathcal{D}^{\bar{m}}_{\mathbb{Z}, \alpha q}$, and another noise vector $\hat{\mathbf{e}} \leftarrow \mathcal{D}^{n\ell_q}_{\mathbb{Z}, s}$, where $s^2 = (\|\bar{\mathbf{e}}\|^2 + \bar{m}(\alpha q)^2) \cdot \omega(\sqrt{\log n})^2$. Then, the algorithm cascades $\bar{\mathbf{e}}$ and $\hat{\mathbf{e}}$ together, and obtains the noise vector $\mathbf{e} = (\bar{\mathbf{e}}, \hat{\mathbf{e}})^T \in \mathbb{Z}^{m \times 1}$.

4) Computes and outputs the ciphertext over $\mathbb{Z}_q^{m\times 1}$,

$$\mathbf{c} = \mathbf{A}_u^T \cdot \mathbf{s} + \mathbf{e} + (\mathbf{0} \,|\, \mathsf{encode}(\mathbf{m})) \,(\mathrm{mod}\ q). \tag{4.2}$$

5) Lastly, outputs the ciphertext as follows $c = (u, \mathbf{c}) \in \mathcal{U} \times \mathbb{Z}_q^{m\times 1}$.

- $\mathbf{m} \leftarrow \mathsf{MP.Dec}(sk, c = (u, \mathbf{c}))$: In order to decrypt the ciphertext $c = (u, \mathbf{c})$ for u, the algorithm proceeds with the following steps:
  1) The algorithm first parses c into u and $\mathbf{c}$. If $u = 0$, then outputs empty value; otherwise the algorithm invokes the algorithm $\mathsf{Invert}(\mathbf{A}_u, \mathbf{c}, \mathbf{R}, h(u) \in \mathbb{Z}_q^{n\times n})$ (Lemma 3.5) and obtains $\mathbf{c} = \mathbf{A}_u^T\mathbf{z}_1 + \mathbf{e}_1$, (i.e., obtains secret vector $\mathbf{z}_1 \in \mathbb{Z}_q^{n\times 1}$ and noise vector $\mathbf{e}_1 = (\bar{\mathbf{e}}_1, \ \hat{\mathbf{e}}_1)^T = \mathbf{c} - \mathbf{A}_u^T\mathbf{z}_1 \in \mathbb{Z}^{m\times 1} < (i.e., < \mathbb{Z}^{\bar{m}\times 1} \times \ Z^{n\ell_q\times 1})$). Then, the algorithm checks whether $\|\bar{\mathbf{e}}_1\| \geq \alpha q\sqrt{\bar{m}}$ or $\|\hat{\mathbf{e}}_1\| \geq \alpha q\sqrt{2\bar{m}n\ell_q}\cdot \omega(\sqrt{\log n})$, and the output is assigned empty value.
  2) Otherwise, the decryption algorithm invokes $\mathsf{Invert}(\mathbf{A}_u, \mathbf{c} - (\mathbf{0} \,|\, \mathsf{encode}(\mathbf{m})^T, \mathbf{R}, h(u))$ and obtains $\mathbf{c} - (\mathbf{0} \,|\, \mathsf{encode}(\mathbf{m})) = \mathbf{A}_u^T \cdot \mathbf{z}_2 + \mathbf{e}_2$ for $\|\bar{\mathbf{e}}_2\| < \alpha q\sqrt{\bar{m}}$ and $\|\hat{\mathbf{e}}_2\| < \alpha q\sqrt{2\bar{m}n\ell_q}\cdot\omega(\sqrt{\log n})$. Let $\mathbf{v} = \mathbf{c} - \mathbf{e}_2 = \mathbf{A}_u^T \cdot \mathbf{z}_2 + (\mathbf{0} \,|\, \mathsf{encode}(\mathbf{m}))^T \,(\mathrm{mod}\ q)$, and the algorithm requires $\mathbf{v}$ to parse $\mathbf{v} = (\bar{\mathbf{v}}, \hat{\mathbf{v}}) \in \mathbb{Z}_q^{\bar{m}\times 1} \times \mathbb{Z}_q^{n\ell_q\times 1}$ for $\bar{\mathbf{v}} \in \Lambda(\bar{\mathbf{A}}^T)$.
  3) Finally, computes and outputs the plaintext $\mathsf{decode}\ (\mathbf{v}^T \cdot [\frac{\mathbf{R}}{\mathbf{I}}] \,(\mathrm{mod}\ q)) \in \{0,1\}^{n\cdot\ell_q}$.

The proofs of security and correctness are similar to that of the IND-CCA1 MP scheme. To be self-contained, we now provide the two important properties via the following Lemma 4.2 and Theorem 4.3 respectively.

Lemma 4.2 (Correctness, [22]). *If the correctness property of the above Micciano-Peikert scheme is hold, then the probability of decryption error is only* $2^{-\Omega(\lambda)}$.

Theorem 4.3 (Security, [22]). *The above Micciano-Peikert scheme is labeled* IND-CCA1*-secure under the hardness of decisional* $\mathsf{LWE}_{n,m,q,\chi}$ *assumption.*

The correctness and security analysis can be obtained in a straightforward method following the work of [22]. To save space, the detailed analysis is omitted, and more details are referred to [22].

4.2 Approximate SPHF via MP Scheme

It is known that the Micciancio-Peikert scheme is labeled IND-CCA1-secure under the decisional LWE assumption. Further, if we introduce a one-time signature scheme, then we can use it to sign the ciphertext under the secret key of the signature, resulting in an IND-CCA2-secure scheme.

Below, we use the Micciancio-Peikert scheme to develop an associated SPHF following the Katz-Vaikuntanathan (KV-SPHF) construction and introducing the deterministic rounding function of [9], we present a neat lattice-based SPHF scheme with adaptive soundness based on Micciancio-Peikert scheme. The scheme is also hereafter referred to as MP-SPHF for convenience.

- $hk \leftarrow \mathsf{HashKG}(\mathsf{params})$. The algorithm draws a random vector $\mathbf{k}$ from $\mathbb{Z}_q^{n\times 1}$. Then, it outputs the hashing key $hk := \mathbf{k} \in \mathbb{Z}_q^{m\times 1}$.
- $ph \leftarrow \mathsf{ProjKG}(\mathsf{params}, hk = \mathbf{k}, pk = \mathbf{A}_u)$. The algorithm first fixes the label u at the beginning, then it takes the hashing key $\mathbf{k}$ and the public key $\mathbf{A}_u = [\bar{\mathbf{A}} \,|\, h(u)\mathbf{G} - \bar{\mathbf{A}}\mathbf{R}] \in \mathbb{Z}_q^{n\times m}$ as input. Afterwards, the algorithm gains the projection key $ph := \mathbf{p} = \mathbf{A}_u \cdot \mathbf{k} \in \mathbb{Z}_q^{n\times 1}$. In this setting, to obtain the "approximate correctness", we modify the public key. However, in MP-SPHF scheme, we adopt the public key $\mathbf{P} := [\bar{\mathbf{A}} \,|\, \mathbf{A}_1] = [\bar{\mathbf{A}} \,|\, -\bar{\mathbf{A}} \cdot \mathbf{R}] \in \mathbb{Z}_q^{n\times m}$ in MP scheme.
- $h \leftarrow \mathsf{Hash}(hk = \mathbf{k}, W := (c, \mathbf{m}))$. The smooth hash function executes as follows:
  1) The algorithm inputs on the hashing key $\mathbf{k}$ and the word W. Notably, W includes a ciphertext $c = (\mathsf{label}, \mathbf{c} \in \mathbb{Z}_q^{m\times 1})$ along with the corresponding message $\mathbf{m}$.
  2) The algorithm executes the following computations:

