Hindawi
Security and Communication Networks
Volume 2020, Article ID 5167832, 12 pages
https://doi.org/10.1155/2020/5167832

Research Article

A Novel RLWE-Based Anonymous Mutual Authentication Protocol for Space Information Network

Junyan Guo and Ye Du

Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University, Beijing, China

Correspondence should be addressed to Ye Du; ydu@bjtu.edu.cn

Received 27 September 2019; Revised 4 February 2020; Accepted 6 May 2020; Published 25 August 2020

Academic Editor: Angel M. Del Rey

Currently, space information network (SIN) has become an increasingly important role in real life. As a large heterogeneous wireless network, SIN can better provide global mobile services to users anytime and anywhere, even in extreme geographic environments. In addition, there is no need to build the communication base-stations every few kilometers on the ground to ensure high service quality, which greatly reduces the construction costs and can be used as an economical communication method in sparsely populated areas. So there is a trend that more and more end users are more likely to get SIN services than traditional terrestrial cellular networks. However, due to the openness and publicity of the satellite wireless channel and the limited resources of the satellite nodes, the privacy and security cannot be perfectly guaranteed and may even be vulnerable to attacks initiated by the adversary such as replay attacks, impersonation attacks, and eavesdropping attacks. To improve the access security of SIN, researchers have proposed a series of authentication protocols based on different cryptographic assumptions. Nevertheless, existing research shows that these protocols cannot meet the requirements of higher and higher security and short authentication delay. In addition, these protocols are mainly based on public key cryptography mechanisms such as DLP and ECDLP, which can be solved by postquantum computers in polynomial time, so these protocols will no longer be secure. To solve the vulnerability of these protocols, in this paper, we propose a new RLWE-based anonymous mutual authentication and key agreement protocol, which guarantees higher security with low computational overhead even in the postquantum era. Detailed security analysis shows that our protocol meets security requirements and is resistant to a variety of known attacks. Besides, combining security comparison and performance analysis, our proposed protocol is more practical than other protocols in SIN.

1. Introduction

With the continuous development of globalization, users are increasingly demanding communication and are required to establish high-quality communication connections with others. In this context, SIN is proposed to meet the needs of users for reliability and availability globalization services [1]. As a large heterogeneous network covering the globe, SIN includes various satellite constellations to provide navigation, communication, weather, reconnaissance, and other services for numerous users. In the future, SIN will not only serve the Earth users, but also serve space satellites and interstellar spacecraft with advanced space technology [2]. Compared with traditional terrestrial cellular networks, SIN has the following three characteristics [3]. First, SIN covers a wider range and can provide the stable signal in cities, villages, and even extreme environments. Second, as long as the device has the ability to communicate with satellites, it can join SIN as an end user or service provider anywhere in the world, which greatly improves scalability. Third, SIN does not require the laying of large amounts of ground facilities like a terrestrial network, and the signals are not affected by the terrain. SIN is a scarce resource for present and future, so the United States and Europe have carried out a series of key national research projects such as Thuraya [4], MUOS [5], ISICOM [6], and TSAT [7].

As a key communications architecture in marine, aerospace, military, and remote IoT applications, confidential and internal information will be transmitted to each other through SIN wireless channels [3]. Because security is

not a primary concern when initially deploying SIN, the current research on security is still insufficient. At present, researchers have begun to shift from researching the functional requirements to the security requirements, and access authentication is the focus of their research [8, 9]. Accessing authentication is the first step for the end user to apply for service to the satellite node; it is even more necessary to take some light mechanisms to ensure the security and the quality of service (QoS). There are two key factors affecting the security and performance of the authentication protocol: one is the openness of the satellite wireless channel, the public channel will result in data being easily captured by adversaries, and the cryptographic mechanism should be used to resist the attack initiated by the malicious node. The other is the authentication delay caused by messages transmission and computing overhead. For delay-sensitive users who require real-time communication, the authentication delay should be as small as possible, even if the signal delay caused by satellites located 500 to 3,000 kilometers above the ground is unavoidable.

Currently, hardware technology is developing rapidly, and satellites can already carry more complex computing devices [10]. The three-party or two-party authentication protocols proposed in other research areas can provide a reference for designing authentication protocols suitable for SIN but cannot be directly applied. For example, in mobile pay-TV systems, [11] proposes a time slot-based key distribution mechanism that divides 24 hours by 45 minutes and arranges them in a binary tree. Service providers can provide up to 16 communication keys for a single user. However, when the number of users increases, it will occupy a lot of storage space of the service provider, which is not suitable for satellite with shortage of resources. In global mobility networks, [12] proposed a roaming authentication scheme based on the chaotic map-based discrete logarithm problem (CMBDLP). When a user and the foreign agent authenticate with each other, they need to transfer information four times with the participation of home agent, which increases the user's waiting time, and this does not meet the SIN low-latency requirement. Other scholars have also proposed many authentication protocols based on difficult problems such as elliptic curve discrete logarithm problem (ECDLP) [13], discrete logarithm problem (DLP) [14], large integer factorization problem (LIFP) [15, 16], and hash function [17], but these protocols still cannot be directly applied to SIN unless high security and efficiency are not considered. Furthermore, with the advent of the post-quantum era, most authentication protocols that rely on public key cryptography have the potential to be resolved in polynomial time. So these authentication protocols may present security risks in the future. Therefore, this paper proposes a novel RLWE-based anonymous mutual authentication protocol for space information network, which can meet the requirements of high security and efficiency. In our protocol, LEO satellites serve as nodes that end users want to access. Terrestrial control station (TCS) provides system registration services for end users and LEO satellites. In the process of mutual authentication between the end user and the LEO satellite, the temporary identity is used so that the true identity is not revealed, and TCS does not participate in the authentication process as an offline node, which greatly reduces the authentication delay. In addition, the authentication protocol is based on the RLWE assumption which has been shown to be as difficult as the worst-case problem in the ideal lattice [18] and is also resistant to quantum computing with less computational overhead [19]. It is worth mentioning that our proposed protocol has the following main contributions:

(1) We propose a novel RLWE-based anonymous mutual authentication protocol that enables both the end user and the LEO satellite to authenticate each other. In addition, TCS does not participate in the authentication process, which greatly reduces the communication delay to better meet the needs of delay-sensitive users.

(2) This is the first anonymous authentication protocol for antiquantum computing in SIN. Moreover, in this paper, detailed security analysis proves that our protocol can meet the security requirements based on the well-known Dolev–Yao threat model [20] and resist various attacks by adversaries.

(3) Considering the linkability of the user identity may be known by the adversary with background knowledge during the temporary identity transmission process, our protocol allows the end user to apply for a new temporary identity from TCS in the public channel, which enhances anonymity and unlinkability.

The rest of this paper is organized as follows: in Section 2, we first briefly discuss related work. Then we show the background information related to the protocol in Section 3. In Section 4, we describe the proposed protocol in detail. Detailed analysis of the security and performance of the proposed protocol is provided in Sections 5 and 6, respectively. Finally, Section 7 presents the conclusion of this paper.

2. Related Work

In recent years, researchers have been enthusiastic about security and privacy issues in the authentication protocol for space information network (SIN). In 1996, Cruickshank [21] proposed the first authentication protocol in satellite networks using the public key cryptosystem to authenticate the legitimate identity of the end user and the satellite, respectively, and uses the symmetric key system to encrypt the communication data. However, the protocol requires terrestrial control station (TCS) to participate during the authentication process, which results in a large delay in the authentication process, and the use of four complicated encryption and decryption operations results in large computational overhead. Then Hwang et al. [22] proposed an authentication protocol that uses the symmetric key to encrypt the message transmission in the mutual authentication phase without the need for a public key mechanism. Although the computational overhead is reduced, the shared key in each authentication is only determined by the TCS, and the end user does not participate in generating the

shared key, without enabling the end user to trust the security of the shared key. To overcome the weaknesses in [22], Chang and Chang [23] proposed an authentication scheme with only hash functions and XOR operations. This scheme greatly reduces the computational complexity and ensures that the shared key is jointly generated by both parties, but still cannot overcome the problem of TCS participating in the authentication process which will result in the increase of TCS computing load and a large delay of authentication. In 2012, Zheng et al. [24] proposed a more effective protocol to reduce the computational complexity of TCS, but proved in [25] that [24] cannot resist the denial of service (DoS) attack and the identity spoofing attack. Recently, Yang et al. [10] proposed a new anonymous fast authentication protocol based on the q-SDH problem and elliptic curve digital signature algorithm (ECDSA) for authenticating roaming users in foreign domains. In the authentication stage, the protocol does not compromise the anonymity of the user's identity and can resist replay attacks, man-in-the-middle attacks, and modification attacks. Unfortunately, the protocol does not have user login authentication, which will result in that if the user's device is stolen, the adversary can use the user's device to impersonate as a legitimate user. Besides, in the postquantum era, the difficult problems that [10] relies on have been proven to be resolved in polynomial time, and the anonymity and communication security will be invalid. Feng et al. [26] first proposed an anonymous authentication protocol based on ideal lattices and resistant to quantum computing. However, there are two defects in [26] that cannot be applied. First, when the adversary intercepts the message sent by the user in authentication phase and the message is directly replayed or modified, then the server will spend a lot of computational overhead to check whether it is a legitimate message. If the adversary sends a large number of forged authentication messages, it will consume a lot of system resources. Secondly, server provides both registration and user authentication functions, while satellite nodes in SIN cannot simultaneously undertake heavy computing and storage tasks due to limited resources. In summary, previous authentication protocols based on hash or classic hard problems cannot meet the anonymity, security, and low-latency requirements. Therefore, in this paper, we design an access authentication and key agreement protocol, which can guarantee the anonymity of users and has lower transmission delay.

