sensors

MDPI

*Article*

# On-Demand Anonymous Access and Roaming Authentication Protocols for 6G Satellite–Ground Integrated Networks

**Ya Tao [1], Haitao Du [2], Jie Xu [1,*], Li Su [2] and Baojiang Cui [1]**

[1] School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China; taoya@bupt.edu.cn (Y.T.); cuibj@bupt.edu.cn (B.C.)
[2] China Mobile Research Institute, Beijing 100032, China; duhaitao@chinamobile.com (H.D.); suli@chinamobile.com (L.S.)
* Correspondence: cheer1107@bupt.edu.cn; Tel.: +86-158-1121-0810

**Abstract:** Satellite–ground integrated networks (SGIN) are in line with 6th generation wireless network technology (6G) requirements. However, security and privacy issues are challenging with heterogeneous networks. Specifically, although 5G authentication and key agreement (AKA) protects terminal anonymity, privacy preserving authentication protocols are still important in satellite networks. Meanwhile, 6G will have a large number of nodes with low energy consumption. The balance between security and performance needs to be investigated. Furthermore, 6G networks will likely belong to different operators. How to optimize the repeated authentication during roaming between different networks is also a key issue. To address these challenges, on-demand anonymous access and novel roaming authentication protocols are presented in this paper. Ordinary nodes implement unlinkable authentication by adopting a bilinear pairing-based short group signature algorithm. When low-energy nodes achieve fast authentication by utilizing the proposed lightweight batch authentication protocol, which can protect malicious nodes from DoS attacks. An efficient cross-domain roaming authentication protocol, which allows terminals to quickly connect to different operator networks, is designed to reduce the authentication delay. The security of our scheme is verified through formal and informal security analysis. Finally, the performance analysis results show that our scheme is feasible.

**Keywords:** satellite–ground integrated network; 6G; privacy preserving; authentication protocol

check for updates

**Citation:** Tao, Y.; Du, H.; Xu, J.; Su, L.; Cui, B. On-Demand Anonymous Access and Roaming Authentication Protocols for 6G Satellite–Ground Integrated Networks. *Sensors* **2023**, *23*, 5075. https://doi.org/10.3390/s23115075

Academic Editors: Yingjie Jay Guo and Ramon Gonzalez Carvajal

Received: 23 April 2023
Revised: 22 May 2023
Accepted: 23 May 2023
Published: 25 May 2023

**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

## 1. Introduction

The fifth generation (5G) of wireless communication technology has promoted the development of the Internet of Things, automatic driving, virtual reality [1], etc. However, challenges still exist [2,3]. Although the terrestrial network has developed unprecedentedly, the seamless coverage of global heterogeneous networks has not been achieved. It is difficult for users to enjoy high-quality network services in many underserved areas (such as mountains, oceans, etc.) and in more than 50% of countries [4]. Oriented toward the 6G network, satellite–ground integrated networks (SGIN), which could provide global coverage and integrated management of satellite and terrestrial networks, have become a hot topic of current research [5–7]. In recent years, Amazon, SpaceX, and other major manufacturers are making great efforts to build satellite networks, which are expected to form a satellite communication system with a network capacity of 10 Tbps [8], providing reliable communication services for SGIN. With 6G, satellites will obtain more computing power through mobile edge computing (MEC) [9] and they will therefore be able to undertake heavier computing tasks.

Nevertheless, due to the high cost of satellite launches and maintenance, it is difficult for large satellite operators to hold all the satellites. There will still be many satellite operators maintaining a small number of satellites and providing personalized services.

Owing to the short coverage time per satellite, users may need to switch operators or switch to a faster terrestrial network. In order to support global coverage, it is important to encourage operators of SGINs to collaborate with each other and provide cross-domain services. In a word, a SGIN is a complex heterogeneous network that combines multiple operators with global coverage.

As shown in Figure 1, the SGIN architecture consists of three segments, i.e., a Satellite Node Segment, Ground Network Segment, and Access User Segment. The Satellite Node Segment consists of geosynchronous earth orbit (GEO) satellites, medium earth orbit (MEO) satellites, and low earth orbit (LEO) satellites. Compared to MEOs and GEOs, LEOs are closer to the ground and therefore have less communication overheads. Thus, only LEO authentication is considered in our architecture. These satellites can be transparent satellites or regenerative satellites. Transparent satellites are only responsible for transmitting messages. While regenerative satellites, which are equipped with gNB-DU, can be applied as a part of a 6G base station to process messages. The Ground Network Segment includes heterogeneous networks, such as large-scale servers, base stations, and satellite ground stations (GSs). Among them, the satellite network operators maintain the satellite ground network, including GSs and network control centers (NCCs). The terrestrial mobile operators maintain terrestrial networks, including the next-generation NodeBs (gNBs) and 6G core networks (6GCs). The Access User Segment consists of the user equipment (UE) in heterogeneous networks, such as mobile UE, IoT UE, and marine UE, which can be located on the ground, in suburban areas (areas with poor signal), at sea, in mountains, and so on. Obviously, these devices have different computing capabilities and diverse security requirements. They should choose suitable network domains and operators according to their own needs.

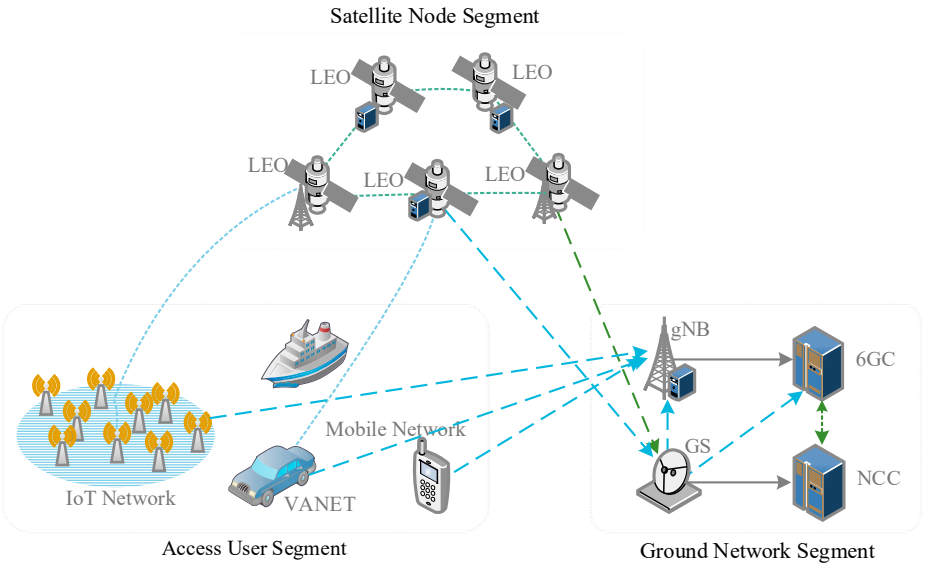


**Figure 1.** SGIN overall architecture.

Since the information in a SGIN is transmitted through a public wireless channel, the user's information is vulnerable to malicious attacks. Access authentication is the first step for UE to connect to the core network, so as to guarantee mutual authentication, forward and backward security, and be resistant to typical attacks. Unlike existing terrestrial networks, there is a large propagation delay (more than 10 ms) between satellite and terrestrial networks. In the traditional satellite network authentication scheme, the satellite generally transmits the forwarding messages [10], which causes a transmission delay in communication at least four times that of satellite and terrestrial networks [11]. Therefore, addressing efficient and secure authentication in SGINs is the first issue.

To make things even more challenging, heterogeneous devices need to be authenticated in this network. Based on our above architecture, the different types of user equipment and

their diverse security requirements pose great challenges to the research of SGIN access authentication. At present, terrestrial networks usually use the authentication and key agreement (AKA) protocol to implement UE access authentication. The 5G AKA proposes to protect the privacy of users by encrypting their permanent identity and transmitting encrypted SUCI. However, there is no privacy authentication for heterogeneous terminals that access via satellite networks. For diverse users in SGIN, a targeted authentication model is required. For example, for IoT nodes with a large scale and relatively weak computing power, anonymous batch authentication with lower complexity is needed to achieve a faster speed. However, the existing batch authentication schemes do not effectively protect user privacy [11] or incur excessive costs [12]. While individual end-users are more computationally capable and care a lot about their privacy and security, they can choose protocols with higher security and better protect their privacy. That is to say, it is difficult to unify these users with different needs.

Meanwhile, due to the existence of multiple operators, SGIN secure roaming authentication also deserves attention. The current situation of many large satellite service providers (such as SpaceX and OneWeb) [13,14] as well as small providers cannot achieve continuous global coverage of signals, which causes many inconveniences for UE to enjoy diverse and stable services. As mentioned above, different network services in 6G may be provided by various operators, and the core network (CN) of satellite and terrestrial networks may be disparate. If a new satellite or terrestrial base station providing service to a subscriber originates from a different operator, the new operator needs to perform roaming authentication with the subscriber. However, the existing 3GPP roaming authentication approach [15] is not applicable for SGINs, because satellite transmission will bring significant time delays. How to provide fast multi-operator cross-network roaming authentication for subscribers is also a key challenge.

Motivated by these challenges, we present an on-demand authentication protocol model for SGINs. In the model, we propose protocols for mutual authentication of UE and satellites, to reduce transmission overheads. Our protocols propose different access authentication scenarios for different users' performance and security requirements, so that UE can have on-demand access. Among these, UE privacy is protected to different degrees. In addition, a roaming authentication protocol is proposed for cross-domain roaming by different operators, in line with the mutual authentication between the UE and satellite. Compared with other related works, we give consideration to authentication and roaming, and provide on-demand access authentication for terminals with different abilities and needs. The contributions of this paper are as follows:

- On-demand privacy-preserving authentication protocols for SGIN: We propose an on-demand access authentication protocol for satellite networks in SGINs based on the protocol architecture. For UE with high security and privacy requirements, an anonymous unlinkable authentication protocol is proposed which ensures UE's unlinkability. For large numbers of UE with demand for short delay times, a batch authentication protocol is proposed. The protocol supports rebatch authentication after authentication failure and can effectively alleviate DoS attacks.
- A lightweight roaming authentication protocol for SGIN: The roaming authentication protocol provides a strategy for roaming between different operator networks for satellite-connected UE in SGINs, which needs to pre-negotiate with the corresponding core network after the last authentication is completed. The UE only needs to complete mutual authentication with the satellite when roaming, thus reducing the propagation delay.

The remainder of this article is organized as follows: We review the related work in Section 2. Then, we introduce the prior knowledge involved in the protocol and SGIN system model in Section 3. The details of our scheme are presented in Section 4. The security of the proposed scheme is proven in Section 5. We compare the performance of related schemes in Section 6. Finally, we summarize the article in Section 7.

