ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 2 - 2026

Design of quantum-resistant lightweight authentication protocol for 6G terminal devices

Jian Zhao[1,*] *and Lisha Xie*[1]

[1] Hunan Vocational College of Railway Technology, Zhuzhou, Hunan, 412006 China

SUMMARY: *The security of current authentication protocols is challenged by quantum attacks, in this regard, this paper designs a security mechanism based on Kyber and Dilithium algorithms to ensure data security, and in order to realize quantum-resistant lightweight authentication, a key negotiation (AKA) protocol is designed to protect the communication between the 6G terminal device and the fog node. The authentication scheme constructed based on Dilithium and Kyber algorithms is tested experimentally, and the experimental results show that the scheme has high security and good performance. The AKA protocol based on Dilithium and Kyber algorithms proposed in this paper is compared and experimented with PSK and ECDHE based on traditional public key certificates, and PBC protocol based on identity, respectively, in terms of communication overhead and key negotiation connection time. Under the premise of guaranteeing security, the number of key negotiation messages to be transmitted by the AKA protocol scheme is 6, which is better than PSK and ECDHE, and the key negotiation connection time of the AKA protocol scheme is 22.61ms, which is significantly better than the other protocol schemes. This scheme significantly reduces the communication overhead and connection delay while ensuring high security, and is suitable for resource-constrained 6G terminal equipment scenarios.*

KEYWORDS: *6G terminal device; quantum attack; lightweight authentication; Dilithium; Kyber; AKA protocol*

1 Introduction

With the popularization of the Internet and digital technology, various types of data and information are stored and transmitted in computer and network systems, which are filled with a large amount of sensitive data and confidential documents. The users and devices in the Internet need to ensure the security of confidential data files through authentication security in the process of communicating and interacting with such sensitive data [1, 2]. Authentication security refers to the process of confirming the user's identity and authorizing his access to the system or resources, which is mainly protected by the authentication protocol of the device. The authentication protocols designed based on the security requirements can prevent unauthorized access, protect the integrity of data and thus improve the reliability and security of the entire device [3, 4].

In recent years, with the gradual commercialization of 5G, researchers around the world have turned their attention to the future 6G research. In the view of many researchers, the entire 6G network is a network driven by artificial intelligence (AI) empowerment, rather than being similar to 5G, which only applies AI technology in some aspects in a limited way, the 6G

*13092787853@163.com
https://doi.org/10.65102/is2026716
Published: 30 April, 2026

network should be a deep integration of the current emerging AI technology and network functions, characterized by “intelligence” [5, 6]. However, if 6G network terminal devices are not protected by proper authentication security, they are vulnerable to network eavesdropping and attacks, which can lead to a large amount of leakage of personal and corporate sensitive information, or even interfere with or disrupt the normal operation of the entire network or system, resulting in serious security and privacy issues [7-10]. Therefore, how to ensure the authentication security of 6G network terminal devices has become a core issue in the field of 6G network security.

In the context of 6G networks, AI and machine learning can be utilized to design systems that appear to be more secure, but in fact the systems are also more vulnerable to attacks.6G network terminal devices are characterized by heterogeneity, dynamics, and diversified access modes, which leads to a climb in the complexity of authentication security, as well as security issues such as quantum threats and network attacks. Je et al. point out that in 6G networks, quantum computing's development directly threatens the security of existing encryption algorithms and may disintegrate the current encryption system of mobile communication [11]. Tom et al. emphasized that quantum computing poses a fundamental threat to cybersecurity, and that the Grover and Shore algorithms can efficiently crack symmetric and asymmetric cryptosystems, respectively, and that the core of the algorithms lies in the use of quantum parallelism to quickly solve classical problems such as integer decomposition and discrete logarithms, which can directly shake the security foundations of the current mainstream cryptographic protocols [12]. In this regard, researchers have conducted research on anti-quantum cryptography. Azmi implemented an anti-quantum cryptographic hash function that combines security, efficiency and scalability, with certain adaptability in post-quantum cryptographic models and its resistance to collision attacks, but the anti-quantum hash still faces challenges [13]. Yu and Huang designed an anti-quantum authentication scheme based on dynamic group signatures to ensure quantum security and utilize non-interactive zero-knowledge proofs to protect user privacy and achieve anonymous cross-chain authentication with low computational overhead [14].

In addition, the authentication protocols for 6G network end devices require lightweight design. Currently, the research on lightweight authentication protocols transitions from traditional lightweight authentication protocols to lightweight authentication protocols for IoT environments, and thereafter enters the research on quantum-resistant lightweight authentication protocols. Reddy and Rao formulated a lightweight authentication protocol for resource-constrained IoT environments by employing hashing and heterogeneous operations, which can effectively resist man-in-the-middle attacks and safeguard the user's anonymity and communication security [15]. Gupta et al. proposed a lightweight authentication protocol and key establishment scheme for smart cities, which ensures two-way authentication between the user and the gateway, taking into account the privacy and security of data transmission, with low computational cost and high energy efficiency [16]. Xu et al. combined aggregated signatures with identity-based encryption to create a lattice-based quantum-resistant lightweight group authentication protocol that eliminates the need for certificate management, effectively resists quantum attacks, and significantly reduces computation, signaling, and communication overheads while safeguarding privacy and security [17]. Tawfeeq et al. design a Dirithium signature-based quantum-resistant lightweight authentication protocol for military UAV networks to provide quantum-secure bidirectional authentication between UAVs, soldiers, and command centers, which is effective against impersonation and side-channel attacks, and has good scalability while dealing with quantum threats [18]. Al-Mekhlafi et al. propose a lattice-based lightweight anti-quantum attack scheme for 5G vehicular networking, which utilizes matrix multiplication instead of traditional bilinear pairs or elliptic curve operations to

efficiently generate and verify signatures, which significantly reduces the computational overhead while guaranteeing security and privacy, and can effectively defend against quantum attacks [19]. Turnip et al. constructed a new type of authentication protocol for 6G networks with quantum resistance by incorporating hybrid post-quantum cryptography based on lattice or hash, AI trust mechanisms, and decentralized authentication, which balances security and efficiency, and provides a feasible transition path to defend against classical and quantum attacks [20]. The study also points out that this protocol has key challenges such as lightweighting and standardization. Therefore, there is a need for continuous quantum-resistant lightweight authentication protocol design for 6G terminal devices.