3. Preliminaries

In this section, we give a review of background information and the notations on RLWE and then briefly describe the system model and threat model that the protocol relies on. Finally, the security requirements are presented.

3.1. Ring Learning with Errors. Let $n = 2^k$, where $k \in \mathbb{Z}$. The rings of polynomials over $\mathbb{Z}$ and $\mathbb{Z}_q$, respectively, are denoted by $\mathbb{Z}[x]$ and $\mathbb{Z}_q[x]$, where q is an odd prime number and $q \bmod 2n = 1$. Consider the two rings $\mathbb{R} = \mathbb{Z}[x]/(x^n + 1)$ and $\mathbb{R}_q = \mathbb{Z}_q[x]/(x^n + 1)$. For any polynomial element y in $\mathbb{R}$ or $\mathbb{R}_q$, denote it by its coefficient vector in $\mathbb{Z}^n$ and $\mathbb{Z}_q^n$, respectively. Given a fixed positive real β, the discrete Gaussian distribution over $\mathbb{R}_q$ is denoted by χ_β. We refer to [19, 27, 28] for a more description of RLWE with the following lemmas.

Lemma 1. *For any two elements* $\mathbf{a}, \mathbf{b} \in \mathbb{R}$, *there exist* $\|\mathbf{a} \cdot \mathbf{b}\| \le \sqrt{n} \cdot \|\mathbf{a}\| \cdot \|\mathbf{b}\|$ *and* $\|\mathbf{a} \cdot \mathbf{b}\|_\infty \le \sqrt{n} \cdot \|\mathbf{a}\|_\infty \cdot \|\mathbf{b}\|_\infty$.

Lemma 2. *Given any positive real* $\beta = \omega(\sqrt{\log n})$, *the* $\Pr_{\mathbf{X} \leftarrow \chi_\beta}[\|\mathbf{X}\| > \beta \cdot \sqrt{n}] \le 2^{-n+1}$ *[29].*

For an odd prime $q > 2$, let $\mathbb{Z}_q = \{-((q-1)/2), \ldots, ((q-1)/2)\}$ and the subset $E = \{-\lfloor q/4 \rfloor, \ldots, \lfloor q/4 \rfloor\}$ as the middle set of $\mathbb{Z}_q$. For any $x \in \mathbb{Z}_q$, the characteristic function Cha of the set E complement is defined as

$$\mathrm{Cha}(x) = \begin{cases} 0, & x \in E, \\ 1, & x \notin E. \end{cases} \tag{1}$$

The auxiliary modular function Mod_2: $\mathbb{Z}_q \times \{0,1\} \longrightarrow \{0,1\}$ is defined as $\mathrm{Mod}_2(v, b) = (((v + b \cdot ((q-1)/2)) \bmod q) \bmod 2)$, where $v \in \mathbb{Z}_q$ and $b = \mathrm{Cha}(v)$, with the following lemma for these two functions.

Lemma 3. *Given an odd prime number q, we have two ring elements* v, $e \in \mathbb{Z}_q$ *such that* $|e| < q/8$. *Then, the equation* $Mod_2(v, cha(v)) = Mod_2(w, cha(v))$ *holds, where* $w = v + 2 \cdot e$.

The two functions Cha and Mod_2 can be extended to the ring $\mathbb{R}_q$ by applying coefficients to ring elements and can also follow the lemmas mentioned above. Given a ring element $\mathbf{v} = (v_0, \ldots, v_{n-1}) \in \mathbb{R}_q$ and $\mathbf{b} = (b_0, \ldots, b_{n-1}) \in \{0,1\}^n$, we have $\mathrm{Cha}(\mathbf{v}) = (\mathrm{Cha}(v_0), \ldots, \mathrm{Cha}(v_{n-1}))$ and $\mathrm{Mod}_2(\mathbf{v}, \mathbf{b}) = (\mathrm{Mod}_2(v_0, b_0), \ldots, \mathrm{Mod}_2(v_{n-1}, b_{n-1}))$. Then, for $\mathbf{w} = \mathbf{v} + 2 \cdot \mathbf{e}$, we have $\mathrm{Mod}_2(\mathbf{w}, \mathrm{Cha}(\mathbf{w})) = \mathrm{Mod}_2(\mathbf{v}, \mathrm{Cha}(\mathbf{w}))$, where the absolute value of each element in $\mathbf{e}$ is less than $q/8$ [30].

Definition 1. Ring learning with errors (RLWE) assumption: let $\mathbb{R}_q$ and χ_β be defined as above. $\mathbf{v}$, $\mathbf{e}$ are randomly selected from $\mathbb{R}_q$ and χ_β, respectively. The RLWE assumption states that it is hard for any *PPT* algorithm to distinguish $\mathbb{R}_q \times \chi_\beta$ from the uniform distribution on $\mathbb{R}_q^2$. The hardness of the RLWE assumption can be reduced to the Shortest Independent Vectors Problem (SIVP) over ideal lattices [31].

3.2. System Model. As shown in Figure 1, SIN contains a total of three types of entities: terrestrial control station (TCS), satellite node, and end user. The following details describe the functions of each entity.

(i) TCS is a control center to provide registration services to end users and satellite nodes. Moreover, TCS is considered a trusted entity with the highest level of firewall and intrusion detection system. Any attack can be detected and taken with corresponding security measures to prevent attacker from intruding into TCS.

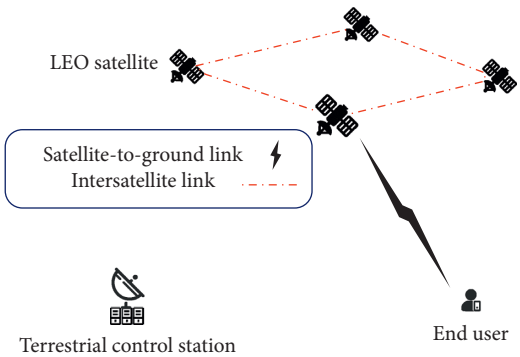

Figure 1: Space information network.

(ii) Satellite node is the service provider for end users in SIN. In order to reduce the delay of users accessing SIN, LEO satellites that are closer to the ground are usually used. LEO satellites are not all legal service providers, and there may be some LEO satellites controlled by malicious adversaries.

(iii) End user is user with the smart device and has the ability to compute, store, and communicate with satellites. The end user will request access to the SIN to get the subscribed service. It needs to be reminded that the smart device is at risk of being lost or stolen.

3.3. Threat Model. In our protocol we make use of the Dolev–Yao threat model, which means that the adversary will control all openness and public channels in SIN. The adversary can arbitrarily monitor, intercept, modify, and replay messages transmitted between nodes and has unlimited storage space to store all the information monitored. The protocol we designed is to allow legitimate nodes to authenticate each other's identity, deny illegal access, and ensure that secrets are not obtained by adversaries under the Dolev–Yao threat model.

3.4. Security Requirements. According to the characteristics of the previously proposed authentication protocol in SIN, a well-designed protocol should meet the following security requirements.

3.4.1. Mutual Authentication. Satellite nodes should have the ability to verify the legal identity of the end user and prevent access by nonlegitimate users. Similarly, the end user should also have the same ability to verify access to legitimate satellites.

3.4.2. Identity Anonymity. The identity of the end user should remain anonymous, and no one other than TCS and the end user himself can know the true identity of the user.

3.4.3. Key Establishment. After successful mutual authentication of the satellite and the end user, they should jointly construct a shared key to protect future communication.

3.4.4. Perfect Forward Secrecy. The authentication protocol also needs to meet the requirement that the shared key leakage does not lead to the previous and future session key leaks.

3.4.5. Attack Resistance. In the authentication process, in order to ensure the accuracy and security of authentication, the protocol should be able to withstand various attacks initiated by the adversary such as replay attack, modification attack, eavesdropping attack, and impersonation attack.

