Quantum2FA: Efficient Quantum-Resistant Two-Factor Authentication Scheme for Mobile Devices

Qingxuan Wang, Ding Wang, Chi Cheng, and Debiao He

Abstract—Smart-card based password authentication has been the most widely used two-factor authentication (2FA) mechanism for security-critical applications (e.g., e-Health, smart grid and e-Commerce) in the past decades, and it is likely to hold its status in the foreseeable future. Hundreds of this type of 2FA schemes have been proposed, yet to our knowledge, most of them are built on the intractability of conventional hard problems (e.g., discrete logarithm problems and integer factoring problems) which are no longer hard in the quantum era. With the recent advancements in quantum computing, the design of secure and efficient smart-card based password authentication schemes against quantum attacks is becoming increasingly urgent. However, it is not as simple as it seems, *how to design such a quantum-resistant 2FA scheme is challenging due to the demanding security requirements and the resource-constrained nature of mobile devices*. In this work, we take the first step towards this issue by proposing Quantum2FA, a practical quantum-resistant smart-card-based password authentication scheme that employs Alkim *et al.*'s lattice-based key exchange and Wang-Wang's "fuzzy-verifier + honeywords" technique (IEEE TDSC'18). Particularly, Quantum2FA can thwart the newly revealed key-reuse attack (ACISP'18, CT-RSA'19) against lattice-based key exchange schemes in two aspects: signal leakage attacks and key mismatch attacks. Specifically, it restricts the necessary conditions (i.e., the attacker must be the initiator of the key exchange) for an adversary to analyze the signal; It introduces honeywords to detect the key mismatches between the smart card and the server, and thus smart card loss attack can be thwarted. We formally prove the security of Quantum2FA under the random oracle model and demonstrate its efficiency through experiments on a 32 MHz 8-bit AVR Embedded Processor. Comparison results show that Quantum2FA is not only more secure but also offers better computation efficiency than the state-of-the-art conventional 2FA schemes.

Index Terms—Two-factor authentication, quantum security, lattice-based cryptography, key-reuse, honeyword

1 Introduction

Smart-card-based password authentication has been the most widely used two-factor authentication (2FA) mechanism for security-critical applications (e.g., e-banking [1], smart grid [2], [3] and e-health), preventing malicious parties from accessing the sensitive resources and services at remote servers. The past three decades have witnessed the fruitful research on this kind of authentication since the seminal work of Chang and Wu at 1991 [4], and a large number of 2FA schemes with varied security [5], privacy [6] and efficiency [7] have been proposed.

Before 2008, most "password + smart card" 2FA schemes (e.g., [8], [9], [10]) were built on the assumption that smart card is tamper resistant and the sensitive data stored in the card is secure. The main design challenge lies in how to resist the traditional attacks such as replay, impersonation and parallel session attacks. However, with the progress in side-channel attacks, which is a kind of cryptanalytic attacks, security parameters stored in the card can be extracted out. When launching side-channel attacks, an attacker will exploit the physical environment of a cryptosystem implementation to recover some leaked secrets [11]. Although some side-channel attacks defense methods like masking [12] and shuffling [13] have been applied to the secure products, there are still ways to attack them. For example, Carbone *et al.* [11] propose a full profiled attack against an RSA implementation, which runs on a processor equipped with the above masking method. More recently, Roche *et al.* [14] successfully clone a legitimate Google Titan security key. These facts show that the parameters stored in security products, such as smart cards or hardware devices, are no longer unconditionally secure. Therefore, it is necessary and realistic to design protocols under the smart card non-tamper-resistant assumption.

- *Qingxuan Wang and Ding Wang are with the College of Cyber Science, Nankai University, Tianjin 300350, China, and with State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China, and also with Tianjin Key Laboratory of Network and Data Security Technology, Nankai University, Tianjin 300350, China. E-mail: wangqingxuan@mail.nankai.edu.cn, wangdingg@yeah.net.*
- *Chi Cheng is with the Hubei Key Laboratory of Intelligent Geo-Information Processing, School of Computer Science, China University of Geosciences, Wuhan 430074, China, also with the State Key Laboratory of Cryptology, Beijing 100878, China, and also with the Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China. E-mail: chengchi@cug.edu.cn.*
- *Debiao He is with the Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China, and also with the State Key Laboratory of Cryptology, Beijing 100878, China. E-mail: hedebiao@163.com.*

Manuscript received 18 May 2021; revised 8 November 2021; accepted 9 November 2021. Date of publication 22 November 2021; date of current version 16 January 2023.
This work was supported in part by the National Nature Science Foundation of China under Grant 62172240, in part by the National Natural Science Foundation of China (NSFC) under Grants 62172374 and 61672029, and in part by the Guangxi Key Laboratory of Trusted Software under Grant KX202038. (Corresponding author: Ding Wang.)
Digital Object Identifier no. 10.1109/TDSC.2021.3129512

Accordingly, considerable attention has been devoted to addressing the question of how to design a two-factor scheme with truly two-factor security under the non-tamper resistant assumption of smart cards, and some notable attempts include [15], [16]. It is worth noting that, as explicated in [5], the non-tamper resistant assumption about smart cards is conditional: only when the smart card is in the possession of the attacker for a relatively long time (e.g., a few hours), can the card be tampered; Otherwise, data in the card is secure. Unfortunately, most of these attempts have been found unable to attain truly two-factor security shortly after they were proposed (see cryptanalysis examples in [17]). The development of this field fall into a vicious "break-fix-break-fix" circle [5].

At IEEE TDSC'18, Wang and Wang [5] managed to break this "break-fix-break-fix" circle by proposing a systematic evaluation framework and the "fuzzy-verifier + honeywords" protocol design technique. Their evaluation framework is composed of a practical adversary model and an evaluation metric with 12 criteria, and has been established to be "indeed very comprehensive and well thought out to be used to critically analyze any further schemes that will be proposed in future" [18].

Actually, Wang and Wang's evaluation framework has already been adopted in a number of recent works (e.g., [6], [19], [20]). The effectiveness of their "fuzzy-verifier + honeywords" technique is demonstrated through 102.6 million real-life passwords, and its wide applicability is shown by applying it to 2FA schemes for various other environments. This technique has also been adopted in many recent works (e.g., [21], [22]). With this technique, Wang and Wang [5] further developed a new smart-card-based password authentication scheme based on the hardness of discrete logarithm problem and showed it could satisfy all the 12 criteria, well solving the long-standing "usability vs. security" tension [23] in the research area of two-factor authentication.

To our knowledge, most of the existing 2FA schemes (including all the above mentioned ones, e.g., [5], [16], [24]) are built on group-based or pairing-based cryptosystems, and thus they are vulnerable in the coming post-quantum era. It is well known that, if there exist large-scale quantum computers that can run Shor's algorithm [25], then current widely used hard problems, such as integer factorization problem, discrete logarithm problem and its elliptic curve counterpart, would be efficiently solved. With the recent advancements in quantum computing [26], [27], [28], standards organizations like IETF, IEEE, and NIST are busy in preparing solutions for securing our digital world in the quantum age. According to NIST's plan in June 2019 [29], the standards on post-quantum cryptography, which can resist attacks from both classical and quantum computers, will be available by 2022-2024.

Now, a natural question arises: *Is it possible to construct an efficient smart-card-based password authentication scheme that is secure in the quantum era?* The past thirty years of research has proved that it is incredibly difficult to design a traditional 2FA scheme (see Fig. 2 of [5]), the design of a practical quantum-resistant 2FA scheme can only be harder. As far as we know, a number of quantum-resistant password-only schemes (e.g., [30], [31], [32]) and authenticated key exchange (AKE) protocols (e.g., [32], [33], [34]) have been proposed, and some are even being standardised [35], yet none of them can be readily applied to build practical quantum-resistant smart-card-based password authentication schemes. On the one hand, such schemes need to meet many security goals (e.g., resistance against smart card loss attack and replay attack) and desirable properties (e.g., user anonymity [36] and repairability) beyond that of quantum-resistant password-only schemes and AKE protocols; On the other hand, smart cards are resource-constrained devices with limited computation capacity and storage space.

Currently, there are many candidate technical routes for constructing post-quantum cryptographic schemes such as hash-based, code-based, multivariate polynomial-based, lattice-based, and supersingular isogeny-based [35], [37]. Among them, the lattice based scheme, especially those exploiting the hardness of the ring-learning with errors problem (Ring-LWE), is a promising choice for smart-card-based password authentication due to its provable security, relatively low computation and communication costs. However, we cannot use these lattice-based key exchange schemes [32], [33], [34] directly to build our Quantum2FA, due to the recently proposed key reuse attacks on lattice-based cryptosystems [38], [39], [40]. When designing a smart-card-based password authentication scheme, we need to store some public key information in the smart card in advance inevitably and this may lead to key reuse attacks. Thus, special care shall be taken to such public key information.

In all we propose Quantum2FA, a secure and efficient smart-card based password authentication scheme, taking the first step towards practical quantum resistant 2FA in the quantum age. The contributions are summarized as follows:

- We provide a design framework for quantum-resistant 2FA in the context of key reuse which is inevitable while applying the key exchange to authentication scenarios. As a result, it is helpful for researchers to employ new key establishment algorithm or develop more new designs with the continuous advancement of the NIST Post-Quantum Cryptography Standardization Process.
- As far as we know, we are the first to build a smart-card based password authentication scheme whose security is on the basis of the Ring-LWE problem. The existence of a quantum connection, from solving a hard problem in ideal lattice in the worst case to solving the Ring-LWE problem in the average case, guarantees the security of the proposed approaches against quantum computers.
- We employ Alkim *et al.*'s lattice-based key exchange [41] and Wang-Wang's "fuzzy-verifier + honeywords" technique [5] to thwart not only all known attacks against conventional two-factor authentication schemes, but also the recently proposed key-reuse attacks (key mismatch and signal leakage) against Ring-LWE-based two-factor authentication for quantum era. Formal security analysis demonstrates that the proposed scheme Quantum2FA is

Fig. 1. Quantum-Resistant Two-factor Authentication.

secure against attacks from both classical and quantum computers under the harshest adversary model.

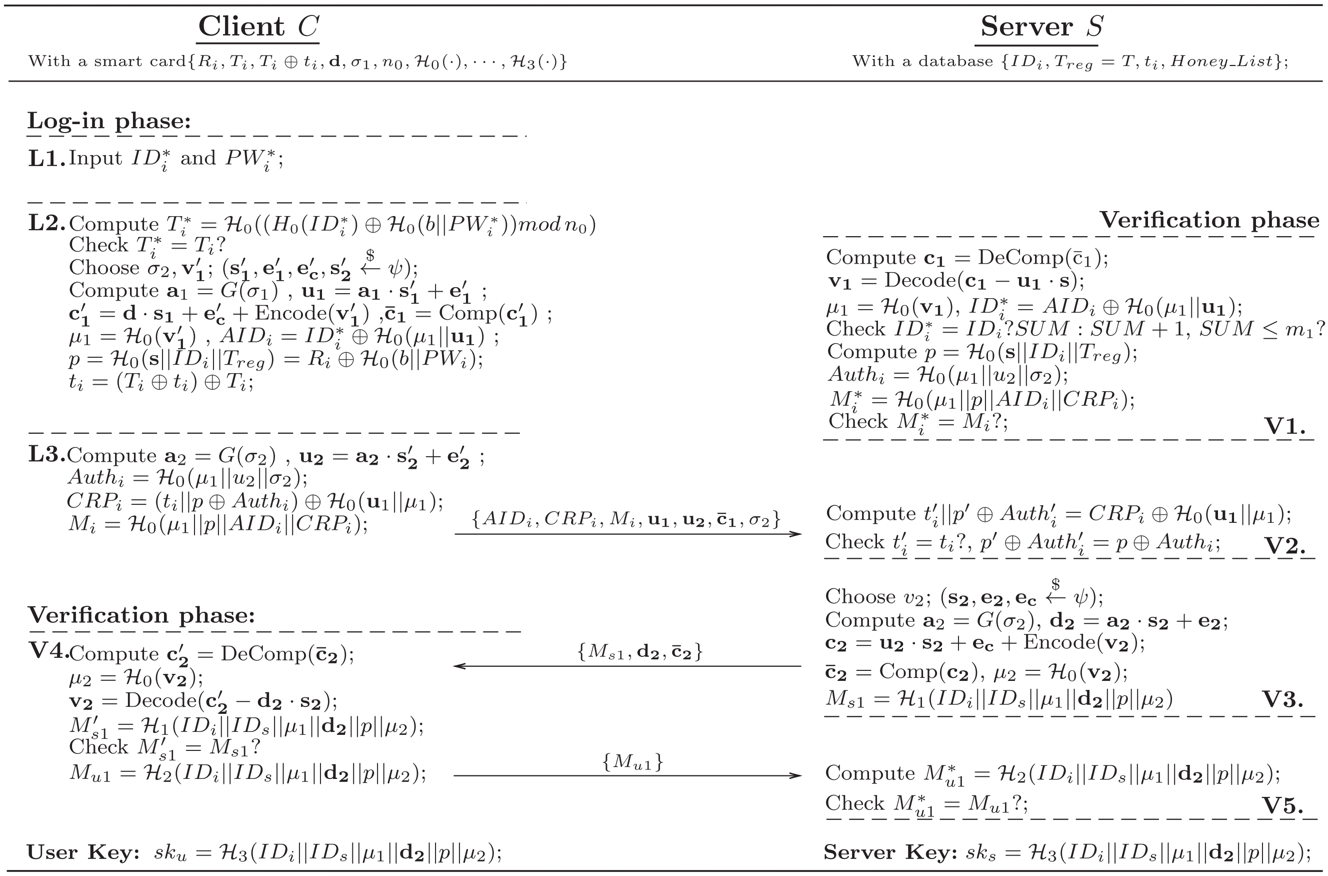
- We implement Quantum2FA on a 32 MHz micro controller. Comparison results show that Quantum2FA is not only more secure (our implementation can provide 101-bit post-quantum bit-security and the strength of 128-bit AES) but also offers better computation efficiency than the state-of-the-art traditional 2FA schemes.