## 2. Related Work

In recent years, researchers have made many contributions to the access authentication of SGINs. Nguyen et al. [2] provided a systematic overview of 6G security and privacy issues. They analyzed the security architecture of 6G and considered the new open authentication protocols (e.g., satellite, sea area) for non-3GPP networks, as one of the priorities for 6G network access security. Zhao et al. [16] made use of the broadcasting function of satellites to propose an efficient and lightweight access authentication scheme, to prevent the burden of "message storms" on satellite authentication. Cui et al. [17] proposed an authentication scheme for heterogeneous B5G networks (including satellite networks) and proposed a user detection scheme based on trust evaluation. Guo et al. [18] proposed an anonymous mutual authentication scheme based on RLWE, which can resist attacks based on quantum computing and guarantee efficiency and security in the post-quantum era. Yao et al. [19] proposed a mutual authentication protocol named IMAS, which introduced group management forms to the satellite, to accomplish the multicast authentication between UE and satellites. Guo et al. [20] proposed an access authentication protocol based on elliptic curve cryptography (ECC), which included three entities: the UE, satellite, and ground station. In addition, the scheme designed a batch handover scheme to reduce overheads.

In the face of the high privacy requirements of users of 6G, anonymous authentication based on aliases or group signature authentication can be used. Although the alias mechanism [21] has good performance in information transmission and privacy protection, users need to store a large number of certificates, which leads to a large amount of overheads [22]. The traditional group signature message will bring some transmission overheads. Boneh et al. [23] proposed a short group signature (SGS) scheme, which allows bilinear pairing to be widely used in modern cryptography. Wasef and Shen [12] proposed a batch authentication scheme based on SGS, so that SGS can be applied to a large number of user authentication scenarios. Alamer [24] proposed a scheme to transform the SGS signature algorithm into a signcryption algorithm, which ensures message integrity and confidentiality.

Owing to the large number of user nodes, researchers have proposed batch authentication. Huang et al. [25] proposed a fast anonymous batch authentication scheme for vehicle networking, which can verify multiple requests at a time and negotiate a session key with the vehicle through broadcast messages. Considering that a failure of batch authentication will lead to the failure of all authentication of a batch of nodes, the author also proposed to rebatch authentication to prevent possible DoS attacks. Lai et al. [26] proposed a lightweight group authentication scheme for M2M networks, and the UE in their scheme can also accomplish rebatch authentication by dichotomizing. Mahmood et al. [27] proposed ECC-based lightweight security without using a batch verification method (LSWBVM). Their method can authenticate a large number of request messages and verify messages one by one.

Due to the presence of multiple SGIN operators, cross-domain roaming authentication has become a research direction. Xue et al. [28] proposed a lightweight group key negotiation protocol based on $(t, n)$ secret sharing and proposed a cross-domain handover authentication scheme. Considering the problems of different operators in the converged network, Liu et al. [29] proposed a decentralized anonymous authentication scheme applied to the cross-operator satellite service scenario, which can carry out cross-domain fast handover authentication and ensure the fairness of billing. Yang et al. [30] proposed an authentication scheme based on group signatures and completed cross-domain roaming of users through advanced negotiation between ground stations and satellites. Guo et al. [31] proposed a new secure roaming authentication and key negotiation protocol called SRAKN, which enables efficient and fast roaming between users, satellites, and foreign terrestrial control stations (FTCS), and finally negotiates secure session keys. Yang et al. [32] proposed a fast handover authentication protocol for high-speed mobile terminals for railways in SGINs. Their method forms a temporary group of terminals in the same compartment and completes the handover based on preset information. Table 1 shows a summary of the

related works described in this paper. We study their schemes according to four aspects of these related works: performance objective, algorithm/scheme, scenario, and motivation.

**Table 1.** Summary of related works *.

| Ref. | Performance Objective | Algorithm/Scheme | Scenario | Motivation |
|---|---|---|---|---|
| [2] | 6G security architecture | - | 6G | Research on 6G security requirements, analyzing 6G security architecture |
| [16] | access authentication | bilinear pairing | SIN | Research on group authentication using satellite broadcasting |
| [17] | unified authentication architecture | - | heterogeneous B5G | Research on a heterogeneous unified authentication architecture |
| [18] | access authentication | RLWE | post-quantum era | Research on access authentication against quantum cryptography |
| [19] | access authentication | bilinear pairing | SAGIN | Research on satellite multicast authentication protocols |
| [20] | access authentication | ECC | SIN | Research on secure access authentication protocols |
| [23] | group signature | SGS | - | Research on unlinkable group signatures |
| [12] | access authentication | SGS | - | Research on batch authentication scheme for SGS |
| [24] | access authentication | SGS | - | Research on signcryption scheme for SGS |
| [25] | batch authentication | ECC | VANET | Research on fast and anonymous batch and rebatch authentication |
| [26] | batch authentication | ECC | M2M | Research on lightweight group and rebatch authentication |
| [27] | access authentication | ECC | IoT | Research on authentication that quickly validates massive requests |
| [28] | access and handover authentication | secret sharing | multi-operator networks | Research on cross-domain handover authentication |
| [29] | access and roaming authentication | ECC and SBC | SGIN | Research on decentralized authentication and charging fairness |
| [30] | access and roaming authentication | SGS | SIN | Research on a user cross-domain roaming scheme |
| [31] | access and roaming authentication | ECC | SIN | Research on a secure roaming and key agreement scheme |
| [32] | access and handover authentication | CRT | HSR and SGIN | Research on efficient authentication and handover of terminals for railways |

* SIN: space information network. B5G: beyond 5G. RLWE: ring learning with errors. SAGIN: space–air–ground integrated network. ECC: elliptic curve cryptography. SGS: short group signature. VANET: vehicular ad hoc network. M2M: machine to machine. IoT: Internet of things. SBC: smart billing contract. CRT: Chinese remainder theorem. HSR: high-speed rail. -: there is no relevant statement about this work.

## 3. Preliminaries and System Model

In this section, we first review the mathematical preliminaries of our scheme, including bilinear pairing and ECDSA. Then, we present the system model and security requirements for SGIN networks. In particular, we describe the model of the SGIN protocol in detail. We describe the security model adopted by the SGIN system and provide security requirements to meet the security assumptions.

### *3.1. Bilinear Pairing*

Let $\mathbb{G}$ be the addictive cyclic group of the prime order $q$ and $\mathbb{G}_T$ be the multiplicative cyclic group of the same prime order $q$. Let $P$ be a generator of group of $\mathbb{G}$. The bilinear pairing $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ satisfies:

- Bilinearity: $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$, where $a, b \in Z_q^*$ and $P,\ Q\ \in \mathbb{G}$;
- Nondegeneracy: $\exists P,\ Q\ \in \mathbb{G}$, let $\hat{e}(P, Q) \neq 1$;
- Computability: $\forall P,\ Q\ \in \mathbb{G}$, $\hat{e}(P, Q)$ can be calculated efficiently.

### *3.2. ECDSA*

The elliptic curve digital signature algorithm (ECDSA) is a simulation of the digital signature algorithm (DSA) based on the elliptic curve cryptography (ECC) algorithm. Let $q$, $P$ be the public parameters as discussed above. ECDSA can be divided into the following three steps [33]:

- $ECKeyGen(\cdot)$: Select a random integer $k$ as the private key, where $k \in [1, q-1]$. Compute the public key $K = kP$. The entity can obtain the key pair $(k, K)$.
- $ECSign(M)$: When receiving the message $M$, the entity first selects a random number $r \in [1, q-1]$, then generates $rP$. Set $(x, y) = rP$ and $R = x\ mod\ q$. Compute $h = SHA1(M)$, then generate $S = (h + k \cdot R)r^{-1}\ mod\ q$. Finally, the entity obtains the signature $(R,\ S)$. It is noted that if one of $R$ or $S$ equals 0, this algorithm should be rerun.
- $ECVerify(M, (R, S))$: When the entity receives $M$ and $(R, S)$, it first checks whether $R$ and $S$ are in the range $[1, q-1]$. If yes, the entity calculates $h' = SHA1(M)$, $(x', y') = h' \cdot S^{-1}P + R \cdot S^{-1}K$ and $R' = x'\ mod\ q$. If $R' = R$, the entity can accept the signature.

### *3.3. System Model*

Referring to the overall architecture shown in Figure 1, the satellite network members for the SGIN in this article include the UE, regenerative LEO with gNB-DU (distributed unit), GS with gNB-CU (centralized unit), and NCC; the terrestrial network members include the UE, gNB, and 6GC. Due to the heterogeneous 6G environment and different user needs, we put forward two different scenarios of satellite network access: (1) An Unlinkable Authentication Scenario, which requires higher anonymity and unlinkability, based on a short group signature [23]; (2) a Batch Authentication Scenario, which includes massive lightweight UE, where their requirement for anonymity is relatively low. We designed different access authentication protocols for the two scenarios. The UE can select corresponding authentication modes based on their requirements. In addition, the proposed scheme provides roaming authentication methods for users who need to roam across domains.

The protocol model of the SGIN is shown in Figure 2. The protocol involved in this paper is represented by orange lines. The dashed lines mean that the channel between two entities is insecure, and the solid lines are the contrary. The UE, connected to the satellite, can access the NCC through the two protocols designed in the following section. In addition, since regenerative satellites are equipped with gNBs, the UE can also access the 6G core network through regenerative satellites, transparent satellites, or gNBs, using the AKA protocol. When it is necessary to handover satellites due to their movement, the satellites plan the handover strategy independently. This process is transparent to the UE. When the UE needs to roam between networks of the same operator or roam across the

networks of different operators, they have to use the protocols of the presented roaming authentication phase according to the situation. The protocol architecture aims to address the required access methods and performance for heterogeneous terminals.

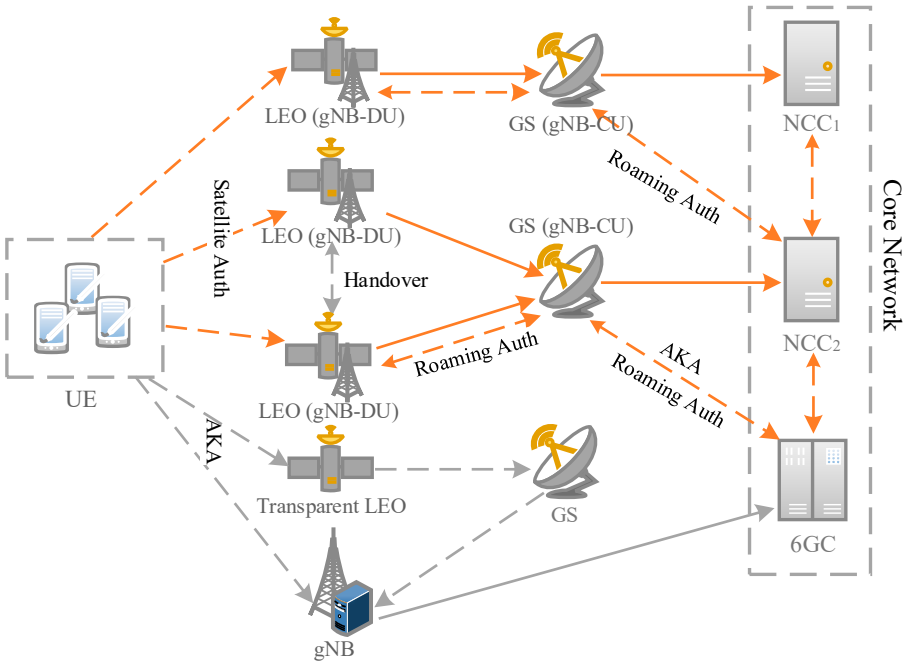


**Figure 2.** SGIN protocol model.

### *3.4. Security Requirement*

The proposed protocol has the following security assumptions:

- Assuming that the NCCs and 6GCs are trusted by UE, LEOs, and GSs. During the initialization phase, NCCs can send secret parameters for future use to the UE and LEOs over trusted channels (e.g., offline channels);
- Assuming that during satellite authentication, the channels between NCCs and GSs, GSs and LEOs, and NCCs and NCCs are trusted, it can be built using SSL or TLS. While using AKA, the satellite channels from UE to gNBs are untrusted, and the channels between gNBs and 6GCs are trusted;
- It is assumed that no trust relationship has been established between the UE and LEOs before access authentication;
- It is assumed that in an unlinkable authentication scenario, regenerative LEOs may track the message of the UE and try to reveal its real information;
- The proposed protocol also needs to meet the following security requirements:
- Forward and Backward Secrecy: Due to the changes in the geographical location and requirements of UE, the proposed scheme should ensure forward and backward security; that is, an attacker cannot obtain the current session key through the information of the previous session [34]. In addition, if the current session is compromised, the attacker cannot affect the security of the previous channel [35];
- Mutual Authentication: The proposed scheme should satisfy mutual authentication; that is, regenerative LEOs can detect and refuse access to an illegal UE. In addition, a UE can also know the legitimacy of access nodes in the system, to avoid potential malicious attacks;
- Key Establishment: The proposed scheme should ensure that the session keys negotiated by the protocol are only shared between the UE and regenerative LEOs;
- User Privacy: In the unlinkable authentication scenario, the UE requires unlinkability; that is, others cannot know whether the information comes from the same UE. In the

batch authentication scenario and roaming authentication phase, the UE is linkable, but others will not be able to learn its identity information.

## 4. The Proposed Scheme

In this section, we give a detailed description of the scheme, which consists of four phases: an initialization phase, anonymous authentication phase, roaming authentication phase, and user revocation phase. Without losing generality, and in order to be more intuitive, we will only focus on a certain set of UE and the corresponding LEOs.

### *4.1. Initialization Phase*

In the initialization phase, the satellite core network servers (i.e., NCCs) first input the security parameter $\lambda \in N$ and generate the system master key $s \in Z_q^*$. Then, the NCC generates $(sk_{LEO}, pk_{LEO})$ for LEOs based on ECC. Moreover, the NCC outputs the public parameter $params = \langle q, \mathbb{G}, \mathbb{G}_T, P, g, \hat{e}, H_1, H_2 \rangle$, where $\mathbb{G}$ is the addictive cyclic group with the generator element $(P, g)$ and the order $q$, $\mathbb{G}_T$ is the multiplicative cycle group with the same order $q$, $\hat{e}$ is the bilinear pairing $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, and $H_1 : \{0,1\}^* \to Z_q^*$ and $H_2 : \{0,1\}^* \times \mathbb{G} \to Z_q^*$ are hash functions. At the same time, the 6GC completes the initial key configuration with the UE. The detailed access authentication protocols for terrestrial networks are beyond the scope of this paper.

After generating the public parameter, the NCC implements Algorithm 1 to generate the group public key $gpk : \langle g, h, u, v, \omega \rangle$ and group master secret key $gmsk : \langle \xi_1, \xi_2 \rangle$, in which $gpk$ needs to be published in an open channel and $gmsk$ needs to be kept secret by the NCC. For each $UE_i$ in the group, the NCC generates $gsk_i : \langle ID_i, \varphi_i, A_i \rangle$ and sends $gsk_i$ to the corresponding UE. The UE needs to use $gsk_i$ as its private key and not disclose it to anyone.

**Algorithm 1:** Initialization

**Input**: a group of user identity, a number of users $N$
**Output**: $(g, h, u, v, \omega, \xi_1, \xi_2)$, $(ID_i,\ \varphi_i, A_i)$

1: Select random numbers $u, v, h \in G$
2: Select random numbers $\xi_1, \xi_2,\ \gamma \in Z_q^*$ such that $\xi_1 \cdot u = \xi_2 \cdot v = h$
3: Sets $\omega = \gamma g$
4: **for** all $UE_i$ with identity $ID_i$ **do**
5: Select a random number $\varphi_i \in Z_q^*$
6: Set $A_i = \frac{1}{\gamma + \varphi_i} g$
7: Store the tuple $(ID_i, \varphi_i, A_i)$
8: **end for**
9: **return** $g,\ u,\ v,\ h,\ \omega,\ \xi_1, \xi_2, (ID_i, \varphi_i, A_i)$

Additionally, the NCC generates the necessary parameters for batch authentication. The steps for batch authentication initialization are shown below:

1. NCC selects a random number $\varepsilon_{LEO} \in Z_q^*$ for each LEO, and generates a batch master key $BMK = s^{-1}P$ and batch public key $BPK_{LEO} = \varepsilon_{LEO}P$ for LEOs. The LEOs should keep $BMK$ and $\varepsilon_{LEO}$ secret.
2. The NCC selects a random number $x_i \in Z_q^*$ for each $UE_i$. Then, the NCC calculates the batch authentication key $BK_i = x_i sP$, $RK_i = e(x_i P, P)$ for the UE. Therefore, the batch authentication key of $UE_i$ is $buk_i : \langle BK_i,\ RK_i \rangle$, and the UE should keep $buk_i$ secret.

Finally, the NCC sends $(\varepsilon_{LEO},\ BMK,\ BPK_{LEO},\ sk_{LEO}, pk_{LEO})$ to the LEOs over a secure channel (e.g., offline channel). Additionally, the NCC sends $(gsk_i, gpk, buk_i)$ to the corresponding UE securely, and saves $ID_i$ and $gsk_i$ in a local user key list (UKL).

### *4.2. Anonymous Authentication Phase*

In this section, we show two scenarios according to the different needs of the UE: an unlinkable authentication scenario and batch authentication scenario. Each UE gives all

the key information required for the two scenarios during the initialization phase, so it can switch scenarios when needed, without reregistering.

#### 4.2.1. Unlinkable Authentication Scenario

In this scenario, we refer to Boneh's [23] short group signature (SGS) algorithm, which is one of the most famous group signatures. The UE in SGS can randomly generate a temporary identity (TID) that is irrelevant to the real $ID_i$. Owing to the unlinkable anonymity of SGS, the UE can use different TIDs in different sessions, and no other entity knows that these TIDs belong to the same UE. The specific protocol process is shown in Figure 3, and its steps are as follows:

1. When the UE wants to access the NCC, the message $M_i = (TID_i||ID_{LEO}||g^{r_1}||TS_1)$ needs to be constructed, where $r_1 \in Z_q^*$ is a random number, $g^{r_1}$ is part of the session key, and $TS_1$ is a timestamp that can resist reply attacks. Taking $M_i$, $gpk$, and $gsk_i$ as input, the UE implements Algorithm 2 and obtains the signature $\sigma_i = \left(T_1, T_2, T_3, c, s_\alpha, s_\beta, s_x, s_\delta, s_\mu\right)$, where $\hat{e}(A_i, g)$, $\hat{e}(h, g)$ and $\hat{e}(h, \omega)$ can be calculated and stored in advance. When the UE needs to revote its secret keys, it needs to recalculate $\hat{e}\left(\hat{A}_i, g\right)$ and $\hat{e}(h, \hat{\omega})$. After completing the construction of plaintext and signature, the UE sends $(M_i, \sigma_i)$ to an appropriate LEO.
2. When receiving s request from the UE, the LEO first authenticates the timestamp $TS_1$ in the message. The LEO generates the current timestamp $TS_1'$ and verifies $TS_1' - TS_1 < \Delta T$, where $\Delta T$ is adjusted according to different network conditions. If it does not meet the conditions, the LEO returns an error message. Otherwise, the LEO validates the accuracy of the signature using Algorithm 3. If the verification passes, the LEO selects the random number $r_2 \in Z_q^*$ and calculates the session key $SK = (g^{r_1})^{r_2}$. If the verification fails, an error message is returned. The LEO signs the message $M_i' = (TID_i||ID_{LEO}||g^{r_2}||TS_2)$ using $ECSign(M_i')$ to obtain the signature $\sigma_i'$. Then, the LEO sends the message $(M_i', \sigma_i')$ to the UE.
3. After receiving the message, the UE first verifies the validity of the timestamp; that is, whether the timestamp satisfies $TS_2' - TS_2 < \Delta T$. If it is valid, the message will be verified by the ECDSA. If verified, the UE calculates the session key $SK = (g^{r_2})^{r_1}$. The key negotiation between the two sides is complete.

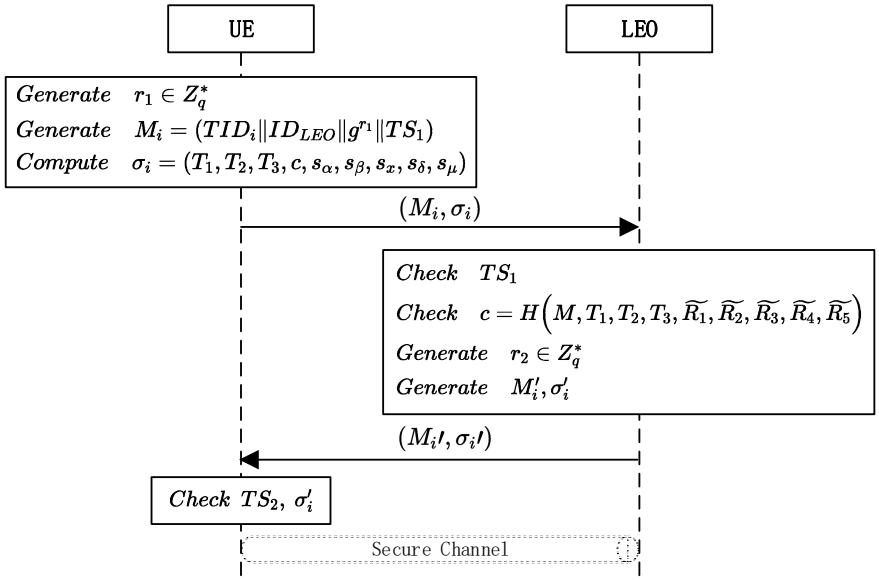


**Figure 3.** Authentication phase in the unlinkable authentication scenario.

**Algorithm 2:** Generating Signature

**Input:** $M_i, gpk, gsk_i$
**Output:** $\sigma_i$

Select random numbers $\alpha, \beta \in Z_q^*$
Set $T_1 = \alpha u$, $T_2 = \beta v$, $T_3 = A_i + (\alpha + \beta)h$
Set $\delta = \alpha\varphi_i$ and $\mu = \beta\varphi_i$
Select random numbers $r_\alpha,\ r_\beta, r_x, r_\delta, r_\mu \in Z_q^*$
Set $R_1 = r_\alpha u$, $R_2 = r_\beta v$, $R_4 = r_x T_1 - r_\delta u$, $R_5 = r_x T_2 - r_\mu v$,
$R_3 = \hat{e}(T_3, g)^{r_x} \hat{e}\Big(h, \Big(-r_\alpha - r_\beta\Big)\omega + \big(-r_\delta - r_\mu\big)g\Big)$
$= \hat{e}(A_i, g)^{r_x}\ \hat{e}(h, g)^{r_x(\alpha+\beta)}\ \hat{e}(h, g)^{-r_\delta - r_\mu}\ \hat{e}(h, \omega)^{-r_\alpha - r_\beta}$
Set $c = H(M, T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5) \in Z_q^*$
Set $s_\alpha = r_\alpha + c$, $s_\beta = r_\beta + c$, $s_x = r_x + cx$, $s_\delta = r_\delta + c$, $s_\mu = r_\mu + c$
**return** $\sigma_i = \Big(T_1, T_2, T_3, c, s_\alpha, s_\beta, s_x, s_\delta, s_\mu\Big)$