4. Our Proposed Protocol

In this section we present a novel RLWE-based anonymous authentication protocol. The detailed protocol description will be introduced in the order of system initialization phase, registration phase, authentication phase, password update phase, and temporary identity update phase.

4.1. System Initialization Phase. In the system initialization phase, TCS generates the master key pair and some system public parameters according to the following steps:

(1) TCS sets system security parameters k

(2) TCS chooses an odd prime number q and an integer n, where n is a power of 2 and $q \bmod 2n = 1$

(3) TCS chooses a discrete Gaussian distribution χ_β and a random ring element $\mathbf{a}$, where β is a fixed positive number and $\mathbf{a} \in \mathbb{R}_q$

(4) TCS randomly samples $\mathbf{s}, \mathbf{e} \longleftarrow \chi_\beta$ and computes the master public key $\mathbf{p}_{\mathrm{TCS}} = \mathbf{a} \cdot \mathbf{s} + 2 \cdot \mathbf{e}$, where $\mathbf{s}$ is the master private key

(5) TCS chooses a security hash function h: $\{0,1\}^* \longrightarrow \{0,1\}^k$

(6) TCS publishes the system parameters $\{q, n, \chi_\beta, \mathbf{a}, \mathbf{p}_{\mathrm{TCS}}, h\}$ to the public and securely stores the master private key $\mathbf{s}$

The system initialization phase is performed once when the system is laid out and not during other phases. Since it is only executed once, the computational overhead of the system initialization phase can be considered negligible.

4.2. Registration Phase. The registration phase is the process by which TCS interacts with trustworthy end users and satellite nodes. In this phase, the satellite and the end user need to submit the true identity ID and other necessary parameters to TCS. It is worth noting that end users also need to generate the temporary identity TID that masks the true identity during the authentication phase. Then TCS generates the parameters needed for mutual authentication in the future for the end user and the satellite node, respectively. We briefly show this process in Figure 2 and the more detailed steps will be described in Algorithm 1 and the rest of this section. We assume that the satellite set is $\mathbf{S}_{LEO}$ with N_1 satellites and the end user set is $\mathbf{S}_{\mathrm{user}}$ with N_2 users. For a clearer presentation, in the following description, we

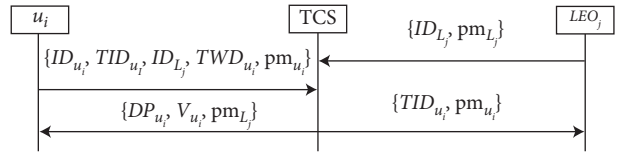

Figure 2: Registration phase.

(1) **LEO$_j$**:
(2) Generates the true identity ID_{L_j};
(3) Randomly samples $\mathbf{sm}_{L_j}, \mathbf{em}_{L_j}$ and computes the master public key $\mathbf{pm}_{L_j} = \mathbf{a} \cdot \mathbf{sm}_{L_j} + 2 \cdot \mathbf{em}_{L_j}$;
(4) Stores the master private key $\mathbf{sm}_{L_j}$;
(5) Submits $\{ID_{L_j}, \mathbf{pm}_{L_j}\}$ to TCS;
(6) **LEO$_j$** END
(7) **u$_i$**:
(8) Generates the true identity ID_{u_i}, temporary-ID TID_{u_i}, password PW_{u_i} and chooses the satellite node ID_{L_j};
(9) Randomly samples $\mathbf{sm}_{u_i}$, $\mathbf{em}_{u_i}$ and computes the master public key $\mathbf{pm}_{u_i} = \mathbf{a} \cdot \mathbf{sm}_{u_i} + 2 \cdot \mathbf{em}_{u_i}$;
(10) Computes $TWD_{u_i} = h(TID_{u_i}, PW_{u_i})$;
(11) Submits $\{ID_{u_i}, TID_{u_1}, ID_{L_j}, TWD_{u_i}, \mathbf{pm}_{u_i}\}$ to TCS;
(12) **u$_i$** END
(13) **TCS**:
(14) Computes $DI_{u_i} = h(TID_{u_i}, \mathbf{s})$, $DP_{u_i} = DI_{u_i} \oplus TWD_{u_i}$, $V_{u_i} = h(TID_{u_i}, TWD_{u_i}, DP_{u_i})$;
(15) Transmits $\{DP_{u_i}, V_{u_i}, \mathbf{pm}_{L_j}\}$ to u_i and $\{TID_{u_i}, \mathbf{pm}_{u_i}\}$ to LEO_j;
(16) Stores $\{ID_{u_i}, TID_{u_i}, \mathbf{pm}_{u_i}\}$ and $\{ID_{L_j}, \mathbf{pm}_{L_j}\}$;
(17) **TCS** END
(18) **u$_i$** stores $\{DP_{u_i}, V_{u_i}, \mathbf{pm}_{L_j}\}$;
(19) LEO_j stores $\{TID_{u_i}, \mathbf{pm}_{u_i}\}$;

Algorithm 1: Registration phase.

simplify the system model with only one end user u_i, one satellite LEO_j, and TCS. Besides, it is necessary to note that the messages in this phase are transmitted in the secure channel.

(1) LEO_j generates the true identity ID_{L_j} and computes the master public key $\mathbf{pm}_{L_j} = \mathbf{a} \cdot \mathbf{sm}_{L_j} + 2 \cdot \mathbf{em}_{L_j}$, where $\mathbf{sm}_{\mathbf{L_j}}$ and $\mathbf{em}_{\mathbf{L_j}}$ are randomly sampled from χ_β. $\mathbf{sm}_{\mathbf{L_j}}$ is assumed to be the master private key and is stored securely by LEO_j. Finally, LEO_j sends the message $\{ID_{L_j}, \mathbf{pm}_{L_j}\}$ to TCS. It needs to be reminded that the identity of the ID_{L_j} satellite node is not private, so there is no need to anonymize it.

(2) u_i generates the true identity ID_{u_i}, temporary identity TID_{u_i}, and password PW_{u_i} and then chooses the satellite node ID_{L_j} which it wants to communicate with. The function of PW_{u_i} is used for login verification and it will be required in authentication phase, password update phase, and temporary identity update phase. Next, u_i computes the master public key $\mathbf{pm}_{u_i} = \mathbf{a} \cdot \mathbf{sm}_{u_i} + 2 \cdot \mathbf{em}_{u_i}$, where $\mathbf{sm}_{\mathbf{u_i}}$ and $\mathbf{em}_{u_i}$ are randomly sampled from χ_β. $\mathbf{sm}_{u_i}$ is assumed to be the master private key and is stored securely by u_i. Then u_i computes $TWD_{u_i} = h(TID_{u_i}, PW_{u_i})$, which is to protect the PW_{u_i} from being known by the TCS. Finally, u_i sends the message $\{ID_{u_i}, TID_{u_i}, ID_{L_j}, TWD_{u_i}, \mathbf{pm}_{u_i}\}$ to TCS.

(3) TCS computes $DI_{u_i} = h(TID_{u_i}, \mathbf{s})$, $DP_{u_i} = DI_{u_i} \oplus TWD_{u_i}$, and $V_{u_i} = h(TID_{u_i}, TWD_{u_i}, DP_{u_i})$ after receiving the registration message of u_i. Then TCS sends the messages $\{DP_{u_i}, V_{u_i}, \mathbf{pm}_{L_j}\}$ to u_i and $\{TID_{u_i}, \mathbf{pm}_{u_i}\}$ to LEO_{L_j}. V_{u_i} is used to check if the temporary identity TID_{u_i} and password PW_{u_i} entered are correct when u_i logs into the smart device. The function of the DP_{u_i} is to enable u_i to update the password PW_{u_i} as wishes and detailed in the password update phase. The function of DI_{u_i} is to prevent u_i from changing TID_{u_i} by himself. If u_i is free to update the temporary identity TID_{u_i}, the legal identity of u_i cannot be distinguished during the authentication phase. By binding the master private key $\mathbf{s}$ of the TCS to TID_{u_i}, even if u_i attempts to update TID_{u_i}, it cannot change DI_{u_i}, DP_{u_i}, and V_{u_i} without TCS. Finally, TCS stores $\{ID_{u_i}, TID_{u_i}, \mathbf{pm}_{u_i}\}$ and $\{ID_{L_j}, \mathbf{pm}_{L_j}\}$.

(4) u_i stores the message $\{DP_{u_i}, V_{u_i}, \mathbf{pm}_{L_j}\}$.

(5) LEO_j stores the message $\{TID_{u_i}, \mathbf{pm}_{u_i}\}$.