2 Related Work

The development of quantum computers has had a considerable impact on traditional cryptography. While symmetric-key schemes like AES are still secure but need to increase the key size, public-key cryptographic schemes like RSA, ECDSA and ECDH are not so lucky, they are no longer secure under attacks from quantum computers. There are several cryptographic approaches to resist quantum attacks, such as hash-based, code-based, multivariate polynomial-based, lattice-based, and supersingular isogeny-based cryptosystem [42]. Among them, the lattice-based cryptosystem is considered to be one of the most promising solutions, which can be best illustrated from the overwhelming fraction of lattice-based candidates submitted and selected into the first round (41% = 26/64), second round (46% = 12/26), and the newly third round (71% = 5/7) competition issued by NIST [35], [43]. Besides, NIST regards lattice-based schemes as the most promising general-purpose algorithms for public-key encryption/KEM.

As one of the basic variants of the lattice-based problem, the Ring-LWE problem has gained significant attention [44]. In particular, the Ring-LWE key exchange [45] provides a good solution for establishing shared keys in a two-factor authentication scheme. Generally, we can divide the lattice-based key exchange into two categories [40]: reconciliation-based ones and encryption-based ones. The first reconciliation-based Ring-LWE key exchange scheme was proposed by Ding *et al.* [46], which asks one side to send an additional signal for the two sides to share an exact same key. Along the line of Ding *et al.* [46], there has been the scheme BCNS [34], which is an modification of the work of Peikert [47], the authenticated key exchange scheme proposed by Zhang *et al.* [33], and NewHope [48], which proposes an efficient reconciliation mechanism. After that, an encryption-based approach NewHope-Simple is given in [41].

The first reconciliation-based Ring-LWE key exchange scheme using passwords was proposed by Ding *et al.* at CT-RSA'17 [32]. It can be seen as a variant of the classic password-based key exchange (PAKE) protocols PAK and PPK [49]. Following Ding *et al.*'s PAKE protocol [32], a number



Client C

With a smart card$\{R_i, T_i, T_i \oplus t_i, \mathbf{d}, \sigma_1, n_0, \mathcal{H}_0(\cdot), \cdots, \mathcal{H}_3(\cdot)\}$

Server S

With a database $\{ID_i, T_{reg} = T, t_i, Honey\_List\}$;

Log-in phase:

L1. Input ID_i^* and PW_i^*;

L2. Compute $T_i^* = \mathcal{H}_0((H_0(ID_i^*) \oplus \mathcal{H}_0(b||PW_i^*)) mod\, n_0)$
Check $T_i^* = T_i$?
Choose $\sigma_2, \mathbf{v_1'}$; $(\mathbf{s_1'}, \mathbf{e_1'}, \mathbf{e_c'}, \mathbf{s_2'} \overset{\$}{\leftarrow} \psi)$;
Compute $\mathbf{a}_1 = G(\sigma_1)$, $\mathbf{u_1} = \mathbf{a_1} \cdot \mathbf{s_1'} + \mathbf{e_1'}$;
$\mathbf{c_1'} = \mathbf{d} \cdot \mathbf{s_1} + \mathbf{e_c'} + \text{Encode}(\mathbf{v_1'})$,$\mathbf{\bar{c}_1} = \text{Comp}(\mathbf{c_1'})$;
$\mu_1 = \mathcal{H}_0(\mathbf{v_1'})$, $AID_i = ID_i^* \oplus \mathcal{H}_0(\mu_1||\mathbf{u_1})$;
$p = \mathcal{H}_0(\mathbf{s}||ID_i||T_{reg}) = R_i \oplus \mathcal{H}_0(b||PW_i)$;
$t_i = (T_i \oplus t_i) \oplus T_i$;

L3. Compute $\mathbf{a}_2 = G(\sigma_2)$, $\mathbf{u_2} = \mathbf{a_2} \cdot \mathbf{s_2'} + \mathbf{e_2'}$;
$Auth_i = \mathcal{H}_0(\mu_1||u_2||\sigma_2)$;
$CRP_i = (t_i||p \oplus Auth_i) \oplus \mathcal{H}_0(\mathbf{u}_1||\mu_1)$;
$M_i = \mathcal{H}_0(\mu_1||p||AID_i||CRP_i)$;

$\xrightarrow{\{AID_i, CRP_i, M_i, \mathbf{u_1}, \mathbf{u_2}, \mathbf{\bar{c}_1}, \sigma_2\}}$

Verification phase

Compute $\mathbf{c_1} = \text{DeComp}(\bar{\text{c}}_1)$;
$\mathbf{v_1} = \text{Decode}(\mathbf{c_1} - \mathbf{u_1} \cdot \mathbf{s})$;
$\mu_1 = \mathcal{H}_0(\mathbf{v_1})$, $ID_i^* = AID_i \oplus \mathcal{H}_0(\mu_1||\mathbf{u_1})$;
Check $ID_i^* = ID_i ? SUM : SUM + 1$, $SUM \leq m_1$?
Compute $p = \mathcal{H}_0(\mathbf{s}||ID_i||T_{reg})$;
$Auth_i = \mathcal{H}_0(\mu_1||u_2||\sigma_2)$;
$M_i^* = \mathcal{H}_0(\mu_1||p||AID_i||CRP_i)$;
Check $M_i^* = M_i$?; **V1.**

Compute $t_i'||p' \oplus Auth_i' = CRP_i \oplus \mathcal{H}_0(\mathbf{u_1}||\mu_1)$;
Check $t_i' = t_i$?, $p' \oplus Auth_i' = p \oplus Auth_i$; **V2.**

Choose v_2; $(\mathbf{s_2}, \mathbf{e_2}, \mathbf{e_c} \overset{\$}{\leftarrow} \psi)$;
Compute $\mathbf{a}_2 = G(\sigma_2)$, $\mathbf{d_2} = \mathbf{a_2} \cdot \mathbf{s_2} + \mathbf{e_2}$;
$\mathbf{c_2} = \mathbf{u_2} \cdot \mathbf{s_2} + \mathbf{e_c} + \text{Encode}(\mathbf{v_2})$;
$\mathbf{\bar{c}_2} = \text{Comp}(\mathbf{c_2})$, $\mu_2 = \mathcal{H}_0(\mathbf{v_2})$;
$M_{s1} = \mathcal{H}_1(ID_i||ID_s||\mu_1||\mathbf{d_2}||p||\mu_2)$ **V3.**

$\xleftarrow{\{M_{s1}, \mathbf{d_2}, \mathbf{\bar{c}_2}\}}$

Verification phase:

V4. Compute $\mathbf{c_2'} = \text{DeComp}(\mathbf{\bar{c}_2})$;
$\mu_2 = \mathcal{H}_0(\mathbf{v_2})$;
$\mathbf{v_2} = \text{Decode}(\mathbf{c_2'} - \mathbf{d_2} \cdot \mathbf{s_2})$;
$M_{s1}' = \mathcal{H}_1(ID_i||ID_s||\mu_1||\mathbf{d_2}||p||\mu_2)$;
Check $M_{s1}' = M_{s1}$?
$M_{u1} = \mathcal{H}_2(ID_i||ID_s||\mu_1||\mathbf{d_2}||p||\mu_2)$;

$\xrightarrow{\{M_{u1}\}}$

Compute $M_{u1}^* = \mathcal{H}_2(ID_i||ID_s||\mu_1||\mathbf{d_2}||p||\mu_2)$;
Check $M_{u1}^* = M_{u1}$?; **V5.**

User Key: $sk_u = \mathcal{H}_3(ID_i||ID_s||\mu_1||\mathbf{d_2}||p||\mu_2)$;

Server Key: $sk_s = \mathcal{H}_3(ID_i||ID_s||\mu_1||\mathbf{d_2}||p||\mu_2)$;

Fig. 2. The Log-in phase and the Verification phase of Quantum2FA. After the verification phase, the user and server share a session key.

of password-based Ring-LWE scheme have been proposed [30], [50]. However, in these reconciliation-based PAKE schemes, if the public key pair is reused by one party, there exists an attack to recover this party's secret keys by plenty of queries to this party [38], [39]. In the context of key reuse, the adversary can play the roles of both party A and party B[1], the former is called signal leakage while the latter is called key mismatch. Inspired by this, the similar key mismatch approach [40], [51], [52] has been applied to the encryption-based PAKE. However, reusing the public-key pair is unavoidable in some applications such as the 0-RTT in TLS 1.3 and smart card-based authentication.

Gao *et al.* [53] proposed a key-reuse mode that adds a randomized error polynomial to the sent signal to resist the key reuse attack, but their improvement can not be readily applied to the smart card-based password authentication scenario. There are two reasons: 1) Two additional polynomial multiplications are added to every key exchange in [53]; 2) The polynomial multiplication operation accounts for most of the computation costs of the Ring-LWE implementations on resource constrained smart card as shown in [54], and the added two multiplications may bring heavy burden to the card. In our proposed scheme, we employ the modified encryption-based approach to achieve authenticated key exchange without bringing additional computation and communication costs to the smart card.

3 Evaluation Framework

In this section, we introduce the evaluation metric and the underlying security model (i.e., capabilities of the adversary). Generally, a practical scheme in the quantum era should meet the following two important criteria:

1) *Quantum security*: the proposed 2FA scheme should be based on hard problems that are secure even under attacks from quantum computers and can also provide the quantum safety for both two kinds of authentication factors.
2) *Efficiency*: the proposed 2FA scheme, while using the resource-constrained smart card, should achieve both low computation and communication costs.

Note that, the security and efficiency requirements under conventional computers are still necessary, and new proposed two-factor authentication schemes shall satisfy them. Fortunately, Wang-Wang [5] have proposed a practical evaluation framework for two-factor authentication under conventional computers, which is composed of 12 evaluation criteria. Thus, we use it as our building block and introduce it here.

C1. *No password verifier table*: The server doesn't maintain a database for storing the registered users' password verifier.

C2. *Password friendly*: The user could choose their password freely and change it at any time.

C3. *No password exposure*: Only the smart card contains the owner's password related information, even the server cannot extract any knowledge about the users' password.

C4. *No smart card loss attack*: The scheme is free from smart card loss attack, i.e., if an unauthorized person captures the user's lost smart card, she can't recover the password or impersonate the user.

C5. *Resistance to known attacks*: The proposed scheme should not only resist the attacks which are designed to attack the post quantum key exchange protocols (signal-leakage attack [39] and the key mismatch attack [38]), but also secure under the traditional attack (offline password guessing attack and stolen verifier attack, etc).

C6. *Sound repairability*: Allowing the user to revoke her smart card without changing her identity.

C7. *Provision of key agreement*: The client and the server can establish a shared session key for subsequent secure communication.

C8. *No clock synchronization*: The proposed scheme should avoid the clock synchronization.

C9. *Timely typo detection*: The smart card will check the correctness of user-input password before interacting with the server.

C10. *Mutual authentication*: The client and server ensure they are communicating with each other.

C11. *User anonymity*: The scheme should protect user's identity and user's activities cannot be traced.

C12. *Forward secrecy*: The leakage of long-term keys cannot affect the security of previous sessions.

We consider the capabilities of the adversary as proposed in the widely accepted security models [5], [55]. To be self-contained, we list them as follows:

- The attacker can fully control the communication channel between the smart card and the server, and launching attacks like monitoring of transmitted messages, or trying to modify and delete them. This accords with the standard Dolev-Yao adversary model [56];
- The attacker can enumerate all the possibilities in the Cartesian product of password space and identity space, $D_{ID} \times D_{PW}$, in an offline way. This is realistic because: 1) Identities are not secrets and often can be easily obtained (e.g., account number or email address) [57]; 2) To facilitate memory, user-chosen passwords are often of low-entropy and belong to a small dictionary [58], [59].
- By using a malicious card reader, the attacker could learn a victim's password; by launching side-channel attacks, the attacker may recover all the data stored in a lost smart card [60], [61]. But an attacker cannot first sniff the victim's password and then exact the data stored in the lost card. Otherwise, an attacker can succeed in a trivial way.
- The attacker could get the keys established in the previous sessions (e.g., due to improper erasure).

Although we have seen much progress in finding quantum algorithms with exponential speedup [26], for security concerns, so far the most influential quantum attacks are still those using Shor's algorithm and Grover's algorithm. We will consider them in this paper:

[1] Here, for a key exchange protocol involving two parties, party A means the initiator of the communication (such as Alice in the Diffie-Hellman scheme); Party B means the responder who receives the message (such as Bob in the DH scheme).

- *Quantum attacks using Shor's algorithm*: A large-scale quantum computer can use Shor's algorithm to solve the integer factorization and discrete log problems efficiently.
- *Quantum attacks using Grover's algorithm*: A large-scale quantum computer can use Grover's algorithm to speedup the search of an unstructured database and collision of hash functions.

Note that, a quantum resistance scheme should also be able to thwart known attacks under conventional attackers. we extend the following attacks to the quantum setting:

- *Key reuse attack:* This is a classic attack against public key cryptography, and it is particularly harmful to lattice based key exchange protocols. In the key reuse context, this attack can be divided into two types[2]: the signal leakage attack [39] and the key mismatch attack [38]. 1) For the signal leakage attack, the adversary plays the role of party A (user) with the abilities to initiate a sequence of key exchange sessions with a malformed private key, then she looks for the signal variations sent by party B (server). 2) When launching the key mismatch attack, the adversary plays the role of party B (server) and set her own private key and errors to special values, then she queries with the victim many times. In this way, the adversary can recover the private key of the targeted entity who does not change the public key and executes the key exchange protocol with her.
- *Offline password guessing attack:* This is a basic attack which could be combined with many other attacks. Since users' choices of passwords often follow a highly skewed distribution [58], an adversary has the chance to find the password in the password dictionary and check them against the password hash in an offline way.
- *Online password guessing attack:* This attack is similar to the offline ways. An adversary first insert the smart card into the card reader and choose a password from the password dictionary. Then she needs to interact with the server to verify whether the chosen password is correct.
- *Smart card loss attack*: An adversary may obtain user's smart card by stealing it or picking it up for a relatively long period. During this period, she may try to change the user's password, guess the password by using the offline password guessing attack or impersonate the user to enjoy the service. Also, she could extract the secret information stored in the smart card through side-channel attacks and returns the breached card back to the user without victim's awareness. When the victim uses the breached card to log-in again, the adversary could intercept the message between the user and the server. Combining these data, the adversary may establish an available cryptographic hash of the password and use it to check the password guess result.
- *Replay attack*: An adversary can store the communication data between the server and the user in the previous authentication sessions. When the next authentication session between the user and server occurs, the adversary may replace all or some specific parts of the communication data.
- *Parallel session attack*: An intruder can interleave messages between different sessions: she exchanges messages with the first victim, and then uses some messages that received in the first session to interact with the second victim. Also, she may exploit some messages received in the second session to feedback to the first session.
- *De-synchronization attack*: To provide user untraceability, serialization is used to update user pseudo-identities. An adversary can block or tamper the communication data between the user and the server. By doing this, the serialization process is destroyed, so that users cannot-log in any more.
- *Stolen verifier attack*: An adversary somehow gets the verification data from the server and uses them to generate legal authentication messages or extract the legal user's password.
- *Reflection attack*: This attack aims to trick the target into providing the answer to its own challenge.
- *Impersonation attack*: An adversary combines all above attacks to impersonate a legal user.

