Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 8811970, 14 pages
https://doi.org/10.1155/2021/8811970

Research Article

An Efficient ECC-Based Authentication Scheme against Clock Asynchronous for Spatial Information Network

Huihui Huang,[1] **Xuyang Miao**,[2] **Zehui Wu**,[1] **and Qiang Wei**[1]

[1]*State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China*
[2]*Henan Key Laboratory of Network Cryptography Technology, Zhengzhou 450001, China*

Correspondence should be addressed to Zehui Wu; wuzehui2010@foxmail.com

Received 17 September 2020; Revised 29 December 2020; Accepted 11 January 2021; Published 2 February 2021

Academic Editor: Haopeng Zhang

With the rapid development of mobile communication technology, the spatial information networks (SIN) have been used for various space tasks' coverage in commercial, meteorology, emergency, and military scenarios. In SIN, one basic issue is to achieve mutual authentication and secret communication among the participants. Although many researches have designed authentication schemes for SIN, they have not considered the situation where the clock is not synchronized as the broad coverage space in wireless environment. In this paper, we disclose several flaws of Altaf et al.'s scheme (2020), in which the main weakness is that a malicious user can easily obtain the master key of the network control center after launching the offline password-guessing attack. Then, we design an authentication scheme against clock asynchronous for SIN by utilizing elliptic curve cryptosystem (ECC) and identity-based cryptography (IBC). Based on a brief introduction to the main design ideas of our scheme, the security protocol analysis tools of Scyther and AVISPA are used to prove that the scheme can resist various existing active and passive attacks. We further discuss our scheme that provides five essential requirements of security properties to design a robust scheme for SIN and is superior in terms of resistance to security functionality and computational performance by comparison with two other representative schemes. As a result, our scheme will be workable and efficient security for mobile users in the actual environment.

1. Introduction

As the pace of human exploration has spread across the entire Earth and even the deep universe, spatial information networks (SIN) have been proposed to meet the rapidly growing needs of mobile communications. SIN is a backbone communication network composed of multiple satellites in orbit and a satellite constellation, which is intrinsically a radio-based transmission medium in the wireless mobile environment [1]. It can provide communication and broadcasting services for various space tasks in professional, commercial, military, and emergency scenarios as it overcomes the shortcomings of geographic and environment limitations in traditional personal communication systems (e.g., LTE-A networks and Wi-Fi) [2]. Therefore, users will be more willing to access SIN to obtain network services, which has become a hot spot in global research today. In SIN, the typical model is the low-earth-orbit satellite communications (LSC) system [2, 3], which consists of the low Earth orbit satellites (LEOS), the ground station/gateway (G), the network control center (NCC), and mobile users (U), as shown in Figure 1.

Recently, quite a lot of access authentication protocols have been designed for LSC system [4–22], but many schemes in the literature only provide unilateral or ineffective properties. For detailed literature research introduction, we refer readers to [21]. In 2020, Altaf et al. [22] proposed a lightweight authentication scheme for LSC system and claimed that it is protected against all possible security threats. However, we discover that their scheme is vulnerable to several drawbacks. Firstly, it has offline password-guessing attack because the smartcard records sensitive data during the registration phase. After lunching the above attack, a malicious user can easily get the secret number of the master control server, which is a crucial parameter to the entire system. Secondly, it is a common

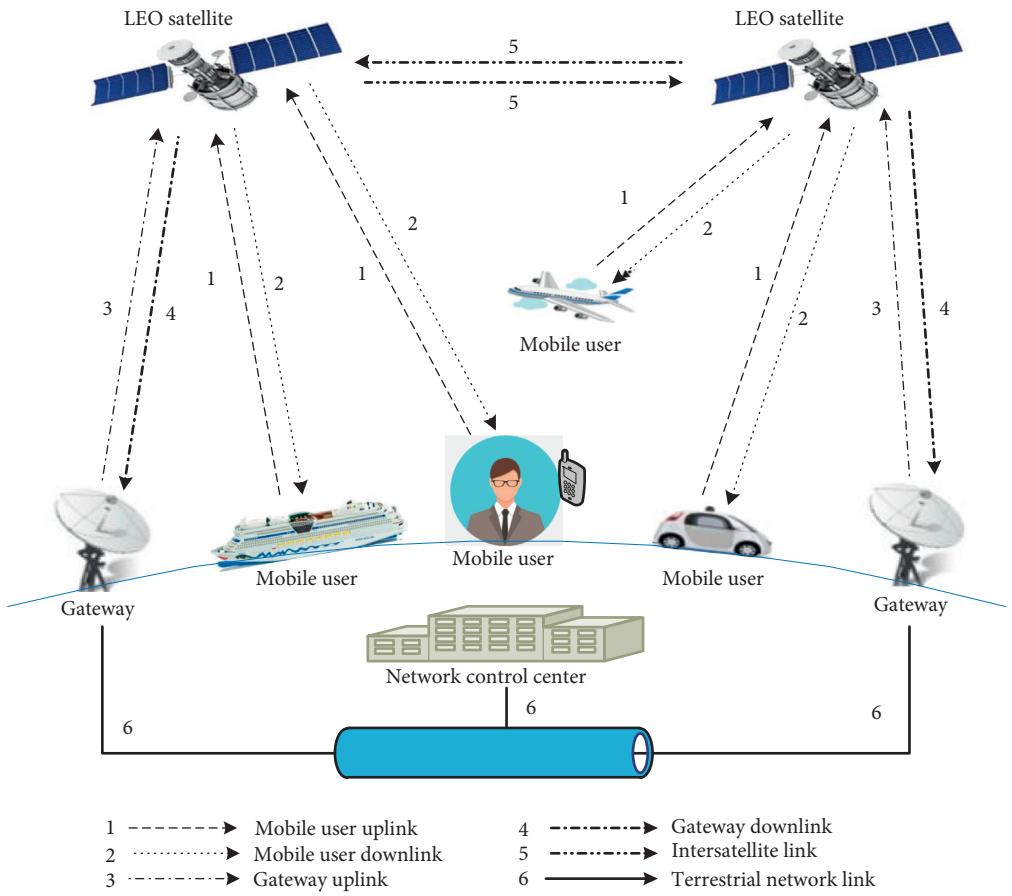


Figure 1: A low Earth orbit satellite communications system.

weakness that most protocols apply the freshness of timestamp to verify the validity of the message, which are not suitable for clock asynchronous environment as highly exposed links and extremely high propagation delay in SIN [23–25]. To overcome the shortcomings, we utilize elliptic curve cryptosystem (ECC) and identity-based cryptography (IBC) to design an efficient authentication scheme against clock asynchronous for SIN.

On the basis of analyzing the related scheme, we summarize several essential requirements of security and functionality which should be premeditated for designing a robust scheme in LSC system:

(1) Valid mutual authentication: All joined entities in a communication system should identify each other and communicational messages should be authenticated to come from the original sender. Compared with identity authentication, information authentication should be verified more carefully because it is usually overlooked and eavesdropped in a LSC system.

(2) Data confidentiality and integrity: It is well accepted to keep the data secrecy. Apart from that, the data integrity is also crucial. To protect the data integrity, an effective scheme should have the ability to detect the data manipulation including insertion, deletion, and substitution.

(3) No sensitive data maintained by the NCC and U: In order to achieve mutual authentication between the NCC and U, a simple scheme should avoid sensitive data stored in both terminals, such as the verifier table in NCC's terminal and the values stored in U's mobile device.

(4) Perfect session key secrecy: All session keys used to encrypt the exchange data between the NCC and U are kept in secret and the compromise of session keys does not divulge the forward/backward session keys.

(5) User's privacy: Any information of U should be protected from outsiders and some vital information, such as password, should be even kept secret from NCC.

(6) Fast computation and few communication costs: On one hand, large storages and high calculations will be worthless in U's lightweight equipment as its resource is constrained; on the other hand, due to the need for multiple forwarding and exposure to the wireless environment, the fewer communication costs each time, the better.

The rest of this paper is organized as follows. Section 2 provides necessary techniques used in this paper. In Section 3, we review Altaf et al.'s scheme and point out the security weaknesses in detail. The proposed scheme is introduced in Section 4. Results and discussion of our scheme are given in Section 5. Finally, we draw some concluding remarks of this paper in Section 6.

2. Technical Background

2.1. Elliptic Curve Cryptosystem (ECC). ECC was firstly put forward by Miller [26] and Koblitz [27] to design public key cryptosystem. It gives more security with less bit size key and faster computation than traditional public key cryptography. For example, Hankerson et al. [28] pointed out that 160-bit ECC and 1024-bit RSA or DLP has the same security level in practice. Hence, it has been widely used in several cryptographic schemes to provide desired level of security and adequate computational efficiency [29–32].

In this section, we just give a simple description of the ECC defined over a prime field F_p . A nonsingular or secure elliptic curve $F_p(a,b)$ over F_p is defined by an equation $y^2 = x^3 + ax + b \pmod p$, $a, b \in F_p$,and with the discriminant $\Delta = 4a^3 + 27b^2 \neq 0 \pmod p$. Then, the set $G_p = \{(x,y) | x, \quad y \in F_p$ and $(x,y) \in E_p(a,b)\} \cup \{O\}$ can form a cyclic additive elliptic curve group, where point O is identity element of G_p . Let P be a base of G_p with an order n, since $nP = O$ for the smallest integer $n > 0$. The point multiplication on group G_p is defined as $kP = P + P + \cdots + P$ (k times). Details of elliptic curve group properties are given in [28].