$$\begin{aligned} h &= \mathsf{Hash}(hk = \mathbf{k}, W := (c, \mathbf{m})) \\ &= R\Big(\Big[\mathbf{c} - (\mathbf{0} \,|\, \mathsf{encode}(\mathbf{m}))^T\Big]^T \cdot \mathbf{k}\Big) \\ &= R\Big((\mathbf{s}^T \cdot \mathbf{A}_u) \cdot \mathbf{k} + \mathbf{e}^T \cdot \mathbf{k} \,(\mathrm{mod}\ q)\Big), \end{aligned} \tag{4.3}$$

  and outputs the hash values over the set $\{0,1\}$, where $\mathbf{e}^T \cdot \mathbf{k}$ is the noise element with the upper bounded $|\mathbf{e}^T\mathbf{k}| \leq \|\mathbf{e}^T\| \cdot \|\mathbf{k}\| \leq (r\sqrt{mn}) \cdot (\alpha q\sqrt{mn}) < \varepsilon/2 \cdot q/4$.
  3) Next, the algorithm obtains the result of $b := h \,(\mathrm{mod}\ 2) \in \{0,1\}$ and outputs $b = 0$ if $h < 0$; otherwise, outputs $b = 1$. Notably, the value h is in $[-(q-1)/2, \cdots, (q-1)/2]$.
- $p = \mathsf{ProjHash}(ph = \mathbf{p}, W := (c, \mathbf{m}); w = \mathbf{s})$. The projection hash function executes the following computations,
  1) On input a projected key $ph = \mathbf{p} \in \mathbb{Z}_q^{n\times 1}$, the word W, and the witness $\mathbf{s} \in \mathbb{Z}_q^{n\times 1}$.
  2) The algorithm gains p by computing

$$\begin{aligned} p &= \mathsf{ProjHash}(ph = \mathbf{p}, W := (c, \mathbf{m}); w = \mathbf{s}) \\ &= R\Big(\mathbf{s}^T \cdot (\mathbf{A}_u\mathbf{k}) \,(\mathrm{mod}\ q)\Big) \in \{0,1\}. \end{aligned} \tag{4.4}$$

  3) Next, the algorithm obtains the result of $b := p \,(\mathrm{mod}\ 2) \in \{0,1\}$, outputs $b = 0$ if $p < 0$; otherwise, outputs $b = 1$.

Theorem 4.4. *The* MP-SPHF *is a smooth projective hash function based on the Micciancio-Peikert scheme.*

Proof. We will now prove Theorem 4.4 via the following two steps, which are used to respectively demonstrate approximate correctness and smoothness. □

Approximate Correctness (or Projection). To prove MP-SPHF satisfy the property of approximate correctness, we only need to prove the equality of $\mathsf{Hash}(hk = \mathbf{k}, W := (c, \mathbf{m})) = \mathsf{ProjHash}(ph = \mathbf{p}, W := (c, \mathbf{m}); w = \mathbf{s})$ with probability greater than $1/2$. In lattice-based setting, the projection of SPHF indicates that the relationship between hk and W from an NP language L (i.e., executing $\mathsf{Hash}(\cdot)$) is equivalent to the

relationship between ph and the witness w for any word in L (i.e., executing $\mathsf{ProjHash}(\cdot)$).

Lemma 4.5 (Approximate Correctness). *If the magnitude of inner product $\langle \mathbf{e}, \mathbf{k} \rangle$ is a small integer for the parameters n, $m \geq n\sqrt{\log q}$, then the outcomes of the following rounding functions 1). $y_{\mathsf{hash}} = R(\mathsf{Hash}(hk = \mathbf{k}, W := (c, \mathbf{m})))$ and 2). $y_{\mathsf{proj}} = R(\mathsf{ProjHash}(ph = \mathbf{p}, W := (c, \mathbf{m}); w = \mathbf{s}))$ are satisfying the following requirement:*

$$\Pr[\mathsf{HD}(y_{\mathsf{hash}}, y_{\mathsf{proj}}) > \varepsilon \cdot v] = \mathrm{negl}(\lambda), \tag{4.5}$$

where $R(x) = \lfloor 2x/q \rfloor \pmod 2$ is the deterministic rounding function.

Proof. For convenience, we use the typical $R(x)$ following the methodology of [9] to round the respective outcomes of $\mathsf{Hash}(\cdot)$ and $\mathsf{ProjHash}(\cdot)$. Calculating the round outcomes, we have that

$$\begin{aligned} & R\Big(\mathsf{Hash}(hk = \mathbf{k}, W := (c, \mathbf{m}))\Big) \\ &= R\Big(\mathsf{ProjHash}(ph = \mathbf{p}, W := (c, \mathbf{m}); w = \mathbf{s})\Big) \\ &= R\Big(\mathbf{s}^T \cdot (\mathbf{A}_u \mathbf{k}) \pmod q\Big). \end{aligned} \tag{4.6}$$

Consider the left side of the above equation, we can view $R(h)$ as a number in $[-\frac{(q-1)}{2}, \cdots, \frac{(q-1)}{2}]$ and output $b \in \{0, 1\}$. Moreover, the noise element $\mathbf{e}^T \cdot \mathbf{k}$ is bounded by $|\mathbf{e}^T \mathbf{k}| \leq \|\mathbf{e}^T\| \cdot \|\mathbf{k}\| \leq (r\sqrt{mn}) \cdot (\alpha q \sqrt{mn}) < \varepsilon/2 \cdot q/4$. Hence, the result of $R(\mathbf{e}^T \mathbf{k})$ is identical with 0. Thus,

$$b = \begin{cases} 0, & \text{if } R(h) < 0; \\ 1, & \text{if } R(h) > 0. \end{cases} \tag{4.7}$$

Consider the right side of the above equation, we have the following results

$$b = \begin{cases} 0, & \text{if } R\big(\mathbf{s}^T \cdot (\mathbf{A}_u \mathbf{k})\big) < 0; \\ 1, & \text{if } R\big(\mathbf{s}^T \cdot (\mathbf{A}_u \mathbf{k})\big) > 0. \end{cases} \tag{4.8}$$

This completes the proof. □

Smoothness. The smoothness property of SPHF implies that the hashing output is independent of ph for any $W \in X \setminus L$. Moreover, the typical deterministic rounding function $R(x) = \lfloor 2x/q \rfloor \pmod 2$ (a.k.a., so-called square-signal function) has harmonic coefficients $\hat{r}_j$ decreasing as $\Theta(1/j)$ in absolute value. Using such a rounding function, one would attempt to invoke the trapdoor inversion for $q/2$ many multiples of c, more details about the case in [9]. Furthermore, $W := (c, \mathbf{m}) \notin L$ (the word not in the NP language) indicates that the ciphertext c is not related to the message $\mathbf{m}$ under $pk = \mathbf{A}_u$. Hence, the statistical distance of the following two distributions is a negligible in λ,

$$\begin{aligned} &1).\ \{(ph, h) \,|\, \mathsf{HashKG}(L) \to \mathbf{k}, \\ &\quad \mathsf{ProjKG}(hk, L, W) \to \mathbf{A}_u \mathbf{k}, \\ &\quad \underline{\mathsf{Hash}(hk, L, W) = (\mathbf{s}^T \mathbf{A}_u + \mathbf{e}^T)\mathbf{k}}\}. \\ &2).\ \{(ph, h) \,|\, \mathsf{HashKG}(L) \to \mathbf{k}, \\ &\quad \mathsf{ProjKG}(hk, L, W) \to \mathbf{A}_u \mathbf{k}, \\ &\quad \underline{h \leftarrow \{0, 1\}}\}. \end{aligned} \tag{4.9}$$

We note that, $\mathsf{Hash}(hk, W) = (\mathbf{s}^T \mathbf{A}_u + \mathbf{e}^T)\mathbf{k}$ given $\mathsf{ProjKG}(hk, pk) = \mathbf{A}_u \mathbf{k}$. Due to $\mathbf{s}$ being the witness (or random vector), no information on $\mathsf{Hash}(hk, W)$ can be provided by $\mathsf{ProjKG}(hk, pk)$, and $\mathsf{Hash}(hk, W)$ is uniformly distributed over $\{0, 1\}$, given $\mathsf{ProjKG}(hk, pk)$.