In all, our security goals and corresponding attacks are comprehensive, since we not only consider the attacks from traditional computers but also quantum threats.

4 Challenges and the Basic Idea

Among hundreds of "password + smart-card" two-factor authentication protocols, the scheme proposed by Wang-Wang [5] is currently considered state-of-the-art and can achieve truly two-factor security with high efficiency. However, they employ the Diffie-Hellman key exchange technique to establish the session key. Thus their scheme is prone to quantum attacks.

Taking into consideration both quantum resistance and efficiency on smart cards, our main idea is to propose a smart card-based password authentication scheme whose security is based on the hardness of Ring-LWE problem. Besides, this scheme also needs to withstand other known attacks that arise in traditional smart-card-based password authentication. Thus, we further adopt the "fuzzy-verify + honeywords" mechanism [5] to detect and prevent the smart card loss attack. More specifically, the "honeywords" strategy [62], a concept of system security, is used to thwart online password guessing attacks by inserting the wrong authenticator p' (a parameter derived from the user's password) into the $Honey\_List$. The threshold of the $Honey\_List$ is m_0. This means that once the number of p' in the $Honey\_List$ exceeds m_0, the server can infer the occurrence of smart card corruption.

The existence of a quantum connection from solving a hard problem in ideal lattice in the worst case to solving the Ring-LWE problem in the average case guarantees the security of Ring-LWE-based approaches against even quantum computers [44]. Meanwhile, as a ring-variant of LWE, the

[2] Informally, the signal-leakage attack is initiated by A and aim to obtain B's private key information; While the key mismatch attack is initiated by B and aim to obtain A's private key information.

Ring-LWE-based approach could perform well on resource-constraint devices such as smart cards. However, when we try to integrate the lattice-based key exchange with the state-of-the-art smart-card-based password authentication (e.g., [5], [16]), there comes key reuse attacks against the Ring-LWE-based key exchange [38], [39] and we have to overcome them.

Similar to other considered attacks, the key reuse attack is essentially a known attack that shall be avoided. The development of public key cryptography seems to be accompanied by key reuse attacks. Back to 1998, Bleichenbacher first proposes the key reuse attack [63], when considering the chosen-plaintext attack (CPA) security of the RSA PKCS. After that, the Diffie-Hellman key exchange scheme is also affected by the key reuse attack [64], [65]. In the post-quantum era, lattice-based cryptosystem presents great potential, three lattice-based KEMs (i.e., CRYSTALS-KYBER, SABER, and NTRU) out of the NIST third-round list [43]. Unfortunately, none of these three lattice-based KEMs is immune to key reuse attacks (more precisely, key mismatch attacks) [66]. For most of the NIST candidate KEM schemes, a CPA secure proposal is first given, then using the Fujisaki-Okamoto (FO) [67] transformation to convert it into a chosen-ciphertext attack (CCA) secure version. Although the FO transformation, which is essentially re-encryption (also the main computing cost), can resist key mismatch attacks, its efficiency is lower than the CPA version [66]. As a result, key reuse cannot be avoided as an important way to improve efficiency. Besides, in the initialization phase of a smart card-based 2FA scheme, the server needs to generate its public-key information for authentication and then store the information in the smart card. That is, the public key information cannot be changed frequently and is supposed to be reused.

As mentioned above, there are two kinds of lattice-based key exchange: the reconciliation-based and the encryption-based scheme. In the reconciliation-based scheme, to share an exact key, one entity needs to send additional signals, which could be used by the attacker to derive the victim's secret key. Specifically, if the public key pairs are reused, the attacker could recover the secret key of the responder by using signal leakage attacks [39]. Therefore, if we use the reconciliation-based approaches directly, the leakage of one entity's secret keys cannot be avoided. On the other side, as we have noticed before, the improvement in [53] by adding additional two polynomial multiplications is not suitable for the resource limited smart card. Another kind of key reuse attack is key mismatch attacks [38], which could be launched by simply knowing whether the two shared keys match or not. Then, by repeatedly querying the victim, the attacker can finally recover the victim's private key. In all, key reuse will bring disastrous consequences to the lattice-based key exchange scheme. However, the requirements of availability and efficiency make key reuse inevitable. This contradiction brings great challenges to our design of quantum resistant smart-card-based two-factor authentication protocol.

To provide the quantum security, at first, our scheme employs an encryption-based scheme to establish the shared keys. But with the development of research on key reuse attacks, more and more signs indicate that this encryption-based scheme may also suffer from key reuse attack[3]. Therefore, we have to do further design. For potential signal leakage attacks, we set the server as the initiator of the authentication process (party A) and the smart card as the responder (party B). In a signal leakage attack [39], the attacker needs to be the initiator (party A) of the key exchange process, she sets her public-key information as some special values, sends them to the victim (party B), and queries a lot of signals which contains sensitive information about the victim's secret keys. Thus only the server can launch the signal leakage attack, but in our scheme, the server has stored its public-key information in the smart card in advance in the registration phase. This means that Quantum2FA can thwart the possible signal leakage attack.

In order to help the server recover the shared secret used to establish the session key. The smart card sends the encrypted data in our scheme, the possible key mismatch attack can only be launched from the smart card (party B) against the server (party A). To further prevent the key mismatch attack, we bind the user's ID with the shared secret obtained through the key exchange and expand the range of "honeywords" by including the number of shared key mismatches between the smart card and the server. Since the most significant feature of the key mismatch attack is that the adversary has to make a large number of queries to the victim. (Quite recently, Qin *et al.*[66] show the lower bounds on the minimum average number of queries needed for key mismatch attacks. For example, 1568 queries are needed for attacking NewHope-512 KEM and the minimum number of queries is 1183 for NTRU hrss701.) And each query will cause a session termination because the shared keys do not match. Once the number of terminations reaches a predetermined threshold (far less than the minimum number of queries required by the adversary), the server infers that an attack has occurred and stops the service. After that, the server will terminate the authentication and ask the smart card holder to register again. In this way, Quantum2FA could detect and prevent the possible key mismatch attack.

5 Our Proposed Scheme Quantum2FA

We now propose Quantum2FA, a Ring-LWE-based two-factor authentication scheme. It is quantum-resistant and efficient on the smart card. We divide the proposed scheme into four parts: registration phase, log-in phase, verification phase, and password-changing phase. For ease of presentation, some intuitive notations are listed in Table 1 and will be used throughout this paper.

Our scheme is defined and operated in a ring of polynomials denoted by $\mathcal{R}_q = \mathbb{Z}_q[x]/(x^n + 1)$, where q is a positive integer and n is a power of 2. Specifically, $\mathbb{Z}_q$ is a ring in which all the integers are operated modulo q, and each element of $\mathcal{R}_q$ is a polynomial with coefficients selected from $\mathbb{Z}_q$ and all the operations are done modulo $x^n + 1$. Sometimes we treat the polynomial $\mathbf{a} = a_0 + a_1x + \cdots + a_{n-1}x^{n-1}$ in $\mathcal{R}_q$ equally in the vector form $(a_0, a_1, \ldots, a_{n-1})$, here each

[3] For example, many NIST second round candidates have been found suffer from the key mismatch attacks, such as LAC, Kyber, SABER, RQC and HQC [68] and the lattice-based KEM Kyber is now a NIST third round candidate.

TABLE 1
Notations and Abbreviations

Symbol	Description
U_i	i^{th} user
S	remote server
n_0	an integer related to the size of password space
$\mathbf{s}$	the private key for server
E_{sk}	a symmetric encryption using sk as the key
$\oplus$	the bitwise XOR operation
$\vert\vert$	string concatenation
$\Rightarrow$	a secure channel
$\rightarrow$	a common channel
SUM	a counter for key mismatch
$Honey\_List$	a list with bogus passwords
$\mathrm{Encode}(\cdot)$	an encoding function in [41]
$\mathrm{Decode}(\cdot)$	a decoding function in [41]
$\mathrm{(De)Comp}(\cdot)$	a (decompress) compress function in [41]

a_i corresponds to the i-th coefficient of $\mathbf{a}$. To base our security on the hardness of Ring-LWE problem, we also need to define a centered binomial distribution ψ over $\mathcal{R}_q$, which can output relatively small and high-entropy vectors.

5.1 Registration Phase

In this phase, the user U_i registers to the server S, which then issues a smart card for U_i. At the beginning of the registration phase, the server S selects a cryptographically secure Pseudorandom Generator (PRG) $G(\cdot) : \mathcal{K}_G \rightarrow \mathbb{Z}_q^n$ and hash functions which map from $\{0,1\}^*$ to $\{0,1\}^{l_i}$. We denote these hash functions as $\mathcal{H}_i(\cdot)$, where for $i = 0, 1, 2, 3$, l_i represents the length of the output of hash functions. Next, we select an integer $2^4 \leq n_0 \leq 2^8$, which is first used in [5] as a fuzzy-verifier to resist against offline guessing. The server maintains a counter SUM that records the number of session terminations between the server and the smart card to prevent the possible key mismatch attacks. Let $\mathbf{s}$ denote the server's private key which is randomly sampled from ψ and σ_1 denotes a randomly chosen seed, then the user registers to the server S as follows:

R1. The user U_i chooses the identity ID_i and its corresponding password PW_i, as well as a randomly selected string b. Then, U_i sends to S the following information in a secure channel:
$U_i \Rightarrow S : \{ID_i, \mathcal{H}_0(b||PW_i)\}$.

R2. On receiving the registration message from U_i at time T, S first picks a random number t_i and computes $T_i = \mathcal{H}_0((\mathcal{H}_0(ID_i) \oplus \mathcal{H}_0(b||PW_i)) \bmod n_0)$. Then S checks whether U_i is a valid user and this is the first time that U_i registers to the server. If so, S first generates a new entry for U_i and then adds $\{ID_i, T_{reg} = T, t_i, Honey\_List = NULL\}$ to the corresponding database. Else S changes the available entry of U_i from updating T_{reg} to T, t_i to the new t_i, and sets $Honey\_List$ as NULL. Then, S calculates $R_i = \mathcal{H}_0(b||PW_i) \oplus \mathcal{H}_0(\mathbf{s}||ID_i||T_{reg})$ and uses the seed σ_1 to generate a polynomial $\mathbf{a_1} = G(\sigma_1)$. Next, by sampling a random polynomial $\mathbf{e_1}$ from ψ, S can compute $\mathbf{d} = \mathbf{a_1} \cdot \mathbf{s} + \mathbf{e}_1$. Finally, S sends to U_i a smart card containing the following information:
$S \Rightarrow U_i : \{R_i, T_i, T_i \oplus t_i, \mathbf{d}, \sigma_1, n_0, \mathcal{H}_0(\cdot), \cdots, \mathcal{H}_3(\cdot)\}$

R3. With the received smart card SC, U_i enters the string b twice to activate it.

5.2 Log-in Phase

This phase is consisting of the following steps:

L1. After inserting the smart card SC into the card reader, the user U_i types in the identity ID_i^* and the corresponding password PW_i^*.

L2. With ID_i^* and PW_i^*, the smart card SC could calculate $T_i^* = \mathcal{H}_0((H_0(ID_i^*) \oplus \mathcal{H}_0(b||PW_i^*)) \bmod n_0)$ and checks the validity of the pair (ID_i^*, PW_i^*) by evaluating $T_i^* = T_i$ holds or not. If not, the log-in request is denied. If yes, SC randomly samples a secret polynomial $\mathbf{s_1'}$, and error polynomials $\mathbf{e_1'}, \mathbf{e_c'}$ from ψ. Then, SC computes the public polynomial $\mathbf{u_1} = \mathbf{a_1} \cdot \mathbf{s_1'} + \mathbf{e_1'}$. After that SC generates a random string $\mathbf{v_1'}$ to compute $\mathbf{c_1'} = \mathbf{d} \cdot \mathbf{s_1} + \mathbf{e_c'} + \mathrm{Encode}(\mathbf{v_1'})$, and $\mathbf{c_1} = \mathrm{Comp}(\mathbf{c_1'})$, where $\mathrm{Encode}(\mathbf{v_1'}) = (\frac{q-1}{2}\mathbf{v_1'}, \frac{q-1}{2}\mathbf{v_1'}, \frac{q-1}{2}\mathbf{v_1'}, \frac{q-1}{2}\mathbf{v_1'})$; equation $\mathbf{c_1} = \mathrm{Comp}(\mathbf{c_1'})$ means that for each $i = 0, 1, \ldots, n-1$, every $\bar{c}_1[i] = \lceil (c_1'[i] \cdot 8)/q \rfloor \bmod 8$. Then, SC computes $\mu_1 = \mathcal{H}_0(\mathbf{v_1'})$ as the shared secret, and parameters $AID_i = ID_i^* \oplus \mathcal{H}_0(\mu_1||\mathbf{u_1})$, $p = \mathcal{H}_0(\mathbf{s}||ID_i||T_{reg}) = R_i \oplus \mathcal{H}_0(b||PW_i)$, $t_i = (T_i \oplus t_i) \oplus T_i$.

