QPTA: Quantum-Safe Privacy-Preserving Multi-Factor Authentication Scheme for Lightweight Devices

Basker Palaniswamy[a] and Arijit Karati[b]
Cryptology and Network Security Lab, Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan

Keywords: Authenticated Key Agreement, Privacy, Multi-Factor Authentication, Smart Healthcare, Quantum-Safe Authentication, Post-Quantum Authentication, Attack Detection Logic.

Abstract: Smart healthcare is ubiquitous to lift the convenience of managing patients' medical records. Accessibility of patients' sensitive data stored in medical servers needs source authenticity. To ensure this, (Karati et al., 2023) proposed a three-factor authentication scheme using physical uncolonable functions. However, the scheme is vulnerable to a quantum adversary. To this end, we design a multi-factor authentication scheme called QPTA resistant to quantum adversaries for a healthcare scenario involving a user and a medical server. QPTA enables choice within the same factor in multi-factor authentication. The security of QPTA is formally analyzed using the "Attack Detection Logic." QPTA is safe from known attacks, including unknown key share and man-in-the-middle attacks. We perform an informal security analysis of QPTA to ensure various security goals and privacy properties, namely anonymity, unlinkability, and conditional traceability. QPTA satisfies comprehensive security features and is suitable for the post-quantum era.

1 INTRODUCTION

Smart healthcare enables physicians to manage patients' data in a way that is quickly accessible, and they can access the medical records of patients by storing the records in a medical server and retrieving those data using hand-held equipment, such as smartphones, tablets, etc. Smart healthcare improves patients' convenience as they do not need to save every record produced in the hospitals. Because they are managed systematically within the hospitals. Though data management is most efficiently handled in Smart healthcare for patients and physicians, the stored and retrieved data on the server require security and privacy against attackers. Fig. 1 shows the conceptual overview of the major phases in the proposed protocol wherein a physician (Alice) securely accesses the data of patients from a medical server (Bob). The security and privacy of the medical records are preserved by ensuring the confidentiality, authenticity, integrity, anonymity, and unlinkability of the records. To preserve these goals, physicians should be authenticated to access the medical records in the presence of an adversary who can potentially manipulate medical records. So, an authenticated key agreement protocol is inevitable for Smart healthcare. Even if an adversary is passive, the curious adversary can monitor patients' health conditions. In the future, adversaries can have quantum computers to bypass the security imposed by classical cryptosystems. Considering the evolving threat to medical records, several authentications have been proposed for the smart healthcare scenario (Karati et al., 2023; Qiu et al., 2020; Banerjee et al., 2020; Kwon et al., 2021). Nevertheless, they do not satisfy at least five security goals (ref. Table 1).

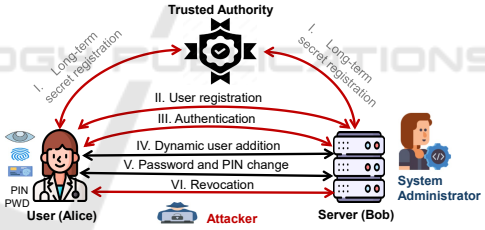

Figure 1: Conceptual overview.

In particular, the work (Banerjee et al., 2020) lacks resistance towards known attacks. Inheriting the same weakness, the research (Qiu et al., 2020) does not allow adding dynamic users. Moreover, the scheme suggests honey lists for handling online password-

[a] https://orcid.org/0000-0002-3661-6048
[b] https://orcid.org/0000-0001-5605-7354

Palaniswamy, B. and Karati, A.
QPTA: Quantum-Safe Privacy-Preserving Multi-Factor Authentication Scheme for Lightweight Devices.
DOI: 10.5220/0012835100003767

In *Proceedings of the 21st International Conference on Security and Cryptography (SECRYPT 2024)*, pages 804-811
ISBN: 978-989-758-709-2; ISSN: 2184-7711

guessing attacks. Whenever the honey lists exceed the threshold value, authorized servers block users. This introduces denial of service (DoS) for the users. For instance, an adversary can target all users at random for the attack. The work (Kwon et al., 2021) does not support user revocation. Consequently, if trusted users misbehave in the network, such users cannot be controlled. The security lapse of the work (Kwon et al., 2021) applies to this work. The research (Karati et al., 2023) has suggested a trifactor authentication scheme for Smart healthcare. However, this scheme has at least two disadvantages: lacking multifactor security and vulnerability to a quantum adversary. Generally, while considering the resistance to a quantum adversary and multifactor security, all these schemes do not meet the requirements. Table 1 presents exhaustive security requirements. Thus, we present our contributions in two-fold.