**Algorithm 3:** Verifying Message

**Input:** $M_i, \sigma, gpk$
**Output:** $\varnothing$

Set $\widetilde{R_1} = -cT_1 + s_\alpha u$, $\widetilde{R_2} = -cT_2 + s_\beta v$,
$\widetilde{R_4} = s_x T_1 - s_\delta u$, $\widetilde{R_5} = s_x T_2 - s_\mu v$,
$\widetilde{R_3} = \hat{e}(s_x T_3, g)\hat{e}(cT_3, \omega)\hat{e}(h, \omega)^{-s_\alpha - s_\beta}\hat{e}(h, g)^{-s_\delta - s_\mu}\hat{e}(g, g)^{-c}$
**If** $c = H(M_i, T_1, T_2, T_3, \widetilde{R_1}, \widetilde{R_2}, \widetilde{R_3}, \widetilde{R_4}, \widetilde{R_5})$ **then**
The signature $\sigma_i$ is valid
**else**
Reject the signature
**end if**

#### 4.2.2. Batch Authentication Scenario

The traditional short group signature scheme uses a batch group signature (BGS) to complete batch authentication. However, due to the insufficient computing power of some devices in 6G heterogeneous networks and the massive terminals, we designed a novel batch authentication protocol to meet the needs of these terminals. In this scenario, users can generate a TID, but it is traceable. A set of UE send their batch authentication message to the LEO, which authenticates all parameters uniformly. If the first batch authentication is successful, the LEO authenticates them and continues the authentication process. If the first authentication fails, a rebatch is required. The specific protocol process is shown in Figure 4, and the detailed steps are as follows:

1. The UE selects random numbers $r_1, k_i, b_i \in Z_q^*$, then calculates $B_i = b_i P$, $U_i = b_i \cdot BPK_{LEO}$. The UE sets the access message as $M_i = (TID_i || ID_{LEO} || B_i || g^{r_1} || TS_1)$, where $TS_1$ is a timestamp. Then, UE calculates the hash value $h_i = H_1(M_i)$ of $M_i$, and sets the batch authentication key $BAK_i = k_i h_i \cdot BK_i$ and $RAK_i = (k_i \cdot RK_i) \oplus H_1(B_i || U_i\ || TS_1)$. Then, the UE receives the signature of batch authentication $\sigma_i = (BAK_i, RAK_i)$. After signing the message, the UE sends $(M_i, \sigma_i)$ to the target LEO;
2. When the LEO receives $(M_i, \sigma_i)$ from the UE, it first checks the validity of the timestamp $TS_1$. If $TS_1$ is legal, the LEO sets $h_i' = H_1(M_i)$, $U_i = \varepsilon_{LEO} \cdot B_i$ and $H_1(B_i || U_i || TS_1)$, then calculates the following equation:

$$e(\sum_{i=1}^{n} BAK_i, BMK) = \prod_{i=1}^{n} h_i(RAK_i \oplus H_1(B_i || U_i || TS_1)) \quad (1)$$

If Equation (1) is true, $n$ UE in the batch authentication group is valid. Otherwise, it means that there are invalid messages in this group. Batch authentication has the advantage of reducing the computational overheads, but once an invalid request occurs in a batch, the authentication will fail. When a malicious attacker continuously

sends invalid information to implement DoS attacks, users may be unable to complete authentication for a long time. Therefore, a rebatch is required to protect the UE's QoS. The algorithm of "divide-and-conquer" (BVDC) [25] can be used. The LEO can use dichotomous validation for a batch authenticated UE, to find the UE that failed validation and return error messages. Although the rebatch may bring computation overheads, it is helpful for improving the overall system efficiency and increasing the verification success rate.

3. For UE that passes the authentication, the LEO constructs $M_i' = (TID_i||ID_{LEO}||g^{r_2}||TS_2)$, where $TS_2$ is a timestamp, $g^{r_2}$ is a session key parameter generated by LEO, and $r_2 \in Z_q^*$ is a number randomly generated. The LEO uses $ECSign(M_i')$ to generate the signature $\sigma_i'$ and send $(M_i', \sigma_i')$ to the UE.
4. After receiving the message, the UE first verifies the validity of the timestamp. The message is then verified by $ECVerify(M_i', \sigma_i')$. If the verification is successful, the UE calculates the session key $SK = (g^{r_2})^{r_1}$, and the LEO calculates the session key $SK = (g^{r_1})^{r_2}$. The key negotiation between the two sides is complete.

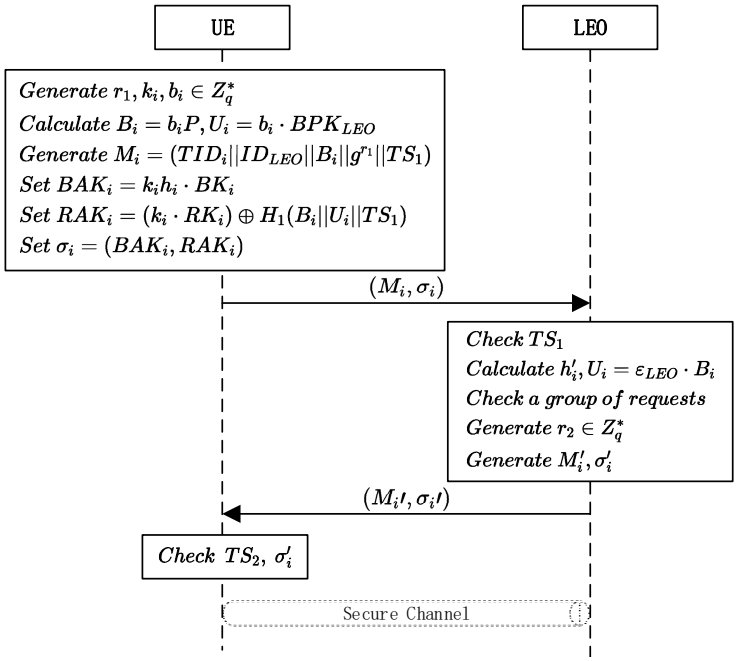


**Figure 4.** Authentication phase in the batch authentication scenario.

### *4.3. Roaming Authentication Phase*

When the UE needs to roam across the network, due to changes in geographical location or network conditions, the roaming authentication phase can be completed. The proposed scheme designs a lightweight roaming authentication protocol to meet the needs of UE. For UE that need to change core network, they must be re-authenticated and negotiate a new session key. To reduce the overheads of roaming authentication, the UE should perform pre-negotiation after the initial authentication or the last roaming.

**Pre-negotiation Phase**: The UE first collects optional satellite information, and then sends its $TID_i$ and $ID_{tLEO}$ to the source core network, namely sCN (i.e., NCCs or 6GCs) through the sLEO, to request roaming authentication tokens. After receiving the UE's request information, the sCN selects the generation key $K_1$. Then, sCN calculates $Ticket_i = SENC_{KDF(psk)}(TID_i||ID_{tLEO}||K_1||ET)$, where $SENC(\cdot)$ is a symmetric encryption function, $KDF(\cdot)$ is a key derivation function, $psk$ is a pre-shared key maintained between the CNs and LEOs, and $ET$ is the expiration time of the token. A secure channel is established between the UE and sLEO during the authentication phase, and secure channels exist between core networks. The sCN encrypts $(Ticket_i, K_1, ET)$ using the key $K_0$ stored between sCN and UE, then returns the message to the UE. When $Ticket_i$ expires or the UE discovers

new suitable LEOs, the UE needs to apply to the sLEO for new tokens. The LEO updates the token issued to the UE when the LEO or CN evaluates that it is necessary to change the *psk*. At the same time, if the tLEO does not belong to the sCN as *Case ii*, the sCN sends the $TID_i$ list to the tCN through the channel, then tCN sends $TID_i$ to tLEO. Otherwise, sCN sends the $TID_i$ list directly to tLEO as *Case i*. The specific negotiation process is shown in the upper part of Figure 5.

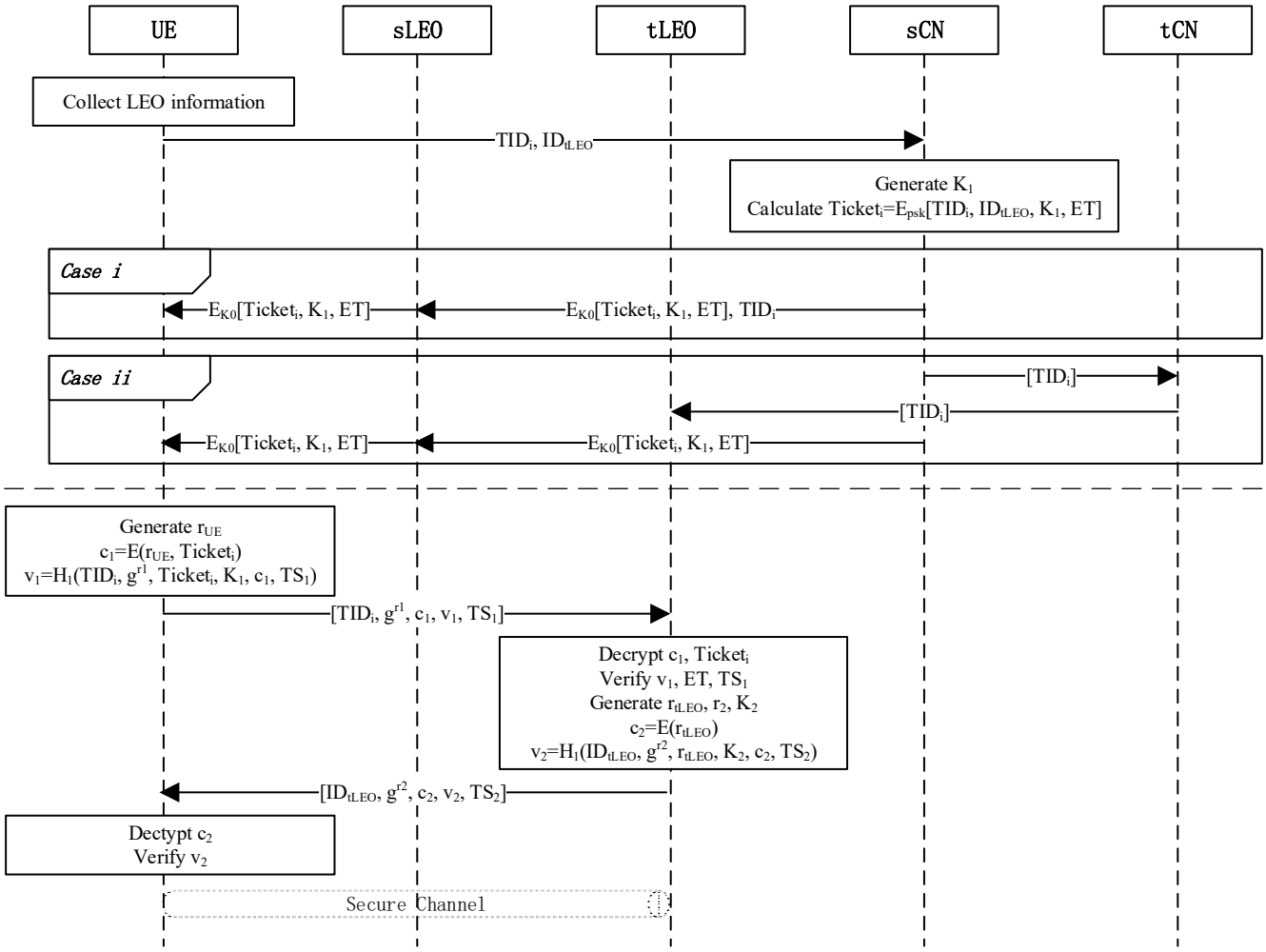


**Figure 5.** Pre-negotiation and roaming authentication phase in SGIN.

**Roaming Authentication Phase**: When the UE needs to roam, the process is as shown in the lower part of Figure 5. The following steps need to be completed:

1. The UE first generates a random number $r_{UE}$, then uses the public key $pk_{tLEO}$ of the tLEO to encrypt $c_1 = AENC_{pk_{tLEO}}(r_{UE}, Ticket_i)$, where $AENC(\cdot)$ is an asymmetric encryption algorithm based on ECC. Then, the UE selects a random number $r_1 \in Z_q^*$ and obtains a timestamp $TS_1$, sets $g^{r_1}$, and generates $v_1 = H_1(TID_i||g^{r_1}||Ticket_i||K_1||c_1||TS_1)$. Finally, the UE sends message $M_1 = (TID_i||g^{r_1}||c_1||v_1||TS_1)$ to the tLEO;
2. Upon receiving the message, the tLEO first checks the timestamp then decrypts $c_1$ using its private key $sk_{tLEO}$ in order to obtain $r_{UE}$ and $Ticket_i$. The tLEO decrypts $Ticket_i$ using $KDF(psk)$ and checks the ID and ET in the token. If it does meet the conditions, the tLEO generates $v_1'$ to verify whether it is equal to $v_1$. If verified, tLEO generates a random number $r_{tLEO}$ and generates $K_2 = KDF(K_1||r_{UE})$. Then, the tLEO calculates $c_2 = SENC_{K_2}(r_{tLEO})$, where $SENC(m)$ is a symmetric encryption algorithm. The tLEO selects a random number $r_2 \in Z_q^*$ and sets $g^{r_2}$. Finally, the

tLEO calculates $v_2 = H_1(ID_{tLEO}||g^{r_2}||r_{tLEO}||K_2||c_2||TS_2)$ and sends the message $M_2 = (ID_{tLEO}||g^{r_2}||c_2||v_2||TS_2)$ to the UE, where $TS_2$ is a timestamp;

3. While receiving the message, UE first checks the timestamp. If checks, UE calculates $K_2 = KDF(K_1||r_{UE})$. Then UE decrypts $c_2$ and gets $r_{tLEO}$. UE calculates $v_2'$ and checks whether $v_2' = v_2$. If it does meet, the new conversation between UE and the tLEO is established. UE and the tLEO get their new session keys $SK = (g^{r_2})^{r_1}$ and $SK = (g^{r_1})^{r_2}$.

### *4.4. User Revocation Phase*

In the case that the UE needs to quit the group or the system needs to revoke the illegal UE authentication in the unlinkable authentication, the algorithm in the user revocation phase is needed. For an illegal UE, the NCC has the right to disclose their real ID and other information through signatures they send out. The private key of the illegal UE can be calculated through the group master key $gmsk : \langle \xi_1, \xi_2 \rangle$ and $(T_1, T_2, T_3)$ in the signature. The NCC can find the real $ID_i$ of the UE by comparing with the user information in UKL. For an illegal UE that requests to quit the group, the NCC performs the operations described above. After that, the NCC creates a revocation list (RL) that contains the key $(\varphi_j, A_j, RK_j)$ of the $UE_j$ to be revoked. The NCC sends the RL to each LEO. The LEOs save the RL and periodically broadcast the latest RL′, which includes $(\varphi_j, A_j)$.

A UE that receives the RL′ updates its private key according to Algorithm 4, where $m$ is the total number of tuples in RL′. Unrevoked UE must run this algorithm until all UEs in RL′ are revoked. After completing the above steps, the unrevoked UE needs to update the pre-stored parameters $\hat{e}(\hat{A}_i, g)$ and $\hat{e}(h, \hat{\omega})$. In addition, for offline UE, they need to request the latest RL′ when they are online. Therefore, the revoked $UE_j$ cannot obtain a new $gsk_j$. The authentication will fail when the $UE_j$ participates in group signature authentication again. When participating in batch authentication, $UE_j$ will also be detected by the LEO, thus prohibiting its access to the network.

**Algorithm 4:** Unrevoked User Update Parameters

**Input:** $gsk_i, RL\prime = \left\{ x_j, A_j \middle| 1 \le j \le m \right\}$
**Output:** $(\varphi_i, \hat{A}_i)$

1: Update $g$ as $\hat{g} = A_j^* = \frac{1}{\varphi_j + \gamma} g$
2: Update $\hat{\omega} = g - \varphi_j \cdot A_j = \gamma \cdot \hat{g}$
3: Update the secret key as $\hat{A}_i = \frac{1}{\varphi_i - \varphi_j} \cdot A_j - \frac{1}{\varphi_i - \varphi_j} \cdot A_i = \frac{1}{\varphi_i + \gamma} \cdot \hat{g}$
4: **return** $(\varphi_i, \hat{A}_i)$

The user revocation process can be carried out offline, which reduces the burden on the UE and prevents delays caused by the LEO checking the RL operation during authentication. The performance analysis of Yang et al. [30] showed that the revocation mechanism of SGS is effective. For the protocol that is unlinkable, this phase allows the UE to exit the group, better managing the network. Although user revocation brings additional computational overheads, it is acceptable and necessary.

## 5. Security Verification

In this section, we use the ProVerif tool to conduct formal verification of the protocol. Then, we complete an informal security analysis.

### *5.1. Formal Analysis Using ProVerif*

We used the ProVerif tool to formalize the proposed protocol in two parts. ProVerif is an automated protocol verification tool that emulates protocols and validates secure protocols against known active and passive attacks. It can handle various encryption primitives, such as key exchange schemes, hash functions, asymmetric encryption, and

symmetric encryption. Since the proposed scheme is based on bilinear mapping, we used equations in ProVerif to add specific rules to analyze the protocol more accurately.

The ProVerif code of the proposed scheme consists of two parts: unlinkable authentication verification and batch authentication verification. The roaming authentication phase is included in each part. All verification results are shown in Figures 6 and 7. Specially, *sk_LEO* and *sk_LEO2* indicate secret keys for different LEOs; and *psk*, *K0*, and *k1* indicate the keys used in the roaming authentication phase described in Section 4.3. For unlinkable authentication verification, (*phi*, *A*) and (*epsilon1*, *epsilon2*) indicate the UE's group secret keys and NCC's group master secret keys in Section 4.1. For batch authentication verification, *s* and *sk_NCC* indicate the secret keys of the NCC, and (*BMK*, *BK*, *RK*) indicate the keys used in batch authentication in Section 4.1. There are injective correspondences between the participants in each step of authentication, pre-negotiation, and roaming authentication. The figures show that the two parts of the proposed scheme are reliable, and the individual keys and the negotiated session keys are secure.

```
Query not attacker(sk_LEO[]) is true.
Query not attacker(sk_LEO2[]) is true.
Query not attacker(phi[]) is true.
Query not attacker(A[]) is true.
Query not attacker(epsilon1[]) is true.
Query not attacker(epsilon2[]) is true.
Query not attacker(psk[]) is true.
Query not attacker(K0[]) is true.
Query not attacker(k1[]) is true.
Query inj-event(endAuthUE(id)) ==> inj-event(beginAuthUE(id)) is true.
Query inj-event(endAuthLEO(id)) ==> inj-event(beginAuthLEO(id)) is true.
Query inj-event(endPreUE(id)) ==> inj-event(beginPreUE(id)) is true.
Query inj-event(endPreLEO1(tid,id)) ==> inj-event(beginPreLEO1(id)) is true.
Query inj-event(endPreCN(tid,id)) ==> inj-event(beginPreCN) is true.
Query inj-event(endRoamUE(id)) ==> inj-event(beginRoamUE(id)) is true.
Query inj-event(endRoamLEO(id)) ==> inj-event(beginRoamLEO(id)) is true.
```

**Figure 6.** Results from the implementations of the unlinkable authentication scenario using ProVerif.

```
Query not attacker(s[]) is true.
Query not attacker(sk_NCC[]) is true.
Query not attacker(sk_LEO[]) is true.
Query not attacker(sk_LEO2[]) is true.
Query not attacker(BMK[]) is true.
Query not attacker(BK[]) is true.
Query not attacker(RK[]) is true.
Query not attacker(psk[]) is true.
Query not attacker(K0[]) is true.
Query not attacker(k1[]) is true.
Query inj-event(endAuthUE(id)) ==> inj-event(beginAuthUE(id)) is true.
Query inj-event(endAuthLEO(id)) ==> inj-event(beginAuthLEO(id)) is true.
Query inj-event(endPreUE(id)) ==> inj-event(beginPreUE(id)) is true.
Query inj-event(endPreLEO1(tid,id)) ==> inj-event(beginPreLEO1(id)) is true.
Query inj-event(endPreCN(tid,id)) ==> inj-event(beginPreCN) is true.
Query inj-event(endRoamUE(id)) ==> inj-event(beginRoamUE(id)) is true.
Query inj-event(endRoamLEO(id)) ==> inj-event(beginRoamLEO(id)) is true.
```

**Figure 7.** Results from the implementations of the batch authentication scenario using ProVerif.

### *5.2. Informal Security Analysis*

In this section, we analyze important security characteristics of the proposed scheme. In the first four sections, we analyze the security characteristics of the proposed scheme. In the subsequent four sections, we analyze attacks that the proposed scheme can defend against.

1. Mutual Authentication and Key Establishment

In a SGIN, authentication passes through at least two interactions. In the first step, the UE sends a message to the LEO, and the UE is authenticated by the LEO. In the second step, the LEO returns the signature of the ECSDA to the UE, and the identity of the LEO is proven by the UE. This completes the process of mutual authentication between the UE and LEO. The keys of both the UE and LEO are issued by the trusted NCC during the initialization phase, and the authenticator holds the public keys of the target nodes, such as $gpk$ and $pk_{LEO}$. Therefore, the fake group members or LEOs cannot be authenticated by the legitimate node.