Hence, the smoothness property of the projective hash function can be concluded.

5 Our One-Round PAKE via MP-SPHF

Since the first SPHFs were initially introduced by Cramer and Shoup [13], various extensions have been proposed. A significant breakthrough was given in [49] (PKC'13) which presented an explicit instantiation of $\mathsf{KV\text{-}SPHF}$ [23] by using the Cramer-Shoup ciphertext. The existing lattice-based constructions of PAKE are only described in [9], [50]. At ASIACRYPT'09, Katz and Vaikuntanathan [9] followed the framework of KOY-GL (i.e., Katz-Ostrovsky-Yung (KOY)[7] and Gennaro-Lindell (GL) [8]), and proposed the first three-round PAKE protocol based on lattices in a variant of the Bellare-Pointcheval-Rogaway model [6]. Benhamuda et al. [15], [50] gave a draft of one-round PAKE over lattices, but we cannot find the detailed security analysis. Importantly, their scheme was withdrawn from eprint due to the weakness of the security proof. Before describing our one-round PAKE based on $\mathsf{MP\text{-}SPHF}$, we first give a high-level of the KOY-GL framework.

1) Each party in the system first sends an encryption of the password synchronously, then the party uses an SPHF taking as input the encrypted password of the partner to generate an output, lastly, the party uses the output to determine whether there exists the same password.
2) The common session key is generated by multiplying two hash values, where the two hash values are coming from the $\mathsf{Hash}(\cdot)$ and $\mathsf{ProjHash}(\cdot)$ with the encrypted password respectively.

Here we remark that, *1).* if the encrypted passwords are the same, the hash function Hash and $\mathsf{ProjHash}$ compute the hash values in their own way but obtain the same results. *2).* Conversely, if the passwords are different, the two hash functions compute the values independently but output the different results, which meets the smoothness feature. In a word, the smoothness property implies that the values computed by the hash function of each player are independent. Therefore, an SPHF on a labeled $\mathrm{IND\text{-}CCA}$-secure encryption scheme is necessary to achieve this aim. But to this aim, the scheme only depends on the encryption and doesn't need the decryption algorithm, thus, no one need to know or store the secret key sk.

5.1 One-Round PAKE Protocol over Lattices

Our protocol follows the framework of Katz and Vaikuntanathan [23] which requires an $\mathrm{IND\text{-}CCA2}$-secure encryption scheme with an associated word-independent $\mathsf{KV\text{-}SPHF}$ for the both sides. As shown in Fig. 4, we initialize the one-round PAKE protocol over lattices by using $\mathrm{IND\text{-}CCA}$-secure encryption scheme. Note that we assume the server keeps the password pw in plain-text just for ease of presentation. In practice, due to the Zipf's law in user-chosen passwords [30],

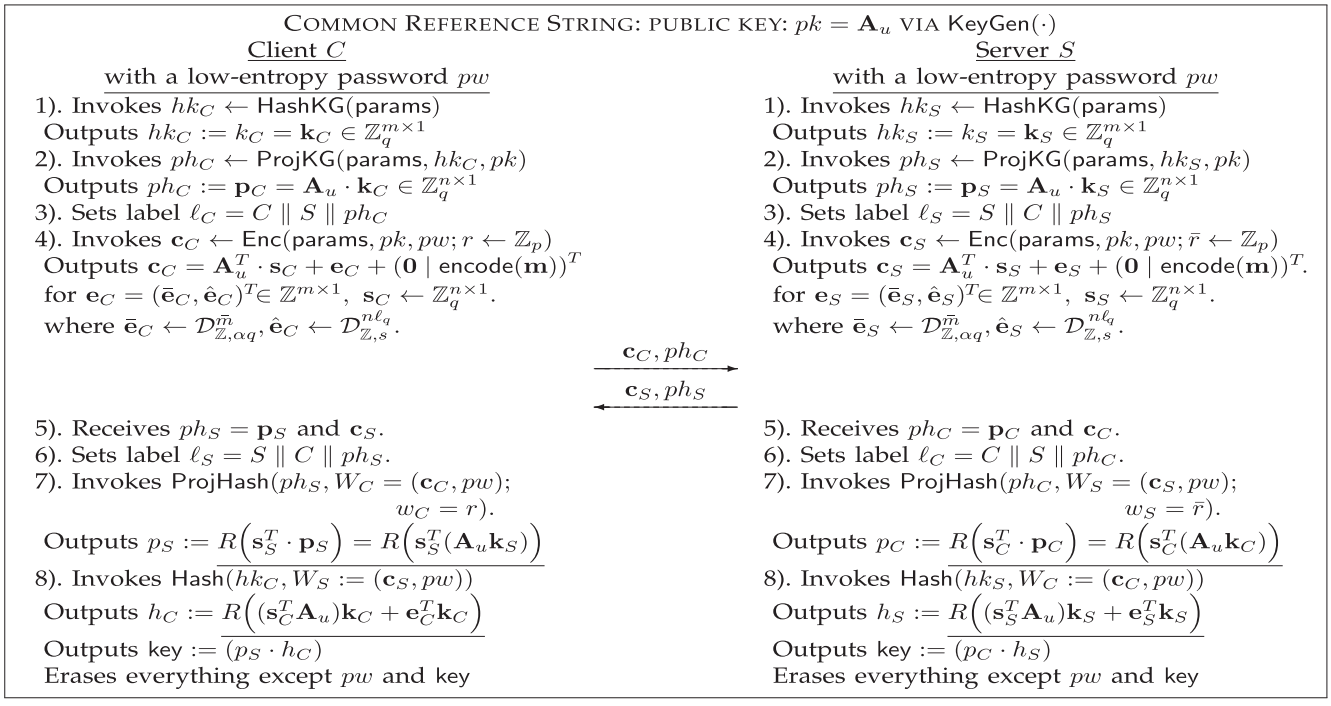

Fig. 4. One-round lattice-based PAKE protocol based on micciano-peikert scheme.

password-storing functions shall be slow enough to resist password guessing attacks. As suggested by the latest NIST SP800-63B "Authentication and Lifecycle Management" guidelines [51], the server shall use a memory-hard hash function (e.g., Scrypt and Lyra2 [52]) to hash the password with a random salt. Furthermore, the correctness of the established session key is implied by the Lemma 4.5.

To our knowledge, once the label is fixed at advanced to some constant, then the resulting scheme degrades to an IND-CPA-secure scheme. To solve this problem, Zhang and Yu [14] provided an idea to achieve an IND-CCA2-secure scheme by combining simulation-sound NIZK over lattices in random oracle setting. But the main trouble is that we don't know how to obtain simulation-sound NIZK over lattices in standard setting. Benhamouda et al. [15] provided a generic transformation using the idea of [37], which can be used to convert an labeled-IND-CCA1-secure scheme to an IND-CCA2-secure scheme. To do so, in the BPR model [6], we first upgrade the labeled-IND-CCA1-secure MP scheme [22] to the IND-CCA2-secure scheme by using a strong one-time signature scheme $\mathsf{Sgn} = (\mathsf{Sgn.Gen}, \mathsf{Sgn.Sign}, \mathsf{Sgn.Ver})$. The client generates the signature for each ciphertext, while the server verifies the correctness of the signature.