L3. SC picks another seed σ_2 and uses it to generate a new polynomial $\mathbf{a_2} = G(\sigma_2)$. Then, SC randomly selects a secret polynomial $\mathbf{s_2'}$ and an error polynomials $\mathbf{e_2'}$ from ψ to compute $\mathbf{u_2} = \mathbf{a_2} \cdot \mathbf{s_2'} + \mathbf{e_2'}$, which are used to establish a shared key between the server and the smart card. SC goes on computing $Auth_i = \mathcal{H}_0(\mu_1||\mathbf{u_2}||\sigma_2)$, $CRP_i = (t_i||p \oplus Auth_i) \oplus \mathcal{H}_0(\mathbf{u}_1||\mu_1)$, $M_i = \mathcal{H}_0(\mu_1||p||AID_i||CRP_i)$. At the end of this phase, SC send the following information to S:
$SC \rightarrow S : \{AID_i, CRP_i, M_i, \mathbf{u_1}, \mathbf{u_2}, \mathbf{c_1}, \sigma_2\}$

5.3 Verification Phase

With the received messages $\{AID_i, CRP_i, M_i, \mathbf{u_1}, \mathbf{u_2}, \mathbf{c_1}, \sigma_2\}$, the server S will perform as follows:

V1. S first calculates $\mathbf{c_1} = \mathrm{DeComp}(\bar{\mathrm{c}}_1)$ and $\mathbf{v_1} = \mathrm{Decode}$ $(\mathbf{c_1} - \mathbf{u_1} \cdot \mathbf{s})$, where $\mathrm{DeComp}(\cdot)$ can be considered as the inverse of the $\mathrm{Comp}(\cdot)$, which means for each $i = 0, 1, \ldots, n-1$, we have $c_1[i] = \lceil (\bar{c}_1[i] \cdot q)/8 \rfloor$. The $\mathrm{Decode}(\cdot)$ can also be seen as the inverse of the $\mathrm{Encode}(\cdot)$, and suppose $\mathbf{c_1} - \mathbf{u_1} \cdot \mathbf{s} = (\mathbf{k_1}, \mathbf{k_2}, \mathbf{k_3}, \mathbf{k_4})$, then each element of $\mathbf{v_1}$ is defined as follows $(j = 0, 1, \ldots, \frac{n}{4} - 1)$:

$$v_1[j] = \begin{cases} 0, & \text{if } \Sigma_{l=1}^{4} |k_l[j] - \lfloor \frac{q}{2} \rfloor| \geq q \\ 1, & \text{else.} \end{cases}$$

Then, server S computes the common secret $\mu_1 = \mathcal{H}_0(\mathbf{v_1})$. After that, S computes $ID_i^* = AID_i \oplus \mathcal{H}_0(\mu_1||\mathbf{u_1})$ and checks whether ID_i^* is equal to the stored ID_i. If not, the session will be interrupted and the counter SUM will be changed to $SUM + 1$. Then, the server will determine if the value of SUM is less than the threshold m_1 (e.g., $m_1 = 10$). If it is greater

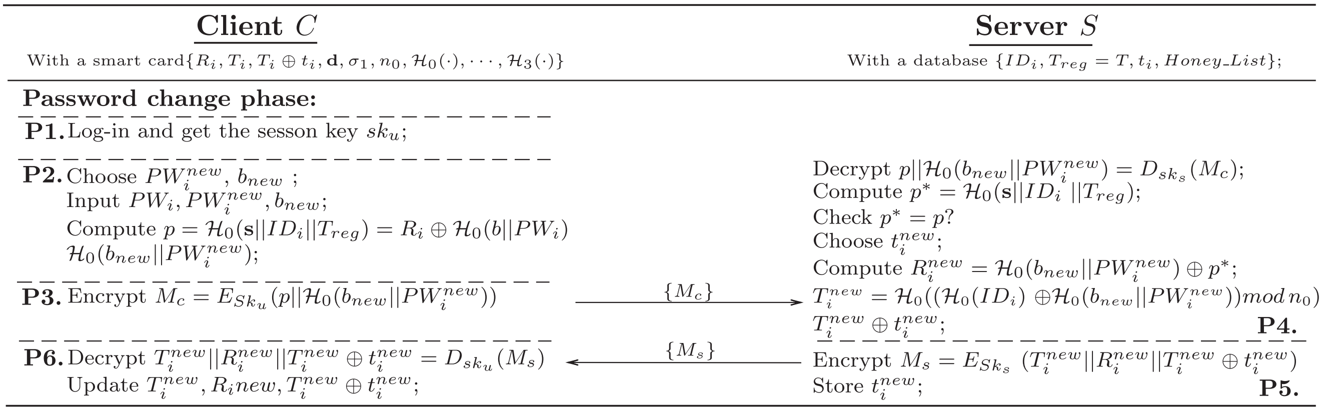

Client C
With a smart card$\{R_i, T_i, T_i \oplus t_i, \mathbf{d}, \sigma_1, n_0, \mathcal{H}_0(\cdot), \cdots, \mathcal{H}_3(\cdot)\}$
Server S
With a database $\{ID_i, T_{reg} = T, t_i, Honey\_List\}$;
Password change phase:
P1. Log-in and get the sesson key sk_u;
P2. Choose PW_i^{new}, b_{new} ;
Input $PW_i, PW_i^{new}, b_{new}$;
Compute $p = \mathcal{H}_0(\mathbf{s}||ID_i||T_{reg}) = R_i \oplus \mathcal{H}_0(b||PW_i)$
$\mathcal{H}_0(b_{new}||PW_i^{new})$;
P3. Encrypt $M_c = E_{Sk_u}(p||\mathcal{H}_0(b_{new}||PW_i^{new}))$
$\{M_c\}$
Decrypt $p||\mathcal{H}_0(b_{new}||PW_i^{new}) = D_{sk_s}(M_c)$;
Compute $p^* = \mathcal{H}_0(\mathbf{s}||ID_i\ ||T_{reg})$;
Check $p^* = p$?
Choose t_i^{new};
Compute $R_i^{new} = \mathcal{H}_0(b_{new}||PW_i^{new}) \oplus p^*$;
$T_i^{new} = \mathcal{H}_0((\mathcal{H}_0(ID_i) \oplus \mathcal{H}_0(b_{new}||PW_i^{new})) mod\, n_0)$;
$T_i^{new} \oplus t_i^{new}$;
P4.
Encrypt $M_s = E_{Sk_s}\ (T_i^{new}||R_i^{new}||T_i^{new} \oplus t_i^{new})$
Store t_i^{new};
P5.
$\{M_s\}$
P6. Decrypt $T_i^{new}||R_i^{new}||T_i^{new} \oplus t_i^{new} = D_{sk_u}(M_s)$
Update $T_i^{new}, R_i new, T_i^{new} \oplus t_i^{new}$;

Fig. 3. Password change phase of the Quantum2FA. In this phase, the user can change her password after logging into the system.

than m_1, the server will stop the service and infer that the probability of the key mismatch attack is $1 - p_0^{m_1}$, where p_0 is the failure probability of the Ring-LWE based key exchange (p_0 is about 2^{-60}, see more details in Section 2.3 of [41]). If yes, S computes $p = \mathcal{H}_0(\mathbf{s}||ID_i||T_{reg})$, $Auth'_i = \mathcal{H}_0(\mu_1||\mathbf{u_2}||\sigma_2)$ and $M_i^* = \mathcal{H}_0(\mu_1||p||AID_i||CRP_i)$ where T_{reg} and ID_i are stored in the entry of the database. If $M_i^* \neq M_i$, the session is terminated. Otherwise, S goes on to the next step.

V2. S computes $t'_i||p' \oplus Auth'_i = CRP_i \oplus \mathcal{H}_0(\mathbf{u_1}||\mu_1)$ and compares t'_i with the stored t_i. Here $t'_i = t_i$ implies that T_i is right, and S rejects the log-in request if they are not equal. If it is right, S goes on checking the equality of the derived $p' \oplus Auth'_i$ and the computed $p \oplus Auth_i$. If they are not equal, S will find that $t'_i = t_i$ but $p' \neq p$. This implies that the smart card may be corrupted. Correspondingly, (1) If the number of the items in the $Honey\_List$ is fewer than m_0(e.g., $m_0 = 10$), S will add p' into the list; (2) If the number of the items in the $Honey\_List$ is reacher than m_0, S treats U$_i$'s card as lost and suspends U_i's card until he re-registers.

V3. S generates a polynomial $\mathbf{a_2} = G(\sigma_2)$, and again. S samples $\mathbf{s_2}$, $\mathbf{e_2}$, and $\mathbf{e_c}$ from ψ and computes $\mathbf{d_2} = \mathbf{a_2} \cdot \mathbf{s_2} + \mathbf{e_2}$. Next, S generates another random key $\mathbf{v_2}$. After this, S computes $\mathbf{c_2} = \mathbf{u_2} \cdot \mathbf{s_2} + \mathbf{e_c} + \text{Encode}(\mathbf{v_2})$, $\mathbf{c_2} = \text{Comp}(\mathbf{c_2})$, and the new shared secret $\mu_2 = \mathcal{H}_0(\mathbf{v_2})$. Finally, S computes $M_{s1} = \mathcal{H}_1(ID_i||ID_s|| \mu_1||\mathbf{d_2}||p||\mu_2)$, and sends to U_i the following: $S \rightarrow U_i$: $\{M_{s1}, \mathbf{d_2}, \mathbf{c_2}\}$

V4. The smart card computes $\mathbf{c'_2} = \text{DeComp}(\mathbf{c_2})$, as soon as receiving the reply message from the server. Then, SC computes the random string $\mathbf{v_2} = \text{Decode}(\mathbf{c'_2} - \mathbf{d_2} \cdot \mathbf{s_2})$. After that, the new shared secret is computed as $\mu_2 = \mathcal{H}_0(\mathbf{v_2})$. Next, SC computes $M'_{s1} = \mathcal{H}_1(ID_i||ID_s|| \mu_1||\mathbf{d_2}||p||\mu_2)$ and compares it with the received M_{s1}. If M'_{s1} = M_{s1} holds, the smart card has authenticated the server S. Finally, U_i computes M_{u1} = $\mathcal{H}_2(ID_i|| ID_s||\mu_1||\mathbf{d_2}||p||\mu_2)$, and the smart card sends it to S the following: $U_i \rightarrow S : \{M_{u1}\}$.

V5. On receiving $\{M_{u1}\}$, S computes $M_{u1}^* = \mathcal{H}_2(ID_i ||ID_s||\mu_1||\mathbf{d_2}||p||\mu_2)$ and then checks if M_{u1}^* equals the received M_{u1}. If this verification holds, S has authenticated the user U_i and the login request is accepted, else the connection is terminated.

V6. Finally, the user U_i and the server S agree on a common session key $sk_u = \mathcal{H}_3(ID_i||ID_s||\mu_1||\ \mathbf{d_2}||p||\mu_2) = sk_s$ and they would use it for future data communications.

5.4 Password Change Phase

When the user U_i needs to change her password, the password change phase is performed as follows:

P1. U_i logs into the server S successfully and agrees on a session key sk_U with it.

P2. U_i chooses a new password PW_i^{new} and a new random string b_{new}, then she inputs PW_i, PW_i^{new} and b_{new} to the card. With these new parameters, the smart card computes $p = \mathcal{H}_0(\mathbf{s}||ID_i||T_{reg}) = R_i \oplus \mathcal{H}_0(b||PW_i)$ and $\mathcal{H}_0(b_{new}||PW_i^{new})$.

P3. The smart card uses the session key sk_u to encrypt the password-changing message $\{p, \mathcal{H}_0(b_{new}|| PW_i^{new})\}$ and sends the encrypted message $M_c = E_{sk_u}(p||\mathcal{H}_0(b_{new}||PW_i^{new}))$ to server S.

P4. S decrypts the received message with the shared session key sk_s and computes $p^* = \mathcal{H}_0(\mathbf{s}||ID_i\ ||T_{reg})$. Then S compares the received p with p^*. If $p \neq p^*$, S denies the password change requirement. Otherwise, S picks a new random number t_i^{new} and computes $T_i^{new} = \mathcal{H}_0((\mathcal{H}_0(ID_i) \oplus \mathcal{H}_0(b_{new}||PW_i^{new})) \bmod n_0)$, $T_i^{new} \oplus t_i^{new}$ and $R_i^{new} = \mathcal{H}_0(b_{new}||PW_i^{new}) \oplus p^*$.

P5. The server S encrypts the new parameter $M_s = E_{sk_s}$ $(T_i^{new}||R_i^{new}||T_i^{new} \oplus t_i^{new})$ and replaces t_i with the newly chosen t_i^{new} in the database. Then, S sends the message M_s to the smart card.

P6. The smart card receives the message and decrypts it. Then, it updates R_i, T_i and $T_i \oplus t_i$ with R_i^{new}, T_i^{new} and $T_i^{new} \oplus t_i^{new}$, respectively.

5.5 Correctness

The correctness of the verification phase is elaborated as follows. At the log-in phase, the user U_i inputs her identity ID_i^* and password PW_i^*. To resist offline password guessing attacks and achieve timely typo detection, the smart card SC first checks the equation $T_i^* \overset{?}{=} T_i$, where $T_i^* = \mathcal{H}_0((H_0(ID_i^*) \oplus \mathcal{H}_0(b||PW_i^*)) \bmod n_0)$ and T_i is stored in the smart card SC. But the establishment of the equation does not mean that the user has entered the accurate password (maybe $\mathcal{H}_0(b||PW_i^*) \equiv \mathcal{H}_0(b||PW_i) \bmod n_0$), due to the use of the "fuzzy-verifier". Therefore, another parameter R_i stored in the smart card is used to check whether the user holds an accurate password PW_i. SC calculates $p^* = R_i \oplus \mathcal{H}_0(b|| PW_i^*) = \mathcal{H}_0(\mathbf{s}||ID_i||T_{reg})$, where $\mathbf{s}$ is the server's private key, ID_i and T_{reg} are stored in the server's database during the registration phase. The server will calculate $p = \mathcal{H}_0(\mathbf{s}|| ID_i||T_{reg})$ immediately when she receives U_i's log-in request and checks the availability of received p^*. Only by entering the accurate password PW_i^* can SC provide the correct p. Thus, the server verifies the user's password.