2. Session Key Security

In each phase of the proposed scheme, no matter in which scenario, the session key is calculated according to the Diffie–Hellman problem through two random numbers generated by the two entities (i.e., UE and LEO). Calculating the session key without knowing the two generators involves solving the discrete logarithm problem on an elliptic curve. At present, it is computationally infeasible to solve the discrete logarithm problem in polynomial time [36].

3. Forward/Backward Security

New session keys $SK = (g^{r_1})^{r_2}$ and $SK = (g^{r_2})^{r_1}$ are negotiated in both authentication and roaming authentication phases of the proposed protocol. There is no computable correlation between the new session key and the session key of other sessions. No entity other than the UE and LEO can calculate the new session key.

4. Privacy and Untraceability

In SGIN unlinkable authentication scenario, it is difficult for an attacker to know the real identity of a UE through the signature. Unless the attacker can obtain the UE's private key and the UKL stored in the NCC, it cannot reveal the UE's identity. Or in another case, the attacker obtains $gmsk$. But these are extremely difficult to do for the attacker. From another perspective, anonymity in the unlinkable authentication scenario is conditional. NCC has the right to recover the UE's private key $A_i$ of the signature through $gmsk$, so as to obtain the real identity of the UE. In addition, the anonymity is untraceable, and the UE can use different $TID$ in different sessions, and the attacker cannot associate two different sessions of the same UE through signatures.

In the batch authentication scenario, the UE is traceable due to the existence of $RK$. However, it is difficult for the attacker to crack the real ID of the UE using the batch authentication key $RAK$. In addition, the UE in the unlinkable authentication scenario does not send information correlating to batch authentication. Even if the UE is changed, the attacker cannot associate the UE in batch authentication scenario with the UE in the unlinkable authentication scenario.

5. Resistance to Replay Attacks

In each scenario, the entity sends a message $M$ that contains a timestamp $TS$. The integrity of the timestamp is protected by a signature, so it is difficult for an attacker to tamper with the timestamp. The other entity verifies the timestamp $TS' - TS < \Delta T$, where $\Delta T$ is the interval that matches the current network condition. Therefore, the authentication party can confirm the freshness of the message and distinguish whether it is under replay attack.

6. Resistance of Impersonation Attacks

Suppose an attacker tries to imitate a legitimate UE, it must have the UE's group private key $gsk_i$, batch private key $buk_i$, or roaming parameters in order to generate a valid signature. However, these private keys are only held by the UE and NCC. It is difficult for the attacker to obtain both private keys. The LEO's private key is only held by the LEO and NCC. If the attacker uses the wrong private key signature, the UE will verify the signature and the validation will fail. In the roaming authentication phase, the attacker needs to obtain the $psk$, modify the information in the token, and send the token to the UE through the established secure channel to complete the attack. This is also difficult for the attacker.

7. Resistance to Man-in-the-Middle Attacks

An attacker attempting a man-in-the-middle attack attempts to intercept the communication between the UE and LEO and imitate the other party in the conversation. However, due to the resistance of the protocol to impersonation attacks, it is difficult for the attacker to successfully achieve this goal, and they therefore cannot complete man-in-the-middle attacks.

8. Resistance to Dos Attacks

The traditional batch authentication protocol does not support a reauthentication algorithm. Therefore, when an attacker launches a DoS attack, the UE of the whole group cannot perform batch authentication. Thus, the LEO cannot know the specific UE who implemented the DoS attack. The batch authentication scenario of the proposed scheme supports the rebatch process. A legitimate UE can pass the authentication without reauthentication, and the LEO can also find the illegal UE that launched attacks. When the number of illegal operations performed by a UE reaches the threshold, the NCC can revoke them.

## 6. Performance Analysis

In this section, the computational complexity and communication overheads of the proposed scheme are analyzed.

### *6.1. Computational Complexity Analysis*

#### 6.1.1. Computational Complexity in the Unlinkable Authentication Scenario

In the unlinkable authentication scenario, we compared the protocol with the work of Feng et al. [37], Alamer [24], and Wasef et al. [12]. The operations involved in these protocols are shown in Table 2, where $\mathbb{G}$ symbolizes the addictive cyclic group of prime order $q$, and $\mathbb{G}_T$ symbolizes multiplicative addictive cyclic group of the same order. Moreover, $g_1$, $g_2$, $g_{T_1}$, and $g_{T_2}$ are generators, where $g_1, g_2 \in \mathbb{G}$ and $g_{T_1}, g_{T_2} \in \mathbb{G}_T$. Parameters $a$ and $b$ are random numbers in $Z_q^*$.

**Table 2.** Comparison of operation time overheads.

| Operation | Description | Time Overload (ms) |
|---|---|---|
| $T_{Pairing}$ | A bilinear pairing $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ | 1.108615 |
| $T_{Add_1}$ | An addition operation $a + b$ | 0.000027 |
| $T_{Add_2}$ | An addition operation $g_1 + g_2$ | 0.002787 |
| $T_{Mul_1}$ | A multiplication operation $ag_1$ | 0.882037 |
| $T_{Mul_2}$ | A multiplication operation $g_{T_1} g_{T_2}$ | 0.000949 |
| $T_{Exp}$ | An exponentiation operation $g_T^a$ | 0.148430 |
| $T_{Hash_1}$ | A hash function $\{0,1\}* \rightarrow Z_q^*$ | 0.000258 |
| $T_{Hash_2}$ | A hash function $\{0,1\}* \rightarrow \mathbb{G}$ | 1.976653 |
| $T_{Hash_3}$ | A hash function $\{0,1\} * \times \mathbb{G} \rightarrow Z_q^*$ | 0.030638 |

In order to analyze the performance of each architecture, we designed an experiment based on the OpenSSL library, GMP library, and PBC library and tested on an Ubuntu 20.04.3 64-bit 4 GB virtual machine with 16 GB memory and a 3.20 GHz 8 core AMD CPU hardware configuration. We tested each operation 10,000 times and calculated their average value. The specific cost of each operation is shown in Table 2. As can be seen from Table 2, $T_{Pairing}$, $T_{Mul_1}$, $T_{Exp}$, $T_{Hash_2}$, and $T_{Hash_3}$ were more time-consuming for all operations. Therefore, in the following analysis of each scheme, attention was only paid to the impact of these five operations on the performance.

A comparison of related works in the unlinkable authentication scenario is shown in Table 3. It should be noted that although a group signature is used, the application scenarios and architectures of these works are different. We only compared the steps of the individual signature and verification. The initialization and registration steps

take place before the entire system starts, and their overheads can be excluded from the authentication overhead.

**Table 3.** Computational overloads in the unlinkable authentication scenario.

| Scheme | Computational Overload (ms) | |
|---|---|---|
| | **Signing** | **Verifying** |
| Feng et al. [37] | $9T_{Mul_1} + 5T_{Exp} = 8.680483$ | $13T_{Mul_1} + 3T_{Pairing} + 2T_{Exp} = 15.089186$ |
| Alamer [24] | $12T_{Mul_1} + 2T_{Pairing} + 2T_{Exp} + T_{Hash_2} = 15.075187$ | $8T_{Mul_1} + 3T_{Pairing} + 5T_{Exp} + T_{Hash_2} = 13.100944$ |
| Wasef et al. [12] | $11T_{Mul_1} + 2T_{Pairing} + T_{Exp} = 12.068067$ | $10T_{Mul_1} + 2T_{Pairing} + 3T_{Exp} = 11.48289$ |
| Proposed | $9T_{Mul_1} + 4T_{Exp} = 8.532053$ | $10T_{Mul_1} + 2T_{Pairing} + 3T_{Exp} = 11.48289$ |

In order to protect the unlinkability, the proposed protocol cannot directly use the UE's private key when the UE sends signatures to the LEO. Otherwise, other entities could verify the signature through the UE's public key and easily trace it. Since entities can know the identity of the satellite through analysis of the orbit of the satellite, there is no need to protect the privacy of the satellite node. Therefore, the LEO can use a signature algorithm with lower cost (i.e., ECDSA) when it sends signatures to the UE. The computational cost of the ECSDA is only the $1T_{Mul_1}$ operation for signature and $2T_{Mul_1}$ operations for verification, whose cost is small compared with other signature algorithms. As shown in Table 3, one-way authentication in the proposed scheme requires $19T_{Mul_1}$, $2T_{Pairing}$, and $7T_{Exp}$. It takes about 20 ms for the signature and verification in the unlinkable authentication scenario, which is less time than the other schemes.

#### 6.1.2. Computational Complexity in the Batch Authentication Scenario

Considering the large number of UEs in some scenarios, it is necessary to reduce the amount of authentication data and the processing time of authentication requests [16]. The batch authentication scenario in the proposed scheme can provide efficient services for these UE and lighten the LEOs' burden. We evaluated the cost of the first batch authentication and the rebatch authentication. We also estimated the computational complexity of the rebatch authentication.

In the evaluation of first batch verification, we compared the proposed scheme with the related works of Xue et al. [11] and Wasef et al. [12]. In order to contrast with the unlinkable authentication scenario, we compared with the work of Wasef et al. [12], based on a batch group signature (BGS). The cost of signing and verifying are shown in Table 4. Although the computational complexity of the signature of the proposed scheme was higher than that of Xue et al.'s work, Xue et al. [11] used a UE private key to complete the signature, and the LEO needs to use the UE's public key to verify the signature. This is detrimental to the privacy protection of the UE, as attackers can easily trace the UE through its public key. Therefore, based on security and privacy considerations, our scheme is more suitable for the proposed SGIN scenario. Additionally, Figure 8 shows that the verification cost of the three schemes varies with the increase in the number of authentication requests. It can be seen that the cost of the proposed scheme is significantly less than that required for BGS and the work of Xue et al. [11].

**Table 4.** Computational overload in the batch authentication scenario.

| Scheme | Computational Overload (ms) | |
|---|---|---|
| | **Signing** | **Verifying** |
| Xue et al. [11] | $T_{Mul_1} + T_{Hash_3} = 0.912675$ | $(n+2)T_{Mul_1} + 3nT_{Hash_3}$ |
| Wasef et al. [12] | $11T_{Mul_1} + 2T_{Pairing} + T_{Exp} = 12.068067$ | $(6n+7)T_{Mul_1} + 2T_{Pairing}$ |
| Proposed | $3T_{Mul_1} + T_{Exp} = 2.794541$ | $nT_{Exp} + T_{Pairing}$ |

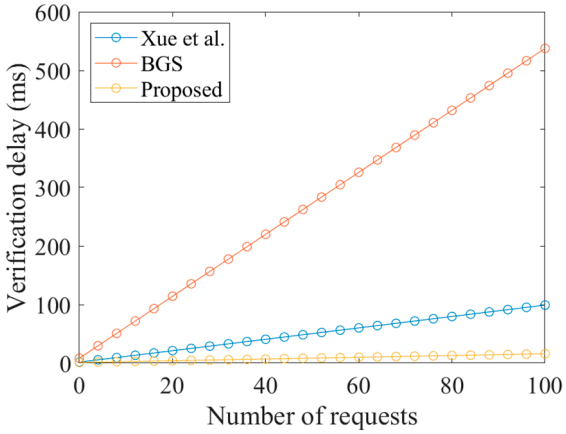


**Figure 8.** Computational cost comparison of batch authentication in the verification phase. Xue et al. [11], BGS [12].