5.2 Security Proof

Theorem 5.1. *The one-round* PAKE *protocol (in short* Π*) in Fig. 4 is secure in the BPR model under the* LWE *assumptions, if the Micciancio-Peikert scheme is* IND-CCA*-secure with an associated* MP-SPHF.

Proof. Our proof follows the approaches of Katz and Vaikuntanathan [23] and Benhamuda et al. [29]. To apply the modularity approach of the proof given in [29, Theorem 4], the main work of this proof is to check whether our primitives (i.e., MP encryption and MP-SPHF) fulfill the same properties. Essentially, we have discussed both correctness and smoothness properties of MP-SPHF in the preceding section, below we detail the rigorous security analysis of the protocol based on MP-SPHF.

We assume that there is a PPT adversary $\mathcal{A}$ attacking our protocol Π. We then construct a sequence of experiments $\mathsf{Expt}.0, \mathsf{Expt}.1, \cdots, \mathsf{Expt}.6$ with the original experiment corresponding to $\mathsf{Expt}.0$. Importantly, we denote $\mathsf{Adv}^{\mathsf{Expt}.i}_{\mathcal{A}}(\lambda)$ as the advantage Adv of $\mathcal{A}$ in experiment Expt. i. To show the adversary is with the desired bound on $\mathsf{Adv}^{\Pi}_{\mathcal{A}} = \mathsf{Adv}^{\mathsf{Expt}.0}_{\mathcal{A}}(\lambda)$, we strictly bound the effect of each change in the experiment on the advantage of $\mathcal{A}$, and then show that

$$\mathsf{Adv}^{\mathsf{Expt}.6}_{\mathcal{A}}(\lambda) \leq C' \cdot Q^{s'}(\lambda). \tag{5.1}$$

Experiment $\mathsf{Expt}.0$. This is the real attack game, whose advantage is quantified as $\mathsf{Adv}^{\mathsf{Expt}.0}_{\mathcal{A}}(\lambda) = \varepsilon$. Then, in order to make the trivial attacks possible, we incrementally change the process of simulation. In this experiment, all honest players have their private input values that can be used by the simulator. Following [23], [29], there exist two types of Send queries:

- $\mathsf{Send}_0(C, i, S)$-query. In this query, the adversary first requires the oracle $\mathsf{Send}_0(\cdot)$ to initiate an execution between an instance Π^i_C of C and an instance of S. Then, C queries S to initiate the execution and S answers the query by a flow and sends to communicate with C.
- $\mathsf{Send}_1(C, i, \mathsf{msg})$-query. msg is sent by the adversary to Π^i_C via the oracle. The oracle provides no answer/response, but defines (or computes) his/her own session key, for possible later Reveal or Test queries from the Π^i_C.

Experiment $\mathsf{Expt}.1$. $\mathsf{Expt}.1$ is identical to $\mathsf{Expt}.0$ except that we modify the strategy how to handle Execute-queries.

In particular, we use the encryption of fake password pw_0 from the Zipf distribution to replace the ciphertext $\mathbf{c}_C$ and $\mathbf{c}_S$ in response to $\mathsf{Execute}(U_C, i, U_S, j)$. Clearly, the fake password pw_0 is not in language L. Moreover, due to participants know the hashing key and projective key, thus they can compute the common session key:

$$\begin{aligned}\mathsf{key}_C &= \mathsf{Hash}(hk_C, W_S := (\mathbf{c}_S, pw))\cdot \\ &\quad \mathsf{ProjHash}(ph_S, W_C = (\mathbf{c}_C, pw); w_C = r) \\ &= \mathsf{Hash}(hk_S, W_C := (\mathbf{c}_C, pw))\cdot \\ &\quad \mathsf{ProjHash}(ph_C, W_S = (\mathbf{c}_S, pw); w_S = \bar{r}) \\ &= \mathsf{key}_S.\end{aligned} \tag{5.2}$$

Due to the soundness property of the SPHF, there is no impact to compute key via the different way using either the initial way or the modified way. In fact, this is indistinguishable property of the probabilitic encryption scheme, for each Execute-query. Thus, we can obtain

$$|\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Expt}.1}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{\mathsf{Expt}.0}(\lambda)| \leq \mathrm{negl}(\lambda) \tag{5.3}$$

by using a series of hybrid hops.

Experiment Expt.2. In the current experiment, the manner how one responds to Execute-queries is modified again. We replace the common session key by sampling a random value from the uniform distribution. At this point, the "password" is not satisfied the requirement of generating the session key, and the indistinguishability property between Expt.2 and Expt.1 is guaranteed by the smoothness, i.e.,

$$|\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Expt}.2}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{\mathsf{Expt}.1}(\lambda)| \leq \mathrm{negl}(\lambda). \tag{5.4}$$

Experiment Expt.3. Expt.3 is identical to Expt.2, except that we change the way of how one deals with Send_1-queries. Concretely, in this experiment, to answer the query of $\mathsf{Send}_1(C, i, \mathsf{msg})$ along with $\mathsf{msg} = (ph_S, \mathbf{c}_S)$, the simulator in the name of the U_C^i introduces a decryption oracle (or knowing the decryption key more precisely) to decrypt the "unused" received message $\mathsf{msg} = (ph_S, \mathbf{c}_S)$. Thus, there are three cases as follows:

1) If msg has been altered (or generated) by the adversary, then one can invoke the decryption oracle to recover the password pw contained in word W by decrypting the ciphertext. Hence, there exists the two following cases:
   a) One can assert that the adversary $\mathcal{A}$ wins the game and the game then is terminated, if they are both correct $W \in L$ and consistent with the values of the receiver ($pw_{C,S} = pw$).
   b) One needs to choose key at random, if they are both incorrect and/or inconsistent with the values of receiver.
2) If the msg is a response to a previous flow from the simulator (or used previously), then the projective key can be obtained by the simulator who knows the hashing key. The next step is that the simulator can use the two hashing keys (the hashing key and the projective hashing key) to generate the session key. Specifically, $\mathsf{key} = \mathsf{Hash}(hk_C, W_S := (\mathbf{c}_S, pw))\cdot \mathsf{ProjHash}(ph_S,\ W_C = (\mathbf{c}_C, pw); w_C = r)$, where the ciphertext $\mathbf{c}_S$ of the server is not generated by using the randomness, which is similar to the case Expt.2.

To facilitate the following analysis, the first case (1a) is denoted as Event Ev, whose probability will be calculated later in Expt.6. Notably, the modification in case (1a) doesn't affect the final result except that it increases the advantage of $\mathcal{A}$. The adaptive-smoothness guarantees that the advantage of the adversary in the second modification of case (1b) only increases in a negligible term due to it being indistinguishable. In addition, the third modification in case (2) has no impact on the way how to compute the key, so finally

$$|\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Expt}.3}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{\mathsf{Expt}.2}(\lambda)| \leq \mathrm{negl}(\lambda). \tag{5.5}$$

Experiment Expt.4. The strategy of how to formulate Send_1-queries response is modified in this experiment. There are two cases appearing after that a "used" message $\mathsf{msg} = (ph_S, \mathbf{c}_S)$ is sent. In more detail:

- If there exists an instance Π_S^j of U_S partnered with an instance Π_C^i of U_C, then set $\mathsf{key} = \mathsf{skey}_C^i = \mathsf{skey}_S^j$.
- Otherwise, one chooses key at random.