The research objective of this paper is to design a set of both secure and efficient authentication schemes for massive smart devices in the future 6G era. By analyzing the anti-quantum characteristics of classical authentication protocols, it is proposed to use Dilithium algorithm and Kyber algorithm to construct an anti-quantum authentication scheme. After that, a system model with a trusted third party, a fog node and a 6G terminal device as the architecture is constructed, and a set of anti-quantum lightweight authentication protocol (AKA) for the 6G terminal device is designed by studying the initialization phase, registration phase and AKA phase of the system. Experiments simulate attacks such as replay attack, tampering attack, man-in-the-middle attack, identity forgery attack and denial of service to test the AKA protocol, and verify the security of the protocol by comparing the message transmission overhead and connection delay of different protocol schemes.

2 2. Method

2.1 Lightweight access certification for 6G terminal equipment

The six scenarios of 6G network security requirements include “immersive communication, extremely reliable and low-latency communication, very large-scale communication, ubiquitous connectivity, communication intelligence integration, and communication sensing integration”. 6G safety key technology is the technical basis for meeting the safety requirements of the six 6G scenarios. Relying on the evolution of traditional safety technology and the application of emerging safety key technology, combined with the ubiquitous arithmetic and data resources in the 6G network, it will provide possibilities for the further development and application of safety key technology.

Design lightweight access authentication protocols and processes to realize cross-domain random access for massive 6G network devices and users under multi-authentication regimes, e.g., reduce the complexity of security protocols by adopting lightweight cryptographic algorithms, simplifying the authentication process, compressing the protocol fields, and reducing the number of interactions or the key hierarchy on the basis of the existing user access. For A-IoT devices such as passive/semi-passive, introduce one-way authentication, default security algorithm configuration without security capability negotiation process, etc. Through lightweight access authentication technology, it can reduce the delay caused by processes including identity authentication and security context switching during service access, and solve the security access problem of a large number of low-capability terminals in extremely large-scale communication scenarios.

2.2 Analysis of anti-quantum properties of classical authentication protocols

The main function of the classic authentication protocol is to ensure the security and reliability of user communications, to ensure security, first of all, to ensure that the source security. The core algorithm is asymmetric encryption algorithm represented by RSA, ECC and other algorithms.

In order to design an authentication protocol with anti-quantum security, it is necessary to analyze the core algorithms of the above authentication protocols for anti-quantum, the current classical asymmetric encryption algorithms, symmetric encryption algorithms and hash summary algorithms affected by quantum algorithms as shown in Table 1. Quantum Shor algorithm based on the principle of quantum Fourier transform can be quickly solved in polynomial time, large integer prime decomposition, elliptic curve discrete logarithm and other problems, which will directly affect the security of the classical asymmetric algorithms such as RSA, ECC, and can not be cracked by increasing the length of the key of such algorithms to restore the difficulty. Therefore, for authentication protocols, they must be designed based on new asymmetric cryptosystems in order to make them have anti-quantum security. For symmetric encryption and decryption algorithms and hash digest algorithms, the current quantum computer mainly improves the efficiency of searching for symmetric cryptographic algorithms keys through the Grover algorithm. However, the algorithm can only accelerate the cracking time, and does not directly reduce the cracking time complexity of the algorithm to polynomial level, so the main idea of the anti-quantum design of the content authentication protocol is to expand the key length based on the existing classical cryptographic system, and through the key length of the algorithm to expand, so that the security of the algorithm can be restored to the security level of the classical era in the quantum era to protect its anti-quantum security.

Table 1: Classical encryption algorithms are compared with quantum computing

Break time complexity	Classic scene	Quantum Shor algorithm	Quantum Grover algorithm
RSA	$O(2^n)$	$O(n)$	$O(2^{n/2})$
ECC	$O(2^n)$	$O(n)$	$O(2^{n/2})$
AES	$O(2^n)$	$O(2^n)$	$O(2^{n/2})$
SHA	$O(2^n)$	$O(2^n)$	$O(2^{n/2})$

2.3 Design of anti-quantum authentication protocols

The anti-quantum authentication protocol designed in this paper is divided into three phases: the signature key pair generation phase, the authentication and key negotiation phase and the negotiation key confirmation phase, and the specific design flow is shown in Figure 1. The communicating parties generate signature public-private key pairs and negotiation public-private key pairs in the first phase, which are used as long-term keys for subsequent authentication. In the second stage, userA signs the public key pk and sends it to userB, which then checks the signature to complete the first one-way authentication and encapsulates the key pk. After that, userB signs the ciphertext c and the authentication message Auth and sends it to userA, which then checks the signature to complete the second one-way authentication, and the two-way authentication is completed, and the identity is confirmed. After the identity confirmation, userA decrypts the ciphertext c and obtains the session key k. In the third

stage, UserA compares the hash with the authentication information to determine whether the decryption operation is correct, if the decryption is wrong, send the error message to UserB, and both parties re-authenticate. If there is no error, both parties can generate the negotiation key K from the two signature messages and the session key k.