4.3. Authentication Phase. As shown in Figure 3, in this phase, u_i and LEO_j mutually authenticate each other's legal identity in accordance with the following steps and negotiate a shared session key to encrypt future communications. It is

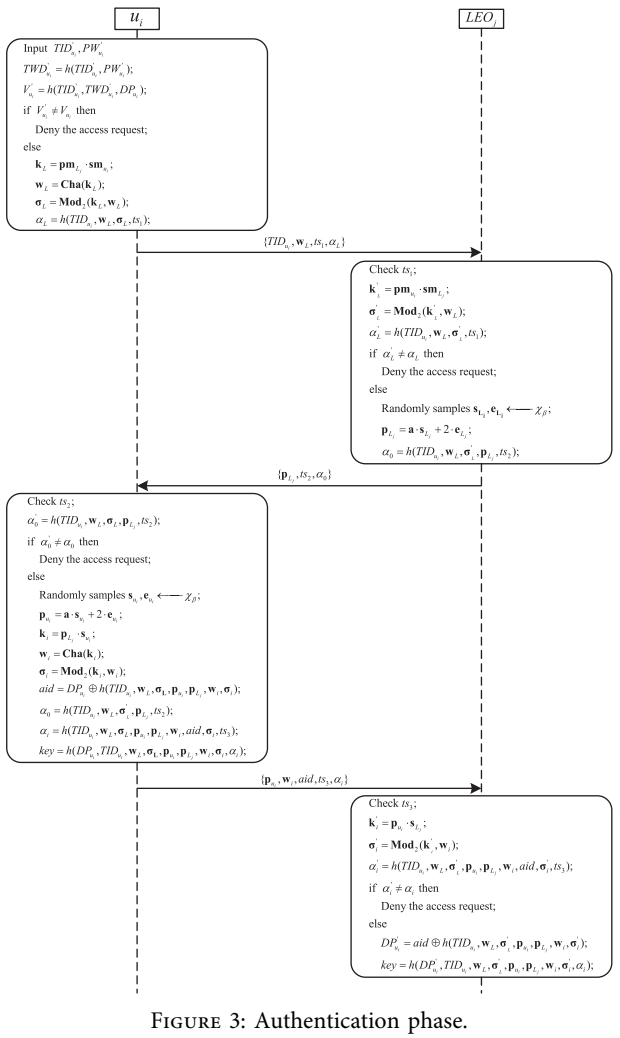

Figure 3: Authentication phase.

worth noting that the three interactive messages are transmitted in the public channel and the messages are transmitted in plaintext.

(1) u_i needs to input the temporary identity TID_{u_i}' and password PW_{u_i}' to login the smart device before requesting authentication. The device computes $TWD_{u_i}' = h(TID_{u_i}', PW_{u_i}')$, $V_{u_i}' = h(TID_{u_i}', TWD_{u_i}', DP_{u_i})$. Then it checks whether $V_{u_i}' = V_{u_i}$. If they are not equal, the access request is terminated immediately; otherwise the following steps are continued according to the protocol. This process is to prevent the device from falling into the adversary and being disguised as u_i. Next, the smart device computes $\mathbf{k}_L = \mathbf{pm}_{L_j} \cdot \mathbf{sm}_{u_i}$, $\mathbf{w}_L = \mathrm{Cha}(\mathbf{k}_L)$, $\sigma_L = \mathrm{Mod}_2(\mathbf{k}_L, \mathbf{w}_L)$, and $\alpha_L = h(TID_{u_i}, \mathbf{w}_L, \sigma_L, ts_1)$. Finally, u_i sends the message $\{TID_{u_i}, \mathbf{w}_L, ts_1, \alpha_L\}$ to LEO_j. ts_1 is the timestamp used to prevent the replay attacks by the adversary. The rest of the timestamps in this paper have the same function as ts_1.

(2) After receiving the message $\{TID_{u_i}, \mathbf{w}_L, ts_1, \alpha_L\}$, LEO_j first checks the timestamp ts_1 and compares it with the current time to see if it is within the time allowed. If the timestamp is not within the allowed range, the access request is denied; otherwise LEO_j computes $\mathbf{k}_L' = \mathbf{pm}_{u_i} \cdot \mathbf{sm}_{L_j}$, $\sigma_L' = \mathrm{Mod}_2(\mathbf{k}_L', \mathbf{w}_L)$, and $\alpha_L' = h(TID_{u_i}, \mathbf{w}_L, \sigma_L', ts_1)$. Then it checks whether $\alpha_L' = \alpha_L$. If they are not equal, the end user who sent the access request is not legitimate and LEO_j rejects the access request; otherwise the next steps of the authentication protocol are continued. Next, LEO_j computes $\mathbf{p}_{L_j} = \mathbf{a} \cdot \mathbf{s}_{L_j} + 2 \cdot \mathbf{e}_{L_j}$ and $\alpha_0 = h(TID_{u_i}, \mathbf{w}_L, \sigma_L', \mathbf{p}_{L_j}, ts_2)$, where $\mathbf{s}_{\mathbf{L_j}}$ and $\mathbf{e}_{\mathbf{L_j}}$ are randomly sampled from χ_β. Finally, LEO_j sends the message $\{\mathbf{p}_{L_j}, ts_2, \alpha_0\}$ to u_i.

(3) After receiving the message $\{\mathbf{p}_{L_j}, ts_2, \alpha_0\}$, u_i first checks the timestamp ts_2 and compares it with the current time to see if it is within the time allowed. If the timestamp is not within the allowed range, the access request is denied; otherwise u_i computes $\alpha_0' = h(TID_{u_i}, \mathbf{w}_L, \sigma_L, \mathbf{p}_{L_j}, ts_2)$. Then u_i checks whether $\alpha_0' = \alpha_0$. If they are not equal, u_i will assume that the satellite requested to access during the authentication phase is not the true LEO_j and actively stops the access request; otherwise it continues to compute $\mathbf{p}_{u_i} = \mathbf{a} \cdot \mathbf{s}_{u_i} + 2 \cdot \mathbf{e}_{u_i}$, $\mathbf{k}_i = \mathbf{p}_{L_j} \cdot \mathbf{s}_{u_i}$, $\sigma_i = \mathrm{Mod}_2(\mathbf{k}_i, \mathbf{w}_i)$, $aid = DP_{u_i} \oplus h(TID_{u_i}, \mathbf{w}_L, \sigma_L, \mathbf{p}_{u_i}, \mathbf{p}_{L_j}, \mathbf{w}_i, \sigma_i)$, $\alpha_i = h(TID_{u_i}, \mathbf{w}_L, \sigma_L, \mathbf{p}_{u_i}, \mathbf{p}_{L_j}, \mathbf{w}_i, aid, \sigma_i, ts_3)$, and the final session key $h(DP_{u_i}, TID_{u_i}, \mathbf{w}_L, \sigma_L, \mathbf{p}_{u_i}, \mathbf{p}_{L_j}, \mathbf{w}_i, \sigma_i, \alpha_i)$. Finally, u_i sends the message $\{\mathbf{p}_{u_i}, \mathbf{w}_i, aid, ts_3, \alpha_i\}$ to LEO_j.

(4) After receiving the message $\{\mathbf{p}_{u_i}, \mathbf{w}_i, aid, ts_3, \alpha_i\}$, LEO_j performs the same timestamp check process as steps 2 and 3. If the timestamp is not within the allowed range, the access request is denied; otherwise LEO_j computes $\mathbf{k}_i' = \mathbf{p}_{u_i} \cdot \mathbf{s}_{L_j}$, $\sigma_i' = \mathrm{Mod}_2(\mathbf{k}_i', \mathbf{w}_i)$, $\alpha_i' = h(TID_{u_i}, \mathbf{w}_L, \sigma_L', \mathbf{p}_{u_i}, \mathbf{p}_{L_j}, \mathbf{w}_i, aid, \sigma_i', ts_3)$. Then LEO_j checks $\alpha_i' = \alpha_i$. If they are not equal, the received message is not sent by the real ID_{u_i} or modified by malicious nodes, and then the access request is immediately terminated; otherwise it computes $DP_{u_i}' = aid \oplus h(TID_{u_i}, \mathbf{w}_L, \sigma_L', \mathbf{p}_{u_i}, \mathbf{p}_{L_j}, \mathbf{w}_i, \sigma_i')$ and the final session key $h(DP_{u_i}', TID_{u_i}, \mathbf{w}_L, \sigma_L', \mathbf{p}_{u_i}, \mathbf{p}_{L_j}, \mathbf{w}_i, \sigma_i', \alpha_i)$.