Remarkably, due to the "used" message is a replay of a previous flow in the first case, thus, the common session key remains identical. In addition, due to adaptive-smoothness in the second case as in [23], [29], even if when ciphertexts and hashing keys are re-used, it does not impact that all the hash values are random looking. Therefore, Expt.4 guarantees that the indistinguishable holds and there exists

$$|\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Expt}.4}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{\mathsf{Expt}.3}(\lambda)| \leq \mathrm{negl}(\lambda). \tag{5.6}$$

Experiment Expt.5. The strategy of how we deal with Send_0-queries are modified in this experiment. One encrypts the fake pw_0 instead of encrypting a real and correct password, (which is similar to Expt.1 for Execute-queries in answering $\mathsf{Send}_0(C, i, S)$). This modification is necessary to simulate the decryption process in Send_1-queries, and therefore the indistinguishability between Expt.5 and Expt.4 holds due to IND-CCA-secure PKE scheme. Therefore, we have

$$|\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Expt}.5}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{\mathsf{Expt}.4}(\lambda)| \leq \mathrm{negl}(\lambda). \tag{5.7}$$

Experiment Expt.6. The experiment Expt.6 is identical to Expt.5, with the exception of using the dummy private inputs hk and ph. In particular, the hashing key hk and the projective hashing key ph do not depend on W, and the distributions of these keys are independent of the auxiliary private inputs. Hence,

$$|\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Expt}.6}(\lambda) - \mathsf{Adv}_{\mathcal{A}}^{\mathsf{Expt}.5}(\lambda)| \leq \mathrm{negl}(\lambda). \tag{5.8}$$

Putting them (i.e., Eq.(5.8)+Eq.(5.7)+$\cdots$+Eq.(5.3)) together, we obtain the following resutl

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Expt}.6}(\lambda) \geq \mathsf{Adv}_{\mathcal{A}}^{\mathsf{Expt}.0}(\lambda) - \mathrm{negl}(\lambda) = \varepsilon - \mathrm{negl}(\lambda). \tag{5.9}$$

TABLE 3
Properties Comparison of Related PAKE Protocols

Scheme	Assumption	Security	SPHF	UC	Rounds(Flows)	Client security	Server security
Katz et al. [7]	DDH	CCA	✙	×	3(3)	CCA	CCA
Gennaro-Lindell[8]	DDH	CCA	GL	×	3(3)	CCA	CCA
Jiang-Gong[16]	DDH	CCA&CPA	GL	×	3(3)	CPA&CCA‡	CCA
Katz-Vaikuntanathan[9]	LWE	CCA	GL	×	3(3)	CCA	CCA
Groce-Katz[18]	general	CCA&CPA	CS	✓	3(3)	CPA&CCA	CCA
Katz-Vaikuntanathan [23]	DLIN	CCA	CS	✓	1(2)	CCA	CCA
Canetti et al. [10]	DDH	CCA	✙	×	3(3)	CCA	CCA
Benhamouda et al. [29]	SXDH	CCA	KV	✓	1(2)	CCA	CCA
Abdalla et al. [19]	general	PCA&CPA	GL& KV	×	2(2)& 1(2)	CPA	PCA
Abdalla et al. [53]	DDH	CCA	✙	×	3(3)	CCA	CCA
Zhang-Yu [14]	LWE	CPA+NIZK	GL	×	2(2)	CCA	CCA
Benhamouda et al. [15]	LWE	CCA	KV	×	1(2)	CCA	CCA
Our scheme	LWE	CCA	KV	×	1(2)	CCA	CCA

The symbol ‡ implies that the corresponding scheme depends on two PKE schemes, one is CPA-secure and the other is CCA-secure;
The symbol ✙ implies that the corresponding scheme achieves a secure PAKE *without using* SPHF.

Actually, the main functionality of $\mathsf{Expt}.6$ is to determine whether the adversary wins Ev, which indicates that the advantage is $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{Expt}.6}(\lambda) = \Pr[Ev]$. Hence, we can obtain the following result $\varepsilon \leq \Pr[Ev] + \mathrm{negl}(\lambda)$. As discussed earlier, event Ev implies that the adversary $\mathcal{A}$ has encrypted the password $(i.e., pw)$, where the password is in the correct ($W \in L$) and consistent with the values of the receiver ($pw_{C,S} = pw$). Actually, we observe that the witnesses (or the randomness) for the honest parties during the simulation phase are never used, thus, the witnesses are chosen at the very end by our assumption, and they are only used to verify if the event Ev will happen:

$$\Pr[Ev] = \Pr[\exists k : pw_{C,S}(k) = pw_k, W \in L]. \tag{5.10}$$

In the above equation, k is the index of the reception of kth Send_1-query. Namely, the simulator is required to guess the private values at the begining, and then once the simulator has correctly guessed these values, it is then required to find a word in NP language. Hence,

$$\Pr[Ev] \leq C' \cdot Q^{s'}(\lambda), \tag{5.11}$$

where $C' \cdot Q^{s'}(\lambda)$ is the best chance of success that the adversary is is capable of finding a word W in L. The detailed advantage analysis about the CDF-Zipf model can be found in [30]. Lastly, combining all the above inequalities, we have

$$\varepsilon \leq C' \cdot Q^{s'}(\lambda) + \mathrm{negl}(\lambda). \tag{5.12}$$

This completes the proof. □

6 Performance Evaluation

We compare the proposed PAKE protocol with several others in this section, and the comparative summary is presented in Table 3. We observe that many follow-up works and optimizations [9], [10], [11], [12] were proposed following the first PAKE scheme of [7]. Hence, we focus only on SPHF-based PAKE protocols. Prior to discussing the comparison results, the notation of *flow* is to denote the unidirectional communication between the participants. Analogously, the notation of *round* is to denote the bidirectional communication between participants. When the messages are from two partners asynchronously, then the notation of round and flow have the same meaning, such as in the one-round (and two-flow) protocol [15], [23], [29]. On the contrary, if the messages are from two partners simultaneously, then each round is constituted by two different flows with distinct directions, such as in the two-round (and two-flow) protocol [14], [19], [54]. More specifically, our PAKE protocol is the first one-round PAKE over lattices with rigorous security analysis.

As shown in Table 3, we note that Benhamuda et al. [15], [50] constructed one-round PAKE over lattices, but their scheme only has a sketchy construction without a detailed security proof. Zhang and Yu [14] proposed a two-round PAKE over the lattice, but their IND-CCA-secure scheme depends on the simulation-sound NIZK proofs. As discussed earlier, constructing an efficient lattice-based simulation-sound NIZK remains an open research problem. Benhamouda et al. [15] also sketched a one-round PAKE framework based on their SPHF, sadly, no detailed construction is presented. Differing from these existing works [14], [15], [50], we present an improved MP-SPHF via the deterministic rounding function $R(\cdot)$, then used the improved MP-SPHF to design the one-round PAKE.

Regarding protocol security, as shown in Table 3, the security of KOY and GL scheme relies on the IND-CCA-secure encryption schemes. In other words, the encryption of the client and the server uses an IND-CCA-secure encryption scheme. However, the client of Jiang-Gong scheme [16] and Groce-Katz scheme [18] discards the IND-CCA-secure encryption scheme in the first round to encrypt the password pw, instead they use an IND-CPA-secure encryption scheme to encrypt the randomness in the first round, under the CRS model. Subsequently, to reduce the number of communication rounds, Abdalla, Benhamouda, and Pointcheval [19] designed a two-round PAKE protocol by using the proposed IND-PCA-secure encryption scheme. Note that, in this case, only the server uses the IND-PCA-secure encryption but the client uses the IND-CPA-secure encryption. However, in our one-round case and previous works, such as [15], [23], both parties (client and server) are required to adopt an IND-CCA-secure encryption scheme.