2.2. Computational Problem

(1) Discrete logarithm problem (DLP): Assume that g is a generator of Z_p^* , where Z_p^* is a finite multiplicative cycle group and p is a prime number. Given $b, g \in Z_p^*$, find a number $a \in Z_p^*$ such that $b = g^a \pmod p$ is difficult.

(2) Elliptic curve discrete logarithm problem (ECDLP): Consider two points $Q, P \in G_p$; then find a number $k \in [1, n-1]$ such that $Q = kP$ is impossible, where G_p is cyclic additive elliptic curve group.

(3) Computational Diffie-Hellman problem (CDHP) on ECC: Assume that there are three known points P, aP, bP for $a, b \in [1, n-1]$, where all belong to group G_p . Then, the computation of abP is also hard to G_p.

(4) Collision attack assumption 1 (k-CAA1) [30]: Assume that there exist a positive integer k and some values $\{P, xP, h_0, (h_1, (x+h_1)^{-1}P), \ldots, (h_k, (x+h_k)^{-1}P)\}$, where $P \in G_p$, $x, h, \in Z_n^*$ with $0 \leq i \leq k$; then it is even difficult to explore the value $(h_0 + x)^{-1}P$.

2.3. Adversarial Model. In this section, we give the threat attack model, where the main reference is Dolev-Yao adversary threat model [33]. In 5.1, we will use this model to simulate the attack on our proposed scheme with the formalization tool Scyther. The result of the attack path is shown in Figure 2. The detailed descriptions of Dolev-Yao adversary threat model are as follows:

(1) Adversary can eavesdrop and intercept all messages passing through the network.

(2) Adversary can store and send the intercepted or self-constructed messages.

(3) Adversary can participate in the operation of the protocol as a legal subject.

In addition to this, since our scheme uses smart cards, we assume that the attacker has the following capability, which can be used to launch offline password-guessing attacks [34]. However, we will give the detailed description of how the proposed scheme can prevent this attack in the overall design idea of our scheme in Section 5.1.

(4) The power analysis or side-channel attacks can help the attacker to extract the secret information stored in user's smart card.

3. Altaf et al.'s Scheme and Its Weaknesses

This section reviews Altaf et al.'s scheme [22] and points out its flaws. As shown in Figure 1, there are mainly 3 types of participants in their scheme: a mobile (U_i), network control center (NCC_s), and the low-Earth-orbit satellites (LEOS_q). The notations used in this paper are defined in Table 1.

3.1. Altaf et al.'s Protocol. Altaf et al.'s protocol consists of registration phase and login and authentication phase, which are presented in Figure 3.

3.1.1. Registration Phase. When wanting to get the servers from NCC_s, U_i has to register to NCC_s as in the following steps:

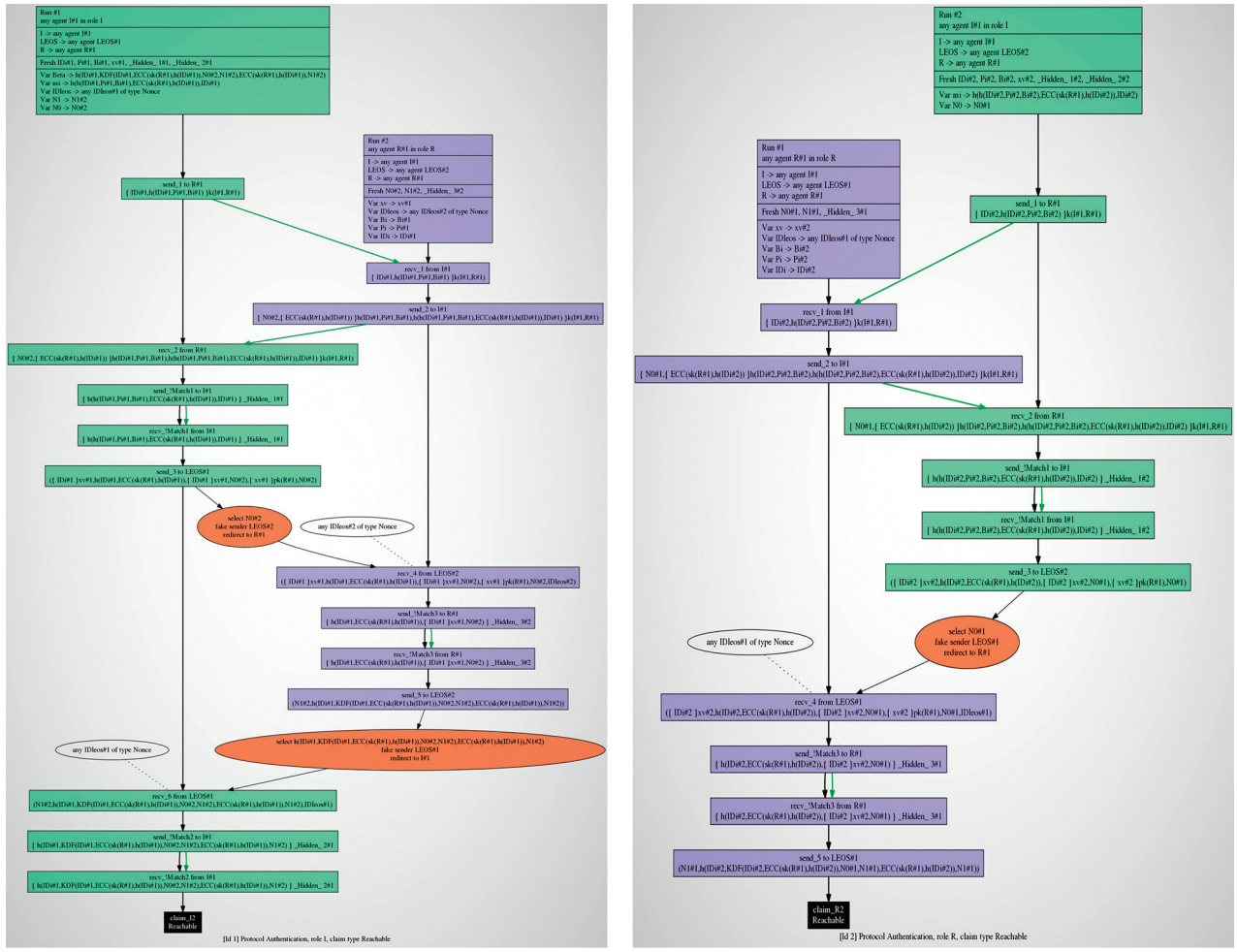
Step 1: U_i chooses his/her identity ID_i and password P_i and generates a random number n_i. Then, U_i calculates $\overline{P}_i = h(ID_i \| P_i \| n_i)$. After that, U_i sends the registration message $\langle \text{ID}_i, \overline{P}_i \rangle$ to NCC_s via a secure channel.

Step 2: After receiving request message from U_i, NCC_s computes the following operations: $X_i = h(\text{ID}_i \| \text{msk})$, $Y_i = X_i \oplus \overline{P}_i$, and $M_i = h(\overline{P}_i \| X_i \| \text{ID}_i)$. Next, NCC_s records the values $\langle Y_i, M_i, h(\cdot) \rangle$ into a smart card SC_i and delivers it to U_i through a secure channel.

Step 3: When getting SC_i , U_i computes $E_i = n_i \oplus h(\text{ID}_i \| P_i)$ and writes the value E_i into SC_i. Finally, SC_i stores the values $\langle Y_i, M_i, E_i, h(\cdot) \rangle$

3.1.2. Login and Authentication Phase. When U_i wishes to login to NCC_s, the following steps are executed between U_i and NCC_s with the help of LEO satellite:

Step 1: U_i inserts SC_i and enters his/her ID_i^*, P_i^* to compute $n_i^* = E_i \oplus h(ID_i^* \| P_i^*)$, $\overline{P}_i^* = h(ID_i^* \| P_i^* \| n_i^*)$, $X_i^* = Y_i \oplus \overline{P}_i^*$, and $M_i^* = h(\overline{P}_i^* \| X_i^* \| ID_i{}^*)$. Then, SC_i



(a) (b)

Figure 2: Attack path under the Dolev-Yao model using Scyther. (a) Path of U_i authenticating NCC_s. (b) Path of NCC_s authenticating U_i.

Table 1: Notations used in this paper.

Symbol	Description
LSC	Low-earth-orbit satellite communications system
U_i	Mobile user
NCC_s	Network control center
LEOS_q	Low-Earth-orbit satellite
Adv	Attacker
ID_i	Identity of U_i
ID_{leos}	Identity of LEOS_q
SC_i	Smart card issued to each specific U_i
P_i	Password of U_i
n_i, b_i	Random numbers generated by U_i
n_s	Random numbers generated by NCC_s
msk, mpk	Private/public key of NCC_s
SK	Share session key
T_1, T_2	Timestamp
P	Point multiplication
$h(\cdot)$	Hash function: $(0,1)^l \longrightarrow (0,1)^n$
$\oplus$	Bitwise XOR operation
$\|$	Concatenate operation

verifies the condition $M_i^* = ?M_i$. If $M_i^* \neq M_i$, SC_i rejects this login of U_i; otherwise, SC_i randomly generates a number b_i to calculate $Q_i = b_i P$, $V_i = b_i$.mpk, $\mathrm{PID}_i = V_i \oplus \mathrm{ID}_i$, and $\mathrm{Auth}_i = h(\mathrm{ID}_i \| B_i \| \mathrm{PID}_i \| T_1)$. Afterwards, SC_i sends $\langle \mathrm{PID}_i, \mathrm{Auth}_i, Q_i, T_1 \rangle$ to NCC_s.