4.4. Password Update Phase. When u_i wants to change the old password $PW_{u_i}^{Old}$ to the new password $PW_{u_i}^{New}$, an attempt as shown in Algorithm 2 is made to perform the password update phase. Firstly, u_i needs to input the temporary identity TID_{u_i} and the correct old password $PW_{u_i}^{Old}$. Then the smart device computes $TWD_{u_i}' = h(TID_{u_i}', PW_{u_i}^{Old})$, $V_{u_i}' = h(TID_{u_i}', TWD_{u_i}', DP_{u_i})$ and checks whether $V_{u_i}' = V_{u_i}$. If they are not equal, the password update phase is terminated immediately. Otherwise, u_i inputs the new password $PW_{u_i}^{New}$ and the smart device computes $TWD_{u_i}^{New} = h(TID_{u_i}, PW_{u_i}^{New})$, $DP_{u_i}^{New} = DP_{u_i}^{Old} \oplus TWD_{u_i}^{Old} \oplus TWD_{u_i}^{New}$, and $V_{u_i}^{New} = h(TID_{u_i}, TWD_{u_i}^{New}, DP_{u_i}^{New})$. Use $DP_{u_i}^{New}$ and $V_{u_i}^{New}$ to replace $DP_{u_i}^{Old}$ and $V_{u_i}^{Old}$, where $V_{u_i}^{New}$ will be used as the

(1) **Input** $TID'_{u_i}, PW^{Old}_{u_i}$
(2) $TWD'_{u_i} = h(TID'_{u_i}, PW^{Old}_{u_i})$;
(3) $V'_{u_i} = h(TID'_{u_i}, TWD'_{u_i}, DP_{u_i})$;
(4) **if** $V'_{u_i} = V_{u_i}$ **then**
(5) Deny the password update;
(6) **else**
(7) **Input** $PW^{New}_{u_i}$
(8) $TWD^{New}_{u_i} = h(TID_{u_i}, PW^{New}_{u_i})$;
(9) $DP^{New}_{u_i} = DP^{Old}_{u_i} \oplus TWD^{Old}_{u_i} \oplus TWD^{New}_{u_i}$;
(10) $V^{New}_{u_i} = h(TID_{u_i}, TWD^{New}_{u_i}, DP^{New}_{u_i})$;
(11) Smart device stores $\{TID_{u_i}, DP^{New}_{u_i}, V^{New}_{u_i}\}$;

Algorithm 2: Password update phase.

new verification value of the login device in the future. Finally, the smart device stores $\{TID_{u_i}, DP^{New}_{u_i}, V^{New}_{u_i}\}$.

4.5. Temporary Identity Update Phase. This phase is only performed when the previous temporary identity $TID^{Old}_{u_i}$ is no longer sufficient for their identity anonymity and security requirements. As shown in Figure 4, u_i finally obtains a new legal temporary identity $TID^{New}_{u_i}$ through two message exchanges and updates the parameters associated with the temporary identity TID such as DI, TWD, DP, and V. We note that all messages between TCS and u_i are transmitted in the public channel.

(1) u_i needs to input the correct TID_{u_i} and PW_{u_i}. After the verification is successful, u_i chooses a new temporary identity $TID^{New}_{u_i}$ and then computes $\mathbf{k}_{\mathrm{TCS}} = \mathbf{p}_{\mathrm{TCS}} \cdot \mathbf{sm}_{u_i}$, $\mathbf{w}_{\mathrm{TCS}} = \mathrm{Cha}(\mathbf{k}_{\mathrm{TCS}})$, $\sigma_{\mathrm{TCS}} = \mathrm{Mod}_2(\mathbf{k}_{\mathrm{TCS}}, \mathbf{w}_{\mathrm{TCS}})$, $\alpha_i = TID^{New}_{u_i} \oplus h(TID^{Old}_{u_i}, \mathbf{w}_{\mathrm{TCS}}, \sigma_{\mathrm{TCS}}, ts_4)$, and $\alpha = h(TID^{Old}_{u_i}, TID^{New}_{u_i}, \mathbf{w}_{\mathrm{TCS}}, \alpha_i, ts_4)$. Finally, u_i sends the message $\{TID^{Old}_{u_i}, \mathbf{w}_{\mathrm{TCS}}, ts_4, \alpha_i, \alpha\}$ to TCS.

(2) After receiving the message $\{TID^{Old}_{u_i}, \mathbf{w}_{\mathrm{TCS}}, ts_4, \alpha_i, \alpha\}$, TCS first checks the timestamp ts_4 and compares it with the current time to see if it is within the time allowed. If ts_4 is valid, TCS computes $\mathbf{k}'_{\mathrm{TCS}} = \mathbf{pm}_{u_i} \cdot \mathbf{s}$, $\sigma'_{\mathrm{TCS}} = \mathrm{Mod}_2(\mathbf{k}'_{\mathrm{TCS}}, \mathbf{w}_{\mathrm{TCS}})$, $TID^{New'}_{u_i} = \alpha_i \oplus h(TID^{Old}_{u_i}, \mathbf{w}_{\mathrm{TCS}}, \sigma'_{\mathrm{TCS}}, ts_4)$, and $\alpha' = h(TID^{Old}_{u_i}, TID^{New'}_{u_i}, \mathbf{w}_{\mathrm{TCS}}, \alpha_i, ts_4)$. Then TCS checks whether $\alpha' = \alpha$. If the check is valid, TCS can not only know the legal identity of the end user, but also know the new temporary identity $TID^{New}_{u_i}$ that u_i wants to update. Next, TCS continues to compute $DI^{New}_{u_i} = h(TID^{New'}_{u_i}, \mathbf{s})$, $DIa = DI^{New}_{u_i} \oplus h(TID^{Old}_{u_i}, TID^{New'}_{u_i}, \sigma'_{TCS})$, and $\alpha_1 = h(TID^{Old}_{u_i}, TID^{New'}_{u_i}, \mathbf{w}_{\mathrm{TCS}}, \sigma'_{\mathrm{TCS}}, DIa, ts_5)$. Finally, TCS sends the message $\{DIa, \alpha_1, ts_5\}$ to u_i and the message $\{TID^{Old}_{u_i}, TID^{New'}_{u_i}\}$ to LEO_j which updates the association between TID_{u_i} and $\mathbf{pm}_{\mathbf{u_i}}$. The message transmits between LEO_j and TCS in the secure channel.

(3) After receiving the message $\{DIa, \alpha_1, ts_5\}$, u_i also needs to first check if ts_5 is within the time allowed. If the check is valid, u_i computes $\alpha'_1 = h(TID^{Old}_{u_i}, TID^{New}_{u_i}, \mathbf{w}_{\mathrm{TCS}}, \sigma_{\mathrm{TCS}}, DIa, ts_5)$. Then u_i checks whether $\alpha'_1 = \alpha_1$. If the check is valid, the entity communicating with it can be identified as TCS. Next u_i continues to compute $DI^{New}_{u_i} = DIa \oplus h(TID^{Old}_{u_i}, TID^{New}_{u_i}, \sigma_{\mathrm{TCS}})$, $TWD^{New}_{u_i} = h(TID^{New}_{u_i}, PW_{u_i})$, and $DP^{New}_{u_i} = DI^{New}_{u_i} \oplus TWD^{New}_{u_i}$. Finally, updating $V^{New}_{u_i} = h(TID^{New'}_{u_i}, TWD^{New'}_{u_i}, DP^{New}_{u_i})$ allows u_i to input the $TID^{New}_{u_i}$ and PW_{u_i} can login the device correctly.

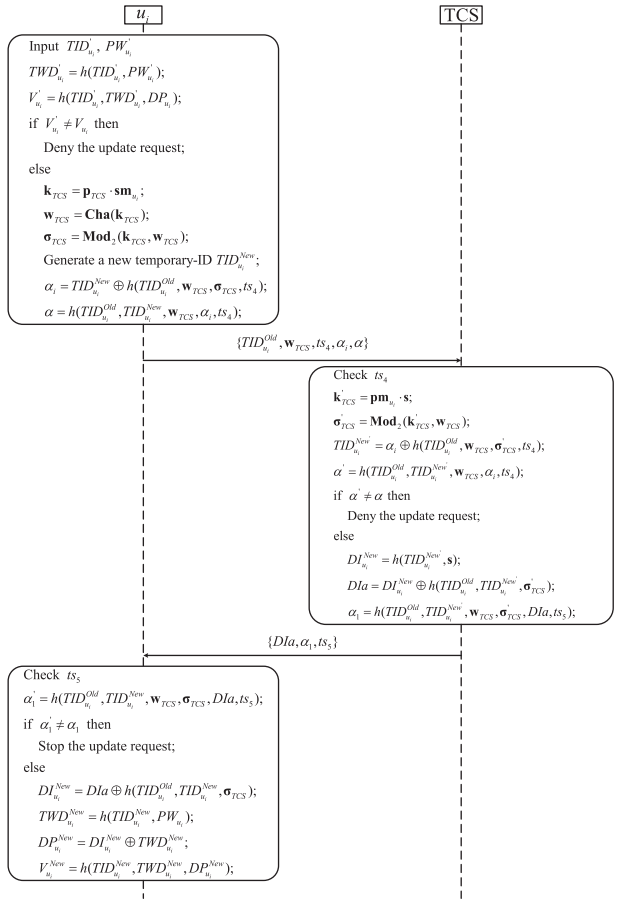

Figure 4: Temporary identity update phase.

