AnFRA: Anonymous and Fast Roaming Authentication for Space Information Network

D. Saritha  M. Venkateswarlu

1 Assistant Professor, Dept. of Master of Computer Applications, Narayana Engineering College, Gudur.
2 PG Scholar, Dept. of Master of Computer Applications, Narayana Engineering College, Gudur.

Abstract—Nowadays, Space Information Network (SIN) has been widely used in real life because of its advantages of communicating anywhere at anytime. This feature is leading to a new trend that traditional wireless users are willing to roam to SIN to obtain a better service. However, the features of exposed links and higher signal latency in SIN make it difficult to design a secure and fast roaming authentication scheme for this new trend. Although some existing researches have been focused on designing secure authentication protocols for SIN or providing roaming authentication protocols for traditional wireless networks, these schemes cannot provide adequate requirements for the roaming communication in SIN and bring in critical issues, such as privacy leakage or intolerable authentication delay. Observing these problems haven’t been well addressed, we design an anonymous and fast roaming authentication scheme for SIN. In our scheme, we utilize the group signature to provide the anonymity for roaming users, and assume that the satellites have limited computing capacity and make them have the defined authentication function to avoid the realtime involvement of the home network control center (HNCC) when authenticating the roaming users. The results of security and performance analysis show that the proposed scheme can provide the required security features, while providing a small authentication delay.

Index Terms—Access authentication, anonymity, roaming, space information network.

I. INTRODUCTION

With the acceleration of the globalization process, the demand for communicating anywhere at anytime is becoming more and more urgent [1]. Space information network (SIN) has been proposed in this background and also already been implemented in real life (e.g., Iridium, Globlastar), which uses artificial earth satellites as relay stations to transmit radio waves to achieve a wider range of communications. In the future, SIN can be developed as an Interplanetary Internet that connects spacecrafts with Earth’s terrestrial Internet to support the future space exploration and ubiquitous Internet access [2]. Compared with the traditional wireless communication systems, such as cellular networks [3] and road networks [4], satellite communication system has the characteristics of global coverage, large capacity, bandwidth-on-demand flexibility and won’t be limited by any complicated geographical conditions between two communication points [5]. Similarly, roaming service is also necessary to be provided by SIN: On the one hand, due to the above appealing features, users in traditional wireless networks are more willing to access SIN to obtain network services, including the roaming service, especially in some extreme conditions, such as in sea, desert, or in earthquake disaster areas, where there is no allocated base station for users to access traditional wireless networks. On
the other hand, providing global roaming in current and next-generation networks to improve network accessibility and roaming quality is an important requirement for nowadays network development [6]. For the security and quality of roaming service, it is critical for SIN to deploy a secure roaming authentication protocol [7]. In traditional wireless networks, roaming authentication protocols can be classified into two types: three-party roaming authentication scheme and two-party roaming authentication scheme. Three-party roaming authentication schemes, such as [8] and [9], usually verify the roaming user at its home server, so that the foreign server cannot learn users’ privacy. However, they need more interactions and cannot be implemented in the SIN architecture, as the SIN has a long propagation delay between satellites and the ground. Even for low earth orbit satellite (LEO) which is closer to the ground, there are still 500 to 2,000 kilometers [10] away from the ground, and accordingly with 10 to 40ms propagation delay. This long propagation delay will bring intolerable authentication delay to these three-party roaming authentication schemes. While twoparty roaming authentication schemes authenticate roaming users without requiring the participation of its home server and usually require less interactions, which can reduce the authentication delay in theory. However, for existing twoparty authentication schemes, they still cannot be deployed directly to SIN. Since they usually have some time-consuming operations (e.g., pairing) of checking revocation list in these schemes, such as [11] and [12]. Meanwhile, the long propagation delay cannot be significantly reduced, as multiple interactions between satellites and ground devices still exist in these schemes. In fact, with the development of satellite hardware technology, satellites have been able to carry complexity computation. Inspired by this new feature, we utilize the satellites as the verifier rather than ground servers, which can largely reduce interactions between the satellites and the ground, so as to lower authentication delay. However, except the long propagation delay challenge, security requirements for the roaming scenario in SIN are also hard to be guaranteed.

Firstly, due to the vulnerability of SIN, some malicious attacks such as interception, modification, replay, and impersonation attacks can easily damage the system [13–16]. Secondly, the highly exposed links of SIN could be utilized by attackers to compromise users’ privacy through eavesdropping the exposed channel [7]. Finally, even the foreign network entities could be potential adversaries, as they may easily disclose users’ privacy by tracking users’ identities and locations. Observing the challenges that the long propagation delay and security vulnerability exist in SIN, and still no existing authentication scheme can be straightly implemented to better solve the problems. In this paper, we propose a group signature based authentication scheme to protect users’ privacy and provide fast access authentication for roaming users. In our scheme, each LEO with certain computing power acts as a verifier to authenticate mobile users when they request to access the SIN, which can largely reduce the authentication delay and interaction messages. Meanwhile, the utilization of group signature can efficiently provide user anonymity, so that users’ privacy won’t be leaked to foreign network entities. Especially, our proposed scheme makes the following main contributions:

1) We strengthen the authentication function of LEO satellites, and proposed a fast roaming authentication scheme, named AnFRA, which can achieve a fast access authentication between users and satellites. Moreover, a pre-negotiation mechanism is implemented to faster the authentication.