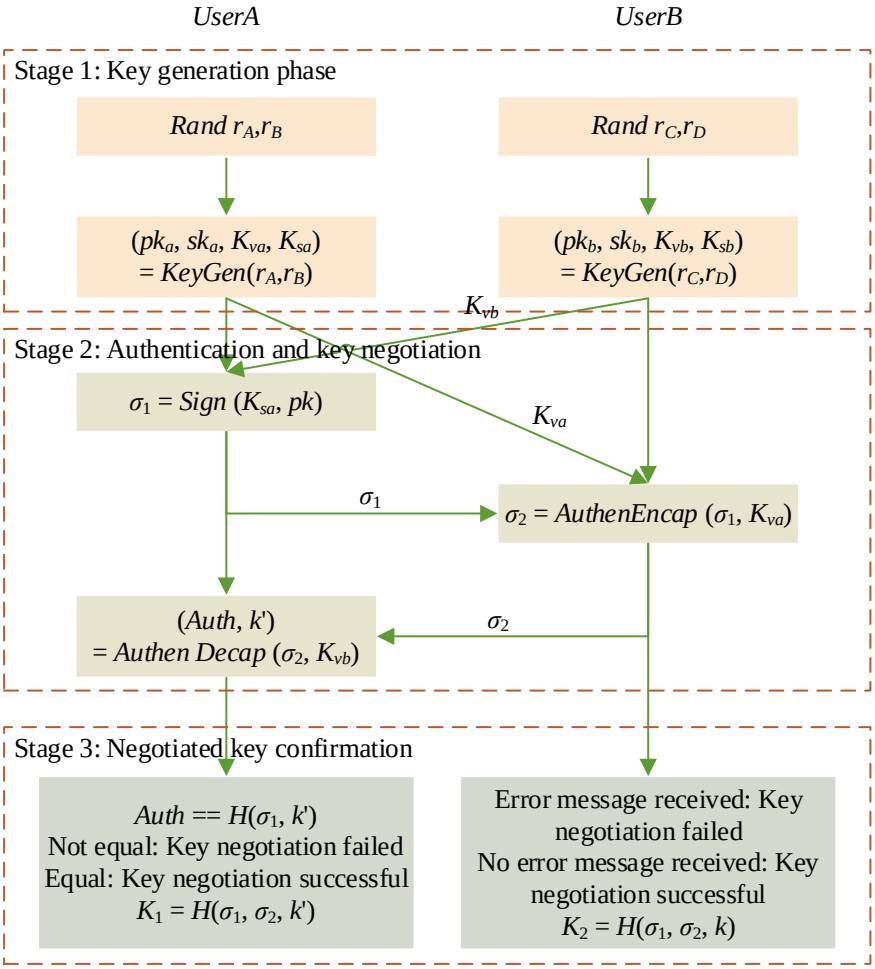


Figure 1: Schematic diagram of the anti-quantum authentication protocol

In this scheme, the authentication key negotiation function constructed based on Dilithium algorithm and Kyber algorithm guarantees the anti-quantum security of the authentication protocol. However, the algorithm lacks a verification mechanism in the key negotiation process, and once the key of both parties will be inconsistent if the unencapsulation is wrong, so this paper adds a layer of checking mechanism, i.e., to verify the correctness of the unencapsulation through the hash function to reduce the probability of the protocol's error. In addition, this paper combines the signature authentication link and key negotiation link into a two-way authentication and key negotiation link, only through two rounds of communication can make the two sides authenticate each other's identities and generate the negotiation key, which reduces the number of communications and improves the efficiency of the algorithm compared to the standard authentication scheme's authentication and then negotiation modes, and at the same time, based on the authentication process generated by the signature and the session key generates the two sides of the negotiation key, which enhances the utilization rate of the message. Utilization, and when an attacker wants to crack this key, cracking the original Kyber algorithm only needs to obtain the private key sk, but cracking this scheme needs to additionally

obtain the signatures of each round of communication and the specific details of the hash algorithm, which improves the difficulty of the attacker to crack the key.

In this paper's anti-quantum authentication protocol link, the main link by the key generation algorithm, signature algorithm, authentication and key encapsulation algorithm and authentication and key unencapsulation algorithm consists of four parts of the main algorithm, the algorithm design is based on the principle of Kyber and Dilithium algorithm for the implementation of the algorithm, which due to the Kyber algorithm to achieve the key expansion rate is large, so in its mathematical implementation is often introduced into compression Because of the large key expansion rate realized by Kyber algorithm, compression and decompression algorithms are often introduced in its mathematical implementation to simplify the computational process data, and the following formula is given:

Compression: for $Compress_q(x,\sigma)$ is defined as follows: input $x \in Z_q$, $\sigma < \lceil \log q \rceil$, and the output of this is:

$$y = round\left(\left(2^{\sigma} / q\right) \cdot x\right) \bmod {}^{+} 2^{\sigma} \tag{1}$$

Decompression: for $Decompress_q(y,\sigma)$ is defined as follows: input $y = Compress_q(x,\sigma)$, output is:

$$x' = round\left((q / 2^{\sigma}) \cdot y\right) \tag{2}$$

Since Kyber algorithm and Dilithium algorithm are based on the operation of polynomial ring, in the key generation algorithm, based on the input random number seed sampling to get the polynomial ring, this algorithm completes the sampling generation of signing key pairs and negotiation key pairs at the same time, which reduces the number of invocations of the process operations such as the random number generation, hash function and so on, and improves the efficiency of key generation.

In the signature algorithm, the signature mainly consists of three vectors, of which c is the hash signature based on the plaintext message, z is the potential signature for the operation process paradigm checking, and h is the rounding vector that contains the rounding information when the signature is generated, the signature c as the signature containing the plaintext message can be used as the main means of signature verification, while the potential signature z and the high judgment vector h are both as auxiliary means to verify the validity of the signature.

The authentication and key encapsulation algorithm mainly accomplishes the three functions of signature verification, key encapsulation, ciphertext and authentication message signing, and needs to check the three sets of vectors separately during signature verification, and after successful verification, it carries out the vector product computation, compression and decompression computation for the public key variable to obtain the ciphertext c and the session key k, and generates the Auth message used for key verification and signing.

The authentication and key unsealing algorithm mainly accomplishes the functions of signature verification and key unsealing, and obtains the ciphertext c after successful verification, and decompresses the vectors to obtain the session key sampling information to generate the session key.

2.4 Lightweight Authentication and Key Negotiation Protocols Resistant to Quantum Attacks

The previous anti-quantum authentication protocol constructed based on Dilithium algorithm and Kyber algorithm is an overall security mechanism for securing 6G terminal devices, and this chapter designs the specific implementation form of the lightweight authentication protocol by constructing the system model and security model.