TABLE 4
Performance Comparison of PAKE Protocols over Lattices†

Scheme	$\|pw\|$	$\|hk\|$	$\|ph\|$	$\|\mathsf{key}\|$	$\|ct\|$
Katz-Vaikuntanathan [9]	ℓ	$O(m\log q)$	$O(n\log q)$	$O(c)$	$\Omega(mn\log q)$
		$\mathbf{e}\in\mathbb{Z}_q^m$	$\mathbf{u}=\mathbf{B}_{\mathsf{VK}}^T\mathbf{e}$	$\mathbf{e}^T[\mathbf{y}-\mathbf{U}(1,pw)^T]$	$\mathbf{y}=\mathbf{A}_{\mathsf{VK}}(\mathbf{s},1,pw)^T+\mathbf{x}$
Zhang-Yu [14]	ℓ	$O(m\log q)$	$O(n\log q)$	$O(c)$	$\Omega(mn\log q)$
		$\mathbf{e}\in\mathbb{Z}_q^m$	$\mathbf{u}=\mathbf{B}^T\mathbf{e}$	$\mathbf{e}^T[\mathbf{y}-\mathbf{U}(1,pw)^T]$	$\mathbf{y}=\mathbf{A}(\mathbf{s},1,pw)^T+\mathbf{x}$
Benhamouda et al. [15]	$n\ell_q$	$O(m\log q)$	$O(n\log q)$	$O(c)$	$\Omega(m\log q)$
		$\mathbf{k}\in\mathbb{Z}_q^m$	$\mathbf{u}=\mathbf{A}_u\mathbf{k}$	$p\cdot h$	$\mathbf{c}=\mathbf{A}_u^T\mathbf{s}+\mathbf{e}+(\mathbf{0}\,\|\,\mathsf{encode}(pw))^T$
Our scheme	$n\ell_q$	$O(m\log q)$	$O(n\log q)$	$O(c)$	$\Omega(m\log q)$

† $\mathbf{A}_{\mathsf{VK}}=[\mathbf{B}_{\mathsf{VK}}\,|\,\mathbf{U}]$; $\mathbf{A}=[\mathbf{B}\,|\,\mathbf{U}]$; $\mathbf{A}_u=[\bar{\mathbf{A}}\,|\,h(u)\mathbf{G}-\bar{\mathbf{A}}\mathbf{R}]\in\mathbb{Z}_q^{n\times m}$; *$\ell$ is random number and* $\ell_q=\lceil\log q\rceil=O(\log q)$.

Table. 4 provides a comparison of our scheme with state-of-the-art [9], [14], [15]. All these schemes have the same magnitude of hk, ph, and key, without considering the size of the pw. Furthermore, except that our PAKE protocol is round-optimal, one can observe that a significant optimization is potentially the magnitude of the ciphertext. In other words, under equivalent conditions, our protocol has a better communication complexity (i.e., $\Omega((m+n)\log q)$) than previous works [14], [23] (i.e., $\Omega((m+mn)\log q)$) and supports a longer password.

7 Extensions and Applications

In this section, we discuss potential extensions to the proposed PAKE protocol.

7.1 Two-Round PAKE via Regev and MP Scheme

Abdalla et al. [19] proposed a cryptographic primitive IND-PCA-secure PKE scheme, which can be used to meet the two-round PAKE requirements. However, they presented only a generic IND-PCA-secure PKE scheme. As every IND-CCA2-secure PKE scheme is also an IND-PCA-secure PKE scheme, in this work we assume their scheme to be secure (also there is no known attack on their scheme, at the time of this research). Notably, the client needs an IND-CPA-secure PKE scheme, but the server needs an IND-PCA-secure PKE scheme. In this setting, we can follow the methodology in [19] and adopt our improved Micciancio-Peikert (i.e., labeled IND-CCA-secure) scheme to instantiate the IND-PCA-secure PKE scheme. Similarly, we use Regev's scheme [25] as the building block, following the SPHF approach of Katz-Vaikuntanathan to design SPHF on the Regev ciphertext. Lastly, we can use it and implement the two-round PAKE via the lattice-based SPHF.

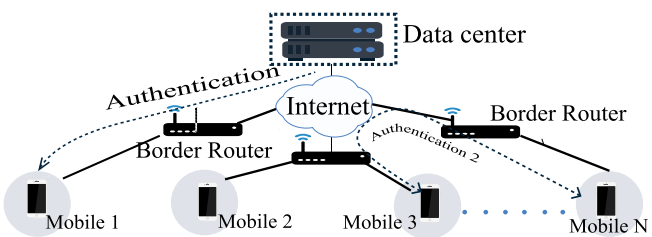

Fig. 5. Potential IoT device authentication.

7.2 UC-Secure One-Round PAKE over Lattices

We can also achieve UC-secure one-round PAKE protocol without increasing the number of rounds. For example, similar to approaches in [17], [18], [23], [55] and in particular [23], we can obtain the UC-secure one-round PAKE protocol over lattices. Key to this is for each participant to generate, apart from its encrypted message (pw) and its hash key hk, the NIZK proof π that participants used to encrypt its hash key hk where $\mathsf{ProjKG}(hk)=ph$. We then can formulate the ideal functionality $\mathcal{F}_{\mathsf{PAKE}}$ for PAKE protocol by utilizing the approach in [17].

7.3 Potential Application: IoT Device Authentication

Password-based authentication is the most common way to verify human users, and it has been a fundamental tool to provide security guarantees, especially for IoT devices (like RFID tags, smart cards, and wireless sensors) as shown in Fig. 5. For example, a number of authentication schemes (e.g., [56], [57]) augment passwords with what a user has (e.g., phones and cards) and/or what a user is (e.g., fingerprints and iris) to achieve multi-factor security for security-critical applications. However, with the advent of quantum computing [58], these conventional schemes that base their security on assumptions such as the intractability of the integer factorization problem and the discrete logarithm problem) will be insecure. Very recently, Banerjee et al. [59] have developed a novel cryptography circuit that can be used to protect low-power IoT devices via lattice-based cryptography in the coming age of quantum computing. Thus, this necessitates the design of quantum-resistant authentication schemes such as our proposed protocol.

We present a high-level description for our system model in Fig. 6. In this model, the resource-constrained IoT devices may connect to the resource-rich server (e.g., the cloud data center) via some public channel on the Internet. The goal of attacker, as shown in Fig. 6, is to gain access to the server. To prevent such an attacker, our one-round PAKE can be used. Specifically, the client and the server first share a low-entropy password pw, and then the client (and the server) gets and sends the projective hashing key ph_C (resp. ph_S), as well as the encryption of the pw under the public key pk_C (resp. pk_S) to each other, where ph is obtained by the projective key generation algorithm ProjKG and the encryption of the pw is obtained via the IND-CCA-secure encryption algorithm. Finally, the both parties (e.g., client and server) compute the session key

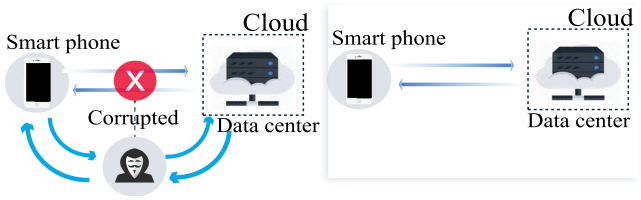

Fig. 6. System model: One-round PAKE for IoTs.

key via the corresponding inner scalar product between ProjHash and Hash.

8 Conclusion

With constant advances in quantum computing, efficient (e.g., round-optimal) quantum resistant authentication protocols are increasingly becoming a real-world necessity. The cryptographic primitive over lattices is one popular post-quantum cryptography. In this paper, we first revisited the methodology of KV-SPHF over lattices. Then, we designed a word-independent lattice-based SPHF (i.e., MP-SPHF) with adaptive smoothness from the Micciano-Peikert scheme [22], and an efficient lattice-based one-round PAKE protocol whose security is also demonstrated. Furthermore, we pointed out the potentials of constructing two-round and UC-secure one-round lattice-based PAKE protocols by combining our MP-SPHF with additional assumptions. Our PAKE protocol is particularly suitable for resource-constrained devices in an IoT environment, where communication cost is a key consideration.