2) Our proposed scheme is based on group signature, which not only makes it possible for satellites to authenticate users without the participation of the home server, but also provides a strong users anonymity and guarantees its security requirements.

3) Considering the distinctive features of SIN, a well designed revocation mechanism is also incorporated into the design of AnFRA to support dynamic user’s revocation. Although the revocation mechanism brings in some additional overhead, it avoids the time cost to implement the revocation list checking when authenticating users. The rest of this paper is organized as follows: we first discuss related works in Section II. Then the preliminaries are demonstrated in Section III. We introduce the system model, security model and
security requirements in Section IV. In Section V, we describe our proposed scheme in details, followed by the analysis of security and performance in Section VI and VII. Finally, Section VIII presents the overall conclusion.

II. RELATED WORK

In this section, we discuss the related works in terms of authentication schemes for SIN and authentication schemes for traditional networks.

A. Authentication Schemes for SIN

In recent years, many studies have done a great deal of work on providing a secure access authentication scheme for space information network (SIN). In 1996, Cruickshank [17] first proposed a security system for satellite networks, which uses a combination of public-key and secret key systems to satisfy the security requirements of mutual authentication and data confidentiality. However, Cruickshank’s scheme needs complex operations of encryption and decryption, and cannot provide user anonymity protection. In order to reduce the computation overhead, Liu et al. [18] proposed a lightweight authentication in satellite networks, in which all the involved computing operations are just the hash function, the bit-wise exclusive-or operation, and the string concatenation operation. The literatures [19] and [20] analyzed the security vulnerabilities in the existing schemes, and proposed their security enhanced authentication for SIN, which can prevent user’s privacy from compromising by malicious attackers. These schemes can provide secure protocols for authentication in SIN, but when implemented in the scenario of roaming to SIN, due to the untrustworthiness of the foreign network and long latency for signal propagation, these schemes may lead to privacy disclosure and intolerable authentication delay.

B. Authentication Schemes for Traditional Networks

Although, public key infrastructure (PKI) has been widely used in traditional networks, its complicated and time consuming certificates management has attracted the concerns of researchers. Therefore, some identity-based authentication schemes have been proposed, such as [21], [22] and [23]. In these schemes, users need to require and store lots of one-time pseudo identities, which is challenging for capacity constrained mobile devices. For preventing illegal access, verifiers have to store each used and revoked pseudo identity in its local memory. However, the satellites have limited storage capacity, this makes identity-based authentication scheme hard to implement to SIN. Moreover, identity-based authentication schemes usually rely on a trusted third party (private key generator(PKG)), which may be a single point of bottle neck or be compromised. Thus, Menmon et al. utilized certificate-less public key cryptography (CL-PKC) to design authentication protocols for GSM [24, 25], in which keys are generated from both the user and the key generation center (KGC). However, these schemes are not designed for the roaming scenario in which the foreign access points are usually untrusted and may compromise users’ privacy. So these schemes are not suitable for the roaming scenario. However, in traditional wireless networks, some roaming authentication schemes have been proposed to provide user authentication in different authentication domains and address the privacy disclosure issue. In 2006, Jiang et al. [8] used the secret-splitting principle and self-certified technologies, and proposed a lightweight roaming authentication protocol for wireless mobile networks to provide the security property of identity anonymity. While some group signature based roaming authentication schemes [11, 26, 27] were proposed to preserve roaming user’s privacy when roaming to an untrusted network. However, due to the long propagation latency between satellite and ground communication point in SIN, directly implementing these roaming protocols to SIN can not address the problems of long authentication delay, especially in [11], a time-consuming pairing operation of checking revocation list is required. Therefore, in this paper, we design a special access authentication protocol for roaming to SIN, which can not only guarantee the anonymity for roaming users, but also can largely reduce the authentication delay.

III. PRELIMINARIES

In this section, we first give a review of background information on bilinear pairing and the security assumption defined on it, then we briefly
describe the definition of elliptic curve digital signature algorithm.

A. Bilinear Pairing

Let $G$ be additive cyclic group of the prime order $p$, and $GT$ be multiplicative cyclic group of the same prime order, and $P$ be a generator of group of $G$. Suppose $G$ and $GT$ are equipped with a pairing, i.e., a non-degenerated and efficiently computable bilinear map $e : G \times G \rightarrow GT$ such that $e(aP_1,bQ_1) = e(P_1,Q_1)^{ab} \in GT$ for all $a,b \in \mathbb{Z}_p^*$ and any $P_1,Q_1 \in G$. We refer to [28, 29] for a more comprehensive description of pairing technique.