2.4.1 System model

The protocol employs a 3-layer network architecture: trusted third parties (TA), fog nodes (FN) and 6G end devices (SM).

(1) TA: possesses sufficient computational and interactive capabilities to accomplish any complex task. In this protocol, TA is mainly responsible for generating the public parameters of the system, the keys of SM and FN, and is not involved in the process of authentication and key negotiation.

(2) FN: deployed in the vicinity of SM, some of the functions of the cloud service are deployed to the FN. FN is able to authenticate the message sent by SM and negotiate a session key.

(3) SM: located at the end-user and used to record the end-user's private information.

2.4.2 Security model

In the protocol, we consider TA to be a trusted entity and an adversary cannot directly attack it. SM and FN are untrustworthy and an adversary can crack either of them to obtain the data stored inside. The security of the protocol is defined using the stochastic language machine model.

Let Π_{TA}^{k}, Π_{SM}^{j}, and Π_{FN}^{i} represent the k, j, and i instances of TA, SM, and FN, respectively, and Π_{Γ}^{i} denote that any participant of the protocol $\Gamma \in \{FN, SM\}$ the i th instance.

The adversary $\mathcal{A}$ is known to have complete control over the information communicated between the two parties, defined by the following random query.

$Execute\left(\Pi_{FN}^{j}, \Pi_{SM}^{i}\right)$: simulates a passive listening attack. When this query is executed, outputs all messages sent between Π_{FN}^{j} and Π_{SM}^{i}.

$Send\left(\Pi_{\Gamma}^{i}, m\right)$: simulates an active attack. When an adversary $\mathcal{A}$ sends a message m disguised as an honest participant, the query is executed according to the steps specified in the protocol and the corresponding result is output.

$Reveal\left(\Pi_{\Gamma}^{i}\right)$: when this query is executed, the session key is output directly if Π_{Γ}^{i} passes the verification. Otherwise, output $\perp$.

$Corrupt\left(\Pi_{\Gamma}^{i}\right)$: simulates a forward-looking attack. When this query is executed, the adversary $\mathcal{A}$ will obtain the key for SM or FN.

$Test\left(\Pi_{\Gamma}^{i}\right)$: when the adversary $\mathcal{A}$ sends the $Test$ query, a random number b is chosen, and if $b=1$, the session key for the instance Π_{Γ}^{i} is output. Otherwise, randomly select a uniformly distributed random number for $\mathcal{A}$, requiring the length of the random number to be the same as the length of b.

Semantic safety (AKA-safety): the query can be executed multiple times, except for $Test$

which can only be executed once at the end. When $Test$ is called, the adversary $\mathcal{A}$ outputs the guessed value b' of b. If $b=b'$, $\mathcal{A}$ guessed correctly and succeeded. Let the event $Succ$ denote that the adversary $\mathcal{A}$ made a $Test$ query and eventually guessed correctly. Let $Pr[Succ]$ denote the probability that the adversary $\mathcal{A}$ wins the game.

Therefore, the advantage of the adversary $\mathcal{A}$, i.e., the advantage of the adversary $\mathcal{A}$ in destroying the semantic security of the key of the protocol P, is denoted as:

$$Adv_{P,\mathcal{A}}^{AKA}(k)=\left|2*\Pr[Succ]-1\right| \tag{3}$$

A protocol P is said to be $AKA-$ secure if $Adv_{P,\mathcal{A}}^{AKA}(k)$ is negligible for any PPT adversary $\mathcal{A}$.

2.4.3 Specific implementations of the AKA protocol

The protocol consists of three parts: the system initialization phase, the registration phase and the AKA phase. The symbols involved in the protocol along with the definitions are shown below:

Trusted Entity: TA.

6G terminal device: SM.

Fog node: FN.

Key of TA: s.

Key of FN: s_j.

Random number generated by TA: n_i.

The value of n_i from the previous session, with initial value $n_i^{-1}=n_i$: n_i^{-1}.

Denote the random numbers generated by SM and FN, respectively: r_1, r_2.

The value of r_1 from the previous session, with initial value: $r_i^{-1}=null$ r_i^{-1}.

Identity identifier of SM: ID_i.

Anonymous identity identifier of SM: AID_i.

Cryptographic identity identifier of SM: EID_i.

Identity identifier of FN: ID_j.

Session key between SM and FN, $SK=\{SK_x, SK_y\}$: SK.

Symmetric encryption and decryption using key k: $E_k(\cdot)/D_k(\cdot)$.

Anti-collision hash function $(i=1,\cdots,4)$: $h(\cdot), H_i(\cdot)$.

(1) System initialization phase

In the initialization phase, TA initializes all the system parameters in the following detailed steps:

1) Given the security parameter k, TA generates its own key s.

2) TA chooses the secure hash function h and $H_i(i=1,\cdots,4)$:

$$\begin{aligned} &h:\{0,1\}^{*}\to\{0,1\}^{k}\\ &H_1:\{0,1\}^{*}\to\{0,1\}^{k_x*m}\\ &H_2:\{0,1\}^{*}\to\{0,1\}^{k_y*m}\\ &H_3:\{0,1\}^{*}\to\{0,1\}^{k_x}\\ &H_4:\{0,1\}^{*}\to\{0,1\}^{k_y} \end{aligned} \tag{4}$$

3) Finally, TA discloses these parameters $\{h,H_1,H_2,H_3,H_4\}$.

(2) Registration Phase

If SM and FN need to communicate, they must register through TA to obtain the corresponding credential information.

Fog Node FN Registration When FN needs to get service from TA or communicate with SM, it first needs to register on TA, the detailed steps are as follows:

1) FN randomly selects the identifier ID_j and sends it to TA over a secure channel.

2) When ID_j is received, TA computes the corresponding key $s_j=h\left(ID_j\|s\right)$ and returns it to FN.

3) FN receives the key s_j and saves it.

6G Terminal Device SM Registration When SM needs to communicate with FN, it first needs to be registered through TA, and the detailed steps are as follows:

1) SM randomly selects the identifier ID_i and sends it to TA over a secure channel.