Our proposed scheme preforms the key exchange twice in the log-in phase and verification phase, respectively. In the log-in phase, smart card chooses a shared secret v_1' and encrypts it as $\mathbf{c_1'} = \mathbf{d} \cdot \mathbf{s_1} + \mathbf{e_c'} + \mathrm{Encode}(\mathbf{v_1'})$. After that, SC sends the compressed cipher $\bar{c_1}$ to server S. Then, S decompresses c_1 and obtains the shared secret by calculating $\mathbf{v_1} = \mathrm{Decode}(\mathbf{c_1} - \mathbf{u_1} \cdot \mathbf{s})$. At last, both of the two parties derive the final shared key μ_1 by hashing the same v_1'. The second key exchange performs in the verification phase which is similar to the first round, and generates μ_2 as the second shared key. Next, in the verification phase V4, SC computes $M_{s1}' = \mathcal{H}_1(ID_i||ID_s||\mu_1||\mathbf{d_2}||p||\mu_2)$ and compares it with the received M_{s1} if $M_{s1}' = M_{s1}$, the user will authenticate the server. Similarly, the server will authenticate the user by comparing $M_{u1}^* = \mathcal{H}_2(ID_i||ID_s||\mu_1||\mathbf{d_2}||p||\mu_2)$ with the received M_{u1}, after this, we achieve the mutual authentication and establish the same session key $sk_U = \mathcal{H}_3(ID_i||ID_s|| \mu_1||\mathbf{d_2}||p||\mu_2) = sk_S$.

6 Security Analysis

In this section, we give a formal security analysis of our proposed Quantum2FA. First, we introduce the formal definition related to the hardness of the Ring-LWE problem, after that, we introduce two attacks that against the LWE/Ring-LWE problem. Finally, we detail the formal security model and give the security proof.

6.1 The Ring-LWE Problem

Recall that $\mathcal{R}_q = \mathbb{Z}_q[x]/(x^n + 1)$ and ψ is a discrete distribution over $\mathcal{R}_q$. We assume that for a security parameter l as its inputs, there is an algorithm $\mathcal{G}$ which outputs (R, R_q, q, n, ψ) according to l. By randomly selected $\mathbf{a} \overset{R}{\leftarrow} R_q$, $\mathbf{s}, \mathbf{e} \overset{R}{\leftarrow} \psi$, and $\mathbf{x}, \mathbf{y} \overset{R}{\leftarrow} R_q$, similar to the Diffie-Hellman assumptions, the decision Ring-LWE problem is to distinguish $(\mathbf{a}, \mathbf{b} = \mathbf{as} + \mathbf{e})$ from $(\mathbf{x}, \mathbf{y})$. More formally, we have:

Definition 1. *Set $\mathcal{R}_q$ and ψ as defined above, if for any probabilistic polynomial-time (PPT) adversary $\mathcal{A}$, the advantage that*

$$Adv_{\mathcal{G}}^{DRLWE}(\mathcal{A}) = |\Pr[\mathcal{A}(R_q, q, n, \chi, \mathbf{a}, \mathbf{b}) = 1] \\ -\Pr[\mathcal{A}(R_q, q, n, \chi, \mathbf{x}, \mathbf{y}) = 1]|$$

is negligible, where $\mathbf{a} \overset{R}{\leftarrow} R_q$, $\mathbf{s}, \mathbf{e} \overset{R}{\leftarrow} \psi$, and each pair $(\mathbf{x}, \mathbf{y})$ is randomly chosen from $R_q \times R_q$. Then we say the decision Ring-LWE problem is hard with $\mathcal{G}$.

6.2 The Quantum Resistance

The security of lattice-based schemes is often depended on the hardness of finding a relatively short vector in the lattice. The BKZ [69] is an algorithm to find short vectors in an n-dimensional lattice. Based on BKZ, there exist two attacks: primal attack and dual attack that against the LWE/Ring-LWE problem.

- *Primal attack:* The primal attack consists of constructing a unique-SVP instance from the LWE problem and solving it using BKZ. By examining how large the block dimension b is required to be for BKZ to find the unique solution, we can demonstrate the security of the Ring-LWE based scheme. The success condition is:>

$$\varsigma\sqrt{b} \leq \delta^{2b-d-1} \cdot q^{m/d}.$$

- *Dual attack:* The dual attack consists of finding a short vector in the dual lattice $\mathbf{w} \in \Lambda' = \{(x, y) \in \mathbb{Z}^m \times \mathbb{Z}^n : \mathbf{A^t x} = \mathbf{y} \bmod q\}$, that is, a short pair $(\mathbf{v}, \mathbf{w})$ such that $\mathbf{v^t A} = \mathbf{w} \bmod q$. The dual attack then uses this as a distinguisher for LWE. The length l of a vector given by the BKZ algorithm is given by $l = \|b_0\|$. To obtain an ε-distinguisher requires running BKZ with block dimension b where

$$-2\pi^2\tau^2 \geq \ln(\varepsilon/4).$$

If an attacker needs an advantage of at least 1/2 to significantly decrease the search space of the agreed key, he must be repeated at least R times where

$$R = max(1, 1/(2^{0.2075b}\varepsilon^2)).$$

See more details in the Section 6.3 and 6.4 of [48], the resistance to these attacks illustrates the quantum security of our Ring-LWE based key exchange scheme.

6.3 Formal Security Model

This section basically models the communications between several players over a channel which can be fully controlled by a PPT adversary $\mathcal{A}$. The players are expected to agree on the session key over this channel and then use it for data transmission. Therefore, to reflect adversary's capabilities, some queries, such as Send-query, Execute-query and Test-query, are given to $\mathcal{A}$ and they are used to model different attacks in the real communication scenarios.

Players. There are a number of participants in a two-factor protocol $\mathcal{P}$, namely, a user $U \in \mathcal{U}$ and a server $S \in \mathcal{S}$. Each of them has several instances called oracles. The user and server instances are denoted by U^i and S^j $(i, j \in \mathbb{Z})$, respectively, and the notation I is represented as any kind of such instances (i.e., $I \in \mathcal{U} \cup \mathcal{S}$).

Passwords and Long-Term Keys. Each user $U \in \mathcal{U}$ with identity ID_u owns a password PW_u which is considered to be drawn from a fixed, non-empty password dictionary $\mathcal{D}$ whose size is $|\mathcal{D}|$. As shown by wang *et al.* [58], the password dictionary obeys Zipf's distribution.Besides, S stores some non-sensitive data and a few necessary public parameters as well as a long-term private key PRK.

Queries. The following queries model the adversary's capabilities in real attacks. Only through the oracle queries does the interaction between the adversary $\mathcal{A}$ and instance I occur.

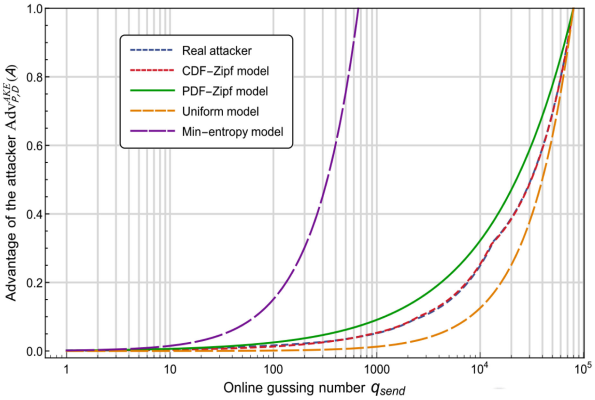


Fig. 4. Online guessing advantages of a real attacker, the uniform-modeled attacker, the min-entropy modeled attacker, the PDF-Zipf modeled attacker and CDF-Zipf modeled attacker (using the 97,415 passwords leaked from Qutar National Bank [72]). Our CDF-Zipf attacker overlaps with the real attacker, indicating a well prediction.

- Send(I, m): We use this query to model the active attacks. When the adversary $\mathcal{A}$ calls this query, a message m will be sent to instance I which will process the message m according to the protocol $\mathcal{P}$. Then the instance I updates its state and gives the output to $\mathcal{A}$.
- Execute(U^i, S^j): We use this query to model the passive attacks, such as eavesdropping. When the adversary $\mathcal{A}$ calls this query the protocol $\mathcal{P}$ will be executed completely and the messages that exchanged between U^i and S^j will be output.
- Test(I): This query is used to define session key's semantic security and can be called only once during the game. When the adversary $\mathcal{A}$ calls this query a bit b will be flipped. If $b = 1$, then the actual session key sk of the instance I is returned to $\mathcal{A}$; otherwise, a random key of the same size is returned.
- Reveal(I): When the adversary $\mathcal{A}$ calls this query the session key sk of instance I will be returned to $\mathcal{A}$.
- Corrupt(I, c): When $\mathcal{A}$ calls this query, one of the two authentication factors of users will be output to $\mathcal{A}$.

- If $I = U, c = 1$, outputs all the security parameters that are stored in the smart card.
- if $I = U, c = 2$, outputs the password PW_U of U.
- if $I = S, c = 1$, outputs the private key PRK and the data stored in S's database.

Partnering. For a pair of instances U^i and S^j, we use the notations sid and pid to denote the session identifier and partner identifier, respectively. U^i and S^j are partnered if they satisfy the following conditions: ① Both instances have accepted. ② $sid_{U^i} = sid_{S^j} = sid$. ③ $pid_{U^i} = S$ and $pid_{S^j} = U$.

Freshness. Capturing the notion of forward secrecy requires freshness to be carefully defined around the corrupt query. We say that an instance I is fresh if: ① I has accepted and computed a session key. ② I and its partner hasn't been asked for a Reveal-query. ③ At most one kind of corrupt query is made to U from the beginning of the game.

Definition of Security. We now turn to actually measuring the probability that the adversary could succeed in breaking $\mathcal{P}$. To tell apart a random string from an instance's true session key sk, $\mathcal{A}$ can ask a polynomial number of Execute-query, Reveal-query and Send-query. At the end of the game, $\mathcal{A}$ outputs a guess bit b' for the bit b involved in the Test-query. Let $Succ(\mathcal{A})$ be the event that $b' = b$, then $\mathcal{A}$'s advantage in winning the game is defined as:

$$Adv_{\mathcal{P}}^{AKE}(\mathcal{A}) = 2Pr[Succ(\mathcal{A})] - 1 = 2Pr[b' = b] - 1$$

As shown in [5], [70], [71], for a two-factor scheme, it is desirable that online password guessing attack shall be $\mathcal{A}$'s best possible strategy to impersonate a party. So, the two-factor protocol $\mathcal{P}$ is said to be semantically secure if for any PPT adversary $\mathcal{A}$ making q_{send} on-line attacks, there exists a negligible function ϵ such that

$$Adv_{\mathcal{P},\mathcal{D}}^{AKE}(\mathcal{A}) \leq C' \cdot q_{send}^{s'} + \epsilon,$$

where $\mathcal{D}$ is the password space which follows Zipf's distribution [58], c' and s' are the Zipf parameters.

Remark 1. In most existing PAKE studies (e.g., [31], [32], [73]) and other kinds of password-based protocols (e.g., two-factor authentication [74] and password authenticated keyword search [75]), passwords are assumed to follow a uniformly random distribution, and the real attacker's advantage Adv is thus formulated as $q_{send}/|D| + \epsilon$, where $|D|$ is the size of the password dictionary D, and q_{send} is the number of $\mathcal{A}$'s active on-line password guessing attempts (which is analogous to q_s in [73], [74], [75]). Instead, we prefer the CDF-Zipf model [5], [58], and the attacker's advantage Adv can be formulated as $C' \cdot q_{send}^{s'} + \epsilon$ for the Zipf parameters C' and s'. Fig. 4 shows that the traditional uniform-model based formulation $q_{send}/|D| + \epsilon$ always significantly underestimates the real attacker's Adv ($\forall q_{send} \in [1, |D|]$). Fortunately, the CDF-Zipf based formulation $C' \cdot q_{send}^{s'} + \epsilon$ well approximates the real attacker's Adv: $\forall q_{send} \in [1, |D|]$, *the largest deviation* between $C' \cdot q_{send}^{s'} + \epsilon$ and Adv is as low as 0.491%. This CDF-Zipf based formulation is drastically more accurate than other occasionally used formulations like the Min-entropy model in [76].

6.4 The Security Proof

Here we show that an adversary $\mathcal{A}$ attacking Quantum2FA is unable to determine the session key sk of a fresh instance (i.e., with no greater advantage than just performing online guessing). For simplicity, the goal of forward-secrecy is not considered here. Our security reduction games are based on the work of Wang-Wang [5].

First, we introduce the *decision Diffie-Hellman-like problem*[4], and a similar problem is introduced in [34] to tell the difference between the tuples containing a truly established key and that with a randomly selected string. More formally we have:

Definition 2. *Set $\mathcal{R}_q$ and ψ the same as in Definition 1, then decision DH-like problem is hard with $\mathcal{G}$, if for a PPT adversary $\mathcal{A}$, the probability that*

[4] Diffie-Hellman-like problem introduced in [34]. Essentially, this is a quantum-resistant hard problem and the hardness of whcih is based on the decision Ring-LWE problem.

$$\begin{aligned} Adv_{\mathcal{G}}^{\text{DDH-like}}(\mathcal{A}) = |&Pr[\mathcal{A}(a,b,b',\bar{c},\mu_2)=1] \\ &- Pr[\mathcal{A}(a,b,b',\bar{c},\mu_2')=1]| \end{aligned}$$

is negligible. Here in the tuple $(a,b,b',\bar{c},\mu_2)$*,* μ_2 *is the actual key computed by the parties, while* μ_2' *is a random string selected from* $\{0,1\}^{\frac{n}{4}}$*.*

Theorem 1. *Let* $\mathcal{D}$ *be a password space that follows Zipf's law,* c' *and* s' *be the the Zipf parameters. Our proposed scheme is denoted by* $\mathcal{P}$*. Let PPT adversary* $\mathcal{A}$ *against the semantic security with a time bound* t *and he makes* q_{send}*,* q_{exe} *queries of type Send, Execute, respectively, and* q_h *queries to the random oracles. For* $t' = t + (q_{send} + q_{exe} + 1)\cdot \tau_m$*, where* τ_m *is the computation time for a multiplication in the ring* $\mathcal{R}_q$*.*

$$Adv_{\mathcal{P},\mathcal{D}}^{AKE}(\mathcal{A}) \le c'\cdot q_{send}^{s'} + 12q_h Adv_{\mathcal{G}}^{\text{DDH-like}}(t') + 2Adv_{\mathcal{G}}^{PRG}(t) + \frac{{q_h}^2 + 6q_{send}}{2^l} + \frac{(q_{send}+q_{exe})^2}{q}.$$

Proof. To begin, we define the following games starting at the real attack game G_0 and ending with G_8. For each game $G_n (n = 0, 1, \cdots, 8)$ we define the following events:

- $Succ_n$ occurs if $\mathcal{A}$ correctly guesses the bit b involved in the Test-query.
- $AskP_n$ occurs if $\mathcal{A}$ correctly computes the parameter p by asking a hash query $\mathcal{H}_0$ on $b||PW_i$ or $\mathbf{s}||ID_i||T_{reg}$.
- $AskM_n$ occurs if $\mathcal{A}$ correctly computes the parameter p and asks a hash query $\mathcal{H}_1$ (or $\mathcal{H}_2$) on $ID_i||ID_s||\mu_1||\mathbf{d_2}||p||\mu_2$.
- $AskH_n$ occurs if $\mathcal{A}$ correctly asks a hash query $\mathcal{H}_i$ $(i = 1, 2, 3)$ on $ID_i||ID_s||\mu_1||\mathbf{d_2}||p||\mu_2$.