Step 2: On receiving the login message from U_i, LEOS_q forwards the message $\langle \mathrm{PID}_i, \mathrm{Auth}_i, Q_i, T_1, \mathrm{ID}_{\mathrm{LEOS}_q} \rangle$ to NCC_s.

Step 3: When getting the message, NCC_s checks the freshness of timestamp by verifying the condition $(T_1 - T_s) \leq ?\Delta T$. If ΔT is not permissible to the NCC_s, this session is terminated; otherwise, NCC_s computes $V_i^* = \mathrm{msk}.Q_i$, $\mathrm{ID}_i = V_i^* \oplus \mathrm{PID}_i$, $B_i = h(\mathrm{ID}_i \| \mathrm{msk})$, and $\mathrm{Auth}_i^* = h(\mathrm{ID}_i \| B_i \| \mathrm{PID}_i \| T_1)$ and checks $\mathrm{Auth}_i^* = ?\mathrm{Auth}_i]$. If $\mathrm{Auth}_i^* = \mathrm{Auth}_i$, NCC_s thinks that U_i is real and generates a random number n_s. Next, it calculates $W_s = n_s \oplus h(\mathrm{ID}_i \| T_2)$, $\mathrm{SK} = h(\mathrm{ID}_i \| X_i \| n_s \| T_2)$, and $\mathrm{Auth}_s = h(\mathrm{ID}_i \| \mathrm{SK} \| X_i \| n_s \| V_i^* \| T_2)$ to send $\langle W_s, \mathrm{Auth}_s, T_2, \mathrm{ID}_{\mathrm{LEOS}_q} \rangle$ to LEOS_q finally.

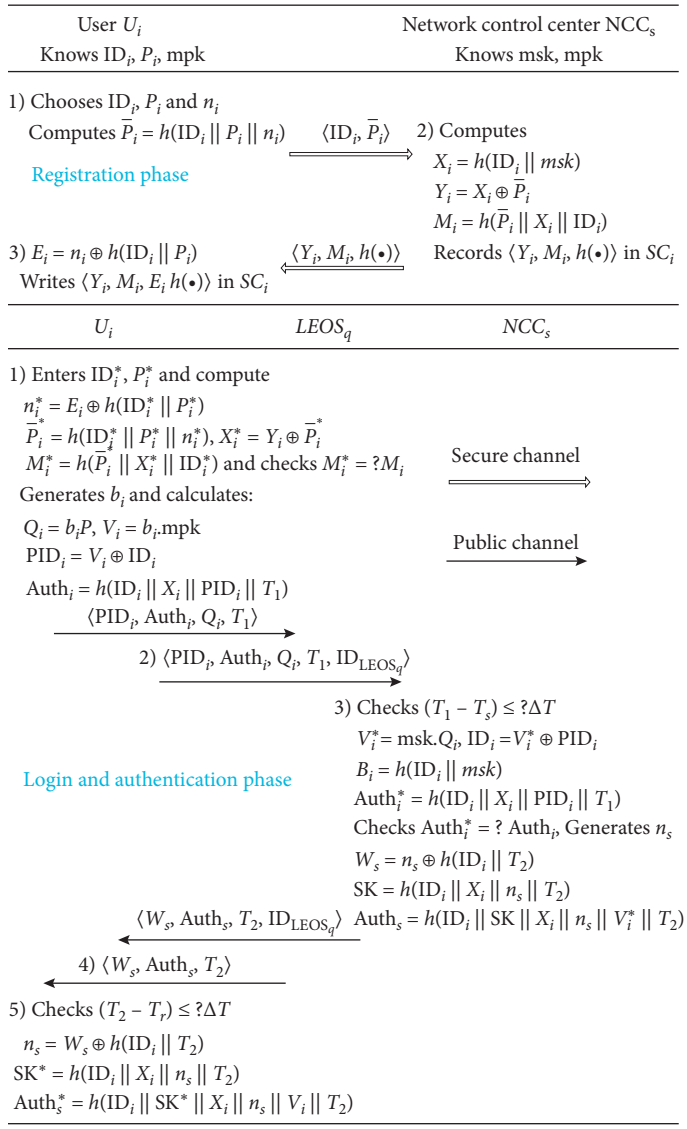


Figure 3: Implementation of Altaf et al.'s scheme.

Step 4: Upon receiving response message, $LEOS_q$ forwards $\langle W_s, Auth_s, T_2 \rangle$ to U_i.

Step 5: After receiving the message, U_i checks the freshness of timestamp by verifying $(T_2 - T_r) \le ?\Delta T$. If $(T_2 - T_r) \le \Delta T$, U_i authenticates NCC_s and computes $n_s = W_s \oplus h(ID_i \| T_2)$, $SK^* = h(ID_i \| X_i \| n_s \| T_2)$, and $Auth_s^* = h(ID_i \| SK^* \| X_i \| n_s \| V_i \| T_2)$. If $Auth_s^* = Auth_s$, U_i and NCC_s achieve mutual authentication and negotiate a shared secret key.

Remark 1. There are two flows in this phase of Altaf et al.'s protocol. The first is the inconsistency operation between the flow chart display in Figure 4 of $\overline{P}_i^* = h(P_i^* \| ID_i^* \| n_i^*)$ and the description in *Step AP1* of $\overline{P}_i^* = h(ID_i^* \| P_i^* \| n_i^*)$. The second is that NCC_s should append $LEOS_q$'s identity information when sending this message $\langle W_s, Auth_s, T_2, ID_{LEOS_q} \rangle$ to $LEOS_q$, instead of $LEOS_q$ sending its own identity to U_i, because $LEOS_q$ only needs to have identity confirmation with NCC_s in the LSC system.

3.2. Security Analysis of Altaf et al.'s Scheme. In this section, we carefully make security analysis of Altaf et al.'s scheme [22]. Firstly, we review the adversarial model in their article, which supposed that Adv has the following abilities when attacking the efficiency and security of their scheme:

(1) Adv can access the full public communication channel to modify, replay, amend, and intercept the confidential information.

(2) Adv can extract the secret information stored in user's smart card with the help of power analysis.

(3) Adv can cheat the user by making the legitimate member of that system.

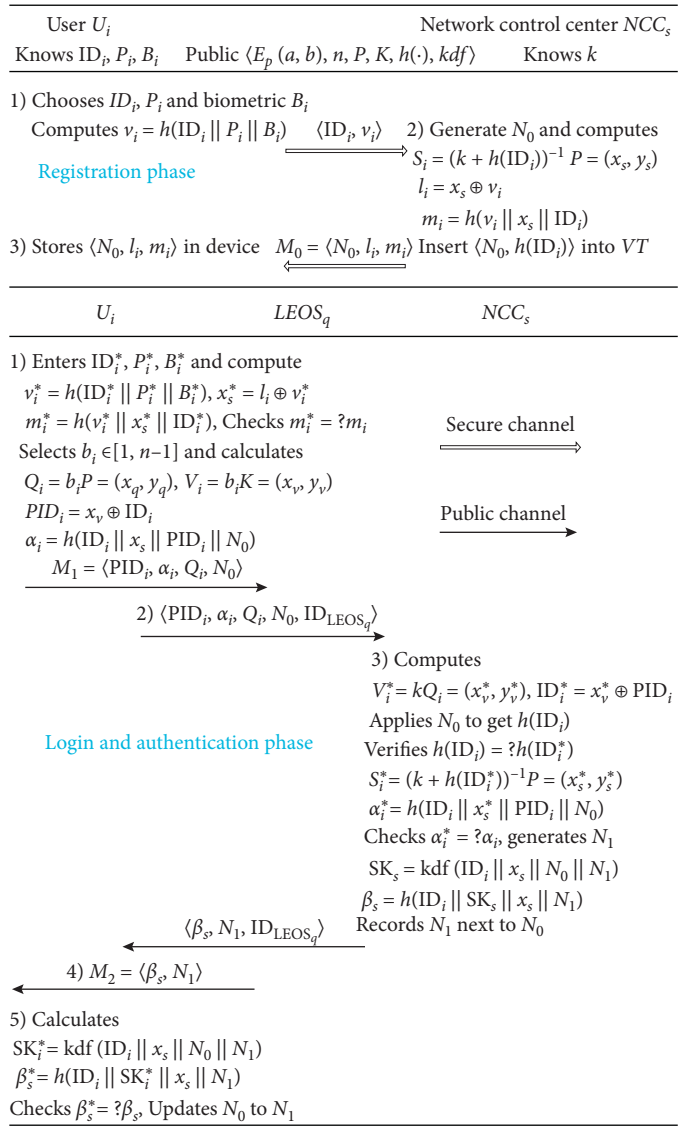


Figure 4: Implementation of two stages in our scheme.

3.2.1. Offline Password-Guessing Attack. This section describes how a malicious attacker can obtain the private key msk of NCC_s after launching the above attack in Altaf et al.'s protocol. The details are given in the following steps:

Step 1: According to the adversarial model of (3), Adv can register in network control center like a normal user. Firstly, Adv computes $\overline{P}_{Adv} = h(ID_{Adv} \| P_{Adv} \| n_{Adv})$ with arbitrary selection of identity ID_{Adv}, password P_{Adv}, and a random number n_{Adv}. Then, Adv records the values $\langle \overline{P}_{Adv}, ID_{Adv}\rangle$ and sends $\langle ID_{Adv}, \overline{P}_{Adv}\rangle$ to NCC_s.