- We design a quantum-safe, multi-factor authentication scheme, QPTA, for a two-party scenario involving a server. It has online password and personal identification number (PIN) changes, user revocation, and dynamic user addition phases.
- We apply the "Attack Detection Logic (ADC)" (Jurcut et al., 2017) to formally analyse QPTA. QPTA is safe from known attacks. Further, informal analysis for various security goals and privacy traits, namely anonymity, unlinkability and conditional traceability confirms the security of QPTA.

The paper is organized as follows. Sections 2 and 3 elaborate on the preliminaries and construction of QPTA. Formal and informal analysis of QPTA is put forth in Section 4. Section 5 presents the efficiency analysis. Section 6 concludes our work.

2 PRELIMINARIES

We describe notations, adversarial models, security goals, and cryptographic primitives at a high level.

2.1 Notations

Let q be a large prime, and let $\mathbb{F}_q = \mathbb{Z}/q\mathbb{Z}$ be the finite field defined by integers bounded by q; it ranges in the limit $[0,\ q-1]$. Let $\mathrm{R_q}$ be the quotient ring formed by $\frac{\mathbb{Z}_q[X]}{f(x)}$, where $f(x) = X^n + 1$ denotes the irreducible cyclotomic polynomial in which n is the power of 2. We use small bold-face to represent polynomials in the text. A polynomial $\mathbf{d} \in \mathrm{R_q}$ is mentioned as $\mathbf{d} = d_0 + d_1x^1 + d_2x^2 + ,...,d_{n-1}x^{n-1}$, where each coefficient of $\mathbf{d}$ is a vector $[\mathbf{d}]_{i \in [0,n]}$. We use $a \longleftarrow_{\$} S/a \in_R \mathbb{Z}_q$ to mention an element (a) is drawn uniformly at random from the set $(S/\mathbb{Z}_q)$. The distribution of the secret polynomial and error polynomial is in $\mathrm{R_{q[C]}}$, where the elements of $\mathrm{R_{q[C]}} \in$ [-C, C]. $\lceil x \rfloor_l = \frac{x-[x]_{2^l}}{2^l}$ denotes l-bit modular rounding of x, where $[x]_{2^l}$ ranges in $(-2^{l-1}, 2^{l-1}]$. $\lceil x \rfloor_l$ is congruent to $x \bmod 2^l$. $\lfloor\ x\ \rfloor$ mentions that x is made to its nearest value. $\|$ is the concatenation operator and $\oplus$ is bitwise exclusive-OR. E and E^{-1} are symmetric encryption and symmetric decryption, respectively. $H(.)$ denotes the hash function. Let n be the security parameter.

2.2 Adversarial Model

We consider extensions to the Dolev-Yao adversarial model (Dolev and Yao, 1983). The adversary can intercept, replay, masquerade, block, and inject new messages. Further, we consider the adversary to be able to obtain past session keys and passwords. The adversary can acquire users' private keys, but it cannot acquire the remaining credentials, such as biometrics and passwords. Also, the adversary is honest-but-curious (HBC) but passive in the HBC role.

2.3 Security Goals

We provide security and privacy goals based on the authenticated key exchange system.

2.3.1 Security Goals

*Entity Authentication (**SG1**):* Using a pre-shared long-term secret, the user and server should identify each other to agree on a short-term secret.
*Session Key Security (**SG2**):* The user and server should derive a matching session key. And the adversary should not know it.
*Implicit Key Authentication (**SG3**):* The user should ensure that it generates the key material.
*Explicit Key Confirmation (**SG4**):* The user must know that the server has accurately derived the session key.
*Known Key Secrecy (**SG5**):* The adversary can not derive remaining session keys by knowing a session key.
*Forward Secrecy (**SG6**):* The leakage of the long-term secret should not expose any past session keys.
*Resilience to Ephemeral Secret Key Leakage (**SG7**):* The leakage of the short-term session-specific secrets should not expose session keys.
*Future Quantum Attack Resilience (**SG8**):* The adversary with quantum computers cannot get session keys.
*Denial of Service Resilience (**SG9**):* Deploying honey lists against online password-guessing attacks should not block the user from accessing data in the server.

*Multifactor Security (**SG10**):* QPTA requires passwords, PINs, fingerprints, iris data, and smart cards for authentication. User can pick among passwords, PINs, fingerprints, and iris data (**SG15**).

2.3.2 Privacy Goals

*Unlinkability (**SG12**):* Any two messages the user sends should not be linked by the adversary.
*Conditional Traceability (**SG13**):* In case of dispute, the trusted authority (TTP) should know the sender identity of every scrambled message.
QPTA offers anonymity (**SG11**) (Karati et al., 2023) and resistance towards known attacks, namely replay, masquerading, main-in-the-middle, password guessing, PIN guessing, stolen smart card verifier and stolen verifier table (**SG14**). **SG20** resists HBC.