Definition I: A bilinear parameter generator $Gen$ is a probabilistic algorithm that takes a security parameter $\kappa$ as input and output a 5-tuple $(p,P,G,GT,e)$. Definition II (q-Strong Diffie-Hellman (q-SDH) Problem): Given a $(q+1)$-tuple $(g,\gamma g,\gamma^2 g,...,\gamma^q g)$, it is difficult to compute a pair $(1 \gamma + x \cdot g,x)$, where $\gamma, x \in \mathbb{Z}_p^g$ is a generator of $G$. An algorithm $A$ has advantage in solving q-SDH in $(G,G)$ if $Pr[A(g,\gamma g,\gamma^2 g,...,\gamma^q g) = (1 \gamma + x \cdot g,x)] \geq \epsilon$. Definition III (Decision Linear Diffie-Hellman Problem): Given $u,v,h,ua,vb,hc \in G$ as input, output yes if $a + b = c$ and no otherwise. More precisely, the advantage algorithm $A$ in deciding the Decision Linear problem in $G$ is defined as: $Adv\ LinearA = |Pr[A(u,v,h,ua,vb,ha+b) = yes : u,v,h \leftarrow G,a,b \leftarrow \mathbb{Z}_p] - Pr[A(u,v,h,ua,vb,ha) = yes : u,v,\tau \leftarrow G,a,b \leftarrow \mathbb{Z}_p]|$.

B. Elliptic Curve Digital Signature Algorithm (ECDSA)

ECDSA is the elliptic curve analogue of the Digital Signature Algorithm (DSA), which mainly contains the following three algorithms in ANSI standard [30]. It should be noted that the SHA-1 algorithm is not secure anymore [31], and recommended to be replaced by other secure hash algorithms such as SHA-256 [32]:

- EC.Keygen(): The key pair of an entity is associated with a particular set of elliptic curve domain parameters, which consist of a suitable chosen elliptic curve $E$ defined over a finite field $\mathbb{F}_q$ of characteristic $p$, and a base point $G \in E(\mathbb{F}_q)$.

To generate the signing/verifying key pair, the entity first selects a random or pseudorandom integer $d$ in the interval $[1,n-1]$ (where $n$ is a sufficiently large prime), then computes $Q = d \cdot G$. Thus, the signing/verifying key pair is $(d,Q)$.

- EC.Sign($d,m$): To sign a message $m$, an entity should implement the steps as follows: (1) Select a random integer $k$ $(1 \leq k \leq n-1)$;

(2) Compute $k \cdot G = (x_1,y_1)$ and $r = x_1 \mod n$. If $r = 0$, go to step (1); (3) Compute $k-1 \mod n$, $e = SHA-1(m)$ and $s = k-1(e + d \cdot r) \mod n$. If $s = 0$, go to step (1); (4) The signature for the message $m$ is $\sigma = (r,s)$. • EC.V erify($Q,\sigma$): To verify the signature $\sigma = (r,s)$, an entity implements the following steps: (1) Verify whether $r$ and $s$ are two integers in the interval $[1,n-1]$. If yes, continue; (2) Compute $e = SHA-1(m)$, $u_1 = e \cdot w \mod n$ and $u_2 = r \cdot w \mod n$; (3) Compute $X = u_1 \cdot G + u_2 \cdot Q$. If $X = O$, reject the signature. Otherwise, compute $v = x_1 \mod n$ where $X = (x_1,y_1)$. Accept the signature if and only if $v = r$.

IV. SYSTEM MODEL, SECURITY MODEL AND SECURITY REQUIREMENTS

In this section, we give the definitions of the system model, security model and security requirements of the roaming authentication scenario in SIN.

A. System Model

![Fig. 1: System Model](image_url)

The trend of providing global roaming in kinds of networks makes it necessary for the SIN to provide roaming service for its roaming users. The roaming scenario in SIN is illustrated in Fig.1. Without loss of generality, we only consider the system model that user roams between the homogeneous SINs, and the scenario of roaming to SIN from other heterogeneous networks (e.g., cellular networks) is the same as this. The system model in our scheme consists of a global offline trusted third party (TTP) and several domains, and each domain contains a network control center.
(NCC), gateway stations (GSs), low earth orbit satellites (LEOs) and mobile users. Following illustrates the functions and duties of each entity:

- TTP is in charge of managing and distributing public/private key pairs for NCCs in different domains. These keys are used for authenticating among these NCCs, so that they can exchange information securely.

- NCC is the management of its network domain. It provides registration and certification for users to access the home/foreign network.

- GS is a middle entity between the NCC and LEOs. It connects to the NCC through the terrestrial networks, and provides a ground interface for LEOs.

- LEO is the access point for users to access the network. With the satellite manufacturing technology advancement, nowadays LEO satellites can have certain computing capacities to execute some complex functions [2].

- Users access the network to obtain its subscription services. In this paper, we consider the scenario where a roaming user is out of its home network and visiting a foreign network.

B. Security Model

The proposed scheme has the following security assumptions:

- The proposed scheme assumes that the TTP is trust worthy for both home network entities and foreign network entities. It is infeasible for any adversary to compromise.

- We assume that there exists a secure channel between NCCs and TTP, NCC and its domain GS, respectively. This secure channel can be constructed by the TLS or SSL protocol.

- We assume that a polynomial time adversary, who can modify or interrupt the interaction messages among the users, FGSs and foreign LEOs (FLEOs), tries to break the proposed anonymous authentication protocol when roaming users access to the foreign network.

- Foreign entities (i.e., FLEOs, FGSs, and FNCC) may be malicious adversaries in our scheme, they may intend to break the proposed scheme to retrieve the roaming user’s identity from user’s access request.

C. Security Requirements

A well-designed roaming authentication should satisfy the following security requirements, which includes not only the security but also the privacy.

- Mutual Authentication: The system should have the ability to detect unauthorized users’ accessing and abort the requests. Meanwhile, users should have the ability to check the legitimacy of the specific access point.