Step 2: After receiving the login message from Adv, NCC_s calculates $X_{Adv} = h(ID_{Adv} \| msk)$, $Y_{Adv} = X_{Adv} \oplus \overline{P}_{Adv}$, and $M_{Adv} = h(\overline{P}_{Adv} \| X_{Adv} \| ID_{Adv})$. Then, it will issue a smart card SC_{Adv} recoding the values $\langle Y_{Adv}, M_{Adv}, h(\cdot)\rangle$ to Adv.

Step 3: Based on the adversarial model of (1), Adv can extract the values Y_{Adv} and $h(\cdot)$ after getting SC_{Adv}. Next, Adv computes $X_{Adv} = Y_{Adv} \oplus \overline{P}_{Adv}$ and records the value X_{Adv}.

Step 4: Since $X_{Adv} = h(ID_{Adv} \| msk)$, Adv can launch the offline password-guessing attack by implementing Algorithm 1.

Although this algorithm may take a long time to execute, Adv will be willing to keep trying as network control center NCC_s utilizes the private key for authenticating all the users, which is a crucial parameter to the whole system. Therefore, the protocol proposed by Altaf et al. is vulnerable to the above attack. The same attack can be also implemented in Sharif et al.'s scheme [21].

3.2.2. Inability to Deal with the Clock Asynchronous Situation. In Altaf et al.'s scheme, the validity of the message $\langle PID_i, Auth_i, Q_i, T_1, ID_{LEOS_q}\rangle$ from U_i is first verified by checking the condition $(T_1 - T_s) \leq ?\Delta T$, where T_1 is the

Input: X_{Adv}, ID_{Adv} and $h(\cdot)$.
Output: msk , which is the private key only known to NCC_s.
(1) Adv generates a random number and takes it as key msk_{tmp}
(2) AdvComputes $X_{\text{tmp}} = h(\text{ID}_{\text{Adv}} \| \text{msk}_{\text{tmp}})$
(3) **If** $X_{tmp} == X_{Adv}$ **then**
Return (msk_{tmp})
else
Go to 1 until correct key is obtained

Algorithm 1: Offline password-guessing attack.

timestamp transmitted from U_i , T_s is the current timestamp of CNN_s, and ΔT is an estimated time delay by the system. As we all know, the propagation delay is usually large between the satellite and the ground. Even for low-Earth-orbit (LEO) satellites, there is a propagation delay of 10 to 40 milliseconds due to the transmission distance of 500 to 2,000 kilometers [2]. What is more, since the satellite only forwards the authentication message, the protocol at least needs two signal transmission delays between U_i and CNN_s (the uplink between mobile user and satellite and the downlink between satellite and ground station, regardless of transmission delay between gateway and NCC_i, based on Figure 1), which will result in unacceptable access delay. Thus, it is very difficult for the entire system to estimate a uniform time delay, which may cause widespread denial of service for users.

Apart from that, users in some professional fields, such as scientists who perform north-south pole expeditions, may not be able to synchronize time with their mobile terminals due to the inability to communicate with synchronous satellites or other reasons in the actual environment. Obviously, Altaf et al.'s scheme is unable to deal with this clock asynchronous situation, while SIN covers all corners of the Earth and even deep space. Moreover, none of the existing schemes [19–21] take this situation into consideration.

4. Our Protocol

We introduce a novel authentication and key agreement protocol against clock asynchronous for SIN in this section. The proposed scheme mainly utilizes elliptic curve cryptosystem (ECC) and identity-based cryptography (IBC) to achieve sufficient security and authentication. There are 4 phases in our enhanced protocol: (1) initialization phase, (2) registration phase, (3) login and authentication phase, and (4) password-change phase. The detailed implementation of the middle two stages is shown in Figure 4.

4.1. Initialization Phase. Since our scheme is based on ECC, this phase is different from prerelevant schemes which can be divided into four steps as follows:

Step 1: NCC_s chooses a secure elliptic curve equation $E_p(a, b)$ and a generator point P of the cyclic additive elliptic curve group G_p with order n , where p is an x-bit prime number.

Step 2: NCC_s selects a random number $k \in [1, n-1]$ as its private key and computes the corresponding public key $K = kP$.

Step 3: NCC_s picks a one-way key derivation function kdf: $(0,1)^j \longrightarrow (0,1)^m$, which is mainly used to generate shared session password.

Step 4: NCC_s publishes $\{E_p(a, b), n, P, K, h(\cdot), \text{kdf}\}$ as the system parameters and keeps its master key k secret.

4.2. Registration Phase. If a mobile user U_i wants to register to the system, this phase is performed only once as follows:

Step 1: U_i freely chooses a valid identity ID_i and password P_i to enter into his/her mobile device, such as a smartphone. The identity ID_i can be combined by any one of U_i's name, e-mail address, social security number, or other identity attributes as his/her public key for a unique signature. Next, the device collects U_i's biometric B_i to compute $v_i = h(\text{ID}_i \| P_i \| B_i)$. Then, the device sends the message $\langle \text{ID}_i, v_i \rangle$ to NCC_s through a secure channel.

Step 2: After receiving the message, the server first calculates $h(\text{ID}_i)$ to check whether ID_i has been registered in the verifier table. If it has been registered, U_i is asked to select a new identity. Otherwise, NCC_s calculates the following operations:

$$\begin{aligned} S_i &= (k + h(\text{ID}_i))^{-1}, \\ P &= (x_s, y_s), \\ l_i &= x_s \oplus v_i, \\ m_i &= h(v_i \| x_s \| \text{ID}_i). \end{aligned} \tag{1}$$

After that, NCC_s inserts $\langle N_0, h(\text{ID}_i) \rangle$ into the verifier table (VT) and delivers the message $M_0 = \langle N_0, l_i, m_i \rangle$ to U_i via the secure channel, where N_0 is a nonce.

Step 3: On getting the response message, U_i stores the values $\langle N_0, l_i, m_i \rangle$ in his/her mobile device for later use in the login process.

Remark 2. The nonce N_0 is a unique value randomly generated by NCC_s and is frequently used to avoid the replay attack. Here, we further apply it as a mechanism to combat the asynchronous clock scenario, which will be discussed at the end of Section 4.3.

Remark 3. There may be two approaches for NCC_s to deliver the message M_0 to U_i: One way is offline method where NCC_s records M_0 into a smartcard and issues it to U_i. The other way is online method where NCC_s connects to U_i through the Internet Key Exchange Protocol version 2 (IKEv2) [35] or Secure Socket Layer (SSL) Protocol [33]. The message will be encrypted for transmission.

4.3. Login and Authentication Phase. This part introduces the user login system process and the mutual authentication between U_i and the NCC_s. The detailed description is as follows:

Step 1: When U_i intends to communicate with others or get service from NCC_s via the LSC system, U_i provides ID_i^*, P_i^*, and B_i^* to his/her mobile device. The device computes

$$\begin{aligned} v_i^* &= h\left(ID_i^* \| P_i^* \| B_i^*\right), \\ x_s^* &= l_i \oplus v_i^*, \\ m_i^* &= h\left(v_i^* \| x_s^* \| \mathrm{ID}_i^*\right), \end{aligned} \tag{2}$$

and it determines whether $m_i^* = ?m_i$ or not, where m_i has been recorded in the device. If $m_i^* = m_i$, the device randomly selects $b_i \in [1, n-1]$ and calculates the operations as follows:

$$\begin{aligned} Q_i &= b_i P = \left(x_q, y_q\right), \\ V_i &= b_i K = \left(x_v, y_v\right), \\ \mathrm{PID}_i &= x_v \oplus \mathrm{ID}_i \\ \alpha_i &= h\left(\mathrm{ID}_i \| x_s \| \mathrm{PID}_i \| N_0\right). \end{aligned} \tag{3}$$

Then, the device sends $M_1 = \langle \mathrm{PID}_i, \alpha_i, Q_i, N_0 \rangle$ to $LEOS_q$.

Step 2: After getting the message, $LEOS_q$ forwards $\langle \mathrm{PID}_i, \alpha_i, Q_i, N_0, \mathrm{ID}_{\mathrm{LEOS}_q} \rangle$ to NCC_s.

Step 3: When obtaining the message, NCC_s calculates

$$\begin{aligned} V_i^* &= kQ_i = \left(x_v^*, y_v^*\right), \\ \mathrm{ID}_i^* &= x_v^* \oplus \mathrm{PID}_i. \end{aligned} \tag{4}$$

Next, NCC_s uses N_0 to find the matching $h(\mathrm{ID}_i)$ in the verifier table. If it is not found, NCC_s refuses the request of U_i; otherwise, it computes

$$\begin{aligned} S_i^* &= \left(k + h\left(\mathrm{ID}_i^*\right)\right)^{-1}, \\ P &= \left(x_s^*, y_s^*\right), \\ \alpha_i^* &= h\left(\mathrm{ID}_i \| x_s^* \| \mathrm{PID}_i \| N_0\right), \end{aligned} \tag{5}$$

and it checks $\alpha_i^* = ?\alpha_i$. If $\alpha_i^* = ?\alpha_i$, NCC_s authenticates *CS* and generates a nonce N_1 to compute the operations

$$\begin{aligned} \mathrm{SK}_s &= \mathrm{kdf}\left(\mathrm{ID}_i \| x_s \| N_0 N_1\right), \\ \beta_s &= h\left(\mathrm{ID}_i \| \mathrm{SK}_s \| x_s \| N_1\right). \end{aligned} \tag{6}$$

Then, NCC_s records nonce N_1 next to N_0 in the verifier table and sends $\langle \beta_s, N_1, \mathrm{ID}_{\mathrm{LEOS}_q} \rangle$ to $LEOS_q$.