2.4 Cryptographic Primitives

QPTA relies on a key derivation function (KDF) (Krawczyk, 2010), physical unclonable function (PUF) (Sahoo et al., 2017), fuzzy extractor (Fan et al., 2023), dynamic accumulator (Benaloh and De Mare, 1993) and a singcrtyption with key encapsulation mechanism (KEM) SETLA-KEM (Gérard and Merckx, 2018). The security of SETLA-KEM depends on the ring learning with error problem (RLWEP) (Lyubashevsky et al., 2010). RLWEP is computationally infeasible to solve using quantum computers.

We use only these algorithms explicitly in QPTA. KDF has two algorithms, namely HKDF.Extract(.) and HKDF.Expand(.). PUF is instantiated as cM-PUF(.). In the fuzzy extractor, FE.Gen(.) is used for generating biometric template creation. The dynamic accumulator has ACC.KeyGen(.), ACC.Acc.Gen(.), ACC.Acc.verify(.), ACC.Acc.Update(.), ACC.WitGen(.) and ACC.Wit.Update(.).

3 CONSTRUCTION

QPTA has six phases: initialization, registration, authentication, authentication credential change, and user revocation. We assume that users and the server have a tamper-proof module that stores long-term secrets securely. Further, the server enforces RBAC to authorize users to access the sensitive data.

3.1 Initialization Phase

We set $q = 2^{25} - 2^{12} + 1$, $n = 512$ ($\{0,1\}^{256}$), $C = 1$, $k = 4$, $pwlist_{i \in \{ID\}} = \emptyset$, $pinlist_{i \in \{ID\}} = \emptyset$ and $hlist_{i \in \{ID\}} = \emptyset$. This work recommends corresponding $AES-256$ and $SHAKE-256$ as the symmetric encryption and hashing algorithms.

3.2 Registration Phase

The registration phase has user registration, smart card registration, and revocation-credential registration sub-phases. We denote the user by $U/Alice$ and Bob does the server by S/Bob interchangeably, and this phase is for registering Alice by Bob.

3.2.1 User Registration

U submits her certificate to S to validate her public key (PK_U). S verifies the public key of U by using the verification key of TTP. S verifies U sufficiently, i.e., verifying the identity proof. U chooses her authentication credentials, namely password (pwd) and PIN (pin) such that the password is an eight-digit data and PIN $\in \mathbb{N}$. S picks a salt $salt \in \{0,1\}^{256}$ and computes $AC_1 = H(ID\|pwd\|salt)$ and $AC_2 = H(ID\|pin\|salt)$. S gets biometrics of U, namely iris (ir) and fingerprint (fp). S gets R_{ir} and R_{fp} using FE.Gen(.). S computes $AC_3 = H(ID\|R_{ir}\|salt)$ and $AC_4 = H(ID\|R_{fp}\|salt)$. S issues a four-digit code word (CW) to U for choosing the preference of authentication credentials. Briefly, CW = "1111" indicates the use of authentication credentials AC_1, AC_2, AC_3 and AC_4. The condition is that out of AC_1 and AC_2, at least one must be chosen by U. Similarly, out of AC_3 and AC_4, at least one must be chosen by U. Public keys and private keys are generated as per the SETLA algorithm. The version for Bob is $\mathbf{a_1}, \mathbf{a_2} \longleftarrow_{\$} \mathrm{R_q}$ (public parameter), $\mathbf{t_{b,1}}$, $\mathbf{t_{b,2}} \in \mathrm{R_q^2}$ (public key) and $\mathbf{s}$, $\mathbf{e_{b,1}}$, $\mathbf{e_{b,2}} \in \mathrm{R_{q[1]}^3}$ (secret key). Similarly, for Alice, TTP generates credentials. The subscript a is carried for Alice. S stores $\langle H(ID)$,$hlist$,AC_1,AC_2,AC_3,AC_4,$pwlist$,$pinlist\rangle$ in its database. Further, S stores $\langle AC_1$,AC_2,AC_3,$AC_4\rangle$ in the hand-held device of the user.

3.2.2 Smart Card Registration

U submits a request for a smart card by choosing a random number ($rnd_1 \in_R \mathbb{Z}_q$)to S. While submitting, the user gives her identity (ID). After getting rnd_1 and ID from U, the server (S) chooses a secret ($sec_1 \in_R \mathbb{Z}_q$). Using $H(.)$, S computes the smart card credentials $SC = H(rnd_1\|ID\|sec_1)$. S ensures that $\langle SC\rangle$ is stored in the smart card.