- Anonymity: Except HNCC, no one could learn user’s real identity from the authentication interactions. The preservation of anonymity should also protect user’s location privacy.

- Unlinkability: To associate multiple authentications cannot get the knowledge whether they are from the same user.

- Key Establishment: The authentication scheme should provide a key agreement protocol to construct a random session key that is only shared between the FGS and the user.

- Forward/Backward Secrecy: It requires that the disclosure of the current session key would not affect the security of its future and previous session keys.

V. PROPOSED SCHEME:

ANFRA In this section, we first give the overview of the proposed anonymous and fast roaming authentication system. In the following, we give a detailed description of the protocol, which mainly consists of five phases: System Initialization, Pre-Negotiation, User Authentication, User Identity Reveal, and Dynamic User Enrollment and Revocation.

A. Overview

The overview of the proposed protocol is shown in Fig.2. After the system initialization, in each
domain, the gateway station (GS) sends a pre-negotiation message to each of its maintained LEOs in advance, which contains a parameter for key negotiation. A mobile user who wants to access the foreign network first needs to request to its home network control center (HNCC) to obtain associated access keys. When the user roams to a foreign network, he/she first sends an access request to the access point of foreign network (i.e., FLEO), which includes a signature that can verify the legality of the user. If the verification is passed, FLEO will send an access response to the user and FGS at the same time. The access response contains two key negotiation parameters that can then be used to build a secure channel between the user and FGS. Additionally, HNCC periodically publishes the revocation list to users, so that the unrevoked users can update their private key to the latest. To verify the legitimacy of roaming users, group signature is introduced in our scheme, in which the HNCC acts as the group manager, and authorizes FLEOs as the verifiers to check whether the access request is signed by an authorized roaming user (a group member). Therefore the HNCC can be offline during the authentication procedures. Thus, the authentication delay and interactions can be largely reduced. Additionally, group signature can provide good anonymity for the roaming users, so that the untrusted foreign network entities are unable to compromise users’ privacy. In addition to verifying the legitimacy of roaming users, a secure protocol also requires to verify the legitimacy of FLEO and FGS. For this purpose, we utilize a conventional digital signature scheme, i.e., Elliptic Curve Digital Signature Algorithm (ECDSA) [30] which is more efficient when compared to RSA signature [33].

B. Details of Our Proposed Scheme

1) System Initialization Phase:

In the system initialization phase, each NCC can be seen as key distribution center (KDC) in its domain, which first generates and assigns ECDSA’s signing/verifying key pairs for its GS and LEO. For clarity and without loss of generality, in the following description, we simplify the system model with only one LEO and GS that are associated with the user’s communication in each domain. And we denote the key pairs for the GS and LEO are (skGS, pkGS) and (skLEO, pkLEO) respectively. Then each NCC works as the group manager, and initializes its group by implementing Algorithm 1. The algorithm takes as input a parameter N, the number of members of the group, and process as follows. Firstly, NCC selects a generator g in the group G at uniformly at random. Then selects random numbers h ∈ G and ξ1,ξ2 ∈ Z∗ p, and sets u,v ∈ G such that u·ξ1 = v·ξ2 = h. Finally NCC selects a random number γ ∈ Z∗ p and computes ω = γ · g. Therefore, the group public key is gpk = (g,h,u,v,ω) which can be broadcast to all LEOs in its domain. And the corresponding private key of the group manager is gmsk = (ξ1,ξ2). When a mobile user Ui registers to its NCC, the NCC first generates a private key tuple gsk[i] = (Ai,xi), where xi ∈ R Z∗ p and Ai = 1 γ+xi · g. Then NCC sends gpk, gsk[i], pkGS, pkLEO and its identity IDNCC to Ui securely. Finally, NCC stores the tuple (IDUi,xi,Ai) in a user index table for revealing user’s identity. It should be noted that Algorithm 1 is only performed once at the beginning of the deployment of system. So we can ignore the computation cost for this phase. Additionally, the global trusted third party (TTP) also generates ECDSA’s signing/verifying key pairs for all NCCs in different domains. We denote the key pairs for FNCC and HNCC as (skFNCC, pkFNCC) and (skHNCC, pkHNCC) respectively, which is used for exchanging information between different domains. The information exchange steps in this phase are presented in Fig.3.

![Fig. 3. Information Exchange between HNCC and FNCC](http://indusedu.org)
timestamp $ts_0$ is within an allowed range compared to current time, and the signature is verified successfully by HNCC, the message will be stored by the HNCC. If a registering user has the roaming requirement, its NCC needs to securely deliver Mpks to the registering user in this phase. Then the HNCC generates message $Mpks = g|h||u|v||IDHNCC||ts_1$, where the IDHNCC is the identity of HNCC, $ts_1$ is a new timestamp. Then HNCC signs it with its private key $sk_{HNCC}$ and sends to FNCC. If the verification for timestamp and signature are passed, FNCC delivers the group public key $gpk$ to all LEOs in its domain.

2) Pre-Negotiation Phase: The pre-negotiation phase as shown in Fig.4 will be implemented between each LEO and GS in each domain. In this phase, each GS sends a prenegotiation message $MGS$ to the LEO. This message contains a parameter $grGS$ ($rGS$ is a random number selected by the GS), which will be utilized in the authentication phase for session key negotiation. A timestamp $ts_2$ is also involved for resisting replay attacks. Moreover, the GS signs the pre-negotiation message with its private signing key $sk_{GS}$ by ECDSA’s signature algorithm as $EC$.