Step 4: After getting the message, $LEOS_q$ forwards $M_2 = \langle \beta_s, N_1 \rangle$ to U_i.

Step 5: On receiving the reply message from $LEOS_q$, U_i computes

$$\begin{aligned} \mathrm{SK}_i^* &= \mathrm{kdf}\left(\mathrm{ID}_i \| x_s \| N_0 \| N_1\right), \\ \beta_s^* &= h\left(\mathrm{ID}_i \| \mathrm{SK}_i^* \| x_s \| N_1\right). \end{aligned} \tag{7}$$

Then, U_i checks $\beta_s^* = ?\beta_s$. If $\beta_s^* = \beta_s$, U_i confirms that NCC_s are authentic and updates N_0 as N_1 in the mobile device. As a result, U_i and NCC_s realize mutual authentication and negotiate a shared secret key:

$$\mathrm{SK} = \mathrm{kdf}\left(\mathrm{ID}_i \| x_s \| N_0 \| N_1\right). \tag{8}$$

Remark 4. We briefly derive the consistency operations in this phase as follows:

$$V_i^* = kQ_i = k\left(b_i P\right) = b_i\left(kP\right) = b_i K = V_i. \tag{9}$$

Remark 5. In Step 3, NCC_s records nonce N_1 next to N_0 rather than updating N_1 as N_0. This means that NCC_s keeps N_0 until receiving U_i's next login message including the value N_1. At this time, NCC_s will produce a new nonce N_2 and then will update N_1 to N_0 and N_2 to N_1. In a word, NCC_s always keep two fresh numbers related to U_i in the verifier table except the first login. The designed mechanism mainly fights against the denial of service attack due to the nonce between U_i and NCC_s being out of sync. We call this scenario "desynchronization challenge" as shown in Figure 5. "Successful authentication" means that U_i has authenticated NCC_s and updated N_0 to N_1. Then, U_i can apply the shared secret key *SK* to encrypt the next traffic with NCC_s, namely, "Data Exchange Phase." The failure indicates U_i cannot authenticate NCC_s because the message M_3 was tampered with by an attacker or interfered with the poor wireless environment. Then, "desynchronization challenge" is invoked. Since NCC_s still holds the fresh number N_0, U_i can continue to send login request information with N_0.

4.4. Password-Change Phase. Whenever U_i wants to update his/her password P_i to a new P_i^{new}, this phase is activated without communication with NCC_s. Firstly, U_i provides ID_i^*, P_i^*, and B_i^* to his/her mobile device and asks for changing the password. Then, the device will automatically perform as follows:

Step 1: The device computes $v_i^* = h(\mathrm{ID}_i^* \| P_i^* \| B_i^*)$, $x_s^* = l_i \oplus v_i^*$, and $m_i^* = h(v_i^* \| x_i^* \| \mathrm{ID}_i^*)$. It checks $m_i^* = ?m_i$. If $m_i^* \neq m_i$, it rejects U_i's password change. Otherwise, the device prompts U_i for a new password P_i^{new}.

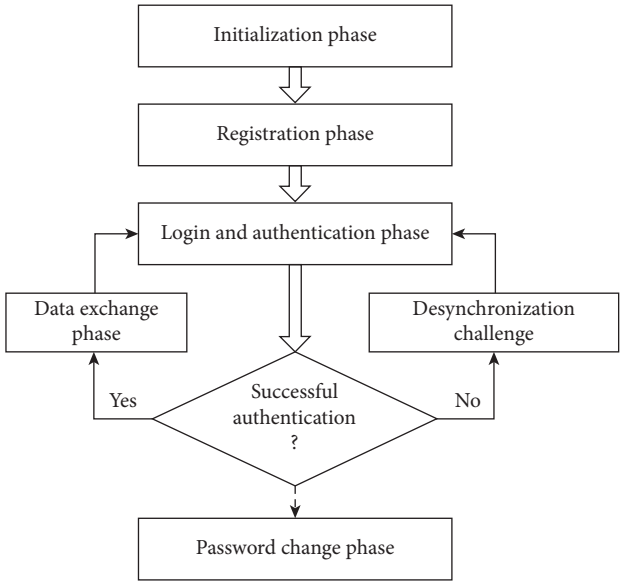


Figure 5: Flow diagram of the new protocol.

Step 2: After U_i enters P_i^{new}, the device calculates $v_i^{\text{new}} = h(\text{ID}_i^* \| P_i^{\text{new}} \| B_i^*)$, $l_i^{\text{new}} = x_s \oplus v_i^{\text{new}}$ and $m_i^{\text{new}} = h(v_i^{\text{new}} \| x_s^* \| \text{ID}_i^*)$. Finally, the device replaces the original $\langle l_i, m_i \rangle$ with $\langle l_i^{\text{new}}, m_i^{\text{new}} \rangle$ in the memory separately.

5. Results and Discussion

This section discusses security analysis of our protocol. Firstly, we give the overall design ideas of the proposed protocol. Then, the two security protocol analysis tools of Scyther and AVISPA simulate the implementation of our scheme. We further analyze how the proposed scheme can fulfill the five essential properties. Finally, the performance comparisons of our protocol with others are described briefly.

5.1. Overall Design Idea of Our Scheme. In our protocol, we utilize two symmetric keys between U_i and NCC_s. The first symmetric secret key is x_s, which is established by calculating $S_i = (k + h(\text{ID}_i))^{-1} P = (x_s, y_s)$ during the user registration process. In addition to the authentication of both parties, x_s is also used to verify the integrity of the messages M_0, M_1, and M_2 and generate the shared secret key SK, which is used to encrypt information in the data exchange phase. If Adv wants to obtain x_s, it can only be obtained through brute-force calculation, because x_s is protected by U_i's real identity ID_i and NCC_s's master key k. However, according to 4th difficult calculation problem K-CAA1 in Section 2.3, it is impossible for Adv to launch brute-force calculation to get this value, such as the offline password-guessing attack used in our attack on Altaf et al.'s scheme.

The second symmetric cipher is x_v, which is secured by calculating ECDLP and CDHP on ECC. On the one hand, Adv obtains Q_i by intercepting the message M_1 through the public channel. According to ECDLP, Adv cannot obtain the value of b_i by calculating $Q_i = b_i P$, and it is also impossible to obtain the server's master key k by operating $K = kP$. On the other hand, if Adv obtains Q_i and K, it is also impossible to obtain V_i, because, according to $Q_i = b_i P$ and $K = kP$ to find $Q_i = b_i(kP)$, this is equivalent to calculating CDHP on ECC. The key x_v is mainly used to protect U_i's real identity ID_i and to realize the verification of the server signature by computing $V_i^* = kQ_i$.

5.2. Simulation Analysis Using Scyther and AVISPA. This section presents simulation of the proposed protocol using widely accepted security protocol analysis tools of Scyther and AVISPA. During simulating the implementation of the scheme, Scyther can detect the reachability of the message among participants and discover the attack path initiated by a pretender. The AVISPA simulation tool sets up various attack models internally to test whether the protocol is SAFE or UNSAFE. The detailed instructions of Scyther can be found in [36, 37] and those of AVISPA can be found in [38, 39], and comparison of these analysis tools can be found in [40].

5.2.1. Simulation Code Description. This section introduces the use of Scyther formal language SPDL (Security Protocol Description Language) and AVISPA formal language HLPSL (High-Level Protocol Specification Language) to model our scheme.

(1) Simulation code in Scyther SPDL: Figure 6 presents the simulation code of our protocol with the Scyther SPDL. Two hash functions and a simulated elliptic curve function (ECC) are defined at the beginning of SPDL simulation code. The ECC is modeled as public key encryption, where NCC_s has a private key k. Next, 3 roles in the scheme are defined: "role I″ simulates U_i; "role R" presents NCC_s; "role LEOS" indicates LEOS. Here, we take U_i role as an example to introduce the SPDL code, which is mainly presented on the left of Figure 6. After defining the variables required for session protocol, user-side operations are mainly represented by the collection of events. The "send" and "recv" events mean that U_i sends a message and receives one, respectively. Lines 16 to 19 indicate the event where U_i receives the message M_0 from NCC_s and checks $m_i^* = ?m_i$ during the login phase. Among them, the 16th line indicates that the symmetric secret key x_s is modeled as ECC function with parameters of NCC_s's private key k and U_i's identity ID_i; line 17 presents that U_i obtains x_s by l_i; then, U_i can receive M_0 and check $m_i^* = ?m_i$, indicated on line 18 and line 19, respectively. Apart from that, the 28th line adds the matching of the verification β_s, which ensures that the attacker cannot construct the message autonomously; the "claim" event in the 30th line is used to describe the authentication of roles and the confidentiality of variables.

```
// The protocol description in Scyther SPDL

hashfunction KDF,h;
function ECC;
protocol Authentication(I,LEOS,R)
{
 role I
 {
  const IDi,Pi,Bi:Nonce;
  var N0,N1,IDleos:Nonce;
  var li,mi,Beta:Ticket;
  fresh xv:Nonce;
  macro vi=h(IDi,Pi,Bi);
  send_1(I,R,{IDi,vi}k(I,R));

  macro xs=ECC(sk(R),h(IDi));
  macro li={xs}vi;
  recv_2(R,I,{N0,li,mi}k(I,R));
  match(mi,h(vi,xs,IDi));

  macro PIDi={IDi}xv;
  macro Q={xv}pk(R);
  macro alpha=h(IDi,xs,PIDi,N0);
  send_3(I,LEOS,PIDi,alpha,Q,N0);

  recv_6(LEOS,I,N1,Beta,IDleos);
  macro SKIR=KDF(IDi,xs,N0,N1);
  match(Beta,h(IDi,SKIR,xs,N1));

  claim(I,Secret,SKIR);

 }
```