Game G_0: This game corresponds to the real attack, in the random oracle model. By definition, we have:

$$Adv_{\mathcal{P}}^{AKE}(\mathcal{A}) = 2Pr[Succ_0] - 1 \tag{1}$$

Game G_1: In this game, we use a truly random string to replace the PRG $G(\cdot)$, which means that the polynomial $\mathbf{a} \xleftarrow{R} \mathbb{Z}_q^n$ instead of $\mathbf{a} \leftarrow G(\sigma)$. The two games G_1 and G_0 are almost the same except the impact of PRG:

$$|Pr[Succ_1] - Pr[Succ_0]| \le Adv_{\mathcal{G}}^{PRG}(t) \tag{2}$$

Game G_2: There are two possible conditions that the collision may appear:

- collisions on the partial transcripts $((AID_i, CRP_i, M_i), M_{s1}, M_{u1})$.
- collisions on the outputs of hash queries.

In this game we exclude the impact of the collision and both probabilities are bounded by the birthday paradox:

$$|Pr[Succ_2] - Pr[Succ_1]| \le \tfrac{(q_{send}+q_{exe})^2}{2q} + \tfrac{{q_h}^2}{2^{l+1}} \tag{3}$$

where $l = min\{l_i\}, i = 1, 2, 3.$

Game G_3: There are some probabilities that the adversary may obtain the correct authenticator M_{s1} or M_{u1} without asking the corresponding hash query $\mathcal{H}_1$ or $\mathcal{H}_2$:

$$|Pr[Succ_3] - Pr[Succ_2]| \le \tfrac{q_{send}}{2^l} \tag{4}$$

Game G_4: We define this game by aborting the game wherein the adversary may have lucky in guessing the correct parameter p (i.e., without asking the corresponding query $\mathcal{H}_0$):

$$|Pr[Succ_4] - Pr[Succ_3]| \le \tfrac{q_{send}}{2^l} \tag{5}$$

Game G_5: We define this game by aborting the game wherein the adversary may have computed the correct parameter p and impersonate as a client or server.

The two games G_4 and G_5 are indistinguishable unless the event $AskP_5$ occurs:

$$|Pr[Succ_5] - Pr[Succ_4]| \le Pr[AskP_5] \tag{6}$$

In order to upper bound $Pr[AskP_5]$, the parameter p is assumed to be correctly computed by the adversary $\mathcal{A}$ in all the following games.

Game G_6: We define this game by aborting the game wherein the adversary may have computed the correct authenticator M_{s1} or M_{u1} (that is, by asking the corresponding hash query $\mathcal{H}_1$ or $\mathcal{H}_2$) and impersonate as a client or a server.

The two games G_6 and G_5 are perfectly indistinguishable unless event $AskM_6$ occurs:

$$|Pr[Succ_6] - Pr[Succ_5]| \le Pr[AskM_6] \tag{7}$$

$$|Pr[AskP_6] - Pr[AskP_5]| \le Pr[AskM_6] \tag{8}$$

Game G_7: In this game, we replace the random oracles $\mathcal{H}_i$ with the private oracles $\mathcal{H}_i'(i = 1, 2, 3)$ and we select a random value k_{ran} which is as the same length as that of μ from $\{0,1\}^n$:

$$\begin{aligned} M_{s2} &= \mathcal{H}_1'(ID_i||ID_s||\mathbf{u_1}||\mathbf{d_2}) \\ M_{u2} &= \mathcal{H}_2'(ID_i||ID_s||\mathbf{u_1}||\mathbf{d_2}) \\ Sk_u = Sk_s &= \mathcal{H}_1'(ID_i||ID_s||\mathbf{u_1}||\mathbf{d_2}) \end{aligned}$$

As a result, the values of $M_{s2}, M_{u2}, Sk_u, Sk_s$ are completely independent from p, μ_1. G_6 and G_5 are indistinguishable unless the event $AskH_7$ occurs:

$$|Pr[Succ_7] - Pr[Succ_6]| \le Pr[AskH_7] \tag{9}$$

$$|Pr[AskP_7] - Pr[AskP_6]| \le Pr[AskH_7] \tag{10}$$

$$|Pr[AskM_7] - Pr[AskM_6]| \le Pr[AskH_7] \tag{11}$$

In order to bound $Pr[AskP_7]$, we need to introduce the following Lemma: □

Lemma 1. *The probabilities of the events $Succ_7$ and $AskP_7$ in this game can be up-bounded by:*

$$|Pr[Succ_7]| = \tfrac{1}{2}, |Pr[AskP_7]| \leq \tfrac{1}{2}C' \cdot q_{send}^{s'} + \tfrac{q_{send}}{2^{l_0}} \tag{12}$$

Proof. In the game G_7, the session keys are computed with private hash oracle unknown to $\mathcal{A}$, and thus $Pr[Succ_7] = \frac{1}{2}$. Since we have avoided the cases where $\mathcal{A}$ have been lucky in guessing p, $\mathcal{A}$ can correctly compute p with the help of either a $Corrupt(I = U^i, 1)$-query or a $Corrupt(I = U^i, 2)$-query. The probability of which is denoted by $Pr[AskP_7withCorr_1]$ ($Pr[AP_7C_1]$ in short) and $Pr[AskP_7withCorr_2]$, respectively.

When $\mathcal{A}$ asks the $Corrupt(I = U^i, 1)$-query, she will get the security parameters stored in the U^i's smart card. In order to get the authenticator p, $\mathcal{A}$ can randomly choose to attack the user U^i or the server S with the probability of $Pr[U] = Pr[S] = \frac{1}{2}$. If $\mathcal{A}$ chooses to attack S, she cannot get the correct p (i.e., $Pr[AP_7C_1S] = 0$), since all of the random oracles are private and she only has the parameters in the smart card. If $\mathcal{A}$ chooses to attack U^i, she can get the p by guessing the correct password. As a result, $Pr[AP_7C_1U] = C' \cdot q_{send}^{s'}$. The $Pr[AP_7C_1]$ is calculated as follows:

$$\begin{aligned} |Pr[AP_7C_1]| &= |Pr[AP_7C_1|U] + Pr[AP_7C_1|S]| \\ &= |Pr[AP_7C_1U] \cdot Pr[U] + Pr[AP_7C_1S] \cdot Pr[S]| \\ &\leq C' \cdot q_{send}^{s'} \cdot \frac{1}{2} + 0 \cdot \frac{1}{2} = \frac{1}{2}C' \cdot q_{send}^{s'} \end{aligned} \tag{13}$$

Thus, we have

$$|Pr[AskP_7withCorr_1]| \leq \frac{1}{2}C' \cdot q_{send}^{s'} \tag{14}$$

$$|Pr[AskP_7withCorr_2]| \leq \frac{q_{send}}{2^{l_0}} \tag{15}$$

Game G_8: In this game, we calculate the parameters $M_{s2}, M_{u2}, Sk_u, Sk_s$ in the same way as that in the game G_7. So we have

$$Pr[AskH_8] = Pr[AskH_7] \tag{16}$$

Notice that $AskH_8$ means that the Adversary $\mathcal{A}$ had queried the random oracles $\mathcal{H}_i (i = 1, 2, 3)$ on $(ID_i||ID_S|| \mu||\mathbf{d_1}||p||DDH\text{-}like(\mathbf{u_2}, \mathbf{d_2}))$ and he will finish this with time bound $t' = t + (q_{send} + q_{exe} + 1) \cdot \tau_m$. By picking randomly in the Λ_A-list we can get the real μ_2 with probability $\frac{1}{q_h}$, thus

$$Pr[AskH_8] \leq q_h Adv_{\mathcal{G}}^{DDH-like}(t'). \tag{17}$$

From Equations (1)–(17), we can complete our proof:

$$\begin{aligned} Adv_{\mathcal{P}}^{AKE}(\mathcal{A}) &= 2Pr[Succ_7] - 1 + 2(Pr[Succ_0] - Pr[Succ_7]) \\ &\leq C' \cdot q_{send}^{s'} + 12 q_h Adv_{\mathcal{G}}^{\mathrm{DDH-like}}(t') + 2Adv_{\mathcal{G}}^{PRG}(t) \\ &\quad + \frac{{q_h}^2 + 6q_{send}}{2^l} + \frac{(q_{send} + q_{exe})^2}{q}. \end{aligned}$$

□

Next, to analyse the security of the mutual authentication, we have:

Theorem 2. *Let $\mathcal{D}$, c', s' and $\mathcal{P}$ defined the same as that in theorem 1. Suppose that there is a PPT adversary $\mathcal{A}$ trying to destroy mutual authentication in a time bound t and he makes q_{send} Send, q_{exe} Execute, and q_h random orale queries, respectively. Then $\mathcal{A}$'s advantage is bounded by*

$$\begin{aligned} Adv_{\mathcal{P},\mathcal{D}}^{AUTH}(\mathcal{A}) \leq \frac{1}{2}c' \cdot q_{send}^{s'} + 5q_h Adv_{\mathcal{G}}^{\mathrm{DDH-like}}(t') + Adv_{\mathcal{G}}^{PRG}(t) \\ + \frac{{q_h}^2 + 6q_{send}}{2^{l+1}} + \frac{(q_{send} + q_{exe})^2}{2q}. \end{aligned}$$

Here $t' = t + (q_{send} + q_{exe} + 1) \cdot \tau_m$, where τ_m is the computation time for a multiplication in the ring $\mathcal{R}_q$.

This proof is similar to the Theorem 1, so we omit it.

7 Further Security Discussion

In this section, we will evaluate the security of the proposed scheme, based on the evaluation criteria and adversary model in Section 3.

7.1 Quantum Security

In order to show the quantum security of our scheme, we start from the prospective that both of the two factors (i.e., password and security parameters stored in the smart card) are quantum secure. In the R2 of the Registration phase, the server stores $R_i = \mathcal{H}_0(b||PW_i) \oplus \mathcal{H}_0(\mathbf{s}||ID_i||T_{reg})$, SC will calculate $p = R_i \oplus \mathcal{H}_0(b||PW_i) = \mathcal{H}_0(\mathbf{s}||ID_i||T_{reg})$, where $\mathbf{s}$ is the server's private key and PW_i is user's password. Therefore, the parameter p is the combination of two authentication factors, called "two-factor authenticator". As mentioned earlier, the encryption based Ring-LWE key exchange could be used to provide quantum resistance. In the log-in phase, smart card chooses a shared secret $\mathbf{v_1'}$ and encrypts it as $\mathbf{c_1'} = \mathbf{d} \cdot \mathbf{s_1} + \mathbf{e_c'} + \mathrm{Encode}(\mathbf{v_1'})$. After that, SC sends the compressed cipher $\bar{c_1}$ to server S. Then, S decompresses c_1 and obtains the shared secret by calculating $\mathbf{v_1} = \mathrm{Decode}(\mathbf{c_1} - \mathbf{u_1} \cdot \mathbf{s})$. At last, both of the two parties derive the final shared key μ_1 by hashing v_1'. Now, we have finished the first key exchange.

When sending the log-in request to the server, SC hides the "two-factor authenticator" p with the shared key μ_1 by computing $CRP_i = (t_i||p \oplus Auth_i) \oplus \mathcal{H}_0(\mathbf{u}_1||\mu_1)$. Since $\mathbf{s}$ is the server's private key, the quantum adversary cannot calculate p directly unless she obtains both the two authentication factors at the same time, which is not allowed. Our security proof formally demonstrates the above process in the random oracle model (ROM). It should be noted that when talking about the quantum security of a post-quantum cryptographic scheme, an alternative method is to prove its security through quantum random oracle (QROM) [77]. To obtain a concrete system, which has been proven to be secure under the ROM, a concrete hash function is needed to replace the random oracle. In this case, a quantum attacker can evaluate the hash function on quantum states (i.e., "in superposition"), which is different from the classical access.

TABLE 2
Timings of Each Cryptographic Operation

Participants	T_M	T_A	T_E	T_P	T_H	T_B	T_h
User	153 ms	157.5 ms	5.6 ms	10.8 ms	13.8 ms	–	–
Server	1.8 ms	3.8 ms	0.03 ms	0.1 ms	0.02 ms	4.7 ms	4.8 ms

Therefore, the proposal of the QROM is aimed to tackle this situation. Boneh *et al.* [77] also pointed out that in quantum settings, one can gain a polynomial speed-up on the collision search against hash functions by using Grover's algorithm (N elements in time $O(\sqrt{N})$), which increases the vulnerability of the hash function. However, Keccak [78], which has been standardized as SHA3 in FIPS-202, offers a concrete hash function SHAKE. Specifically, SHAKE128 could offer 128-bits of post-quantum security against collisions and preimage attacks [48]. This confirms us that with the advancement of post-quantum research, QROM is not the only way to provide quantum security. Moreover, hash-based cryptosystem has been accepted as one of the candidate technical routes for constructing post-quantum cryptographic schemes [35], [37]. In all, the most efficient constructions in lattice-based cryptography are proved in the random oracle model and there do exist post-quantum schemes, such as GPV [79], proved secure under the QROM, but they were proved secure in ROM. As the first step to quantum safe 2FA, our work follows this route and we will focus on providing security proofs under QROM for the post-quantum schemes in future works.