Algorithm 1: Group Initialization

| Input: the number of users $N$ |
| Select a random generator $g \in_R G$ |
| Select a random number $h \in_R G$ |
| Select $\xi_1, \xi_2 \in_R Z_{p^2}$ |
| Set $u, v \in G$, such that $\xi_1 \cdot u = \xi_2 \cdot v = h$ |
| Select $\gamma \in R Z_{p^2}$ |
| Set $\omega = \gamma \cdot g$ |
| set the group public key as $gpk = (g, h, u, v, \omega)$ |
| set the group private key as $gsk = (\xi_1, \xi_2)$ |
| foreach user $U_i (1 \leq i \leq N)$ do |
| Select $x_i \in R Z_{p^2}$ |
| Set $A_i = x_i \cdot g$ |
| set the private key tuple as $gsk[i] = (A_i, x_i)$ |
| Send $gpk, gsk[i], pk_{LEO}, pk_{GS}, ID_{HNCC}$ to $U_i$ |
| Store the tuple $(ID_{HNCC}, x_i, A_i)$ in the user index table; |
| end |

![Fig. 4. Pre-Negotiation Phase](image)

(1) As in Algorithm 2, Ui firstly generates an access request, which contains an access request message and the corresponding signature. The access request message is $MU_i = TID_i||ID_{FLEO}||ID_{HNCC}||gr_i||ts_3$, where $TID_i$ is a temporary identity (not correlated in any way with the user’s real identity), $r_i$ is a random number selected by Ui, IDHNCC is the identity of Ui’s home NCC. IDFLEO is the identity of the FLEO that Ui is going to communicate to. Ui can infer the FLEO’s identity by utilizing their orbit parameters [34] which are obtained in the system initialization phase. And a timestamp $ts_3$ is also
generated and added to resist the replay attacks. The signature is \( \sigma_{Ui} = (T1, T2, T3, c, s_\alpha, s_\beta, s_x, s_\delta1, s_\delta2) \), in which, T1, T2, T3 are different computing results, c is a hash value, s_\alpha, s_\beta, s_x, s_\delta1, s_\delta2 are selected random numbers. The detailed signature generation process is shown in Algorithm 2 which is based on the group signature algorithm as that in [35]. After generating the access request \( M_{UI}||\sigma_{UI} \), the user sends it to the corresponding FLEO. It should be noted that some operations (e.g., e(h, w), e(h, g), e(g, g)) in the algorithm can be pre-computed and cached by the user to speed up user authentication. (2) For each received access request, the specific FLEO verifies the message and generates the access response message by implementing Algorithm 3. Firstly, the FLEO checks whether the timestamp ts3 is within an allowed range compared with its current time. If it is positive, the FLEO then checks whether the signature is valid. If the verification isn’t passed, the FLEO will reject the access request; otherwise, the FLEO reads the corresponding pre-negotiation message which has been cached in the pre-negotiation phase, and generates an access response message as MFLEO = TIDUI||IDFLEO||IDFGS||grUi||grFGS||ts4, where ts4 is a new timestamp, IDFLEO is FLEO’s identity. Then FLEO signs the response message MFLEO by its private key skFLEO. Finally, the FLEO sends the access response message with the corresponding signature to the roaming user and FGS. (3) Upon receiving the access response message from the FLEO, the user and FGS respectively implement Algorithm 4 to establish a secure channel between them. In this algorithm, the user and FGS first check the timestamp ts4. Then they verify the signature \( \sigma_{FLEO} \) by implementing \( \text{EC.Verify}(pkFLEO, \sigma_{FLEO}) \). If the verification is successfully passed, the user can compute the session key \( SK = (grFGS)r_{Ui} \), while the FGS obtains the session key \( SK \) by computing \( SK = (gr_{Ui})r_{FGS} \). 4) User Identity Reveal Phase: To reveal \( Ui \)’s identity, the HNCC collects the access message \( M_{UI} \) and its signature \( \sigma_{UI} = (T1, T2, T3, c, s_\alpha, s_\beta, s_x, s_\delta1, s_\delta2) \) from the FLEO. By inputting the group public key \( gpk = (g, h, u, v, \omega) \) and the corresponding group manager’s private key \( gmsk = (\xi1, \xi2) \), the signature reveal process can be implemented as that described in Algorithm 5. In this algorithm, HNCC first verifies whether the \( \sigma_{UI} \) is a valid signature on \( M_{UI} \), if it returns false, the signature reveal process will be stopped; otherwise, HNCC can compute the user’s private key \( A_i \) as \( A_i = T3 - \xi2 \cdot T1 - \xi2 \cdot T2 \). Then HNCC can further retrieve the user real identity \( ID_{UI} \) by looking up the user index.
the HNCC first selects a random number $x_{\text{new}} \in \mathbb{R}^*_p$ and computes $A_{\text{new}} = 1^{\gamma + x_{\text{new}} \cdot g}$. Then the HNCC sends $U_{\text{new}}$’s private key $(A_{\text{new}}, x_{\text{new}})$ and other system parameters (i.e., $g, u, v, h, o, p, k_{\text{FLEO}}, ID_{\text{HNCC}}, \text{orbit parameters}$) to the user securely. It is worth noting that there is no additional operation for the original users in the system when a new user registers to the system. However, some users may leave out of the system due to key loss, illegally usage, etc. To revoke these users, HNCC should periodically deliver a revocation lists which contains revoked users’ private keys (e.g., $(A_j, x_j)$) to unrevoked users. Considering the dynamic and unstable users and topology of SIN, we design a mechanism for revocation lists distribution as the Fig. 6. Since satellites and mobile users are energy-limited, we adopt the incremental update of revocation lists (RL), that is, HNCC periodically broadcasts the increased entries to the FGS, which is then broadcast to the online users through FLEO as well. Meanwhile, FGS stores the full revocation lists in local memory during this process. To make sure the full revocation lists are synchronous among all FGSs in different domains, once a FGS doesn’t receive the incremental revocation lists from other domain NCC at the broadcast time, it should request to the NCC for the incremental RL. However, some offline users who may miss one or several updates of incremental RL, should firstly request the missing part of revocation list entries from FGS. After receiving the request from the offline user, the FGS looks up its local full RL to retrieve the missing revocation list entries for user, and sends back to the offline user. After these procedures, both the online and offline users can accept the latest revocation list entries for them.