(a)

```
 role R
 {
  var IDi,Pi,Bi,IDleos,xv:Nonce;
  var vi:Ticket;
  var PIDi,alpha,Q:Ticket;
  fresh N0,N1:Nonce;

  recv_1(I,R,{IDi,vi}k(I,R));

  macro xs=ECC(sk(R),h(IDi));
  macro li={xs}vi;
  macro mi=h(vi,xs,IDi);
  send_2(R,I,{N0,li,mi}k(I,R));

  recv_4(LEOS,R,PIDi,alpha,Q,N0,IDleos);
  match(alpha,h(IDi,xs,PIDi,N0));
  macro SKIR=KDF(IDi,xs,N0,N1);
  macro Beta=h(IDi,SKIR,xs,N1);
  send_5(R,LEOS,N1,Beta);

  claim(R,Secret,SKIR);

 }
 role LEOS
 {
  const IDleos:Nonce;
  var PIDi,alpha,Q,Beta,delta:Ticket;
  var N0,N1,IDi,Bi,Pi,xv:Nonce;
  recv_3(I,LEOS,PIDi,alpha,Q,N0);
  send_4(LEOS,R,PIDi,alpha,Q,N0,IDleos);
  recv_5(R,LEOS,N1,Beta);
  send_6(LEOS,I,N1,Beta,IDleos);
 }
}
```

(b)

Figure 6: Simulation code in Scyther SPDL.

```
protocol Sate;
identifiers
J,L,S:user;
IDi,Pi,Bi,Xv,IDleos:number;
N0,N1:number;

KDF,H,ECC:function;
Ks: public_key;
messages

1.J=>S:IDi,H(IDi,Pi,Bi)
2.S=>J:N0,{ECC(IDi,Ks')}(H(IDi,Pi,Bi)),H
(H(IDi,Pi,Bi),ECC(IDi,Ks'),IDi)
3.J->L:{ECC(IDi,Ks')}Xv,{Xv}Ks,N0
4.L->S:{ECC(IDi,Ks')}Xv,{Xv}Ks,N0,IDleos
5.S->L:N1,H(IDi,KDF(IDi,ECC(Ks',H(
IDi)),N0,N1))
6.L->J:N1,H(IDi,KDF(IDi,ECC(Ks',H(
IDi)),N0,N1)),IDleos

knowledge
J   :J,L,S,KDF,H,ECC,Ks,IDi,Pi,Bi;
L   :J,L,S,KDF,H,ECC;
S   :J,L,S,KDF,H,ECC,Ks;

session_instances

[J:useri,L:leos,S:server,H:h,KDF:kdf,ECC
:ecc,IDi:id,Pi:pi,Bi:bi];

intruder_knowledge
useri,leos,server,h,kdf,ecc,id;
```

(a)

```
role role_J(J:agent,L:agent,S:agent,KDF:hash_
func,H:hash_func,ECC:hash_func,Ks:public_key,
IDi:text,Pi:text,Bi:text,Key_set_J_S:(
symmetric_key) set,Key_set_S_J:(
symmetric_key) set,SND,RCV:channel(dy))
played_by J
def=
    local
        State:nat,Xv:text,IDleos:text,N1:text
        ,N0:text,Key_2:symmetric_key,Key_1:sy
        mmetric_key
    init
        State := 0
    transition
        1. State=0 /\ RCV(start) =|>
        State':=1 /\ Key_1':=new() /\
        Key_set_J_S':=cons(
        Key_1',Key_set_J_S) /\ SND({IDi.H(
        IDi.Pi.Bi)}_Key_1')
        2. State=1 /\ in(Key_2',Key_set_S_J)
        /\ RCV({N0'.{ECC(IDi.inv(Ks))}_H(
        IDi.Pi.Bi).H(H(IDi.Pi.Bi).ECC(IDi.inv
        (Ks)).IDi)}_Key_2') =|> State':=2 /\
        Key_set_S_J':=delete(
        Key_2',Key_set_S_J) /\ Xv':=new() /\
        SND({ECC(IDi.inv(
        Ks))}_Xv'.{Xv'}_Ks.N0')
        6. State=2 /\ RCV(N1'.H(IDi.KDF(
        IDi.ECC(inv(Ks).H(
        IDi)).N0.N1')).IDleos') =|>
        State':=3 /\ SND(H(N0.N1'.KDF(IDi.ECC
        (inv(Ks).H(IDi)).N0.N1')))/\ secret(
        KDF(ECC(inv(Ks).H(
        IDi)).N0.N1),sec1,{J,S})
end role
```

(b)

Figure 7: Simulation code of our protocol using AVISPA. (a) Simulation code in CAS+. (b) Part simulation code in HDLS.

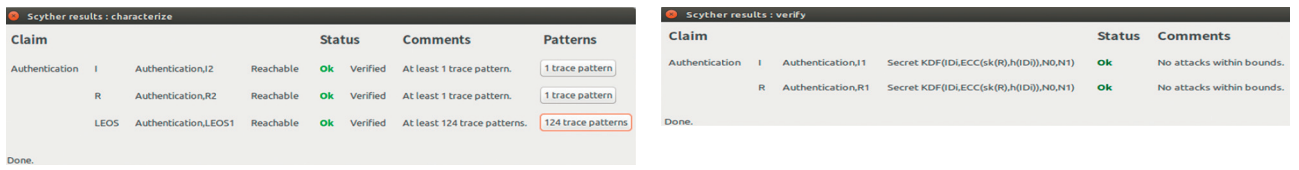
(2) Simulation code in AVISPA HLPSL: In the HLPSL modeling of our protocol, we first formalize the protocol in CAS+ specification language, as shown in Figure 7(a), and then use the SPAN (Security Protocol ANimator for AVISPA) to automatically convert the CAS+ file into the HLPSL format code in Figure 7(b). The following briefly describes the simulation CAS+ code of our scheme. After defining variables in Figure 7(a), the modeling is basically the same as the Scyther modeling, and the XOR and ECC operations are both expressed by approximate operations. Then, using the Alice-Bob message format, the protocol execution process is clear. Among them, "J", "L," and "S" present U_i, LEOS, and NCC_s, respectively; "=>" means encrypted channel, "->" means open channel, and "'" represents the inverse function; for example, "Ks'" is the private key of NCC_s, while "ks" is the public key here. In lines 19 to 21, each line represents the parameters U_i, LEOS, and NCC_s known during the protocol execution process. The 28th line defines the knowledge of intruder when attacking the security of our scheme. After generating the HLPSL format file from the CAS+ file, we manually add the verification target "secret(KDF(ECC(inv(Ks).H(IDi)).N0.N1),sec1,J,S)" in both U_i and NCC_s roles and then generate the final HLPSL format code that simulates our protocol. Since the number of HDLS language lines after conversion is relatively large, here we only give U_i role code in Figure 7(b).

5.2.2. Simulation Results. This section first presents the simulation results of our protocol using Scyther, which is shown in Figure 8. Figure 8(a) is the output report for verifying the reachability of messages among participants and Figure 8(b) presents the attack path search result of shared session secret key SK. All the analysis results prove that there is no problem in our formalization process, which means that U_i and NCC_s can securely convey the message and believe in the confidentiality of the negotiated shared session key SK in our scheme. Then, we verify whether there is an adversary attack on the protocol, that is, the vulnerability of the protocol message being obtained by the adversary. Figure 2 shows output path under the Dolev-Yao adversary threat model [41]. The analysis result indicates that, in the process of mutual authentication between U_i and NCC_s, the protocol has a LEOS impersonation attack because LEOS only forwards message and has not been authenticated in the scheme. However, due to the limitations of the nonce N_0 and the nonce N_1 and the verification message codes α_i and β_s, the attacker cannot construct the message independently and can only replay the message between U_i and NCC_s this time. Therefore, Scyther test results show that our proposed protocol does not have any threat under various active and passive attacks.

Next, we introduce the results of AVISPA analysis. The two back-end analysis results of OFMC and Atse provided by AVISPA are shown in Figure 9, which are both safe (SUMMARY SAFE). These demonstration results indicate that our protocol can achieve the expected security goals. Figure 10 shows the protocol flow chart under intruder simulation. The intruder can obtain the knowledge after the simulation attack is presented in Figure 11. From Figure 11, we can see that Adv obtains values such as N_0, N_1, and $\mathrm{ID}_{\mathrm{LEOS}_q}$ by eavesdropping messages transmitted via open channel, but there is no effective attack path. Thus, AVISPA test results also prove that our scheme can resist the various existing active and passive attacks.

5.3. Security Properties Analysis. This section describes the essential security properties of our scheme, which we summarize above for designing a robust scheme for a LSC system.