3.2.3 Revocation-Credential Registration

For U, S uses ACC.KeyGen(.) for generating a symmetric key (hk). Initially, S make revoca-

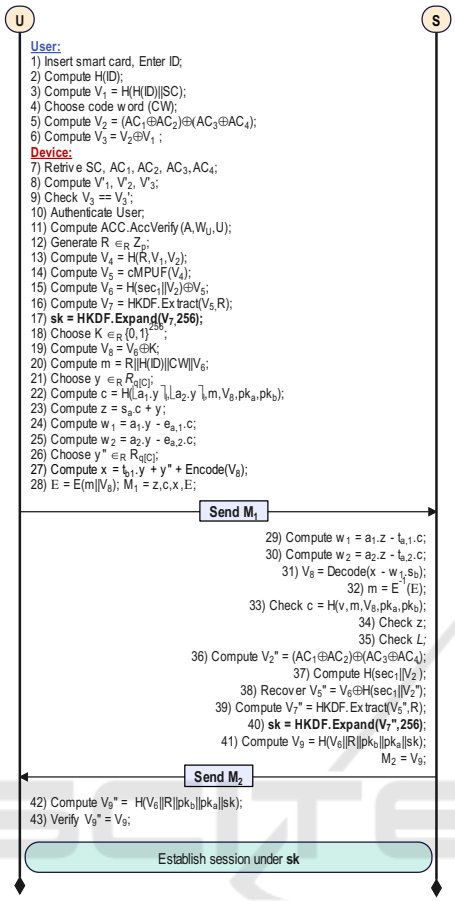

Figure 2: MSC of QPTA.

tion list ($\mathcal{L}$) as null. It generates an accumulator for $\mathcal{L}$ as A using ACC.Acc.Gen(.). The server generates non-membership witness for U as W_U, W_U using ACC.WitGen(.). The server makes A and W_U available on the public ledger. S assigns two variables T_{tol} and T_{exp} for tracking the smart card's tolerance limit and expiry time. S stores $\langle H(ID),AC_1,AC_2,AC_3,AC_4,T_{tol},T_{exp}\rangle$ in its database.

When the registration procedure is complete with the user, S stores $\langle H(ID),sec_1,salt,hk,sk_U,SC\rangle$ in its tamper-proof module. Moreover, $\langle H(ID),AC_1,AC_2,AC_3,AC_4,hk,SC,salt\rangle$ is stored in the tamper-proof module of the user device.

3.3 Authentication Phase

Fig. 2 shows the authenticated key exchange steps between U and S in the message sequence chart (MSC).
User Authentication : The user authenticates with her device, initially. The user inserts her smart card and enters her ID (Step 1). The user computes $H(ID)$ and $V_1 = H(H(ID)||SC)$. According to the preferences, U selects the code word (CW), and computes $V_2 = (AC_1 \oplus AC_2) \oplus (AC_3 \oplus AC_4)$ and $V_3 = V_2 \oplus V_1$. Recall that out of AC_1 and AC_2, U has to choose at least one choice; out of AC_3 and AC_4, U has to select at least one choice. Based on the choice, the code word (CW) is assigned. For example, for all choices, the code word is CW = 1111. Seeing CW, S retrieves $\langle SC,AC_1,AC_2,AC_3,AC_4\rangle$ from its memory. Using SC, the user's device computes $V_1^{'}$. Using authentication credentials, the device calculates $V_2^{'}$. Then, it computes $V_3^{'}$. The device authenticates the user if $V_3 = V_3^{'}$.
$\mathbf{U} \longrightarrow \mathbf{S}$: Upon successful authentication, U executes $ACC.AccVerify(A,W_U,U)$ to check she is not revoked. Once the device authenticates the user, the user performs the following steps by accessing the stored credentials from the device. Hereafter, the device and user are the same. U selects a random number R (nonce) (Step 12) and computes $V_4 = H(R,V_1,V_2)$. Using PUF, the user computes $V_5 = cMPUF(V_4)$. U computes $V_6 = H(sec_1||V_2) \oplus V_5$. U computes $V_7 = HKDF.Extract(V_5,R)$ to derive the session key. U computes the session key $sk = HKDF.Expand(V_7,256)$.At Step 18, U generates K (nonce). To make K sender and receiver bound, Step 19 is done by U as $V_8 = V_6 \oplus K$. Now, the message m is assembled by U as $m = R||H(D)||CW||V_6$. U draws $\mathbf{y}$ uniformly at random from the error distribution. Then, U computes the hash $\mathbf{c} = H(\lfloor a_1.y \rceil_l, \lfloor a_2.y \rceil_l, m, V_8, pk_a, pk_b)$. $\mathbf{z}$ is encrypted as $s_a.c + y$. U computes $\mathbf{w_1} = a_1.y - e_{a,1}.c$ and $\mathbf{w_2} = a_2.y - e_{a,2}.c$. Choosing $\mathbf{y}^{"}$, V_8 is encrypted, where $Encode(V_8)$. Then, m is encrypted under V_8 using symmetric encryption algorithm as $E = E(m||V_8)$. Message $(M_1 = z,c,x,E)$ is transmitted from U to S.
$\mathbf{S} \longrightarrow \mathbf{U}$: Upon receiving M_1 from U, S computes $\mathbf{w_1} = a_1.z - t_{a,1}.c$ and $\mathbf{w_2} = a_2.z - t_{a,2}.c$. S decodes V_8, where $V_8 \in \{0, 1\}$. S gets the message (m) by decrypting E. S checks c; it checks whether z is within the limit of the error distribution. Seeing $H(ID)$, S checks the revocation list ($\mathcal{L}$) to know whether $H(ID)$ is revoked. Observing CW, S computes $V_2^{"} = (AC_1 \oplus AC_2) \oplus (AC_3 \oplus AC_4)$. Retrieving (sec_1), S computes $H(sec_1||V_2^{"})$. S recovers the key material $V_5^{"}$. To derive the session key (sk), S computes $V_7^{"}$. To make sk uniform, S executes $HKDF.Expand(V_7^{"},256)$. As the key confirmation step, S computes $V_9 = H(V_6||R||pk_b||pk_a||sk)$ and makes it M_2. Then it forwards M_2 to U. U checks V_9. If the confirmation is successful, U and S form a secure channel under the session key sk.