After obtaining the latest revocation lists, both the online and offline users can update their private key to the latest.

Following process shows the updating operations for user $U_i$ to revoke a user $U_j$ in his/her revocation lists at a time. By repeating the process $r$ times, the user can revoke $r$ users on his/her revocation lists.

1. Update $g$ as $\hat{g} = A_j \cdot \frac{1}{x_j} \cdot g$.
2. Update the public key $\omega$ as $\hat{\omega} = g - x_j \cdot A_j = \gamma \cdot \hat{g}$.
3. Update $A_i$ (for $i \neq j$) as $\hat{A}_i = \frac{1}{x_i - x_j} \cdot A_j - \frac{1}{x_i - x_j} \cdot A_i = \frac{1}{x_i - x_j} \cdot \hat{g}$.
4. Set the new private key as $(\hat{A}_i, x_i)$.

Although the revocation-support mechanism brings in some additional overhead of communication and computation, users can perform the operations offline, and the performance analysis (shown in Section VII-C) also shows that our revocation mechanism is efficient. Since the number of GSs in each domain is very small and the foreign network entities FLEO and FGS only need to perform a few simple operations in our scheme, such as forwarding and storing RL, it is practical to deploy this revocation

---

**Algorithm 4: Secure Channel Establishing**

**Input:** Access Response $M_{1i}[\|p_{\text{FLEO}}, \text{FLEO}'s public key $pk_{\text{FLEO}}$, parameter $r_{\text{H}}, r_{\text{T}}$]

**Output:** Session key $SK$;

1. Check whether timestamp $t_{\text{H}}$ is within an allowed range;
2. If the verification of $t_{\text{H}}$ is not passed then reject the access response;
3. Else check the signature by $ECVerify(pk_{\text{FLEO}}, \sigma_{\text{FLEO}})$;
4. If not passed then drop the access response;
5. Set $SK = (g^{\gamma \cdot r_{\text{H}}})^{r_{\text{T}}}$ or $SK = (g^{\gamma \cdot r_{\text{H}}})^{r_{\text{T}}}$;
6. Return $SK$;
7. End

---

**Algorithm 5: Signature Reveal**

**Input:** Login Request $M_{2i} || \sigma_{\text{H}}, \text{group public key } gpk$; group manager’s private key $gmask$;

**Output:** Real identity $ID_{\text{H}}$;

1. Check the signature $\sigma_{\text{H}}$;
2. If $\sigma_{\text{H}}$ is not valid then stop the process;
3. Else compute user’s private key $A_i = T_3 - T_1 - T_2$;
4. Retrieve user real identity $ID_{\text{H}}$ by $A_i$ in the user index table;
5. Return Real identity $ID_{\text{H}}$;
6. End
mechanism in actual SIN system. More importantly, the designing revocation mechanism helps to accelerate the authentication procedure, since the verifier (FLEO) is no longer necessary to perform the time-consuming operation of checking revocation list when verifying users (as in some existing schemes).

VI. SECURITY ANALYSIS

In this section, we analyze the security of the proposed AnFRA to verify whether the security requirements introduced in Section IV-C have been satisfied.

A. Mutual Authentication User authenticates the identity of FLEO by the challenge-response pair (grUi,σFLEO), where σFLEO = EC.Sign(skL,MFLEO). Since the Elliptic Curve Digital Signature Algorithm (ECDSA) has been proven secure under the assumption that the discrete logarithm problem is hard and that the hash function employed is a random function [30], without knowing the private key skL, it is infeasible to forge a valid signature on Ui’s freshly generated challenge grUi with non-negligible probability. Moreover, the identity of FLEO and its public key pkFLEO have been bound and published to users by trusted TTP during the system initialization phase. Therefore, any other LEOs cannot cheat by using different public keys or different identities. And the user authentication is achieved by another challenge-response pair: (grUi ,ts3, σUi). Only a legitimate group member can generate a valid group signature on Ui’s challenge{grUi ,ts3}. Thus, mutual authentication is achieved between user and FLEO. And the mutual authentication between FLEO and FGS is also achieved through the challenge-response pairs: (grFGS,ts2,σFGS) and (grFGS,ts4,σFLEO). Because only the legitimate FGS and FLEO can generate a valid signature on challenge {grFGS,ts2} and {grFGS,ts4} respectively.