5.3.1. Valid Mutual Authentication. In our protocol, NCC_s first verifies the authenticity of ID_i's identityID_i by verifying $h(\mathrm{ID}_i) = ?h(\mathrm{ID}_i^*)$ and then verifies the validity of U_i by checking $\alpha_i^* = ?\alpha_i$. The conditions for $h(\mathrm{ID}_i) = h(\mathrm{ID}_i^*)$ and



(a) (b)

Figure 8: Simulation results of the Scyther tool. (a) Message reachability analysis report of our protocol. (b) SK confidentiality analysis report of our protocol.

(a) (b)

Figure 9: The results of OFMC and CL-Atse using AVISPA. (a) OFMC output. (b) CL-Atse output.

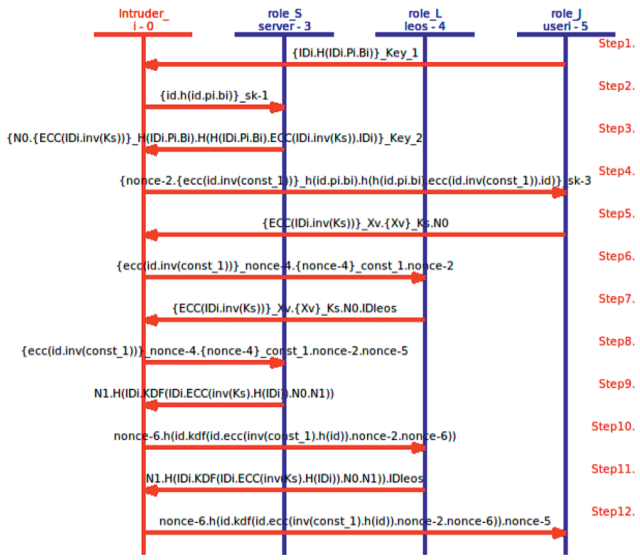


Figure 10: Flow chart under intruder simulation with AVISPA.

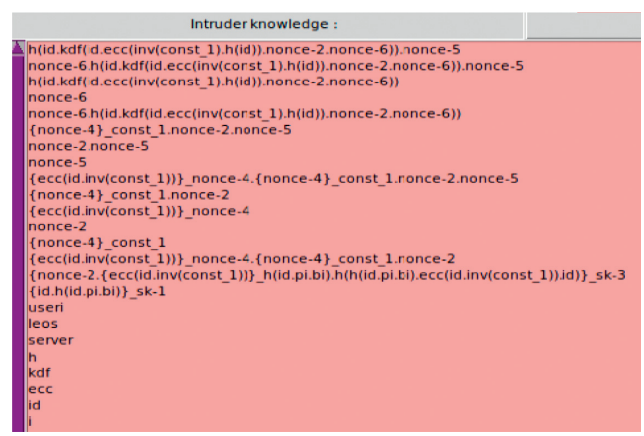


Figure 11: Intruder knowledge after simulation attack with AVISPA.

$\alpha_i^* = \alpha_i$ are all guaranteed by the symmetric keys x_v and x_s, where only real U_i can calculate these two secret values as analyzed in the previous section. So, NCC_s can effectively authenticate U_i. Simultaneously, U_i authenticates NCC_s by verifying $\beta_s^* = ?\beta_s$, which directly involves U_i's real identity ID_i and the symmetric keys x_s. ID_i and x_s are only calculated by knowing NCC_s's master key k. Therefore, our scheme provides valid mutual authentication to avoid the impersonation attack.

5.3.2. Data Confidentiality and Integrity. The proposed scheme needs to protect three types of data: the random number b_i selected by U_i, the identity ID_i of U_i, and the shared session key SK. b_i is protected by ECDLP as discussed in 5.1; ID_i is encrypted and transmitted by the symmetric

secret key x_v through the operation $\mathrm{PID}_i = x_v \oplus \mathrm{ID}_i$; SK is bound with ID_i and the symmetric key x_s in one-way key derivation operation $\mathrm{SK} = \mathrm{kdf}(\mathrm{ID}_i \| x_S \| N_0 \| N_1)$. For data integrity, it refers to the ability to quickly discover whether messages M_0, M_1, and M_2 are inserted, replaced, and deleted. M_0 stored in U_i's mobile equipment is verified by the condition $m_i^* = ?m_i$; M_1 and M_2 are verified by checking $\alpha_i^* = ?\alpha_i$ and $\beta_s^* = ?\beta_s$, respectively. Moreover, the values m_i, α_i, and β_s are all bound to the symmetric key x_s via hash function $h(\cdot)$. Thus, the new scheme ensures the data confidentiality and integrity in all aspects.

5.3.3. No Sensitive Data Maintained by NCC_s and U_i. There are only $\langle N_0, h(\mathrm{ID}_i)\rangle$ in NCC_s's verifier tables and $\langle N_0, l_i, m_i\rangle$ stored in U_i's device in our scheme. N_0 is just a nonce and is refreshed every session. As for $h(\mathrm{ID}_i)$, Adv may obtain the user's identity ID through offline password-guessing attacks after penetrating attack to the NCC_s's inside. However, it is meaningless for the entire system as Adv cannot acquire any clues about the U_i's password and NCC_s's master key. For data stored in U_i's mobile device, l_i and m_i are only used in the beginning of U_i's login phase and neither reveals the key parameters of the system. So, none of sensitive data are maintained by NCC_s and U_i in our scheme.

5.3.4. Perfect Session Key Secrecy. The shared session key is derived from $\mathrm{SK}_i = \mathrm{kdf}(\mathrm{ID}_i \| x_S \| N_0 \| N_1)$, which is enclosed by the nonce N_0 and the nonce N_1, which are refreshed every session. Therefore, even if jth session key SK_i^j is leaked this time, it will not cause the previous SK_i^{j-1} or next session key SK_i^{j+1} to be compromised.

5.3.5. User's Privacy. The scheme mainly involves three types of personal privacy of U_i: identity ID_i, password P_i, and biometric B_i. At first, U_i registers by submitting $\langle \mathrm{ID}_i, v_i\rangle$, where $v_i = h(\mathrm{ID}_i \| P_i \| tB_i)$, in order to keep the secret password P_i and biometric B_i of U_i from NCC_s. Then, during the login and authentication phase, the pseudoidentity PID_i, which is derived from $\mathrm{PID}_i = x_v \oplus \mathrm{ID}_i$, is transmitted via public channel instead of the real ID_i. Therefore, any privacy information related to U_i is enclosed in our scheme.

5.4. Performance Comparisons. In the following, we concretely compare our protocol with the other two protocols [21, 22] in terms of resistance to security functionality and computational performance. In 2019, Sharif et al. [21] compared their proposed protocol with 6 other related protocols in detail in terms of security features and computing performance and claimed that their protocol had obvious advantages in security features. Similarly, Altaf et al. [22] pointed out that their protocol had great advantages compared with the other 4 protocols in 2020. So, we simply give a comparison with these two representative articles. Moreover, Sharif et al.'s scheme is also designed using elliptic curve cryptography, and Altaf et al.'s scheme also uses public key and secret key algorithms, which is not specified in their article.

Table 2: Security functionality comparisons.

Security functionality	Proposed	[22]	[21]
Valid mutual authentication	Yes	Yes	Yes
Data confidentiality and integrity	Yes	Yes	Yes
No sensitive tables kept by NCC and user	Yes	Yes	Yes
Perfect session key secrecy	Yes	Yes	Yes
User's privacy	Yes	Yes	Yes
Protection of the biometric	Yes	No	Yes
Resist offline password-guessing attack	Yes	No	No
Against clock asynchronous situation	Yes	No	No

Table 3: Running time of cryptographic elements using OpenSSL library [21].

Element	Time	Description
T_h	0.121 μs	256-bit hash function (16 size blocks)
T_x	<0.001 μs	The bitwise XOR operation
T_p	2199 μs	384-bit nistp384 ECC point multiplication

In Table 2, we list the 6 general security properties and 2 security attacks for designing a robust authentication protocol for SIN. In addition to the above 6 functions, the newly proposed protocol can resist other attacks, such as impersonation attack, DoS attack, man-in-the-middle attack, smart card loss attack, and replay attack. The results in Table 2 show the superiorities of our protocol in terms of resisting offline password-guessing attack and against clock asynchronous situation.

As we all know, NCC_s always has no limitation with enough powerful servers. Although the most expensive operation is the point multiplication elliptic curve in the related protocol, it takes only 46 microseconds to execute the 160-bit elliptic curve point multiplication on the Intel Core-i7 processor [42]. Therefore, we only compare the efficiencies of different operations in U_i's mobile device in Table 3, which refers to Table 11 in [21]. For the convenience of evaluating the computational cost, we assume that the public key and secret key algorithms in Altaf et al.'s scheme [22] are also elliptic curve cryptography, as mainly considering 160-bit ECC has the same security level as 1024-bit RSA or DLP in practice. In addition, it is generally accepted that XOR operation execution time can be ignored, as it consumes very little time.

Furthermore, we have considered communication cost in the last line of Table 4. We suppose that each length of parameter is roughly the same in [21]: the size of a random number/nonce to be 64 bits, a hash output to be 256 bits, an identifier/timestamp to be 32 bits, and an ECC point to be 384 bits for the communication cost as we also use this length in calculation time. In our scheme, the LEOS receives $M_1 = \langle \mathrm{PID}_i, \alpha_i, Q_i, N_0\rangle$ from U_i's login request message and sends $M_2 = \langle \beta_s, N_1\rangle$ to U_i at last. Thus, the total communication cost bits of $M_1 = \langle 32, 256, (384, 384), 64\rangle$ and $M_2 = \langle 256, 64\rangle$ are 1440 bits. Table 4 demonstrates that our protocol is more efficient than the other two protocols as it uses the least times and far less communication cost bits.