Acknowledgments

The author Zengpeng Li would like to thank Hong-Sheng Zhou for his valuable discussions and feedback at the early stage of this project, and his encouragement for pursuing this work. This research was supported in part by the National Natural Science Foundation of China under Grant No. 61802006 and No. 61802214, in part by the China Postdoctoral Science Foundation under Grants No. 2018M640026 and No. 2019T120019, in part by the National Natural Science Foundation of Shandong province, China under Grant No. ZR2019BF009, and in part by the Applied Basic Research Project of Qingdao under Grant No.19-6-2-6-cg.

References

[1] A. Everspaugh, R. Chatterjee, S. Scott, A. Juels, and T. Ristenpart, "The pythia PRF service," in *Proc. USENIX Secur.*, 2015, pp. 547–562.
[2] J. Schneider, N. Fleischhacker, D. Schröder, and M. Backes, "Efficient cryptographic password hardening services from partially oblivious commitments," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 1192–1203.
[3] A. Yang, X. Tan, J. Baek, and D. S. Wong, "A new ADS-B authentication framework based on efficient hierarchical identity-based signature with batch verification," *IEEE Trans. Serv. Comput.*, vol. 10, no. 2, pp. 165–175, Mar./Apr. 2017.
[4] X. Wu, R. Jiang, and B. K. Bhargava, "On the security of data access control for multiauthority cloud storage systems," *IEEE Trans. Serv. Comput.*, vol. 10, no. 2, pp. 258–272, Mar./Apr. 2017.
[5] S. M. Bellovin and M. Merritt, "Encrypted key exchange: Password-based protocols secure against dictionary attacks," in *Proc. IEEE Comput. Soc. Symp. Res. Secur. Privacy*, 1992, pp. 72–84.
[6] M. Bellare, D. Pointcheval, and P. Rogaway, "Authenticated key exchange secure against dictionary attacks," in *Proc. Int. Conf. Theory Appl. Cryptographic Tech.*, 2000, pp. 139–155.
[7] J. Katz, R. Ostrovsky, and M. Yung, "Efficient password-authenticated key exchange using human-memorable passwords," in *Proc. Int. Conf. Theory Appl. Cryptographic Tech.*, 2001, pp. 475–494.
[8] R. Gennaro and Y. Lindell, "A framework for password-based authenticated key exchange," in *Proc. Int. Conf. Theory Appl. Cryptographic Tech.*, 2003, pp. 524–543.
[9] J. Katz and V. Vaikuntanathan, "Smooth projective hashing and password-based authenticated key exchange from lattices," in *Proc. Int. Conf. Theory Appl. Cryptology Inf. Secur.*, 2009, pp. 636–652.
[10] R. Canetti, D. Dachman-Soled, V. Vaikuntanathan, and H. Wee, "Efficient password authenticated key exchange via oblivious transfer," in *Proc. Int. Workshop Public Key Cryptography*, 2012, pp. 449–466.
[11] S. Jarecki, H. Krawczyk, and J. Xu, "OPAQUE: An asymmetric PAKE protocol secure against pre-computation attacks," in *Proc. Annu. Int. Conf. Theory Appl. Cryptographic Tech.*, 2018, pp. 456–486.
[12] P. Dupont, J. Hesse, D. Pointcheval, L. Reyzin, and S. Yakoubov, "Fuzzy password-authenticated key exchange," in *Proc. Annu. Int. Conf. Theory Appl. Cryptographic Tech.*, 2018, pp. 393–424.
[13] R. Cramer and V. Shoup, "Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption," in *Proc. Int. Conf. Theory Appl. Cryptographic Tech.*, 2002, pp. 45–64.
[14] J. Zhang and Y. Yu, "Two-round PAKE from approximate SPH and instantiations from lattices," in *Proc. Int. Conf. Theory Appl. Cryptology Inf. Secur.*, 2017, pp. 37–67.
[15] F. Benhamouda, O. Blazy, L. Ducas, and W. Quach, "Hash proof systems over lattices revisited," in *Proc. IACR Int. Workshop Public Key Cryptography*, 2018, pp. 644–674.
[16] S. Jiang and G. Gong, "Password based key exchange with mutual authentication," in *Proc. Int. Workshop Select. Areas Cryptography*, 2004, pp. 267–279.
[17] R. Canetti, S. Halevi, J. Katz, Y. Lindell, and P. D. MacKenzie, "Universally composable password-based key exchange," in *Proc. Annu. Int. Conf. Theory Appl. Cryptographic Tech.*, 2005, pp. 404–421.
[18] A. Groce and J. Katz, "A new framework for efficient password-based authenticated key exchange," in *Proc. Proc. 17th ACM Conf. Comput. Commun. Secur.*, 2010, pp. 516–525.
[19] M. Abdalla, F. Benhamouda, and D. Pointcheval, "Public-key encryption indistinguishable under plaintext-checkable attacks," in *Proc. IACR Int. Workshop Public Key Cryptography*, 2015, pp. 332–352.
[20] C. Peikert, "Public-key cryptosystems from the worst-case shortest vector problem: Extended abstract," in *Proc. 41st Annu. ACM Symp. Theory Comput.*, 2009, pp. 333–342.
[21] A. Sahai, "Non-malleable non-interactive zero knowledge and adaptive chosen-ciphertext security," in *Proc. 40th Annu. Symp. Found. Comput. Sci.*, 1999, pp. 543–553.
[22] D. Micciancio and C. Peikert, "Trapdoors for lattices: Simpler, tighter, faster, smaller," in *Proc. Annu. Int. Conf. Theory Appl. Cryptographic Tech.*, 2012, pp. 700–718.
[23] J. Katz and V. Vaikuntanathan, "Round-optimal password-based authenticated key exchange," in *Proc. Theory Cryptography Conf.*, 2011, pp. 293–310.
[24] A. Rosen and G. Segev, "Chosen-ciphertext security via correlated products," in *Proc. Theory Cryptography Conf.*, 2009, pp. 419–436.
[25] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," in *Proc. Proc. 37th Annu. ACM Symp. Theory Comput.*, 2005, pp. 84–93.
[26] C. Gentry, C. Peikert, and V. Vaikuntanathan, "Trapdoors for hard lattices and new cryptographic constructions," in *Proc. 40th Annu. ACM Symp. Theory Comput.*, 2008, pp. 197–206.
[27] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proc. 41st Annu. ACM Symp. Theory Comput.*, 2009, pp. 169–178.
[28] Z. Li, C. Ma, and D. Wang, "Achieving multi-hop PRE via branching program," *IEEE Trans. Cloud Comput.*, pp. 1–14, 2017, doi:10.1109/TCC.2017.2764082.
[29] F. Benhamouda, O. Blazy, C. Chevalier, D. Pointcheval, and D. Vergnaud, "New techniques for sphfs and efficient one-round PAKE protocols," in *Proc. Annu. Cryptology Conf.*, 2013, pp. 449–475.
[30] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, "Zipf's law in passwords," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 11, pp. 2776–2791, Nov. 2017.
[31] D. Wang and P. Wang, "Two birds with one stone: Two-factor authentication with security beyond conventional bound," *IEEE Trans. Dependable Secur. Comput.*, vol. 15, no. 4, pp. 708–722, Jul./Aug. 2018.