B. Conditional Anonymity and Unlinkability In our protocol, the signature on the access message MUi = {TIDUi||IDFLEO||IDHNCC||grUi||ts3} is the form of group signature [30], which satisfies the security requirements of anonymity, that is, given the group signature σUi of the access message, it is computationally difficult to identify the actual signer by the entity who does not have the group manager private key gmsk. However, this anonymity is conditional. The real identity of signature can be revealed by HNCC by carrying out the User Identity Reveal Phase: given a signature σ = {T1,T2,T3,c,sα,sδ1,sδ2}, HNCC uses its private key gmsk = (ξ1,ξ2) to reveal the user’s private key A by computing A = T3−ξ1·T1−ξ2·T2, and retrieves the user’s real identity IDUi by looking up the user index table. Thus, conditional privacy is achieved. Moreover, user unlinkability is achieved: given two group signature σ1 and σ2, according to the verification procedure, it is computationally hard to decide whether these two valid signatures are computed by the same group member.

C. Key establishment and forward/backward secrecy In each session, the session key SK is computed from key negotiation parameters grUi and grFGS. Computing session key SK from these two parameters without knowing rUi and rFGS is equivalent to solve the discrete logarithmic problem (DLP), which is known that it is computationally infeasible [36]. So the session key cannot be derived by any adversary. Besides, the key forward/backward secrecy is mainly achieved by the independence of the session key SK in different sessions. In our scheme, each session is used different fresh random number rUi for key establishment, which makes the independence of each session key possible, that means an attacker cannot acquire the next or the previous session key even though he/she has obtained the current session key.

D. Resistance of modification attacks Suppose that an attacker intercepts the access request MUi||σUi and modifies it. On the one hand, since the attacker does not possess the private key of the user, the attacker does not have any ability to compute a valid σ0Ui on a modification message M0 Ui. Once the FLEO verifies the signature, the signature would be found to be invalid. On the other hand, if an attacker modifies the access response MFLEO||σFLEO as M0FLEO||σ0FLEO. Without FLEO’s private key skFLEO, the forging signature σ0FLEO could not be passed from the verification algorithm EC.Verify(pkFLEO,σ0FLEO). As a result, AnFRA successfully prevents the unauthorized modifications.
E. Resistance of replay attacks

It is noted that the access request message $M_{Ui} = TID_{Ui}||ID_{FLEO}||ID_{HNCC}||gr_{Ui}||ts_3$ contains a timestamp $ts_3$, which is hashed to get

$$c = H(M,T_1,T_2,T_3,R_1,R_2,R_3,R_4,R_5).$$

Then, it would be signed as

$$\sigma_{Ui} = (T_1,T_2,T_3,c,s_{\alpha},s_{\beta},s_x,s_{\delta_1},s_{\delta_2}).$$

Because of the above steps, the timestamp cannot be modified and replaced.

So the access request would be rejected if FLEO checks the timestamp $ts_3$ is invalid. And the access response message $ML$ is also appended a timestamp $ts_4$ and signed by EC.Sign $(sk_{FLEO},MFLEO)$, which guarantees the timestamp cannot be modified and replaced. So the replaying messages could be found by checking the timestamps and signatures. Therefore, the proposed scheme is able to resist replay attacks.

F. Resistance of impersonation and man-in-the-middle attacks

An attacker may impersonate a legal user by forging an authentication request. However, available private keys (e.g., $(x_i,A_i)$) are only mastered by legal users, an attacker without a valid private key is unable to forge a valid signature (e.g., $\sigma_{Ui}$) with non-negligible probability. Therefore, the attacker cannot impersonate a legal user. In addition, FLEO makes a signature $\sigma_{FLEO}$ for each message $MFLEO$. The impersonation for a legal FLEO would be failed, when the user checks the signature by algorithm $EC_V erify(pk_{FLEO},\sigma_{FLEO})$, where the signing key for $\sigma_{FLEO}$ is only held by legal FLEO. Thus, AnFRA is secure against impersonation attacks. A man-in-the-middle attacker tries to trick two parties into a three-party communication. According to the proof of impersonation attacks, a man-in-the-middle attacker fails to impersonate both a legal user to FLEO and a legal FLEO to the user. Therefore, our protocol is secure against the man-in-the-middle attacks.

VII. PERFORMANCE ANALYSIS

In this section, we analyze the performance of our scheme for authentication delay, communication overhead and revocation overhead.