7.2 Key Reuse Attack

In the registration phase, S computes his public key $\mathbf{d} = \mathbf{a_1} \cdot \mathbf{s} + \mathbf{e}_1$ and stores it in U_i's smart card. To perform the user authentication, this is inevitable and results the key reuse attack. For encryption-based key exchange scheme, the signal that reveals the private key information refers to the cipher $\mathbf{c_1}$. In the log-in phase, SC calculates $\mathbf{c_1} = \mathrm{Comp}(\mathbf{c_1'})$, since the adversary must be the signal receiver, in this way, the adversary can only be the person who sends the signal. Additionally, in the log-in phase, SC computes $AID_i = ID_i^* \oplus \mathcal{H}_0(\mu_1 || \mathbf{u_1})$ where μ_1 is the shared key. When receiving the log-in request, S checks the shared key's validity by verifying whether $ID_i^* = ID_i$?. If it is found invalid, S will abort this session and record the number of failures in the counter SUM, we set the threshold $m_1 = 10$ which is much lower than the actual attack (e.g, Qin *et al.* [66]., 1,568 queries). Since the calculation of AID_i only related to μ_1 and the low failure probability of the original scheme (see Section 5.3), the failure of the verification indicates the occurrence of the key mismatch attack.

7.3 Traditional Security Goals

The server only maintains a table $\{ID_i, T_{reg}\}$ of U_i (C1 and C3). U_i chooses his own password and can change it in the password change phase (C2 and C6). By employing the "fuzzy verify" $T_i = \mathcal{H}_0((\mathcal{H}_0(ID_i) \oplus \mathcal{H}_0\ (b||PW_i)) \bmod n_0)$, we can solve the smart card loss problem (C4 and C9) and offline dictionary attack (C5). To avoid clock synchronization (C8), a nonce based mechanism is designed to provide the freshness of the message. In the verification phase V4, SC computes $M'_{s1} = \mathcal{H}_1(ID_i||ID_s||\mu_1||\mathbf{d_2}||p||\mu_2)$ and compares it with the received M_{s1}, if $M'_{s1} = M_{s1}$, the user will authenticate the server. Similarly, the server will authenticate the user by comparing $M^*_{u1} = \mathcal{H}_2(ID_i||ID_s\ ||\mu_1||\mathbf{d_2}||p||\mu_2)$ with the received M_{u1}, after this, we achieve the mutual authentication (C10). We use session-variant pseudo-identity AID_i to achieve user anonymity (C11). To achieve forward secrecy (C12), the DH-like key exchange is adopted. The session key established at the end of the verification phase is $sk_U = \mathcal{H}_3(ID_i||ID_s||\mu_1||\mathbf{d_2}||p||\mu_2) = sk_S$, where $\mu_2 = \mathcal{H}_0(\mathbf{v_2})$ and $\mathbf{v_2}$ is a random string that changes with each new session. As a result, our scheme can provide forward secrecy.

8 Performance Evaluation

We now evaluate the performance of our scheme Quantum2FA. Our scheme performs computations in the ring $\mathcal{R}_q = \mathbb{Z}_q[x]/(x^n + 1)$ with the dimension $n = 512$ and $q = 12289$. We use the centered binomial distribution ψ with parameter $k = 8$, which is obtained by setting $\psi = \Sigma_{i=1}^{8} b_i - b'_i$, where b_i and b'_i are randomly selected from $\{0, 1\}$.

8.1 Performance Comparison

We implement our proposed scheme on a ATxmega128A1 micro controller which is equipped with a 128 Kb FLASH program memory, and a 8 Kb SRAM. The micro controller supports an 128-bit AES hardware crypto-accelerator and can offer a maximum clock speed of 32 MHz. Our server is a laptop which is equipped with an Intel Core i7 processor running at 2.7 GHz and an 8 GB RAM.

To speedup the polynomial multiplications, we use the NTT (Number Theorem Transform), which is a widely used technique and optimized with LUT-based optimization techniques in [86]. We compare the performance of Quantum2FA with eight state-of-the-art relevant schemes [5], [16], [80], [81], [82], [83], [84], [85]. To give an intuitive grasp of the resource consumption of each scheme, in the following we present the main computational time. Let T_B, T_h, T_H, T_A, T_M, T_E and T_P denote the time complexity for bilinear-pairing, map-to-point hash, hash, modular exponentiation, elliptic curve point multiplication, block encryption and polynomial multiplication, respectively. Other lightweight operations such as additions are omitted. We list the computation time for related cryptographic operations on different platforms in Table 2. The comparison results are summarized in Table 3. Meanwhile, Table 3 shows that Quantum2FA not only satisfies all the 12 evaluation criteria proposed by Wang-Wang [5], but also enjoys quantum security. While all the other schemes are short of some important criteria.

8.2 Practicality Discussion

As illustrated in Tables 2 and 3, our scheme Quantum2FA is efficient on both of the user and server sides. This attributes to the lightweight lattice-based operations which guarantees the quantum security while providing high efficiency. The main time-consuming operation of the RLWE-based cryptosystem is coefficient-wise multiplication of two polynomials and this can be accelerated by the NTT. Besides, many scholars are committed to proposing more optimized methods to achieve better performance on various platforms [86], [87], [88], [89].

TABLE 3
Evaluation of Our Scheme With Comparison Among Relevant Authentication Schemes

Scheme	Year	Ref.	User side	Server side	Cost	No verifier table (C1)	Password friendly (C2)	No password exposure (C3)	No smart card loss problem (C4)	Resistance to known attacks (C5)	Sound repairability (C6)	Provision of key agreement (C7)	No clock synchronization (C8)	Timely typo detection (C9)	Mutual authentication (C10)	User anonymity (C11)	Forward secrecy (C12)	Quantum Security
Xie et al.	2017	[16]	$3T_A + 6T_H$	$3T_A + 2T_E + 5T_H$	$566.9ms$	✓	✓	✓	✓	✓	✓	✓	×	×	✓	✓	✓	×
Zhang et al.	2017	[80]	$3T_E + 6T_M$	$3T_E + 8T_M$	$949.3ms$	✓	×	×	✓	×	✓	✓	✓	×	✓	×	✓	×
Das et al.	2018	[81]	$3T_M + 15T_H$	$2T_M + 16T_H$	$669.8ms$	✓	✓	✓	✓	×	✓	✓	×	✓	✓	×	✓	×
Wang et al.	2018	[5]	$3T_A + 9T_H$	$3T_A + 7T_H$	$608.2ms$	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×
Li et al.	2019	[82]	$2T_A + 9T_H$	$4T_A + T_B + T_h$	$463.9ms$	✓	✓	✓	✓	×	✓	✓	✓	✓	✓	✓	✓	×
Das et al.	2019	[83]	$2T_A + 9T_H$	$2T_A + 8T_H$	$446.9ms$	✓	✓	×	×	×	×	✓	×	✓	✓	✓	✓	×
Amin et al.	2020	[84]	$2T_E + 6T_H + 2T_M$	$2T_M + 5T_H + 2T_E$	$403.8ms$	×	✓	×	✓	×	✓	✓	×	×	✓	×	×	×
Majid et al.	2020	[85]	$2T_M + T_E + 10T_H$	$2T_M + 3T_E + 7T_H$	$453.4ms$	✓	✓	✓	×	×	✓	✓	×	×	✓	✓	✓	×
Our scheme	2021	–	$4T_P + 12T_H$	$3T_P + 9T_H$	$209.3ms$	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

[†] *The details of criteria C1∼C12 are referred to Sec. 3;* [‡] *Note that "✓" means achieving the corresponding goal, while "×" not.*

For example, Seo *et al.* proposed a modular reduction method in their work [86], which is a generic approach for any primes for lattice-based schemes. Furthermore, lattice-based cryptosystems have been the most promising choice in the post-quantum era candidate algorithm [43]. Seo *et al.* [86] optimize NTT and random sampling operations on low-end 8-bit AVR micro controllers and point out that "code-based cryptography and multivariate-based cryptography have the long key problem. Hash-based cryptography has long signature problem. Isogeny-based cryptography has long execution timing. Among them, the lattice-based cryptography is considered as one of the most promising candidates for post-quantum cryptography due to its reasonably small key size, small cipher text size, and short execution timing". All this indicates that Ring-LWE based cryptosystems are most promising to be implemented efficiently on resource-constrained devices such as IoT devices and smart cards. We highlight that our Quantum2FA also provides a design framework for two-factor protocols with quantum security in the context of key reuse attack, implying that with the advancement of post-quantum cryptography standards and the improvement of smart card performance, our solution can serves as a guideline for future work.

9 Conclusion

In this paper, we have proposed Quantum2FA, a secure and efficient smart-card-based password authentication scheme for the mobile devices. As far as we know, Quantum2FA is the first 2FA scheme that is secure against attacks from both quantum and conventional computers. We have found that the main challenge in designing Ring-LWE based 2FA schemes lies in how to thwart the recently proposed key reuse attack against lattice-based key exchanges, because existing countermeasures cannot be readily applied to 2FA schemes. By extending the usage of "honeywords" [72] and adapting the modified Alkim et al's [41] encryption-based approach. We have figured out an effective solution to this problem. Besides, Quantum2FA also provides a design framework for quantum-resistant 2FA in the context of key reuse. We highlight that a desirable smart-card-based password authentication scheme should not only satisfy the widely-accepted 12 evaluation criteria proposed by Wang-Wang [5], but also meet two additional, important goals (i.e., quantum secure and high efficiency). Considering the rapid development of quantum-computing, we believe that achieving practical two-factor authentication is of broad interest, and our work constitutes an important step forward in this direction and will spark interest for new quantum-resistant 2FA research.

References

[1] M. Bond, O. Choudary, S. J. Murdoch, S. Skorobogatov, and R. Anderson, "Chip and skim: Cloning EMV cards with the pre-play attack," in *Proc. IEEE Symp. Secur. Privacy*, 2014, pp. 49–64.

[2] X. Huang, Y. Xiang, E. Bertino, J. Zhou, and L. Xu, "Robust multi-factor authentication for fragile communications," *IEEE Trans. Depend. Sec. Comput.*, vol. 11, no. 6, pp. 568–581, Nov./Dec. 2014.

[3] Z. Yang, C. Jin, Y. Tian, J. Lai, and J. Zhou, "LIS: Lightweight signature schemes for continuous message authentication in cyber-physical systems," in *Proc. ACM ASIA Conf. Comput. Commun. Secur.*, 2020, pp. 719–731.

[4] C. Chang and T. Wu, "Remote password authentication with smart cards," *IEEE Proc. E-Comput. Digital Techn.*, vol. 138, no. 3, pp. 165–168, 1991.

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

[6] P. Gope, A. K. Das, N. Kumar, and Y. Cheng, "Lightweight and physically secure anonymous mutual authentication protocol for real-time data access in industrial wireless sensor networks," *IEEE Trans. Ind. Inform.*, vol. 15, no. 9, pp. 4957–4968, Sep. 2019.

[7] X. Li, W. Qiu, D. Zheng, K. Chen, and J. Li, "Anonymity enhancement on robust and efficient password-authenticated key agreement using smart cards," *IEEE Trans. Ind. Electr.*, vol. 57, no. 2, pp. 793–800, Feb. 2010.

[8] H. Sun, "An efficient remote use authentication scheme using smart cards," *IEEE Trans. Consum. Electron.*, vol. 46, no. 4, pp. 958–961, Nov. 2000.

[9] M. Hwang and L. Li, "A new remote user authentication scheme using smart cards," *IEEE Trans. Consum. Electron.*, vol. 46, no. 1, pp. 28–30, Feb. 2000.

[10] I. Liao, C. Lee, and M. Hwang, "A password authentication scheme over insecure networks," *J. Comput. Syst. Sci.*, vol. 72, no. 4, pp. 727–740, 2006.

[11] M. Carbone *et al.*, "Deep learning to evaluate secure RSA implementations," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2019, no. 2, pp. 132–161, 2019.

[12] J. Coron, "Resistance against differential power analysis for elliptic curve cryptosystems," in *Proc. Int. Workshop Cryptographic Hardware Embedded Syst.*, 1999, pp. 292–302.

[13] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks - Revealing the Secrets of Smart Cards*. Berlin, Germany: Springer, 2007.

[14] T. Roche, V. Lomné, C. Mutschler, and L. Imbert, "A side journey to titan," in *Proc. USENIX Secur. Symp.*, 2021, pp. 231–248.

[15] G. Yang, D. S. Wong, H. Wang, and X. Deng, "Two-factor mutual authentication based on smart cards and passwords," *J. Comput. Syst. Sci.*, vol. 74, no. 7, pp. 1160–1172, 2008.

[16] 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. Inform. Foren. Secur.*, vol. 12, no. 6, pp. 1382–1392, Jun. 2017.

[17] C. Ma, D. Wang, and S. Zhao, "Security flaws in two improved remote user authentication schemes using smart cards," *Int. J. Commun. Syst.*, vol. 27, no. 10, pp. 2215–2227, 2014.

[18] K. Hussain, N. Jhanjhi, H. Mati-ur Rahman , J. Hussain, and M. H. Islam, "Using a systematic framework to critically analyze proposed smart card based two factor authentication schemes," *J. KSU. Comput. Inf. Sci.*, vol. 33, no. 4, pp. 417–425, 2019.

[19] E. Erdem and M. T. Sandıkkaya, "Otpaas—One time password as a service," *IEEE Trans. Inform. Forensics Secur.*, vol. 14, no. 3, pp. 743–756, Mar. 2019.

[20] J. W. Byun, "A generic multifactor authenticated key exchange with physical unclonable function," *Secur. Commun. Netw.*, vol. 2019, pp. 5935292:1–5935292:15, 2019.

[21] M. Gupta and N. S. Chaudhari, "Anonymous two factor authentication protocol for roaming service in global mobility network with security beyond traditional limit," *Ad Hoc Netw.*, vol. 84, pp. 56–67, 2019.

[22] X. Li, J. Peng, M. S. Obaidat, F. Wu, M. K. Khan, and C. Chen, "A secure three-factor user authentication protocol with forward secrecy for wireless medical sensor network systems," *IEEE Syst. J.*, vol. 14, no. 1, pp. 39–50, Mar. 2020.

[23] D. Wang, D. He, P. Wang, and C. Chu, "Anonymous two-factor authentication in distributed systems: Certain goals are beyond attainment," *IEEE Trans. Depend. Sec. Comput.*, vol. 12, no. 4, pp. 428–442, Jul./Aug. 2015.