3.4 Dynamic User Addition Phase

Whenever a new or existing user approaches S, it executes the following procedure. S checks whether the user is revoked. If the user is revoked, S aborts further process. If not, then it checks the expiry time of the smart card (T_{exp}). Suppose $T_{exp} = True$; then it reissues a new smart card to the user by following the steps in Sections 3.2.1, 3.2.2, and 3.2.3. When the expiry time is $False$, S checks T_{tol} of the user and initiates the procedure in Section 3.5 for changing the password and PIN. This phase contributes to SG17.

3.5 Password and PIN Change Phase

U picks $\langle cw, pwd/pin\ or\ pwd, pin\rangle$ and forwards $E(cw||sk)$ to S. S retrieves cw as $E^{-1}(E(cw||sk))$. S checks $hlist$ and T_{tol} for U. When T_{tol} exceeds its upper limit, S facilitates U the password and PIN via email. S records all passwords and PINs of U in $pwlist$ and $pinlist$, respectively. Whenever any user is targeted for an online password-guessing attack by the attackers, S stores the passwords tried by the attackers in $hlist$ for the user. Moreover, the index of $hlist$ is held in T_{tol}. When T_{tol} is filled, say 100, then S puts the targeted user in a blocklist and informs U that she is temporarily blocked. U can check $\mathcal{L}$ and execute $ACC.Acc.Verify(.)$ to know whether or not she is not blocked permanently. In this way, due to the online password-guessing attack, DoS is handled. For changing biometric credentials (AC_3 and AC_4), U has to contact S in person as it is sensitive information. This phase contributes to SG18.

3.6 Revocation Phase

Whenever the user misbehavior is beyond the threshold value or dies, the server uses $ACC.Acc.Update(.)$ algorithm for updating A as A^* and $ACC.WitUpdate(.)$ for updating W_U as W_U^*. S updates $\mathcal{L}$ s.t. $U \in \mathcal{L}$. With this information, the user can check whether or not she is revoked (Step 11 of Fig. 2). This phase contributes to SG19.

4 SECURITY ANALYSIS

This section uses "Attack Detection Logic (ADC)" to formally analyze QPTA (SG16) (Jurcut et al., 2017).

4.1 Formal Analysis

ADC has a rich set of twenty-two rules in first-order logic under four categories: message freshness, message symmetries, challenge-response handshake construction and signed messages. These rules are built based on twenty-one axioms that have sub-axioms (Jurcut et al., 2017). We discuss only the applicable rules. Further, we use the same rule number as specified in the work (Jurcut et al., 2017) to have a one-to-one correspondence.

4.1.1 Applying ADC

We examine only the applicable rules. Freshness rules R1.1, R1.2 and R1.3 are not activated. Symmetry rules R2.1 is not activated. In signed statement rules, rules R3.1∼3 are not activated. Under the handshake rule, R4.5.1∼2, and R4.7 are not activated. It is proved that freshness attacks and interleaving session attacks are infeasible in QPTA, so it is safe from the known attacks. Precisely, QPTA is secured against the insider threat. Attacks, such as replay, masquerading and man-in-the-middle are infeasible in QPTA.