Table 4: Operations and communication cost comparison among related schemes.

	Ours	Altaf et al. [22]	Sharif et al. [21]
Registration phase	$1T_h$	$2T_h + 1T_x$	$2T_h + 2T_x$
Login and authentication phase	$5T_h + 2T_x + 2T_p$	$7T_h + 4T_x + 2T_p$	$7T_h + 5T_x + 3T_p$
Total execution time	4198.726 μs	4399.089 μs	6598.089 μs
Communication cost bits	1440	1440	2112

6. Concluding Remarks

In this article, we deeply study the authentication schemes for SIN. We disclose that Altaf et al.'s protocol has fatal security drawbacks and point out that many protocols in the literature cannot handle the clock asynchronous situation. Because SIN mainly transmits information through wireless signals and covers a wide range of services, it often happens when users' mobile devices cannot achieve clock asynchronous situation in the actual environment. To overcome these security challenges, we introduced a lightweight pseudonym identity-based authentication and key agreement scheme using ECC and IBC. To strengthen the security of our scheme, we first introduced the main ideas of the entire protocol design. Then, the security protocol analysis tools of AVISPA and Scyther are both utilized to simulate the proposed scheme and the analysis results prove that our scheme can resist the various existing attacks. We further discussed essential security properties of the proposed scheme, which meets the requirements to design a robust scheme for a LSC system. Moreover, we concretely compare our protocol with the relatsed protocols in terms of security requirements, computational performance, and computational cost. All the results show that our protocol is superior to the other two representative protocols. Actually, our protocol will be well suited for mobile users in SIN.

Data Availability

The corresponding author shall keep the analysis and full simulation code set. If necessary, the data set can be requested from the corresponding author for reasonable requirements.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Key Research and Development project (no. 2019QY0501) and the National Natural Science Foundation of China (no. 2904020211).

References

[1] M. Liu, S. Sun, and J. Wang, "Research on reliable transmission protocol for spatial information networks," *ICIC Express Letters*, vol. 7, pp. 2183–2188, 2013.

[2] W. Meng, K. Xue, J. Xu et al., "Low-latency authentication against satellite compromising for space information network," 2018.

[3] A. Z. M. Shahriar, M. Atiquzzaman, and W. Ivancic, "Network Mobility in satellite networks: architecture and the protocol," *International Journal of Communication Systems*, vol. 26, no. 2, pp. 177–197, 2013.

[4] H. S. Cruickshank, "A security system for satellite networks," 1996.

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

[6] Y.-F. Chang and C.-C. Chang, "An efficient authentication protocol for mobile satellite communication systems," *Acm Sigops Operating Systems Review*, vol. 39, no. 1, pp. 70–84, 2005.

[7] T.-H. Chen, W.-B. Lee, and H.-B. Chen, "A self-verification authentication mechanism for mobile satellite communication systems," *Computers & Electrical Engineering*, vol. 35, no. 1, pp. 41–48, 2009.

[8] I. Lasc, R. Dojen, and T. Coffey, "Countering jamming attacks against an authentication and key agreement protocol for mobile satellite communications," *Computers & Electrical Engineering*, vol. 37, no. 2, pp. 160–168, 2011.

[9] I. Lasc, "A simple and efficient authentication scheme for mobile satellite communication systems," *International Journal of Satellite Communications & Networking*, vol. 30, no. 1, pp. 29–38, 2011.

[10] R. A. Abouhogail, "Verification of authentication protocols for mobile satellite communication systems," *The Egyptian Journal of Remote Sensing and Space Science*, vol. 17, no. 2, pp. 171–177, 2014.

[11] C. C. Chang, T. F. Cheng, and H. L. Wu, "An authentication and key agreement protocol for satellite communications," *International Journal of Communication Systems*, vol. 27, no. 10, pp. 1994–2006, 2015.

[12] X. Li, J. Niu, M. Khurram Khan, and J. Liao, "An enhanced smart card based remote user password authentication scheme," *Journal of Network and Computer Applications*, vol. 36, no. 5, pp. 1365–1371, 2013.

[13] C.-L. Chen, K.-W. Cheng, Y.-L. Chen, I.-C. Chang, and C.-C. Lee, "An improvement on the self-verification authentication mechanism for A mobile satellite communication system," *Applied Mathematics & Information Sciences*, vol. 8, no. 1L, pp. 97–106, 2014.

[14] Y. Zhang, J. Chen, and B. Huang, "Security analysis of an authentication and key agreement protocol for satellite communications," *International Journal of Communication Systems*, vol. 27, no. 12, pp. 4300–4306, 2015.

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

[16] H.-Y. Lin, "Efficient dynamic authentication for mobile satellite communication systems without verification table," *International Journal of Satellite Communications and Networking*, vol. 34, no. 1, pp. 3–10, 2016.

[17] Y. Zhang, J. Chen, and B. Huang, "An improved authentication scheme for mobile satellite communication systems,"

International Journal of Satellite Communications and Networking, vol. 33, no. 2, pp. 135–146, 2015.

[18] L. Yan, Y. Chang, and S. Zhang, "Comments on "An improved authentication scheme for mobile satellite communication systems"" *International Journal of Electronic Security and Digital Forensics*, vol. 9, no. 4, p. 396, 2017.

[19] M. Qi and J. Chen, "An enhanced authentication with key agreement scheme for satellite communication systems," *International Journal of Satellite Communications and Networking*, vol. 36, no. 3, pp. 296–304, 2017.

[20] S. Xu, X. Liu, M. Ma et al., "An improved mutual authentication protocol based on perfect forward secrecy for satellite communications," *International Journal of Satellite Communications and Networking*, vol. 38, no. 1, 2020.

[21] A. Ostad-Sharif, D. Abbasinezhad-Mood, and M. Nikooghadam, "Efficient utilization of elliptic curve cryptography in design of a three-factor authentication protocol for satellite communications," *Computer Communications*, vol. 38, p. 147, 2019.

[22] I. Altaf, M. A. Saleem, K. Mahmood et al., "A lightweight key agreement and authentication scheme for satellite-communication systems," *IEEE Access*, vol. 38, 2020.

[23] D. Yiltas and A. H. Zaim, "Evaluation of call blocking probabilities in LEO satellite networks," *International Journal of Satellite Communications and Networking*, vol. 34, 2009.

[24] E. Rendon-Morales, J. Mata-Díaz, J. Alins, J. L. Muñoz, and O. Esparza, "Performance evaluation of selected Transmission Control Protocol variants over a digital video broadcasting-second generation broadband satellite multimedia system with QoS," *International Journal of Communication Systems*, vol. 26, no. 12, pp. 1579–1598, 2013.

[25] S. Al Khanjari, B. Arafeh, K. Day et al., "Bandwidth borrowing-based QoS approach for adaptive call admission control in multiclass traffic wireless cellular networks," *International Journal of Communication Systems*, vol. 25, 2013.

[26] V. S. Miller, "Use of elliptic curves in cryptography," 1986.

[27] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, p. 203, 1987.

[28] D. Hankerson, A. J. Menezes, and S. A. Vanstone, "Guide to elliptic curve cryptography," 2004.

[29] S. H. Islam and G. P. Biswas, "A more efficient and secure ID-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem," *Journal of Systems and Software*, vol. 84, no. 11, pp. 1892–1898, 2011.

[30] H. Debiao, C. Jianhua, and H. Jin, "An ID-based client authentication with key agreement protocol for mobile client-server environment on ECC with provable security," *Information Fusion*, vol. 13, no. 3, pp. 223–230, 2012.

[31] Z. Li, J. Kang, R. Yu, D. Ye, Q. Deng, and Y. Zhang, "Consortium blockchain for secure energy trading in industrial Internet of things," *IEEE Transactions on Industrial Informatics*, vol. 12, 2017.

[32] S. A. Chaudhry, H. Naqvi, K. Mahmood et al., "An improved remote user authentication scheme using elliptic curve cryptography," *Wireless Personal Communications*, vol. 22, 2016.

[33] C. Heinrich, "Transport layer security (TLS)," *RFC*, vol. 31, no. 4, p. 2009, 2005.

[34] C.-G. Ma, D. Wang, and S.-D. Zhao, "Security flaws in two improved remote user authentication schemes using smart cards," *International Journal of Communication Systems*, vol. 27, no. 10, pp. 2215–2227, 2014.

[35] C. Kaufman, "Internet key exchange (IKEv2) protocol," *RFC*, vol. 43, 2005.

[36] Scyther Team, "Scyther tool," 2014, https://people.cispa.io/cas.cremers/scyther/index.html.

[37] C. J. F. Cremers, *The Scyther Tool: Verification, Falsification, and Analysis of Security Protocols*, Springer, Berlin, Heidelberg, 2008.

[38] AVISPA Team, "AVISPA tool," 2014, http://www.avispa-project.org.

[39] L. Viganò, "Automated security protocol analysis with the AVISPA tool," *Electronic Notes in Theoretical Computer Science*, vol. 155, 2006.

[40] C. J. F. Cremers, P. Lafourcade, and P. Nadeau, "Comparing state spaces in automatic security protocol analysis," 2009.

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

[42] A. M. Dariush and N. Morteza, "Efficient design of a novel ECC-based public key scheme for medical data protection by utilization of NanoPi fire," *IEEE Transactions on Reliability*, vol. 15, 2018.