5. Security Analysis

In this section, we analyze and discuss the security requirements of our protocol and prove that our protocol is sufficiently secure to resist insider attacks, replay attacks, modification attacks, eavesdropping attacks, and impersonation attacks. As shown in Table 1, we also compared the security attributes with other related protocols.

5.1. Mutual Authentication. In step 2 of authentication phase, LEO_j authenticates the legal identity of u_i by checking $\alpha_L' = \alpha_L$, where $\alpha_L' = h(TID_{u_i}, \mathbf{w}_L, \sigma_L', ts_1)$. Since LEO_j has securely stored the temporary identity TID_{u_i} and the public key $\mathbf{pm}_{\mathbf{u_i}}$ in registration phase, only the user whose temporary identity is TID_{u_i} and has the public-private key pair $\{\mathbf{pm}_{u_i}, \mathbf{sm}_{u_i}\}$ can compute the same σ_L as LEO_j and then can pass the check. No one can compute the matching private key $\mathbf{sm}_{\mathbf{u_i}}$ by TID_{u_i} and $\mathbf{pm}_{\mathbf{u_i}}$ unless the SIVP assumption can be solved with *PPT* algorithm. Similarly, in step 3 of authentication phase, u_i authenticates the legal identity of LEO_j by checking $\alpha_0' = \alpha_0$, where $\alpha_0 = h(TID_{u_i}, \mathbf{w}_L, \sigma_L, \mathbf{p}_{L_j}, ts_2)$. Since u_i has securely stored the ID_{L_j} and the public key , the adversary can only disguise as LEO_j if it solves the SIVP assumption in polynomial time. In addition, as [26], the secure hash function is used to ensure the integrity of the messages in the public channel transmission. Therefore, the authentication protocol can meet the security requirements of mutual authentication.

5.2. Identity Anonymity. In the whole system, only u_i and TCS know the true identity ID_{u_i}. Whether in the authentication phase or in the temporary identity update phase, the temporary identity TID_{u_i} is transmitted in the public channel. ID_{u_i} has no relevance to TID_{u_i} and cannot be inferred from TID_{u_i}. The adversary can only obtain the true identity of the user from TCS which preserves the relationship between TID_{u_i} and ID_{u_i}. However, the highest level of security protection of TCS makes it impossible for adversary. Therefore, the authentication protocol can meet the security requirements of identity anonymity.

5.3. Key Establishment. In step 3 and step 4 of authentication phase, u_i and LEO_j independently generated the final shared session key $h(DP_{u_i}, TID_{u_i}, \mathbf{w}_L, \sigma_L, \mathbf{p}_{u_i}, \mathbf{p}_{L_j}, \mathbf{w}_i, \sigma_i, \alpha_i)$, where nonpublic parameters $\sigma_L = \mathrm{Mod}_2(\mathbf{k}_L, \mathbf{w}_L)$ and $\sigma_i = \mathrm{Mod}_2(\mathbf{k}_i, \mathbf{w}_i)$ require both parties to participate to compute $\mathbf{k}$ and avoid the shared key being determined by the single party. Moreover, the adversary cannot guess $\mathbf{k}$ from the public key unless there is a more efficient algorithm to solve the SIVP assumption. Therefore, the authentication protocol can meet the security requirements of key establishment.

5.4. Perfect Forward Secrecy. In our protocol, the shared key negotiated by the two parties is $h(DP_{u_i}, TID_{u_i}, \mathbf{w}_L, \sigma_\mathbf{L}, \mathbf{p}_{u_i}, \mathbf{p}_{L_j}, \mathbf{w}_i, \sigma_i, \alpha_i)$, except for $\sigma_\mathbf{L}$, σ_i, and DP_{u_i}, which are parameters that can be directly intercepted in the public channel. $\sigma_\mathbf{L}$ and DP_{u_i} can be regarded as long-term secrets of u_i and LEO_j. In addition, σ_i is generated by $\mathbf{s}_{u_i}$ and $\mathbf{e}_{u_i}$ randomly sampled from χ_β at each authentication. Even if long-term secrets are captured by the adversary, due to the randomness of $\mathbf{s}_{u_i}$ and $\mathbf{e}_{u_i}$, the adversary cannot know the previous session key. Therefore, the authentication protocol provides perfect forward secrecy.

5.5. Login Authentication. Only after entering the correct temporary identity TID and password PW can the user perform the access authentication in accordance with the steps. When the user's device is lost, the adversary cannot use the device, which avoids the security threat of the adversary pretending to be a legitimate user. Moreover, it can not only deny malicious access directly at the device side, but also reduce the computational cost of SIN.

5.6. Resistance of Insider Attacks. Although TCS is a trusted entity in the SIN, it is inevitable that the possibility of insiders stealing the user's password exists. During the registration phase, the user did not submit the password PW directly to TCS but $TWD = h(TID, PW)$. Due to the one-way security of the hash function h, insiders cannot get PW from TWD. Therefore, the authentication protocol can meet the security requirements of resistance of insider attacks.

5.7. Resistance of Replay Attacks. It is noted that each message transmission contains a timestamp ts, which is hashed with σ. σ can only be known by both parties to the authentication and the adversary cannot know, which ensures that the message cannot be modified. So even if the adversary replays the authentication message, the user or satellite node can check whether it is the replay attack in two steps. First, check if the timestamp ts is within the allowed range; then compute the hash value α of the message and compare it with $\alpha\prime$ sent by the other party. Even if the adversary modifies the timestamp ts in the message and passes the first step of checking but cannot get σ, it is impossible to forge the correct hash value with the modified ts. Furthermore, parameters $\{\mathbf{s}_{L_j}, \mathbf{e}_{L_j}, \mathbf{s}_{u_i}, \mathbf{e}_{u_i}\}$ are randomly sampled, which results in different public keys $\mathbf{p}$ and hash values α for each session. u_i and LEO_j can also detect replay attacks by verifying these parameters. Therefore, the authentication protocol can meet the security requirements of resistance to replay attacks.

5.8. Resistance of Modification Attacks. During the authentication phase, each step contains a final message hash α which is the hash value of the message transmitted this time and some key data previously negotiated. For example, the message $\{\mathbf{p}_{L_j}, ts_2, \alpha_0\}$ is transmitted by LEO_j to u_i in the second step, where $\alpha_0 = h(TID_{u_i}, \mathbf{w}_L, \sigma_L, \mathbf{p}_{L_j}, ts_2)$. α_0 contains not only the message $\mathbf{p}_{L_j}$ and the timestamp ts_2 but also the previously negotiated parameters TID_{u_i}, $\mathbf{w}_L$, and $\sigma_\mathbf{L}$. Due

Table 1: Comparison of security attributes.

Security attribute	[21]	[22]	[23]	[24]	[10]	Ours
Mutual authentication	✓	✓	✓	✓	✓	✓
Identity anonymity	✗	✓	✓	✓	✓	✓
Key establishment	✓	✓	✓	✓	✓	✓
Perfect forward secrecy	✓	✗	✗	✓	✓	✓
Login authentication	✗	✗	✗	✗	✗	✓
Insider attack	✓	✓	✓	✓	✓	✓
Replay attack	✓	✓	✓	✓	✓	✓
Modification attack	✓	✓	✓	✓	✓	✓
Eavesdropping attack	✓	✓	✓	✓	✓	✓
Impersonation attack	✓	✗	✓	✗	✓	✓

to the security features of the hash function h, any changes to the message can be verified. Therefore, the authentication protocol can meet the security requirements of resistance of modification attacks.

5.9. Resistance of Eavesdropping Attacks. In our whole protocol, the adversary can only obtain data such as TID_{u_i}, $\mathbf{w}_L$, $\mathbf{p}_{L_j}$, $\mathbf{p}_{u_i}$, $\mathbf{w}_i$, and a series of hash values and timestamps. Due to the security of the hash function, the adversary cannot get any useful information from the hash value. In addition, the final shared key is $h(DP_{u_i}, TID_{u_i}, \mathbf{w}_L, \sigma_{\mathbf{L}}, \mathbf{p}_{u_i}, \mathbf{p}_{L_j}, \mathbf{w}_i, \sigma_i, \alpha_i)$, where $\sigma_{\mathbf{L}}$ and σ_i cannot compute by the known parameters such as TID_{u_i}, $\mathbf{w}_L$, $\mathbf{p}_{L_j}$, $\mathbf{p}_{u_i}$, $\mathbf{w}_i$ unless the SIVP assumption is solved. Therefore, the authentication protocol can meet the security requirements against eavesdropping attacks.