4.2 Informal Analysis

Proposition 1: *QPTA satisfies known key secrecy.* Adversaries knowing previous session keys cannot compute the current session key because of the requirement of PUF. In addition, every session key is derived as the function of authentication credentials $f(AC_1, AC_2, AC_3, AC_4)$, so without knowing these authentication credentials the adversaries cannot compute the current session key. Further, at S, to derive the current session key, V_5" is required, but, this needs the knowledge of sec_1, so by knowing previous session keys, an adversary cannot derive **sk**.

Proposition 2: *QPTA ensures forward secrecy.* Every session key material V_6 is obfuscated under $H(sec_1||V_2)$ while sending to S, where V_2 is the chosen authentication credentials and sec_1 is stored in the tamper-proof module. Even if the long-term secret of Bob is compromised, the security of the session key is preserved for the current session because of the hiding of the key material under the authentication credentials. Note that the compromise of long-term secrets corresponds to either the secret key of Alice or Bob, but not all secrets, such as authentication credentials. Compromise of all secrets in the tamper-proof module of Bob not only leads to a total break but also paves the way for the exposure of sensitive information (biometric details) of users.

Proposition 3: *QPTA assures resilience to DoS attack.* QPTA tracks password-guessing attacks using $hlist$. $hlist$ makes an entry for every wrong password. Whenever the index of $hlist$ exceeds T_{tol}, the targeted user is blocked for a temporary time using RBAC. The user can confirm this by checking $\mathcal{L}$ and execut-

ing $ACC.Acc.Verify(A, WU, U)$. When the verification result is *true*, U know the fact that she has been blocked for some time, and she can try later or reach the server. QPTA assures resilience to DoS.

Proposition 4: *QPTA assures resilience to leakage of ephemeral secret keys.* Every x in M_1 is encrypted under the public key of Bob, so the leakage of K of every session does not risk the security of QPTA since the adversary is restricted by the requirement of the long-term secret key ($\mathbf{sk_b}$) for decrypting x of M_1.

Proposition 5: *QPTA prevents stolen smartcard-verifier attack.* An adversary with the smart card cannot pass the authentication steps. Despite knowing SC and computing $V_1 = H(H(ID)||SC)$, the adversary cannot find $V_2 = (AC_1 \oplus AC_2) \oplus (AC_3 \oplus AC_4)$ as they are authentication credentials stored in the devise of U. Thus, it resits stolen smart card verifier attacks.

Proposition 6: *QPTA prevents stolen verifier table attack.* An adversary who knows the database of S may use the credentials to violate the authentication of U, however, this attempt is futile for the adversary. To compute sk, the adversary needs $\mathbf{sk_b}$ so that it can recover K. Further, to recover $V_5^{"}$, it requires sec_1 for computing $H(sec_1||V_2^{"})$. Consequently, the adversary is unsuccessful in the stolen verifier table attack.

Proposition 7: *QPTA prevents password and PIN guessing attacks with overwhelming probability.* The adversary may acquire the password and PIN of U via offline guessing attack, nonetheless, due to the requirement of at least one choice out of $Ac_3 \oplus AC_4$ (biometrics) in Step 5, it cannot succeed in the user authentication with probability more than $\frac{1}{2^n}$.

Proposition 8: *QPTA assures anonymity with non-negligible probability.* Let WIN be an event in which the adversary successfully violates the anonymity. Since every M_1 is encrypted under the public key of Bob $\mathbf{pk_a}$ and it is randomized by nonce R, the probability of WIN is bounded by $Pr(WIN) \leq \frac{1}{2^n}$.

Proposition 9: *QPTA ensures unlinkability with non-negligible probability.* The adversary has to link at least any two M_1 and M_2 to violate unlinkability. Because R is a nonce, every time m changes by at least the probability of $1 - \frac{1}{2^n}$, the resulting signcryptext changes with probability $1 - \frac{1}{2^n}$. The adversarial probability of linking any two M_1 and M_2 is $\frac{1}{2^n}$.

Proposition 10: *QPTA ensures conditional traceability.* At Bob, after successful decryption of $\mathbf{x}$, if the verification of $\mathbf{c} = H(v, m, V_8, pk_a, pk_b)$ is *false*, Bob can report M_1 to the trusted third party. The trusted third party can confirm this by computing $\mathbf{w_1} = \mathbf{a_1}.\mathbf{z} - \mathbf{t_{a,1}}.\mathbf{c}$, $\mathbf{w_2} = \mathbf{a_2}.\mathbf{z} - \mathbf{t_{a,2}}.\mathbf{c}$, $V_8 = Decode(\mathbf{x} - \mathbf{w_1}.\mathbf{s_b})$ and $m = E^{-1}(E)$ and verifying $\mathbf{c} = H(v, m, V_8, pk_a, pk_b)$ from M_1. Note that to do this, TTP needs the server's secret key.