[24] 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 Techn.*, 2018, pp. 456–486.

[25] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proc. Annu. Symp. Found. Comput. Sci., 1994*, pp. 124–134.

[26] J. Preskill, "Quantum computing in the NISQ era and beyond," *Quantum*, vol. 2, pp. 79–99, 2018.

[27] L. Gyongyosi and S. Imre, "A survey on quantum computing technology," *Comput. Sci. Rev.*, vol. 31, pp. 51–71, 2019.

[28] L. Gyongyosi, S. Imre, and H. V. Nguyen, "A survey on quantum channel capacities," *IEEE Commun. Surv. Tutorials*, vol. 20, no. 2, pp. 1149–1205, Sep. 2018.

[29] NIST, "Post-quantum cryptography workshops and timeline," June 2019, [Online]. Available: https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Workshops-and-Timeline.

[30] Z. Li and D. Wang, "Two-round pake protocol over lattices without NIZK," in *Proc. Int. Conf. Inf. Secur. Cryptol.*, pp. 138–159.

[31] J. Zhang and Y. Yu, "Two-round pake from approximate SPH and instantiations from lattices," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, 2017, pp. 37–67.

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

[33] J. Zhang, Z. Zhang, J. Ding, M. Snook, and Ö. Dagdelen, "Authenticated key exchange from ideal lattices," in *Proc. Annual Int. Conf. Theory Appl. Cryptographic Techn.*, 2015, pp. 719–751.

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

[35] G. Alagic *et al.*, "Status report on the first round of the NIST post-quantum cryptography standardization process," 2019. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/ir/2019/NIST.IR.8240.pdf

[36] Y. Yang, H. Lu, J. K. Liu, J. Weng, Y. Zhang, and J. Zhou, "Credential wrapping: from anonymous password authentication to anonymous biometric authentication," in *Proc. ACM ASIA Conf. Comput. Commun. Secur.*, 2016, pp. 141–151.

[37] C. Cheng, R. Lu, A. Petzoldt, and T. Takagi, "Securing the internet of things in a quantum world," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 116–120, Feb. 2017.

[38] J. Ding, S. Fluhrer, and S. Rv, "Complete attack on RLWE key exchange with reused keys, without signal leakage," in *Proc. Australas. Conf. Inf. Secur. Privacy*, 2018, pp. 467–486.

[39] J. Ding, S. Alsayigh, R. Saraswathy, S. Fluhrer, and X. Lin, "Leakage of signal function with reused keys in RLWE key exchange," in *Proc. Int. Conf. Commun.*, 2017, pp. 1–6.

[40] A. Bauer, H. Gilbert, G. Renault, and R. M.,"Assessment of the key-reuse resilience of newhope," in *Proc. Cryptographers' Track RSA Conf.*, 2019, pp. 272–292.

[41] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe, "Newhope without reconciliation," *IACR Cryptol. ePrint Arch.*, vol. 2016, pp. 1157–1166, 2016.

[42] S. Pirandola *et al.*, "Advances in quantum cryptography," *Adv. Opt. Photon.*, vol. 12, no. 4, pp. 1012–1236, 2020.

[43] G. Alagic *et al.*, "Status report on the second round of the nist post-quantum cryptography standardization process," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. 8309, Jul., 2020.

[44] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," in *Proc. Adv. Cryptol.*, 2010, pp. 1–23.

[45] A. Fujioka, K. Suzuki, K. Xagawa, and Y. K, "Strongly secure authenticated key exchange from factoring, codes, and lattices," in *Proc. Public Key Cryptogr.*, 2012, pp. 467–484.

[46] J. Ding, "A simple provably secure key exchange scheme based on the learning with errors problem," *IACR Cryptol. ePrint Arch.*, vol. 2012, pp. 688–703, 2012.

[47] C. Peikert, "Lattice cryptography for the internet," in *Proc. Post-Quantum Cryptogr.*, 2014, pp. 197–219.

[48] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe, "Post-quantum key exchange—A new hope," in *Proc. USENIX Secur. Symp.*, 2016, pp. 327–343.

[49] J. Becerra, V. Iovino, D. Ostrev, P. Sala, and S. M, "Tightly-secure pak(E)," in *Proc. Int. Conf. Cryptol. Netw. Secur.*, 2018, pp. 27–48.

[50] X. Gao, J. Ding, J. Liu, and L. Li, "Post-quantum secure remote password protocol from RLWE problem," in *Proc. Int. Conf. Inf. Secur. Cryptol.*, 2017, pp. 99–116.

[51] Y. Qin, C. Cheng, and J. Ding, "A complete and optimized key mismatch attack on NIST candidate newhope," in *Proc. Eur. Symp. Res. Comput. Secur.*, 2019, pp. 504–520.

[52] S. Okada, Y. Wang, and T. Takagi, "Improving key mismatch attack on newhope with fewer queries," *in Proc. ACISP*, pp. 505–524, 2020.

[53] X. Gao, J. Ding, L. Li, and J. Liu, "Practical randomized RLWE-based key exchange against signal leakage attack," *IEEE Trans. Comput.*, vol. 67, no. 11, pp. 1584–1593, Nov. 2018.

[54] Y. Yuan, C. Cheng, S. Kiyomoto, Y. Miyake, and T. Takagi, "Portable implementation of lattice-based cryptography using javascript," *Int. J. Netw. Comput.*, vol. 6, no. 2, pp. 309–327, 2016.

[55] M. Sookhak *et al.*, "Remote data auditing in cloud computing environments: A survey, taxonomy, and open issues," *ACM Comput. Surveys*, vol. 47, no. 4, pp. 1–34, 2015.

[56] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inform. Theory*, vol. 29, no. 2, pp. 198–208, Mar. 1983.

[57] J. Bonneau, M. Just, and G. Matthews, "What is in a name?" in *Proc. Front. Comput.*, 2010, pp. 98–113.

[58] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, "ZIPF's law in passwords," *IEEE Trans. Inform. Foren. Secur.*, vol. 12, no. 11, pp. 2776–2791, Nov. 2017.

[59] J. Bonneau, "The science of guessing: Analyzing an anonymized corpus of 70 million passwords," in *Proc. IEEE Symp. Secur. Privacy, 2012*, pp. 538–552.

[60] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proc. Adv. Cryptol.*, 1999, pp. 789–789.

[61] T. H. Kim, C. Kim, and I. Park, "Side channel analysis attacks using AM demodulation on commercial smart cards with seed," *J. Syst. Soft.*, vol. 85, no. 12, pp. 2899–2908, 2012.

[62] A. Juels and R. L. Rivest, "Honeywords: making password-cracking detectable," in *Proc. ACM Conf. Comput. Commun. Secur.*, 2013, pp. 145–160.

[63] D. Bleichenbacher, "Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1," in *Proc. Adv. Cryptol.*, 1998, pp. 1–12.

[64] C. Hall, I. Goldberg, and B. Schneier, "Reaction attacks against several public-key cryptosystems," *Proc. Int. Conf. Inf. Commun. Secur.*, vol. 1726, pp. 2–12, 1999.

[65] A. Menezes and B. Ustaoglu, "On reusing ephemeral keys in Diffie-Hellman key agreement protocols," *Int. J. Appl. Cryptogr.*, vol. 2, no. 2, pp. 154–158, 2010.

[66] Y. Qin, C. Cheng, X. Zhang, Y. Pan, L. Hu, and J. Ding, "A systematic approach and analysis of key mismatch attacks on cpa-secure lattice-based NIST candidate Kems," in *Proc. ASIA Adv. Cryptol.*, 2021, pp. 92–121.

[67] E. Fujisaki and T. Okamoto, "Secure integration of asymmetric and symmetric encryption schemes," in *Proc. Annu. Int. Cryptol. Conf.*, 1999, pp. 537–554.

[68] L. Huguenin-Dumittan and S. Vaudenay, "Classical misuse attacks on NIST round 2 PQC," in *Proc. Appl. Cryptogr. Netw. Secur.*, 2020, pp. 208–227.

[69] C. Schnorr and M. Euchner, "Lattice basis reduction: Improved practical algorithms and solving subset sum problems," *Math. Program.*, vol. 66, pp. 181–199, 1994.

[70] E. Bresson, O. Chevassut, and D. Pointcheval, "Security proofs for an efficient password-based key exchange," in *Proc. ACM Conf. Comput. Commun. Secur.*, 2003, pp. 241–250.

[71] P. MacKenzie , "The pak suite: Protocols for password-authenticated key exchange," *IEEE P1363.2*, 2002, [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.20.5299

[72] D. Wang, H. Cheng, P. Wang, J. Yan, and X. Huang, "A security analysis of honeywords," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2018, pp. 1–15.

[73] F. Benhamouda, O. Blazy, C. Chevalier, D. Pointcheval, and D. Vergnaud, "New techniques for SPHFS and efficient one-round pake protocols," in *Proc. Adv. Cryptol.*, 2013, pp. 449–475.

[74] 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 Cryptogr.*, 2018, pp. 431–461.

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

[76] 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.

[77] D. Boneh, Ö. Dagdelen, M. Fischlin, A. Lehmann, C. Schaffner, and M. Zhandry, "Random oracles in a quantum world," in *Proc. ASIA Adv. Cryptology*, 2011, pp. 41–69.

[78] G. Bertoni, J. Daemen, M. Peeters, and G. V. Assche, "Keccak," in *Proc. Int. Conf. Theory Appl. Cryptographic Techn.*, 2013, pp. 313–314.

[79] 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.

[80] R. Zhang, Y. Xiao, S. Sun, and H. Ma, "Efficient multi-factor authenticated key exchange scheme for mobile communications," *IEEE Trans. Dependable Secur. Comput.*, vol. 16, no. 4, pp. 625–634, Jul./Aug. 2019.

[81] 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. Dependable Secur. Comput.*, vol. 17, no. 6, pp. 1133–1146, Nov./Dec. 2020.

[82] W. Li, L. Xuelian, J. Gao, and H. Y. Wang, "Design of secure authenticated key management protocol for cloud computing environments," *IEEE Trans. Depend. Sec. Comput.*, vol. 18, no. 3, pp. 1276–1290, 1 May/Jun. 2021.

[83] M. Karuppiah *et al.*, "Secure remote user mutual authentication scheme with key agreement for cloud environment," *Mobile Netw. Appl.*, vol. 24, no. 3, pp. 1046–1062, 2019.

[84] S. Rajamanickam, S. Vollala, R. Amin, and N. Ramasubramanian, "Insider attack protection: Lightweight password-based authentication techniques using ECC," *IEEE Syst. J.*, vol. 14, no. 2, pp. 1972–1983, Jun. 2020.

[85] D. Abbasinezhad-Mood , S. M. Mazinani, M. Nikooghadam, and A. O. Sharif, "Efficient provably-secure dynamic id-based authenticated key agreement scheme with enhanced security provision," *IEEE Trans. Dependable Secur. Comput.*, early access, Sep. 18, 2020, doi: 10.1109/TDSC.2020.3024654.

[86] H. Seo *et al.*, "Fast number theoretic transform for ring-LWE on 8-bit AVR embedded processor," *Sensors*, vol. 20, no. 7, 2020, Art. no. 2039.

[87] A. Boorghany and R. Jalili, "Implementation and comparison of lattice-based identification protocols on smart cards and microcontrollers," *IACR Cryptol. ePrint Arch.*, vol. 2014, pp. 78–97, 2014.

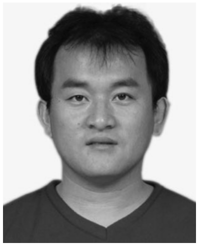
[88] Z. Liu *et al.*, "High-performance ideal lattice-based cryptography on 8-bit AVR microcontrollers," *ACM Trans. Embed. Comput. Syst.*, vol. 16, no. 4, pp. 117:1–117:24, 2017.

[89] T. Pöppelmann, T. Oder, and T. Güneysu, "High-performance ideal lattice-based cryptography on 8-bit atxmega microcontrollers," in *Proc. Int. Conf. Cryptol. Inf. Secur. Latin Amer.*, 2015, pp. 346–365.

Qingxuan Wang received the MS degree in information security from the China University of Geosciences, Wuhan, China, in June 2020. He is currently working toward the PhD degree with the College of Cyber science, Nankai University, Tianjin, China. He has authored or coauthored papers at venues, including *IEEE Transactions on Services Computing* and *Journal of Systems Architecture*. His research interests include applied cryptography and password-based authentication.

Ding Wang received the PhD degree in information security from Peking University in 2017. He was supported by the Boya Postdoctoral Fellowship with Peking University from 2017 to 2019. He is currently a full professor with Nankai University. As the first author or the corresponding author, he has authored or coauthored more than 60 papers at venues, including IEEE S&P, ACM CCS, NDSS, USENIX Security, *IEEE Transactions on Dependable and Secure Computing*, and *IEEE Transactions on Information Forensics and Security*. His research interests include passwords, authentication, and provable security. His research has been reported by more than 200 medias, including Daily Mail, Forbes, IEEE Spectrum and Communications of the ACM, appeared in the Elsevier 2017 Article Selection Celebrating Computer Science Research in China, and resulted in the revision of the authentication guideline NIST SP800-63-2. He was the PC chair or a TPC member for more than 60 international conferences, including PETS 2022, ACSAC 2021/2020, ACM AsiaCCS 2022/2021, IEEE CNS 2020, IFIP SEC 2018-2021, and ICICS 2018-2022. He was the recipient of ACM China Outstanding Doctoral Dissertation Award, Best Paper Award at INSCRYPT 2018, and the First Prize of Natural Science Award of Ministry of Education.

Chi Cheng received the PhD degree in information and communication engineering from the Huazhong University of Science and Technology, Wuhan, China, in December 2013. He is currently an associate professor with the School of Computer Science, China University of Geosciences, Wuhan, China. From 2014 to 2016, he was an international research fellow of the Japan Society for the Promotion of Science, Institute of Mathematics for Industry, Kyushu University, Japan. His research interests include cryptography and information security and especially lattice-based cryptography and its applications.



Debiao He received the PhD degree in applied mathematics from the School of Mathematics and Statistics, Wuhan University, in 2009. He is currently a professor with the School of Cyber Science and Engineering, Wuhan University. His research interests include cryptography and information security, and cryptographic protocols.