5.10. Resistance of Impersonation Attacks. An adversary may impersonate a legitimate user or satellite node to submit or respond to access requests. However, the adversary can only obtain the public key of both parties, but cannot obtain the private key for authentication and negotiation of shared key, so it is impossible to forge hash values to pass authentication. Therefore, the authentication protocol can meet the security requirements of resistance to impersonation attacks.

6. Performance Analysis

In this section, we present the performance analysis of the protocol for authentication delay and communication overhead. Because our proposed protocol is the first postquantum anonymous authentication scheme for SIN, we just choose to compare it with the classic authentication scheme [23] and the latest proposed protocol [10]. Furthermore, to make the comparison more intuitive and consider the practicality of the protocol for smart devices, we set the parameters in the protocol as [19]. The integer n is 1024 and the odd prime number q is 12289. For discrete Gaussian distribution parameter β is set to $\log \beta = 17.1$. Finally, choose the secure hash function SHA3 with output of 512 bits.

6.1. Authentication Delay. Authentication delay refers to the sum of the total computing time and the transmission time of both devices from the beginning to the end in the mutual authentication phase. Before discussing the authentication delay of our proposed protocol in detail, we need to use the following symbols to represent the average time overhead caused by different operations. T_{Ge} is used to represent the sampling time from the discrete Gaussian distribution χ_β. T_{smul} and T_{pmul} represent multiplication with scalar and multiplication time in $\mathbb{R}_q$, respectively. T_{pma} represents the time of the multiplication and addition operation in $\mathbb{R}_q$. T_{Cha} indicates the time when the Cha function is executed once. T_{h} is the time when the SHA3 hash function is executed once. To better analyze the performance of our proposed protocol, we quote the overhead time of various computation operations in [26]. The satellite node and the end user in the proposed protocol correspond to the server and the mobile device in [26], respectively. The machine is equipped with 3.4 GHz Intel Core i7-6700 processor and 8 GB RAM as the satellite node and the end user with 1.4 GHz Quad-core Exynos 4412 processor and 1 GB RAM. Both parties used the LatticeCrypto library and the MIRACL library when implementing the protocol. The experimental results are shown in Table 2 and it is worth mentioning that the computation overhead of Mod_2 function is small enough to be neglected. In addition, since LEO satellites are usually located 500 km–3000 km from the ground, it is reasonable to set the single message delivery time to $T_{u-LEO} = 5\mathrm{ms}$. The following is the analysis of the computing time according to the steps in the authentication phase.

(1) In the first step, after TID'_{u_i} and PW'_{u_i} entered by u_i, the device performs two hash operations to check whether they are correct. If the verification passes, u_i also needs to perform one multiplication operation in $\mathbb{R}_q$, one Cha function in $\mathbb{R}_q$, and one hash operation. Therefore, the total computing time overhead of u_i is $3 \times T_{\mathrm{h_u}} + T_{\mathrm{pmul_u}} + T_{\mathrm{Cha_u}} = 591.459\mathrm{ns}$.

(2) In the second step, after receiving the message, LEO_j first needs to perform one multiplication operation in $\mathbb{R}_q$ and one hash operation to check the validity of the message. If the verification passes, it also needs to continue to perform two random sampling operations in χ_β, one multiplication with scalar in $\mathbb{R}_q$, one multiplication and addition operation in $\mathbb{R}_q$, and one hash operation. Therefore, the total computing time

Table 2: Execution times of RLWE-related operations.

Notation	Satellite node (ns)	End user (ns)
T_{Ge}	73.503	561.483
T_{smul}	0.298	6.655
T_{pmul}	0.307	13.052
T_{pma}	2.549	29.505
T_{Cha}	0.689	35.515
T_{h}	14.09	180.964

overhead of LEO_j is $2 \times T_{\mathrm{h_L}} + T_{\mathrm{pmul_L}} + 2 \times T_{\mathrm{Ge_L}} + T_{\mathrm{smul_L}} + T_{\mathrm{pma_L}} = 178.34ns$.

(3) Next, after u_i receives the response message from LEO_j, u_i first performs one hash operation to check the validity of the message. If the verification passes, u_i also needs to perform two sampling operations in χ_β, one multiplication with scalar in $\mathbb{R}_q$, one multiplication and addition operation in $\mathbb{R}_q$, one multiplication time in $\mathbb{R}_q$, one Cha function in $\mathbb{R}_q$, and three hash operations. Therefore, the total computing time overhead of u_i is $4 \times T_{\mathrm{h_u}} + 2 \times T_{\mathrm{Ge_u}} + T_{\mathrm{smul_u}} + T_{\mathrm{pma_u}} + T_{\mathrm{pmul_u}} + T_{\mathrm{Cha_u}} = 1931.549$ns.

(4) Finally, after LEO_j receives the response message from u_i, it first performs one multiplication operation in $\mathbb{R}_q$ and one hash operation to check the validity of the message. If the verification passes, LEO_j also needs to continue to perform two hash operations. Therefore, the total computing time overhead of LEO_j is $3 \times T_{\mathrm{h_L}} + T_{\mathrm{pmul_L}} = 42.577$ns.

In general, the two parties of authenticating the identity and building the session key need to execute 2523.008 ns at the end user and 220.917 ns at the satellite node, respectively. The total computing time required for the protocol is 2743.925 ns. Besides, the three messaging times required for the authentication phase are $3 \times T_{u-LEO} = 15$ms. So the authentication delay of our proposed protocol is 15.003 ms and the computing time is only a small part of the authentication delay.

In [23], the computing time depends on the maximum number of accesses N and the $j-$th access. In order to achieve the shortest time, we set $N = 1$, $j = 1$. Therefore, both the end user and the satellite node of the authentication process need to perform four hash operations and the computing times aare $4 \times T_{\mathrm{h_u}} = 723.856$ns and $4 \times T_{\mathrm{h_L}} = 56.36$ns, respectively. Besides, the five messaging times required for the authentication phase are $5 \times T_{u-LEO} = 25$ms. So, the authentication delay of [23] is 25.001 ms. However, in general, in order to reduce the computational task of TCS, N is set to a larger value which makes the authentication delay greatly increase and the performance of the protocol degrade.

In [10], the protocol is based on the $q-Strong$ Diffie–Hellman problem [32] where the execution of the protocol requires pairing operation, multiplication operation, and exponentiation operation in the additive cyclic group. The experimental results of these operations in [26] are as described in Table 3. In the authentication phase, the end user needs to execute the total of nine multiplication operations, three exponentiation operations, two pairing operations, and one hash operation, so the computing time of end user is $9 \times T_{\mathrm{mul_u}} + 3 \times T_{\mathrm{exp_u}} + 2 \times T_{\mathrm{pair_u}} + T_{\mathrm{h_u}} = 115.485$ms. The satellite node executes five multiplication operations, four exponentiation operations, two pairing operations, and one hash operation, so the computing time of satellite node is $5 \times T_{\mathrm{mul_L}} + 4 \times T_{\mathrm{exp_L}} + 2 \times T_{\mathrm{pair_L}} + T_{\mathrm{h_L}} = 18.202$ms. The total computing time required for the protocol is 133.687 ms. Besides, the two messaging times required for the authentication phase are $2 \times T_{u-LEO} = 10$ms. So, the authentication delay of [10] is 143.687 ms.

Table 3: Execution times of $q-Strong$ DH related operations.

Notation	Satellite node (ms)	End user (ms)
T_{pair}	8.262	52.533
T_{mul}	0.002	0.012
T_{exp}	0.417	3.437

Table 4: Comparison of authentication delay.

	Satellite node	End user	Transmission time	Total
[23]	56.36 ns	723.856 ns	25 ms	25.001 ms
[10]	18.202 ms	115.485 ms	10 ms	143.687 ms
Our	220.917 ns	2523.008 ns	15 ms	15.003 ms

Table 4 shows the comparison of the authentication delays of our proposed protocol with the other two protocols. It can be seen from the table that the authentication delay of our proposed protocol is significantly lower than [10, 23], which is a more effective authentication scheme.

6.2. Communication Overhead. According to the security of the protocol and the support for the device mentioned earlier, we set the size of elements in $\mathbb{R}_q$ to 4096 bits, the output of the SHA3 function is 512 bits, and the length of the identity and the timestamp is 100 bits. There are three message transmissions in the mutual authentication phase of our protocol. First, u_i sends $\{TID_{u_i}, \mathbf{w}_L, ts_1, \alpha_L\}$ to LEO_j. Then, LEO_j responds $\{\mathbf{p}_{L_j}, ts_2, \alpha_0\}$ to u_i. Finally, u_i sends a confirmation message $\{\mathbf{p}_{u_i}, \mathbf{w}_i, aid, ts_3, \alpha_i\}$ to the LEO_j. Therefore, the total communication overhead of the protocol is $(100 + 4096 + 100 + 512) + (4096 + 100 + 512) + (4096 + 4096 + 512 + 100 + 512) = 18832$ bits.