5 EFFICIENCY ANALYSIS

We compare computation, transmission, and storage costs and safety traits of QPTA with related schemes W1 (Banerjee et al., 2020), W2 (Qiu et al., 2020), W3 (Kwon et al., 2021) and W4 (Karati et al., 2023).

5.1 Security Features

The work (Qiu et al., 2020) does not hold resilience to stolen data and password attacks, stolen smartcard and password attacks and stolen verifier table attacks. Further, (Banerjee et al., 2020) the work fails to resist stolen data and password attacks (Karati et al., 2023). Notably, Table 1 shows that QPTA satisfies at least five security features more than the existing schemes. QPTA ensures the comprehensive satisfaction of twenty security goals.

5.2 Computation Cost

In Table 2, exclusive-OR operations, and encoding and decoding operations are left because their cost is negligible. Based on Table 2, compared to the most recent work (Karati et al., 2023), QPTA has a higher computation cost, but it attains enhanced security features at the cost of increased computation.

5.3 Communication and Storage Costs

We assume the following: a salt, random number and ID of 16 bytes, biometric data, symmetric key and a hash of 32 bytes, and public and secret key of 256 bytes. In QPTA, S stores $\langle H(ID), AC_1, AC_2, AC_3, AC_4 \rangle$ = 160 bytes in its database. $\langle SC \rangle$ is stored in the smart card, which is 32 bytes. $\langle H(ID), AC_1, AC_2, AC_3, AC_4, hk, SC, salt \rangle$ = 240 bytes is stored in U. $\langle H(ID), sec_1, salt, hk, sk_U, SC \rangle$ = 400 bytes. In total, QPTA consumes 800 bytes for storage. regarding communication cost, M_1 consumes 576 bytes, and M_2 consumes 32 bytes. QPTA consumes 608 bytes as the communication cost. Figure 3 shows the storage and communication costs of all schemes. QPTA is quantum-safe, whereas the remaining schemes are vulnerable to quantum adversaries.

6 CONCLUSION

This paper presented a quantum-safe privacy-preserving multi-factor authentication scheme called QPTA for lightweight devices. QPTA ensured multi-factor security. QPTA facilitated users to choose more

Table 1: Security features comparison.

Schemes	Security features																			
	SG1	SG2	SG3	SG4	SG5	SG6	SG7	SG8	SG9	SG10	SG11	SG12	SG13	SG14	SG15	SG16	SG17	SG18	SG19	SG20
W1 (Banerjee et al., 2020)	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓	✗	✗	✓	✓	✓	✓	♣
W2 (Qiu et al., 2020)	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓	✗	✗	✓	✗	✓	✓	♣
W3 (Kwon et al., 2021)	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓	✓	✗	✗	✓	✓	✓	✗	♣
W4 (Karati et al., 2023)	✓	✓	✓	✗	✓	✓	✓	✗	✗	✗	✓	✓	✓	✓	✗	✗	✓	✓	✓	♣
QPTA	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

✓: Satisfied; ✗: Not satisfied; ♣: Not applicable; SG1: Entity authentication; SG2:Session key security; SG3:Implicit key authentication; SG4:Explicit key confirmation; SG5:Known key secrecy; SG6:Forward secrecy; SG7:Resilience to ephemeral secret key leakage; SG8:Future quantum attack resilience; SG9:Denial of service resilience; SG10:Multifactor security; SG11:Anonymity SG12:Unlinkability; SG13:Conditional traceability; SG14:Resistance to known attacks; SG15:Choice within same factor; SG16:Formal analysis; G17:Dynamic user addition; G18:Online password and PIN change; G19:User revocation; G20:Resilience to HBC.

Table 2: Computation cost comparison.

Schemes	Registeration overhead	Authentication overahead	Total overhead
W1 (Banerjee et al., 2020)	T_{FE}+4T_H	T_{FE}+25T_H	2T_{FE}+29T_H
W2 (Qiu et al., 2020)	T_{FE}+T_{ECM}+4T_H	T_{FE}+6T_{ECM}+19T_H	2T_{FE}+7T_{ECM}+23T_H
W3 (Kwon et al., 2021)	T_{FE}+5T_H	2T_{FE}+T_{PUF}+32T_H	3T_{FE}+T_{PUF}+37T_H
W4 (Karati et al., 2023)	T_{FE}+T_{PUF}+T_{ECM} +T_{SE}+2T_H	T_{FE}+T_{PUF}+2T_{ECM} +2T_{SE}+3T_{SD}+7T_H	2T_{FE}+2T_{PUF}+3T_{ECM} +3T_{SE}+3T_{SD}+9T_H
QPTA	2T_{FE}+4T_{PM}+ 4T_{PA}+4T_H	T_{PUF}+9T_{PM}+7T_{PAS}+ T_{SE}+T_{SD}+15T_H	2T_{FE}+T_{PUF}+13T_{PM}+ 11T_{PAS}+T_{SE}+T_{SD}+19T_H