2) TA chooses a random number $n_i\in Z_q^*$, and then computes SM 's key $s_i=h\left(ID_i\|s\right)$, $t_i=h\left(s\oplus s_i\oplus n_i\right)$ and encrypts the identity $EID_i=E_{h(t_i\oplus s_i)}\left(ID_i\oplus s_i\right)$. Also, initialize n_i^{-1} and r_i^{-1} $n_i^{-1}=n_i$ and $r_i^{-1}=null$.

Finally, TA sends $\left\{s_i,n_i,t_i\right\}$ to SM. Send $\left\{EID_i,n_i,n_i^{-1},r_i^{-1}\right\}$ to FN.

3) After receiving this information, SM stores the received information in the anti-jamming device. FN adds the received information to the validation table.

(3) Authentication and key negotiation phase

In this phase, SM and FN authenticate each other. If both parties are authentic and valid, SM and FN will negotiate a session key. At the same time, they also update the corresponding secret information. The detailed steps are as follows:

1) Operation of SM

SM selects a random number $r_1\in Z_q^*$ and computes $R_1=r_1\oplus n_i$, $AID_i=ID_i\oplus r_1$, , $T_i=t_i\oplus h\left(ID_i\|r_1\right)$ and $\alpha=h\left(ID_i\oplus s_i\oplus t_i\oplus r_1\oplus n_i\right)$

Next, SM sends the message $MSG_1=\left\{R_1,AID_i,T_i,\alpha\right\}$ to FN over the open channel.

2) After receiving the message MSG_1, FN decompresses r_1 using the stored n_i $r_1=R_1\oplus n_i$, calculates $ID_i=AID_i\oplus r_1$, $t_i=T_i\oplus h\left(ID_i\|r_1\right)$, to obtain the key s_i for SM, $s_i=D_{h(t_i\oplus s_i)}(EID_i)\oplus ID_i$.

FN checks if the equation $\alpha=h\left(ID_i\oplus s_i\oplus t_i\oplus r_1\oplus n_i\right)$ is valid, and if it isn't, uses the stored n_i^{-1} Instead of n_i, recalculate ID_i, t_i and s_i and compare again. Then compare $r_1=r_i^{-1}$ to see if they are equal, and if they are, FN aborts the session. The main purpose of

r_1 is to ensure that the adversary is not able to use the values from the previous session for playback attacks.

FN generates the random number $r_2 \in Z_q^*$, computes $R_2 = r_2 \oplus r_1$, $SK_x = H_1(s_i \| r_1 \| r_2)$, and $SK_y = H_2(s_i \| r_1 \| r_2)$, $A = H_3(r_1 \| r_2)$, and $B = H_4(r_1 \| r_2)$, such that the session key $SK = \{SK_x, SK_y\}$. Next, FN computes $Z = A \cdot SK_x \oplus B \cdot SK_y \oplus v$.

FN checks whether n_i or n_i^{-1} was used in the calculation step above, if n_i, update $n_i^{-1} = n_i$, $n_i = h(n_i \oplus r_1 \oplus r_2)$. If n_i^{-1}, $n_i = h(n_i^{-1} \oplus r_1 \oplus r_2)$. Also update $r_i^{-1} = r_1$, $t_i = h(t_i \oplus n_i)$ and $EID_i = E_{h(t_i \oplus s_i)}(ID_i \oplus s_i)$.

Finally, FN sends the message $MSG_2 = \{R_2, Z\}$ to SM.

3) Receiving the message MSG_2, SM computes $r_2 = R_2 \oplus r_1$, $SK_y = H_2(s_i \| r_1 \| r_2)$, $A = H_3(r_1 \| r_2)$ and $B = H_4(r_1 \| r_2)$. Check whether the inequality $Hwt(A \cdot SK_x \oplus B \cdot SK_y \oplus Z) \le \mu \cdot m$ holds, where $\mu \in (\eta, 1/2)$.

If the inequality does not hold, SM aborts the session. Otherwise, FN passes the verification of SM while having the same session key $\{SK_x, SK_y\}$.

Finally, SM updates $n_i = h(n_i \oplus r_1 \oplus r_2)$ and $t_i = h(t_i \oplus n_i)$.

3 Results and Discussion

3.1 Security analysis

According to the system security analysis, the security comparison results of this paper's scheme, i.e., the authentication scheme constructed based on Dilithium algorithm and Kyber algorithm with other authentication schemes can be derived, and the comparison results of scheme security are shown in Table 2. The security is analyzed in six aspects, namely, anonymity, protection against replay attack (RA), protection against tampering attack (TA), protection against man-in-the-middle attack (MITM), protection against identity forgery attack (ISF), and protection against denial-of-service attack (DoS), respectively. By comparing with the lattice-based authentication scheme (Scheme1), hash-based authentication scheme (Scheme2), and multivariate-based authentication scheme (Scheme3), it can be seen that this paper's scheme can satisfy the security requirements of typical authentication scenarios, and it has a very significant advantage in resisting different types of attacks, and is able to provide systematic security assurance.

Table 2: Solution safety comparison results

Scheme	Anonymity	RA	TA	MITM	ISF	DoS
Scheme1	√	×	√	√	×	×
Scheme2	×	√	√	×	√	×
Scheme3	×	×	√	√	×	√
Ours	√	√	√	√	√	√

3.2 Performance analysis

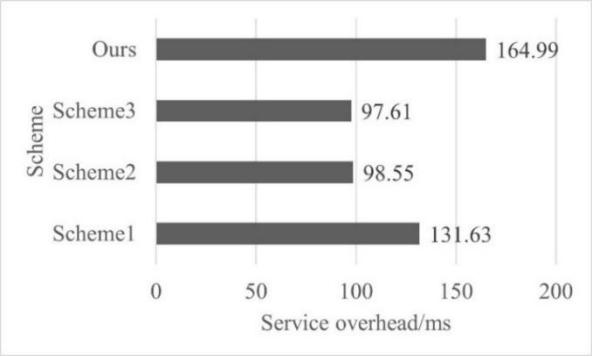
3.2.1 Computational overhead

In this paper, on a Windows 11 system configured with Intel(R) Core (TM) i5-1340P 1.90 GHz and 16GB RAM, we compute the operation parameters of the commonly used cryptographic primitives in the scheme: TGm, TGa, TSe, and Th denote the ECC dot-multiplication, ECC dot-addition, symmetric encryption/ decryption, and hash function operations, respectively. Considering that the time of the different-or operation and concatenation operation is negligible compared with the time of the cryptographic primitive operation, it is not counted in this paper. The comparative analysis of the cryptographic primitive execution time of the authentication scheme proposed in this paper in the authentication phase and other schemes is shown in Table 3. The computational overhead of this paper's scheme on the client side is significantly lower than Scheme1~3, thus it has lightweight user authentication overhead and can complete the authentication on the client side faster. The computation time overhead on the server side is slightly larger compared to Scheme1~3.