In terms of rebatch authentication, it is assumed that there is at most a 1% vulnerable UE in a group containing 1000 UE for batch authentication, then the maximum number of UEs breached in this group is $N_{ckd} = N_{all} \times 1\% = 1000 \times 1\% = 10$. Given that the number of request messages in a batch is $N_{req}$, the probability that $N_{req}$ requests contain exactly $i$ invalid requests can be expressed using the hypergeometric distribution as

$$Pr\{X = i\} = \frac{\binom{N_{all} - N_{ckd}}{N_{req} - i}\binom{N_{ckd}}{i}}{\binom{N_{all}}{N_{req}}} \tag{2}$$

Assuming that event $A$ means rebatch authentication is required to successfully verify valid requests, the probability of event $A$ can be expressed as

$$\begin{aligned} Pr\{A\} &= Pr\{i = 1\} + \cdots + Pr\{i = 10\} \\ &= \frac{\binom{N_{all} - N_{ckd}}{N_{req} - 1}\binom{N_{ckd}}{1}}{\binom{N_{all}}{N_{req}}} + \cdots + \frac{\binom{N_{all} - N_{ckd}}{N_{req} - 10}\binom{N_{ckd}}{10}}{\binom{N_{all}}{N_{req}}} \\ &= \frac{\sum_{i=1}^{10}\binom{N_{all} - N_{ckd}}{N_{req} - i}\binom{N_{ckd}}{i}}{\binom{N_{all}}{N_{req}}} \end{aligned} \tag{3}$$

According to Equation (3), when there are only one or two invalid requests ($i = 1$ or $i = 2$) in a batch, the probability of rebatch authentication is extremely small. However, in the case of DoS attacks by malicious UEs, rebatch authentication can protect a legitimate UE from authentication failure for a long time.

The proposed scheme in the first authentication needs $nT_{Exp}$ and $1T_{Pairing}$ operation. The LEO calculates $h_i(RAK_i \oplus H_1(B_i||U_i||TS_1))$ for each UE in the first authentication phase, thus only the $T_{Pairing}$ operation is required. Therefore, the computational cost of the worst and average cases of rebatch authentication is analyzed below:

1. Worst case: According to the rebatch algorithm proposed by Huang et al. [25], we assume that the worst case is where invalid requests are always in the detected batch. Assuming there are $n$ requests in a batch, it takes $log_2 n$ times the calculation in the worst case. Therefore, the worst-case total batch validation time for a valid request is

$$T_{wst} = T_{fst} + 2 \times log_2 n \times T_{reb} \tag{4}$$

where $T_{wst}$ is the time required for the worst case, $T_{fst}$ is the time required for the first batch verification, and $T_{reb}$ is the time required for completion of the rebatch algorithm.

2. Average case: Calculating the total validation cost in all cases divided by the number of possible cases gives the validation time required for the average case:

$$T_{ave} = T_{fst} + \frac{1}{log_2 n + 1} \sum_{i=2}^{log_2 n} T_{reb} \quad (5)$$

Figure 9 shows the total validation complexity for the worst case, average case, and first batch authentication with 0 to 1000 UE requests. As can be seen from the figure, although the complexity of rebatch authentication in the worst case is relatively high, the cost of rebatch authentication in the average case is close to the complexity of the initial batch authentication, and the computational complexity does not increase rapidly with the increase of the number of requests. Considering the possibility of DoS attacks leading to large-scale UE authentication failure, rebatch authentication is necessary.

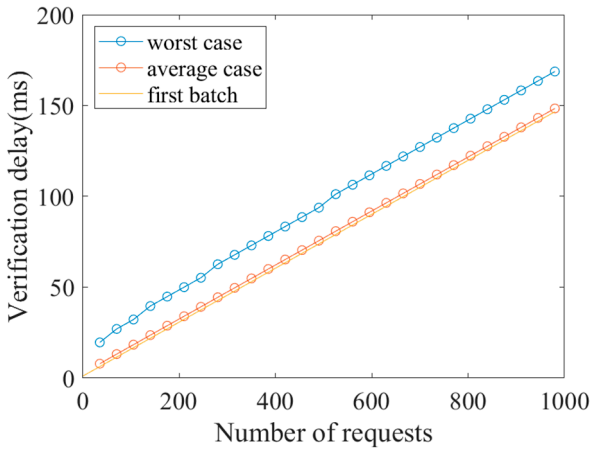


**Figure 9.** Case comparison for rebatch authentication.

#### 6.1.3. Computational Complexity in Roaming Authentication

In the process of roaming authentication, we use symmetric encryption to ensure the efficiency of the scheme. The AES CBC algorithm was tested using OpenSSL library in the same environment. We tested the symmetric encryption and decryption algorithms 10,000 times and took their average values: the encryption algorithm costs 0.000112 ms, and the decryption algorithm costs 0.000110 ms. Assuming that the times required for symmetric encryption of the AES algorithm are $T_{senc}$ and $T_{sdec}$, and compared with the overheads in Table 2, they are negligible. In order to be consistent with the anonymous authentication phase, roaming authentication uses the same session key generation method. The proposed roaming authentication protocol involves $2T_{Exp}$ operations to negotiate the session key. It was assumed that the time required for the encryption algorithm in asymmetric encryption is $T_{aenc}$ and the time required for the decryption algorithm is $T_{adec}$. After 10,000 calculations using the OpenSSL library, the average value was obtained: $T_{aenc}$ = 0.015079 ms, $T_{adec}$ = 0.028344 ms. Then the computational cost required for the UE to send a message to LEO was $T_{Exp}$ = 0.148430 ms. The computational cost for the LEO to send a message to the UE was $T_{aenc} + T_{Exp}$ = 0.176774 ms. It was shown that the total roaming authentication calculation cost is far less than the cost of re-authentication.

### *6.2. Communication Overhead Analysis*

According to the simulation results, the size of elements $L_G$ and $L_Z$ in $\mathbb{G}$ and $Z_q^*$ are both 16 bytes. Assuming that the identity length $L_{ID}$ and time message length $L_T$ are both 12 bytes. The hash function length $L_H$ is 32 bytes.

In the unlinkable authentication scenario, the UE needs to send an LEO signature of $3L_G + 6L_Z$ = 144 bytes and plaintext of $2L_{ID} + L_G + L_T$ = 52 bytes. When the LEO completes authentication, 84 bytes of information will be returned. The signature information for the schemes of both Feng et al. [37] and Wasef et al. [12] was $3L_G + 6L_Z$ = 144 bytes, while the schemes of Alamer [24] required $6L_G + 6L_Z$ = 192 bytes. Since other SGS-based schemes have different architectures from the proposed scheme, we tried to unify their certification scenarios and situations for a better analysis. It was assumed that at least the ID and time stamp are required in these schemes. The trust value of the scheme of Feng et al. [37] was 4 bytes according to their article. A comparison of communication overhead from the UE to the access point is shown in Figure 10. As can be seen from the figure, the communication overhead of the proposed scheme was slightly smaller than the schemes of Alamer [24] and Wasef et al. [12], while it was equal to Feng et al. [37].

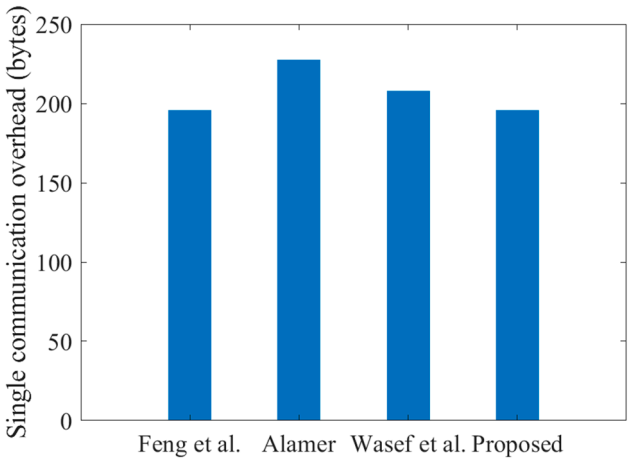


**Figure 10.** Communication overhead comparison in the unlinkable authentication scenario. Feng et al. [37], Alamer [24], Wasef et al. [12].

In batch authentication, the UE needs to send a $4L_G + 2L_{ID} + L_T$ = 64 bytes message to the LEO, and the LEO returns a message of 84 bytes. A comparison was made with the communication overheads of the authentication protocol of Xue et al. [11] and Wasef et al. [12]. The results are shown in Table 5. According to the analysis results, the overall message length of the proposed scheme is shorter than that of the other related works.

**Table 5.** Communication overhead comparison in the batch authentication scenario.

| Scheme | Message Length (Bytes) | |
|---|---|---|
| | UE to LEO | LEO to UE |
| Xue et al. [11] | $2L_{ID} + 2L_T + 2L_G + L_Z + 2L_{ID} + L_G = 144$ | $L_T + 2L_{ID} + 3L_G + L_Z = 100$ |
| Wasef et al. [12] | $2L_{ID} + 4L_G + 6L_Z + L_T = 166$ | $2L_{ID} + L_T + L_G + 2L_Z = 84$ |
| Proposed | $4L_G + 2L_{ID} + L_T = 64$ | $2L_{ID} + L_T + L_G + 2L_Z = 84$ |

Assuming that the plaintext length is $16n$ bytes (i.e., $n$ is divisible by 16), then the ciphertext length is $16n$ bytes in AES encryption. If the plaintext length is $16n + m$ bytes and $m < 16$, the ciphertext length is $16(n + 1)$ bytes. In the roaming authentication phase, $K_1$ is key with a length $L_r$ = 16 bytes, and the size of $Ticket_i$ is $L_{ticket} = 16 * (2L_{ID} + L_r + L_T)/16$ = 64 bytes. Due to the use of symmetric encryption for key negotiation, we needed to calculate its communication cost separately. The size of $c_1$ sent by UE to LEO is $L_{c_1} = 16(L_r + L_{ticket})/16$ = 80 bytes, and the total message length is $L_{ID} + L_G + L_{c_1} + L_H + L_T$ = 152 bytes. The size of $c_2$ is $L_{c_2} = 16 * (L_{ID} + L_r)/16$ = 32 bytes, the message size sent by the LEO to the UE is $L_{ID} + L_G + L_{c_2} + L_H + L_T$ = 104 bytes.

Despite the higher communication overheads of roaming authentication, it drastically reduces the computational latency, which is acceptable.

## 7. Conclusions

In this paper, we investigated on-demand access and roaming authentication protocols in a multi-operator heterogeneous scenario in a satellite–ground integrated network. We have proposed a scheme that includes anonymous unlinkable and batch authentication protocols, as well as fast roaming authentication. Specifically, users with higher privacy requirements are suitable for the unlinkable authentication scenario, where the scheme can provide anonymity and unlinkability; a large number of users with higher efficiency requirements are suitable for the batch authentication scenario, where the scheme provides traceable anonymity. In addition, for users who need to switch between multi-operator networks, this scheme provides a cross-domain fast roaming authentication solution. The proposed protocol delegates the authentication task to the satellite, which significantly reduces the transmission delay in the SGIN. We performed a formal analysis using ProVerif and an informal analysis to prove the security of the scheme. In addition, we evaluated the performance of the scheme and the results showed that our scheme is effective.