T_{PUF}: Time to execute one PUF; T_{FE}: Time to run either FE.Gen(.) or FE.Rec(.); T_{ECM}: Time to execute one extended chaotic map; T_{SE}/T_{SD}:Time for one symmetric encryption/one symmetric decryption; T_H: Time for one hash operation; T_{PM}: Time for one polynomial multiplication; T_{PAS}: Time for one polynomial addition/polynomial subtraction.

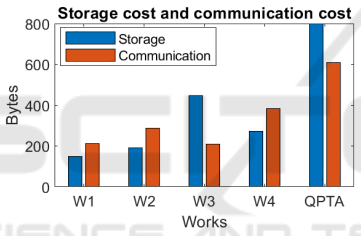

Figure 3: Storage and communication costs.

than one choice within a factor. The security of QPTA relied on the security of the underlying signcryption scheme, physical unclonable functions, key derivation function, dynamic accumulator, and fuzzy extractor. QPTA was formally analyzed using the "Attack Detection Logic." The formal analysis revealed that QPTA is free from freshness attacks and interleaving session attacks. It is free from known attacks, such as replay, masquerading, and man-in-the-middle attacks. QPTA was informally analyzed for various security goals and privacy properties. QPTA assured comprehensive twenty security goals. QPTA is suitable for adoption as it satisfies the pos-quantum security trait.

QPTA offers a security solution for a two-party case (physicians and medical server), neglecting patients' involvement. In the future, we will extend QPTA to a tri-party scenario, where patients' authentic data are stored in the medical server. Further, it will be implemented on a practical testbed mimicking the real-world scenario.

ACKNOWLEDGEMENTS

This work was supported in part by the National Science and Technology Council (NSTC) under grants NSTC 12-2811-E-110-019 and 112-2221-E-110-027.

REFERENCES

Banerjee, S., Odelu, V., Das, A. K., Chattopadhyay, S., and Park, Y. (2020). An efficient, anonymous and robust authentication scheme for smart home environments. *Sensors*, 20(4).

Benaloh, J. and De Mare, M. (1993). One-way accumulators: A decentralized alternative to digital signatures. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 274–285. Springer.

Dolev, D. and Yao, A. (1983). On the security of public key protocols. *IEEE Trans. Inf. Theory*, 29(2):198–208.

Fan, C.-I., Karati, A., and Wu, S.-L. (2023). A privacy-aware provably secure smart card authentication protocol based on physically unclonable functions. *IEEE Trans. Dependable Secure Comput.*, pages 1–13.

Gérard, F. and Merckx, K. (2018). Setla: Signature and encryption from lattices. In *International Conference on Cryptology and Network Security*, pages 299–320. Springer.

Jurcut, A., Coffey, T., and Dojen, R. (2017). A novel security protocol attack detection logic with unique fault discovery capability for freshness attacks and interleaving session attacks. *IEEE Trans. Dependable Secure Comput.*, 16(6):969–983.

Karati, A., Chang, Y.-S., and Chen., T.-Y. (2023). Robust three-factor lightweight authentication based on extended chaotic maps for portable resource-constrained devices. In *Proc. of the 20th International Conference on Security and Cryptography - SECRYPT*, pages 673–682. INSTICC, SciTePress.

Krawczyk, H. (2010). Cryptographic extraction and key derivation: The hkdf scheme. In *Annual Cryptology Conference*, pages 631–648. Springer.

Kwon, D., Park, Y., and Park, Y. (2021). Provably secure three-factor-based mutual authentication scheme with puf for wireless medical sensor networks. *Sensors*, 21(18).

Lyubashevsky, V., Peikert, C., and Regev, O. (2010). On ideal lattices and learning with errors over rings. In *Advances in Cryptology–EUROCRYPT 2010: 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30–June 3, 2010*, pages 1–23. Springer.

Qiu, S., Wang, D., Xu, G., and Kumari, S. (2020). Practical and provably secure three-factor authentication protocol based on extended chaotic-maps for mobile lightweight devices. *IEEE Trans. Dependable Secure Comput.*, 19(2):1338–1351.

Sahoo, D. P., Mukhopadhyay, D., Chakraborty, R. S., and Nguyen, P. H. (2017). A multiplexer-based arbiter puf composition with enhanced reliability and security. *IEEE Trans. on Computers*, 67(3):403–417.