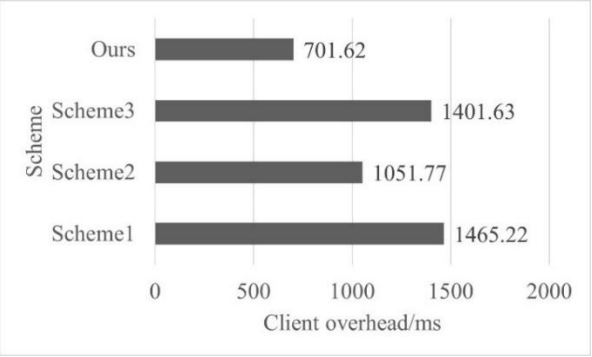
Table 3: Analysis of the time comparison of password primitives

Scheme	Client side	Service end
Scheme1	4TGm+TGa+2TSe+5Th	4TGm+TGa+1TSe+4Th
Scheme2	3TGm+2Ga+6Th	2TGm+1Ga+5Th
Scheme3	4TGm+6Th	2TGm+4Th
Ours	1TGm	6TGm+4TSe+6Th

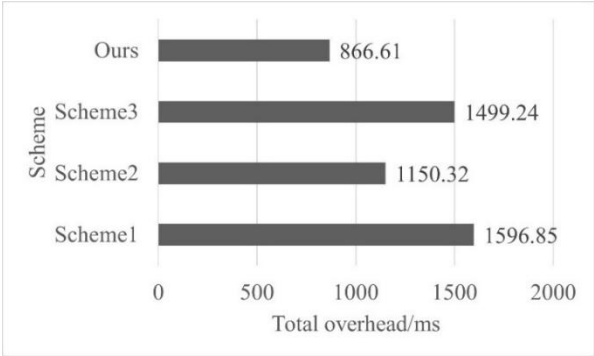
The experimental results of the computational overhead of this paper's scheme compared with other schemes are shown in Fig. 2, and (a)~(c) denote the comparison of server-side overhead, client-side overhead and total overhead, respectively. It can be seen that this paper's scheme has a very significant computational overhead advantage on the client side, and at the same time, the total computational overhead is lower than Scheme1~3 by 730.24, 283.71, and 632.63, respectively.



(a)Service overhead contrast



(b) Service overhead contrast



(c) Total overhead contrast

Figure 2: Compare the calculation overhead of different schemes

3.2.2 Communication overhead

The communication overhead in the hash operation result is 165bit, timestamp is 30bit, random number is 165bit, public key cryptography algorithm key and ID length are 165bit. According to the length of the above data, the comparison result of the communication overhead of the different schemes is shown in Figure 3. In this paper, the communication overhead of the client in the authentication phase is 963, which is smaller compared to Scheme1~3. The communication overhead at the server side is 2627, which is slightly larger compared to Scheme2, but this is due to the fact that this paper's scheme interacts with more server-side subjects, which increases a certain amount of communication overhead. Overall, compared with other authentication schemes, the communication overhead of this paper's scheme is at a lower level, and it can well meet the communication requirements in different authentication scenarios.

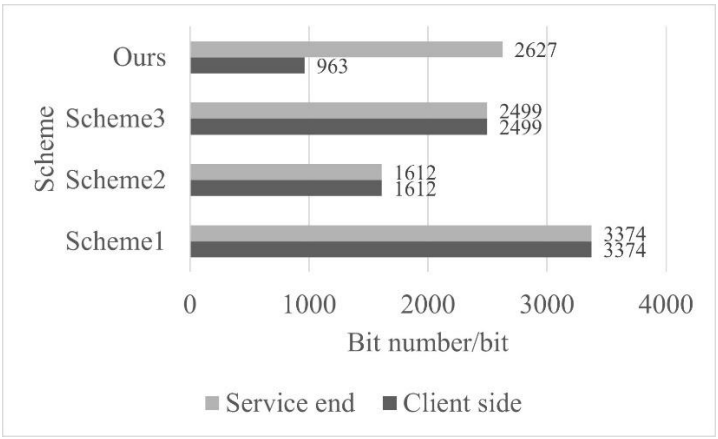


Figure 3: Comparison of communication overhead of different schemes

3.2.3 Storage overhead

The storage overhead of this paper's scheme is compared with Scheme1~3, and the comparison results are shown in Fig. 4. It can be seen that the client-side storage overhead of this paper's scheme is smaller than that of Scheme1~3, which is only 0.05, but the server-side overhead is larger than that of Scheme1~3, which is 0.28. Considering that this paper's scheme has a significant advantage in resisting security threats, the slightly higher server-side storage overhead is acceptable. By comprehensively comparing with Scheme1~3 in terms of computation overhead, communication overhead and storage overhead, this paper's scheme shows smaller computation, communication overhead and acceptable storage overhead, and at

the same time has more perfect security performance, reflecting the unique advantages of lightweight distributed trusted authentication scheme. The security and performance analysis and experimental results show that this paper's scheme is efficient, secure and advanced.