In [23], both parties need to transmit six hash values and six identity messages, so the total transmission overhead is $6 \times 512 + 6 \times 100 = 3672$ bits. In [10], assume that the length of element in cycle group and all signatures is 160 bits. Therefore, according to the communication overhead in [10], it can be concluded that the total of 2480 bits of messages needs to be transmitted during the authentication phase.

It is worth noting that the communication overhead of the proposed protocol is greater than other protocols because the protocol is based on RLWE, and a common flaw in the ring $\mathbb{R}_q$ is that the key size is larger than the traditional encryption system. Considering that satellite network bandwidth has grown significantly and the authentication delay is small enough for delay-sensitive users, so we believe

that our proposed protocol is a practicable scheme which is superior to other protocols.

7. Conclusion

In this paper we propose a novel RLWE-based anonymous mutual authentication protocol for SIN which is an anti-quantum computing protocol. In the security analysis, we elaborated that the proposed protocol can meet the security requirements of SIN access authentication and compare the security with other related protocols. The analysis results show that our protocol is more secure than other protocols. Moreover, in the performance analysis, it is further stated that the authentication delay of our proposed protocol is very small. Although the communication overhead is slightly larger, the protocol we proposed under the trade-off communication delay and communication overhead is practical for SIN in the future.

Data Availability

The data used to support the findings of this study are included within the paper.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This paper was supported by the National Natural Science Foundation of China (Grant no. 61672092).

References

[1] D. Li, X. Shen, J. Gong, J. Zhang, and J. Lu, "On construction of China's space information network," *Geo-Spatial Information Science Wuhan University*, vol. 40, no. 6, pp. 711–715, 2015.

[2] S. Yao, J. Guan, Z. Yan, and K. Xu, "Si-STIN: a smart identifier framework for space and terrestrial integrated network," *IEEE Network*, vol. 33, no. 1, pp. 8–14, 2019.

[3] J. Guo and Y. Du, "Fog service in space information network: architecture, use case, security and challenges," *IEEE Access*, vol. 8, pp. 11104–11115, 2020.

[4] D. A. Sunderland, G. L. Duncan, B. J. Rasmussen et al., "Megagate ASICs for the thuraya satellite digital signal processor," in *Proceedings of the Interantional 630 Symposium on Quality Electronic Design*, San Jose, CA, USA, March 2002.

[5] G. Zunich, J. S. Sadowsky, N. Butts, and W. A. Brown, "MUOS point-to-point power control," in *Proceedings of the MILCOM 2009 - 2009 IEEE Military Communications Conference*, Boston, MA, USA, October 2009.

[6] A. Vanelli-Coralli, G. E. Corazza, M. Luglio, and S. Cioni, "The ISICOM architecture," in *Proceedings of the 2009 International Workshop on Satellite and Space Communications*, IEEE, Tuscany, Italy, pp. 104–108, September 2009.

[7] J. Pulliam, Y. Zambre, A. Karmarkar, V. Mehta, and M. Everett, "TSAT network architecture," in *Proceedings of the MILCOM 2008 - 2008 IEEE Military Communications Conference*, San Diego, CA, USA, November 2008.

[8] C. Jiang, X. Wang, J. Wang, H.-H. Chen, and Y. Ren, "Security in space information networks," *IEEE Communications Magazine*, vol. 53, no. 8, pp. 82–88, 2015.

[9] D. He, X. Li, S. Chan, J. Gao, and M. Guizani, "Security analysis of a space-based wireless network," *IEEE Network*, vol. 33, no. 1, pp. 36–43, 2019.

[10] Q. Yang, X. Kaiping, X. Jie, W. Jiajie, L. Fenghua, and Y. Nenghai, "AnFRA: anonymous and fast roaming authentication for space information network," *IEEE Transactions on Information Forensics & Security*, vol. 14, no. 2, pp. 486–497, 2018.

[11] H. Arshad, M. Nikooghadam, S. Avezverdi, and M. Nazari, "Design and FPGA implementation of an efficient security mechanism for mobile pay-tv systems," *International Journal of Communication Systems*, vol. 30, no. 15, Article ID e3305, 2017.

[12] A. Ostad-Sharif, A. Babamohammadi, D. Abbasinezhad-Mood, and M. Nikooghadam, "Efficient privacy-preserving authentication scheme for roaming consumer in global mobility networks," *International Journal of Communication Systems*, vol. 32, no. 5, Article ID e3904, 2019.

[13] A. Irshad, M. Sher, M. S. Faisal, A. Ghani, M. Ul Hassan, and S. Ashraf Ch, "A secure authentication scheme for session initiation protocol by using ECC on the basis of the tang and liu scheme," *Security and Communication Networks*, vol. 7, no. 8, pp. 1210–1218, 2014.

[14] L. Harn and J. Ren, "Generalized digital certificate for user authentication and key establishment for secure communications," *IEEE Transactions on Wireless Communications*, vol. 10, no. 7, pp. 2372–2379, 2011.

[15] C. Meshram, "An efficient id-based cryptographic encryption based on discrete logarithm problem and integer factorization problem," *Information Processing Letters*, vol. 115, no. 2, pp. 351–358, 2015.

[16] C. Meshram, S. A. Meshram, and M. Zhang, "An id-based cryptographic mechanisms based on GDLP and IFP," *Information Processing Letters*, vol. 112, no. 19, pp. 753–758, 2012.

[17] H. Guo, Y. Gao, T. Xu, X. Zhang, and J. Ye, "A secure and efficient three-factor multi-gateway authentication protocol for wireless sensor networks," *Ad Hoc Networks*, vol. 95, Article ID 101965, 2019.

[18] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *Journal of the ACM*, vol. 56, no. 6, pp. 1–40, 2009.

[19] J. Zhang, Z. Zhang, J. Ding, M. Snook, and O. Dagdelen, "Authenticated key exchange from ideal lattices," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 719–751, Springer, Berlin, Germany, 2015.

[20] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.

[21] H. Cruickshank, "A security system for satellite networks," in *Proceedings of the Fifth International Conference on Satellite Systems for Mobile Communications and Navigation, 1996*, pp. 187–190, London, UK, May 1996.

[22] M.-S. Hwang, C.-C. Yang, and C.-Y. Shiu, "An authentication scheme for mobile satellite communication systems," *ACM SIGOPS Operating Systems Review*, vol. 37, no. 4, pp. 42–47, 2003.

[23] Y.-F. Chang and C.-C. Chang, "An efficient authentication protocol for mobile satellite communication systems," *ACM*

SIGOPS Operating Systems Review, vol. 39, no. 1, pp. 70–84, 2005.

[24] G. Zheng, H.-T. Ma, C. Cheng, and Y.-C. Tu, "Design and logical analysis on the access authentication scheme for satellite mobile communication networks," *Iet Information Security*, vol. 6, no. 1, pp. 6–13, 2012.

[25] W. Zhao, A. Zhang, J. Li, X. Wu, and Y. Liu, "Analysis and design of an authentication protocol for space information network," in *Proceedings of the MILCOM 2016 - 2016 IEEE Military Communications Conference*, Baltimore, MD, USA, November 2016.

[26] Q. Feng, H. Debiao, Z. Sherali, K. Neeraj, and L. Kaitai, "Ideal lattice-based anonymous authentication protocol for mobile devices," *IEEE Systems Journal*, vol. 13, no. 3, pp. 2775–2785, 2019.

[27] Z. Brakerski, A. Langlois, C. Peikert, O. Regev, and D. Stehlé, "Classical hardness of learning with errors," 2013, https://arxiv.org/abs/1306.0281.

[28] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 1–23, Springer, Berlin, Germany, 2010.

[29] J. Ding, X. Xie, and X. Lin, "A simple provably secure key exchange scheme based on the learning with errors problem," *IACR Cryptology EPrint Archive*, vol. 2012, p. 688, 2012.

[30] J. Ding, S. Alsayigh, J. Lancrenon, R. V. Saraswathy, and M. Snook, "Provably secure password authenticated key exchange based on rlwe for the postquantum world," in *Proceedings of the Cryptographers Track at the Rsa Conference*, Cham, Germany, 2017.

[31] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," *Journal of the ACM*, vol. 60, no. 6, pp. 1–35, 2013.

[32] S. D. Galbraith, K. G. Paterson, and N. P. Smart, "Pairings for cryptographers," *Discrete Applied Mathematics*, vol. 156, no. 16, pp. 3113–3121, 2008.