A. Authentication Delay

The authentication delay is defined as the total time costs during the whole authentication process, including the time costs of computations and signal propagation. In this paper, we denote the time costs of signal propagation between the user and FLEO, FLEO and FGS, FGS and HNCC as $T_{U-FL}$, $T_{FL-FGS}$, $T_{FGS-HNCC}$, respectively. Since the FLEOs are 500 to 2,000 kilometers away from ground [10], it is reasonable to set $T_{U-FL} = T_{FL-FGS} = 10ms$ and $T_{FGS-HNCC} = 5ms$. We investigate the time costs of the primitive cryptography operations using OpenSSL library [37] on Intel P IV 3 GHz processor. And the experiment results in [26] show that the time costs for performing a pairing operation, multiplication, and exponentiation are $T_{mul} = 0.376ms$, $T_{exp} = 0.387ms$ and $T_{pair} = 11.903ms$, respectively. Based on these information, we compares the authentication delay in AnFRA and the related work [8, 19–21, 23, 27] in Table I. It is noted that all these related works are not originally designed for roaming authentication in SIN, so we make some appropriate modifications for them. In general, the verifier in [8, 19, 20] is moved to HNCC so that users’ sensitive information (e.g., real identity) won’t be leaked to foreign network entities, and the verifier in [21, 23, 27] is moved to FGS to reduce the authentication delay while protecting users’ privacy. From the table, it can be seen that a successful roaming authentication in AnFRA needs 14 multiplication operations, 6 exponentiation operations, 1 pairing operation, and 2 signal propagation time between user and FLEO. Totally, it requires almost 40ms for a roaming authentication, which is significantly faster than the related works. Additionally, we give the comparison of the computation costs for
components with the related works. Most three-party authentication schemes only use the lightweight cryptography algorithm, such as hash, symmetric encryption. Their computation costs can be negligible when compared to pairing operation. Therefore, we just compare our work with some two-party authentication schemes as in Table II. From the table, we can see that the total computation costs for our scheme is more efficient than [23] and [27], and closer to [21].

B. Communication Overhead To compare our scheme with the above different authentication schemes in terms of communication overhead, we set the length of all identities and time stamps as 100 bits, the length of random numbers and encryption messages as 1024 bits, and all signatures and |G| are 160 bits. Based on these assumptions, we compare the communication overhead in terms of UserFLEO, FLEO-FGS, FGS-HNCC and Total respectively. The results are shown in Table III. It can be seen that the proposed AnFRA is more efficient than most of the related works in terms of User-FLEO, FLEO-FGS, FGS-HNCC, or Total.

<table>
<thead>
<tr>
<th>TABLE III</th>
<th>COMPARISON OF COMMUNICATION OVERHEAD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>User-FLEO</td>
</tr>
<tr>
<td>[8]</td>
<td>3272</td>
</tr>
<tr>
<td>[19]</td>
<td>2248</td>
</tr>
<tr>
<td>[20]</td>
<td>3272</td>
</tr>
<tr>
<td>[21]</td>
<td>3812</td>
</tr>
<tr>
<td>[23]</td>
<td>3568</td>
</tr>
<tr>
<td>[27]</td>
<td>940</td>
</tr>
<tr>
<td>AnFRA</td>
<td>1600</td>
</tr>
</tbody>
</table>

*The communication overhead is represented in bits.

C. Revocation Overhead As shown in the dynamic user enrollment and revocation phase in Section V-B5, a user needs to perform 2 multiplication operations to update its private key, while a FLEO requires one multiplication operation to update the public verifying key. And the FGS and HNCC do not need to perform any computing operation. Fig. 7(a) shows the computation overhead of revocation for a user and FLEO (assume 10 users are revoked one time). It is noted that our scheme is efficient for the user and FLEO, only 0.75s will be cost for a user during a month when the update frequency is 100 per month, and FLEO is less. For evaluating the total communication costs for revocation overhead of the whole system, we depict Fig. 7(b) to show the variation of total communication costs in terms of the number of revocation entries in RL, where the elements in G and Zp are 160 bits, and an additional index in RL is 10 bits. It can be seen that the whole system’s communication costs are increasing with the increase of entries number. However, it is acceptable since only approximately 12 MB costs for an online user and 16 MB costs for an offline user when the number of entries in RL is 100.

VIII. CONCLUSION

Space information network (SIN) can break regional restrictions and provide wider coverage comparing with traditional Internet. The trend of roaming to SIN will be a new feature of the future network, which calls for designing a new roaming authentication scheme for SIN. While challenges exist for designing a roaming authentication system for SIN due to its special environment (e.g., the dynamic and unstable topology, the highly exposed links, the long latency). Motivated by the importance of user authentication delay and anonymity for roaming in SIN, we design an anonymous and fast roaming authentication protocol (named AnFRA). In AnFRA, we utilize the group signature and emphasize the authentication of foreign LEO (FLEO), that means the FLEO can directly authorize roaming users to access the foreign network without the realtime involvement of home network control center (HNCC) and without privacy disclosure. Moreover, a revocation mechanism designed specifically for the system is incorporated into the roaming authentication scheme to support users revocation. Although a small amount of overhead is brought in owing to the revocation mechanism, it can largely reduce the authentication delay. In addition, the system satisfies a set of more strict security features, while enjoys a lower authentication delay and less communication overhead.

ACKNOWLEDGMENT

The authors sincerely thank the anonymous referees for their valuable suggestions that have led to the present improved version of the original
manuscript. This work is partly supported by the National Key Research and Development Program of China under Grant No. 2016YFB0800301, the National Natural Science Foundation of China under Grant No. 91538203, Youth Innovation Promotion Association CAS under Grant No. 2016394 and the Fundamental Research Funds for the Central Universities. Kaiping Xue is the corresponding author of this paper.

REFERENCES