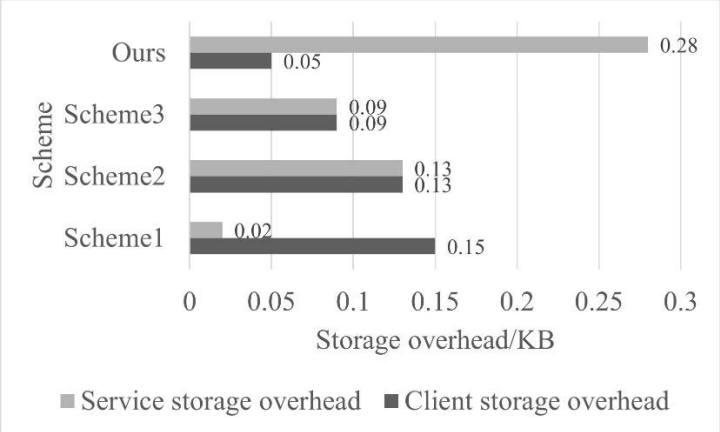


Figure 4: Comparison of storage overhead for different schemes

3.3 Experimental validation

The Anti-Quantum Lightweight Authentication Protocol (AKA) based on Ubuntu 18.04.3, CentOS 3.10.0 and Deepin 5.4.50 systems are implemented respectively. The client sends the crypto suite defined in this paper to the server via Cli-entHello message, and the server selects this crypto suite to implement the key negotiation method proposed in this paper. If the server does not select this cipher suite it can also jump to other cipher suites, in this paper PSK and ECDHE are used as two alternate suites.

The experimental environment is a Dell desktop with a Core(TM) i7-9700 processor having 32GB of RAM, the type of virtual machine is VMware Workstation Pro with 4GB of virtual memory allocated to the virtual machine, and the virtual machine system is Ubuntu 18.04.3, CentOS 3.10.0, and Deepin 5.4.50.By combining the AKA protocol based on Dilithium and Kyber algorithm designed in this paper with the two alternate suites of AKA PSK, ECDHE and Identity based PBC protocol schemes, the performance of the scheme designed in this paper is examined by comparing their message transmission overhead and connection latency.

3.3.1 Message transmission overhead

The messages transmitted by the 4 schemes interaction are captured by wireshark respectively, comparing the number of message bytes generated during the 4 kinds of key negotiation process, and the message transmission overhead of the 4 schemes is shown in Table 4. Firstly, it can be seen that the number of interactions for both the 2 suites of AKA alternate is 5 times, while the scheme proposed in this paper and the identity-based PBC protocol scheme both reduce the number of interactions by 2 times. Secondly, in these interaction processes, due to the elimination of the certificate sending and verification processes, the number of key agreement messages required to be transmitted in the scheme designed in this paper and the PBC protocol scheme based on identity identification is also reduced. Compared with PSK, the number of messages sent is reduced by 4, and compared with ECDHE, it is reduced by 11, which greatly reduces the communication traffic in the key agreement process. Moreover, the key agreement message byte of the scheme proposed in this paper is slightly smaller than that of the PBC protocol scheme based on identity identification.

Table 4: Messaging overhead for the four schemes

Scheme	Frequency of interaction	Handshake number	Handshake message/B
PSK	5	10	1829
ECDHE	5	17	5741
PBC	3	6	2396
AKA	3	6	2344

3.3.2 Connection time

Separately measured the connection time of the four schemes in three operating systems (Ubuntu, CentOS, Deepin) with linux as the kernel, the measurement method is to record the time used from the beginning of sending the first message to the establishment of the connection, and the connection latency of the four schemes is measured several times, and finally the average value is obtained, the results of the comparison of the four schemes' connection time under the three operating systems The results of the comparison of the connection times of the four schemes under three operating systems are shown in Fig. 5. Because PSK is directly based on the key agreed by both parties to encrypt the communication, so this scheme saves the time spent in the key negotiation process to calculate the key, and the PSK scheme does not need to certify the certificate, which also saves the connection time, and the connection can be completed in a smaller delay, but the PSK scheme is less secure, and the ECDHE scheme needs to parse the certificate for authentication, so it takes time to connect. The ECDHE scheme needs to parse the certificate for authentication, so it spends more time and completes the connection with the largest delay.The PBC scheme also realizes the key negotiation process based on no certificate, which saves the connection time, but this scheme does not take into account the problem that the PKG nodes suffer from an attack that leads to the leakage of the system's private key and the legitimate user's private key. In contrast, the scheme proposed in this paper eliminates the certificate verification process and also establishes the connection with relatively small latency while considering the security.The key negotiation connection time of the AKA protocol scheme is 22.61ms, and the scheme proposed in this paper reduces the connection latency by 44.08% compared to the ECDHE scheme. Compared with the PBC scheme, which uses bilinear pair operation to obtain the shared key and subsequent session keys, this paper adopts the faster discrete logarithm operation, so the key negotiation connection time is shorter.

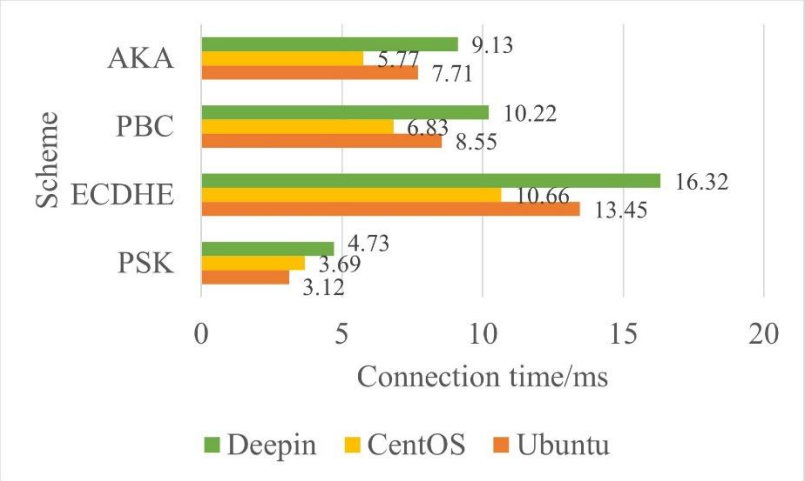


Figure 5: Four scheme connection time comparison

4 Conclusion

The study designs an anti-quantum lightweight authentication protocol (AKA) in the context of 6G end devices. The authentication scheme based on Dilithium and Kyber algorithms secures the protocol in six aspects: anonymity, resistance to replay attacks, resistance to tampering attacks, resistance to man-in-the-middle attacks, resistance to identity forgery attacks and resistance to denial of service attacks. The total computation overhead (866.61ms), client communication overhead (963bit) and client storage overhead (0.05KB) of this paper's scheme are better than the comparison scheme.AKA protocol not only reduces the communication traffic during key negotiation, but also saves the time spent on calculating the key during key negotiation, and reduces the connection latency compared with ECDHE protocol by 44.08%. This paper is designed to guarantee the security and reliability of next generation mobile communication systems.