To enhance our research, we intend to study short group signature algorithms in more depth in our future work and propose more secure and more efficient protocols for unlinkability authentication scenarios based on zero-knowledge proofs. In addition, the application scenario of 6G will be more complex than 5G, and it is a challenge to cope with the fair billing of multiple operators and balance their interests. Especially, the introduction of eSIMs brings novel security risks and conflicts of interest. Thus, in the future, we will further consider the interests of multi-operator scenarios and try to find a suitable solution to authentication.

**Author Contributions:** Conceptualization, Y.T. and J.X.; methodology, Y.T. and J.X.; software, Y.T.; validation, B.C.; formal analysis, Y.T.; investigation, Y.T.; resources, B.C.; data curation, Y.T.; writing—original draft preparation, Y.T.; writing—review and editing, J.X. and B.C.; visualization, Y.T.; supervision, B.C., L.S. and H.D.; project administration, B.C. and H.D.; funding acquisition, L.S. and H.D. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was funded by Beijing University of Posts and Telecommunications-China Mobile Research Institute Joint Innovation Center.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Erol-Kantarci, M.; Sukhmani, S. Caching and computing at the edge for mobile augmented reality and virtual reality (AR/VR) in 5G. In *Ad Hoc Networks, Proceedings of the 9th International Conference, AdHocNets 2017, Niagara Falls, ON, Canada, 28–29 September 2017*; Springer: Cham, Switzerland, 2018; pp. 169–177.
2. Nguyen, V.-L.; Lin, P.-C.; Cheng, B.-C.; Hwang, R.-H.; Lin, Y.-D. Security and privacy for 6G: A survey on prospective technologies and challenges. *IEEE Commun. Surv. Tutor.* **2021**, *23*, 2384–2428. [CrossRef]
3. Wang, Y.; Su, Z.; Guo, S.; Dai, M.; Luan, T.H.; Liu, Y. A Survey on Digital Twins: Architecture, Enabling Technologies, Security and Privacy, and Future Prospects. *IEEE Internet Things J.* **2023**. [CrossRef]
4. Zhang, Z.; Zhang, W.; Tseng, F.-H. Satellite mobile edge computing: Improving QoS of high-speed satellite-terrestrial networks using edge computing techniques. *IEEE Netw.* **2019**, *33*, 70–76. [CrossRef]
5. Yang, H.; Alphones, A.; Xiong, Z.; Niyato, D.; Zhao, J.; Wu, K. Artificial-intelligence-enabled intelligent 6G networks. *IEEE Netw.* **2020**, *34*, 272–280. [CrossRef]
6. Guo, H.; Li, J.; Liu, J.; Tian, N.; Kato, N. A survey on space-air-ground-sea integrated network security in 6G. *IEEE Commun. Surv. Tutor.* **2021**, *24*, 53–87. [CrossRef]
7. Zhou, D.; Sheng, M.; Li, J.; Han, Z. Aerospace Integrated Networks Innovation for Empowering 6G: A Survey and Future Challenges. *IEEE Commun. Surv. Tutor.* **2023**, *25*, 975–1019. [CrossRef]

8. Pachler, N.; del Portillo, I.; Crawley, E.F.; Cameron, B.G. An updated comparison of four low earth orbit satellite constellation systems to provide global broadband. In Proceedings of the 2021 IEEE International Conference on Communications Workshops (ICC Workshops), Montreal, QC, Canada, 14–23 June 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–7.
9. Xie, R.; Tang, Q.; Wang, Q.; Liu, X.; Yu, F.R.; Huang, T. Satellite-terrestrial integrated edge computing networks: Architecture, challenges, and open issues. *IEEE Netw.* **2020**, *34*, 224–231. [CrossRef]
10. Hwang, M.-S.; Yang, C.-C.; Shiu, C.-Y. An authentication scheme for mobile satellite communication systems. *ACM SIGOPS Oper. Syst. Rev.* **2003**, *37*, 42–47. [CrossRef]
11. Xue, K.; Meng, W.; Li, S.; Wei, D.S.; Zhou, H.; Yu, N. A secure and efficient access and handover authentication protocol for Internet of Things in space information networks. *IEEE Internet Things J.* **2019**, *6*, 5485–5499. [CrossRef]
12. Wasef, A.; Shen, X. Efficient group signature scheme supporting batch verification for securing vehicular networks. In Proceedings of the 2010 IEEE International Conference on Communications, Cape Town, South Africa, 23–27 May 2010; IEEE: Piscataway, NJ, USA, 2010; pp. 1–5.
13. Di, B.; Zhang, H.; Song, L.; Li, Y.; Li, G.Y. Ultra-dense LEO: Integrating terrestrial-satellite networks into 5G and beyond for data offloading. *IEEE Trans. Wirel. Commun.* **2018**, *18*, 47–62. [CrossRef]
14. Li, J.; Lu, H.; Xue, K.; Zhang, Y. Temporal netgrid model-based dynamic routing in large-scale small satellite networks. *IEEE Trans. Veh. Technol.* **2019**, *68*, 6009–6021. [CrossRef]
15. 3GPP. System architecture for the 5G System (5GS). In *3GPP Sophia Antipolis*; 3GPP: Valbonne, France, 2021.
16. Zhao, B.; Liu, P.; Wang, X.; You, I. Toward efficient authentication for space-air-ground integrated Internet of things. *Int. J. Distrib. Sens. Netw.* **2019**, *15*, 1550147719860390. [CrossRef]
17. Cui, Q.; Zhu, Z.; Ni, W.; Tao, X.; Zhang, P. Edge-Intelligence-Empowered, Unified Authentication and Trust Evaluation for Heterogeneous Beyond 5G Systems. *IEEE Wirel. Commun.* **2021**, *28*, 78–85. [CrossRef]
18. Guo, J.; Du, Y. A Novel RLWE-Based Anonymous Mutual Authentication Protocol for Space Information Network. *Secur. Commun. Netw.* **2020**, *2020*, 5167832. [CrossRef]
19. Yao, S.; Guan, J.; Wu, Y.; Xu, K.; Xu, M. Toward secure and lightweight access authentication in sagins. *IEEE Wirel. Commun.* **2020**, *27*, 75–81. [CrossRef]
20. Guo, J.; Du, Y.; Zhang, Y.; Li, M. A provably secure ECC-based access and handover authentication protocol for space information networks. *J. Netw. Comput. Appl.* **2021**, *193*, 103183. [CrossRef]
21. Sun, J.; Zhang, C.; Fang, Y. An id-based framework achieving privacy and non-repudiation in vehicular ad hoc networks. In Proceedings of the MILCOM 2007-IEEE Military Communications Conference, Orlando, FL, USA, 29–31 October 2007; IEEE: Piscataway, NJ, USA, 2007; pp. 1–7.
22. Zhang, L.; Li, C.; Li, Y.; Luo, Q.; Zhu, R. Group signature based privacy protection algorithm for mobile ad hoc network. In Proceedings of the 2017 IEEE International Conference on Information and Automation (ICIA), Macao, China, 18–20 July 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 947–952.
23. Boneh, D.; Boyen, X.; Shacham, H. Short group signatures. In Proceedings of the 24th Annual International Cryptology Conference, Santa Barbara, CA, USA, 15–19 August 2004; Springer: Berlin/Heidelberg, Germany, 2004; pp. 41–55.
24. Alamer, A. An efficient group signcryption scheme supporting batch verification for securing transmitted data in the Internet of Things. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *14*, 5885–5902. [CrossRef]
25. Huang, J.-L.; Yeh, L.-Y.; Chien, H.-Y. ABAKA: An anonymous batch authenticated and key agreement scheme for value-added services in vehicular ad hoc networks. *IEEE Trans. Veh. Technol.* **2010**, *60*, 248–262. [CrossRef]
26. Lai, C.; Lu, R.; Zheng, D.; Li, H.; Shen, X.S. GLARM: Group-based lightweight authentication scheme for resource-constrained machine to machine communications. *Comput. Netw.* **2016**, *99*, 66–81. [CrossRef]
27. Al-Shareeda, M.A.; Anbar, M.; Alazzawi, M.A.; Manickam, S.; Al-Hiti, A.S. LSWBVM: A lightweight security without using batch verification method scheme for a vehicle ad hoc network. *IEEE Access* **2020**, *8*, 170507–170518. [CrossRef]
28. Xue, K.; Meng, W.; Zhou, H.; Wei, D.S.; Guizani, M. A lightweight and secure group key based handover authentication protocol for the software-defined space information network. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 3673–3684. [CrossRef]
29. Liu, X.; Yang, A.; Huang, C.; Li, Y.; Li, T.; Li, M. Decentralized Anonymous Authentication With Fair Billing for Space-Ground Integrated Networks. *IEEE Trans. Veh. Technol.* **2021**, *70*, 7764–7777. [CrossRef]
30. Yang, Q.; Xue, K.; Xu, J.; Wang, J.; Li, F.; Yu, N. AnFRA: Anonymous and fast roaming authentication for space information network. *IEEE Trans. Inf. Forensics Secur.* **2018**, *14*, 486–497. [CrossRef]
31. Guo, J.; Du, Y.; Sun, Z.; Wu, R.; Wu, X.; Zhang, L.; Zheng, T. SRAKN: Secure roaming authentication and key negotiation protocol for space information network. *Comput. Commun.* **2023**, *206*, 22–37. [CrossRef]
32. Yang, Y.; Cao, J.; Ma, R.; Cheng, L.; Chen, L.; Niu, B.; Li, H. FHAP: Fast Handover Authentication Protocol for High-Speed Mobile Terminals in 5G Satellite-Terrestrial Integrated Networks. *IEEE Internet Things J.* **2023**. [CrossRef]
33. Johnson, D.; Menezes, A.; Vanstone, S. The elliptic curve digital signature algorithm (ECDSA). *Int. J. Inf. Secur.* **2001**, *1*, 36–63. [CrossRef]
34. Hu, J.; Yang, L.-L.; Hanzo, L. Energy-efficient cross-layer design of wireless mesh networks for content sharing in online social networks. *IEEE Trans. Veh. Technol.* **2017**, *66*, 8495–8509. [CrossRef]

35. Guo, Y.; Li, X.; Yousefi'zadeh, H.; Jafarkhani, H. UAV-aided cross-layer routing for MANETs. In Proceedings of the 2012 IEEE Wireless Communications and Networking Conference (WCNC), Paris, France, 1–4 April 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 2928–2933.
36. Diffie, W.; Hellman, M. New directions in cryptography. *IEEE Trans. Inf. Theory* **1976**, *22*, 644–654. [CrossRef]
37. Feng, W.; Yan, Z.; Xie, H. Anonymous authentication on trust in pervasive social networking based on group signature. *IEEE Access* **2017**, *5*, 6236–6246. [CrossRef]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.