[32] S. Jarecki, H. Krawczyk, M. Shirvanian, and N. Saxena, "Two-factor authentication with end-to-end password security," in *Proc. IACR Int. Workshop Public Key Cryptography*, 2018, pp. 431–461.
[33] Q. Xie, D. S. Wong, G. Wang, X. Tan, K. Chen, and L. Fang, "Provably secure dynamic id-based anonymous two-factor authenticated key exchange protocol with extended security model," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 6, pp. 1382–1392, Jun. 2017.
[34] F. Hao, R. Metere, S. F. Shahandashti, and C. Dong, "Analyzing and patching SPEKE in ISO/IEC," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 11, pp. 2844–2855, Nov. 2018.
[35] X. Yi, F.-Y. Rao, Z. Tari, F. Hao, E. Bertino, I. Khalil, and A. Y. Zomaya, "Id2s password-authenticated key exchange protocols," *IEEE Trans. Comput.*, vol. 65, no. 12, pp. 3687–3701, Dec. 2016.
[36] R. Canetti, "Universally composable security: A new paradigm for cryptographic protocols," in *Proc. 42nd IEEE Symp. Found. Comput. Sci.*, 2001, pp. 136–145.
[37] D. Dolev, C. Dwork, and M. Naor, "Non-malleable cryptography (extended abstract)," in *Proc. 23rd Annu. ACM Symp. Theory Comput.*, 1991, pp. 542–552.
[38] M. Naor and M. Yung, "Public-key cryptosystems provably secure against chosen ciphertext attacks," in *Proc. 22nd Annu. ACM Symp. Theory Comput.*, 1990, pp. 427–437.
[39] E. Kiltz, K. Pietrzak, M. Stam, and M. Yung, "A new randomness extraction paradigm for hybrid encryption," in *Proc. Annu. Int. Conf. Theory Appl. Cryptographic Tech.*, 2009, pp. 590–609.
[40] K. Kurosawa and Y. Desmedt, "A new paradigm of hybrid encryption scheme," in *Proc. Annu. Int. Cryptology Conf.*, 2004, pp. 426–442.
[41] C. Peikert and B. Waters, "Lossy trapdoor functions and their applications," in *Proc. 40th Annu. ACM Symp. Theory Comput.*, 2008, pp. 187–196.
[42] P. Mukherjee and D. Wichs, "Two round multiparty computation via multi-key FHE," in *Proc. Part II 35th Annu. Int. Conf. Adv. Cryptology*, 2016, pp. 735–763.
[43] R. Chen, Y. Mu, G. Yang, F. Guo, and X. Wang, "Dual-server public-key encryption with keyword search for secure cloud storage," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 4, pp. 789–798, Apr. 2016.
[44] D. Wang and P. Wang, "On the implications of Zipf's law in passwords," in *Proc. Eur. Symp. Res. Comput. Secur.*, 2016, pp. 111–131.
[45] D. Wang, P. Wang, D. He, and Y. Tian, "Birthday, name and bifacial-security: Understanding passwords of chinese web users," in *Proc. USENIX Secur.*, 2019, pp. 1537–1555.
[46] C. Peikert and B. Waters, "Lossy trapdoor functions and their applications," *SIAM J. Comput.*, vol. 40, no. 6, pp. 1803–1844, 2011.
[47] S. Agrawal, D. Boneh, and X. Boyen, "Efficient lattice (H)IBE in the standard model," in *Proc. 29th Annu. Int. Conf. Theory Appl. Cryptographic Tech.*, 2010, pp. 553–572.
[48] D. Boneh, R. Canetti, S. Halevi, and J. Katz, "Chosen-ciphertext security from identity-based encryption," *SIAM J. Comput.*, vol. 36, no. 5, pp. 1301–1328, 2007.
[49] F. B. Hamouda, O. Blazy, C. Chevalier, D. Pointcheval, and D. Vergnaud, "Efficient uc-secure authenticated key-exchange for algebraic languages," in *Proc. Int. Workshop Public Key Cryptography*, 2013, pp. 272–291.
[50] O. Blazy, C. Chevalier, L. Ducas, and J. Pan, "Exact smooth projective hash function based on lwe," IACR Cryptology ePrint Archive, 2013/821. 2013. [Online]. Available: http://eprint.iacr.org/2013/821.pdf
[51] P. A. Grassi, E. M. Newton, R. A. Perlner, A. R. Regenscheid, W. E. Burr, and J. P. Richer, "NIST 800–63B digital identity guidelines: Authentication and lifecycle management," National Institute of Standards and Technology, McLean, VA, USA, Tech. Rep., 2017. [Online]. Available: https://doi.org/10.6028/NIST.SP.800-63b
[52] E. R. Andrade, M. A. Simplicio, P. S. Barreto, and P. C. dos Santos, "Lyra2: Efficient password hashing with high security against time-memory trade-offs," *IEEE Trans. Comput.*, vol. 65, no. 10, pp. 3096–3108, Oct. 2016.
[53] M. Abdalla, F. Benhamouda, and P. MacKenzie, "Security of the J-PAKE password-authenticated key exchange protocol," in *Proc. IEEE 2015Symp. Secur. Privacy*, 2015, pp. 571–587.
[54] Z. Li and D. Wang, "Two-round PAKE protocol over lattices without NIZK," in *Proc. Int. Conf. Inf. Secur. Cryptology*, 2018, pp. 138–159.
[55] M. Abdalla, D. Catalano, C. Chevalier, and D. Pointcheval, "Efficient two-party password-based key exchange protocols in the UC framework," in *Proc. The Cryptopgraphers' Track RSA Conf. Top. Cryptology*, 2008, pp. 335–351.
[56] D. Wang, W. Li, and P. Wang, "Measuring two-factor authentication schemes for real-time data access in industrial wireless sensor networks," *IEEE Trans. Ind. Inf.*, vol. 14, no. 9, pp. 4081–4092, Sep. 2018.
[57] J. Srinivas, A. K. Das, M. Wazid, and N. Kumar, "Anonymous lightweight chaotic map-based authenticated key agreement protocol for industrial internet of things," *IEEE Trans. Depend. Secur. Comput.*, p. 1, 2018, doi: 10.1109/TDSC.2018.2857811.
[58] J. Preskill, "Quantum computing in the NISQ era and beyond," *arXiv: 1801.00862*, vol. 2, pp. 1–20, 2018.
[59] U. Banerjee, A. Pathak, and A. P. Chandrakasan, "An energy-efficient configurable lattice cryptography processor for the quantum-secure internet of things," in *Proc. IEEE Int. Solid- State Circuits Conf.*, 2019, pp. 46–48.

Zengpeng Li received the PhD degree in Computer Science and Technology from Harbin Engineering University, China, in 2018. During his doctoral program, he was a long-term PhD visiting student (and researcher assistant) with the University of Auckland, NZ and Virginia Commonwealth University, USA, respectively, from 2015 to 2017. Currently, he is a lecturer with Qingdao University, China, and a postdoctoral researcher with Lancaster University, UK. His research focuses on lattice-based cryptography, distributed secure computing, cryptographic protocol, and password-based cryptography.

Ding Wang (M'17) received the PhD degree in information security from Peking University, China, in 2017. Currently, he is a lecturer and supported by the "Boya Postdoctoral Fellowship" at Peking University. He has been involved in the community as a TPC member and a reviewer for more than 50 international conferences and journals. His works appear in prestigious venues like ACM CCS, Usenix Security, NDSS, ESORICS, DSN, the *IEEE Transactions on Dependable and Secure Computing* and the *IEEE Transactions on Information Forensics and Security*, and seven of them are selected as "ESI highly cited papers". He received the 2018 ACM China Doctoral Dissertation Award, and the 2017 China Computer Federation (CCF) Outstanding Doctoral Dissertation Award. His research interests include password, authentication and provable security. He is a member of the IEEE.