About the Author

Jian Zhao (1983-7) male, Han, from Yuncheng, Shandong Province, holds a master's degree and is an associate professor. His research focuses on ubiquitous vocational education, cybersecurity, and related fields.

Lisha Xie (1984-6) female, Han, from Zhuzhou, Hunan Province, holds a master's degree and works as an engineer. Her research focuses on ubiquitous network security, intellectual property protection, and related applied fields.

References

[1] Wu, L., Wang, J., Choo, K. K. R., & He, D. (2018). Secure key agreement and key protection for mobile device user authentication. IEEE Transactions on Information Forensics and Security, 14(2), 319-330.

[2] Adhikari, T. (2024). Advancing zero trust network authentication: Innovations in privacy-preserving authentication mechanisms. Comput. Sci. Eng, 1, 1-22.

[3] Dolan, E., & Widayanti, R. (2022). Implementation of authentication systems on hotspot network users to improve computer network security. International Journal of Cyber and IT Service Management, 2(1), 88-94.

[4] Suresh Kumar, V., Ibrahim Khalaf, O., Raman Chandan, R., Bsoul, Q., Kant Gupta, S., Zawaideh, F., ... & Salama AbdElminaam, D. (2024). Implementation of a novel secured authentication protocol for cyber security applications. Scientific Reports, 14(1), 25708.

[5] Dogra, A., Jha, R. K., & Jain, S. (2020). A survey on beyond 5G network with the advent of 6G: Architecture and emerging technologies. IEEE access, 9, 67512-67547.

[6] Azari, M. M., Solanki, S., Chatzinotas, S., Kodheli, O., Sallouha, H., Colpaert, A., ... & Ottersten, B. (2022). Evolution of non-terrestrial networks from 5G to 6G: A survey. IEEE communications surveys & tutorials, 24(4), 2633-2672.

[7] Abdel Hakeem, S. A., Hussein, H. H., & Kim, H. (2022). Security requirements and challenges of 6G technologies and applications. Sensors, 22(5), 1969.

[8] Ren, Z., Li, X., Jiang, Q., Wang, Y., Ma, J., & Miao, C. (2022). Network slicing in 6G: An authentication framework for unattended terminals. IEEE Network, 37(1), 78-86.

[9] Kazmi, S. H. A., Hassan, R., Qamar, F., Nisar, K., & Ibrahim, A. A. A. (2023). Security concepts in emerging 6G communication: Threats, countermeasures, authentication techniques and research directions. Symmetry, 15(6), 1147.

[10] Pandi, S., Albert, A. J., Thapa, K. N. K., & Krishnaprasanna, R. (2024). A novel enhanced security architecture for sixth generation (6G) cellular networks using authentication and acknowledgement (AA) approach. Results in Engineering, 21, 101669.

[11] Je, D., Jung, J., & Choi, S. (2021). Toward 6G security: technology trends, threats, and solutions. IEEE Communications Standards Magazine, 5(3), 64-71.

[12] Tom, J. J., Anebo, N. P., Onyekwelu, B. A., Wilfred, A., & Eyo, R. E. (2023). Quantum computers and algorithms: a threat to classical cryptographic systems. Int. J. Eng. Adv. Technol, 12(5), 25-38.

[13] Azmi, S. K. (2024). Cryptographic Hashing Beyond SHA: Designing collision-resistant, quantum-resilient hash functions. International Journal of Science and Research Archive, 12(2), 3119-3127.

[14] Yu, H., & Huang, M. (2025). Anti-quantum cross-chain identity authentication approach using dynamic group signature. Frontiers of Information Technology & Electronic Engineering, 26(5), 742-752.

[15] Reddy, M., & Rao, K. (2024). A Lightweight Symmetric Cryptography based User Authentication Protocol for IoT based Applications. Scalable Computing: Practice and Experience, 25(3), 1647-1657.

[16] Gupta, S., Alharbi, F., Alshahrani, R., Kumar Arya, P., Vyas, S., Elkamchouchi, D. H., & Soufiene, B. O. (2023). Secure and lightweight authentication protocol for privacy preserving communications in smart city applications. Sustainability, 15(6), 5346.

[17] Xu, P., Wu, H., Tao, X., Wang, C., Chen, D., & Nan, G. (2024). Anti-quantum certificateless group authentication for massive accessing IoT devices. IEEE Internet of Things Journal, 11(9), 16561-16577.

[18] Tawfeeq, N. H., Yousif, M., Al-Shareeda, M. A., Almaiah, M. A., & Shehab, R. (2025). Lightweight and quantum-resistant authentication for the internet of drones (iod) using dilithium signatures. International Journal of Innovative Research and Scientific Studies, 8(2), 2842-2853.

[19] Al-Mekhlafi, Z. G., Al-Shareeda, M. A., Manickam, S., Mohammed, B. A., & Qtaish, A. (2023). Lattice-based lightweight quantum resistant scheme in 5G-enabled vehicular networks. Mathematics, 11(2), 399.

[20] Turnip, T. N., Andersen, B., & Vargas-Rosales, C. (2025). Towards 6G Authentication and Key Agreement Protocol: A Survey on Hybrid Post Quantum Cryptography. IEEE Communications Surveys & Tutorials.