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 Wei1

1State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China
2Henan 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

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.

1.1. Historical Survey

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 launching 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
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.

Figure 1: A low Earth orbit satellite communications system.
The rest of this paper is organized as follows. Section 2 provides necessary techniques used in this paper. In Section 3, we review Altfa 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 $E_p(a, b)$ over $F_p$ is defined by an equation $y^2 = x^3 + ax + b (mod p)$, $a, b \in F_p$, and with the discriminant $\Delta = 4a^3 + 27b^2 \neq 0 (mod p)$. Then, the set $G_p = \{(x, y) | x, y \in F_p$ 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 $n \geq 1$. The point multiplication on group $G_p$ is defined as $kP = P + P + \cdots + P (k \text{ times})$. 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 $b = g^a (mod p)$ 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 $P \in G_p$, $x, h_i \in Z_p^*$ 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. Altfa et al.’s Scheme and Its Weaknesses

This section reviews Altfa 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$_s$). The notations used in this paper are defined in Table 1.

3.1. Altfa et al.’s Protocol. Altfa 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 $Y_i = h,ID_i||n_i$. After that, $U_i$ sends the registration message $\langle ID_i, Y_i \rangle$ to NCC$_s$ via a secure channel.

Step 2: After receiving request message from $U_i$, NCC$_s$ computes the following operations: $X_i = h,ID_i||msk$, $Y_i = X_i \oplus P_i$, and $M_i = h(P_i||X_i||ID_i)$. 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 $E_i = n_i \oplus h(ID_i||P_i)$ 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 $n_i^* = E_i \oplus h(ID_i^*||P_i^*)$, $P_i^* = h(ID_i^*||P_i^*||n_i^*)$, $X_i^* = Y_i \oplus P_i^*$, and $M_i^* = h(P_i^*||X_i^*||ID_i^*)$. Then, SC$_i$
verifies the condition $M_i^* \neq M_i$. If $M_i^* \neq M_i$, SC rejects this login of $U_i$; otherwise, SC randomly generates a number $b_i$ to calculate $Q_i = b_i P_i$, $V_i = b_i \cdot mpk$, $PID_i = V_i \oplus ID_i$, and $Auth_i = h(ID_i \| B_i \| PID_i \| T_1)$. Afterwards, SC sends $\langle PID_i, Auth_i, Q_i, T_1 \rangle$ to NCC.

Step 2: On receiving the login message from $U_i$, LEOS forwards the message $\langle PID_i, Auth_i, Q_i, T_1 \rangle$ to NCC.

Step 3: When getting the message, NCC checks the freshness of timestamp by verifying the condition $(T_1 - T_s) \leq \Delta T$. If $\Delta T$ is not permissible to the NCC, this session is terminated; otherwise, NCC computes $V_i^* = msk_i Q_i$, $ID_i = V_i^* \oplus PID_i$, and $B_i = h(ID_i \| B_i \| PID_i \| T_1)$ and checks $Auth_i^* = ?Auth_i$. If $Auth_i^* = Auth_i$, NCC thinks that $U_i$ is real and generates a random number $n_s$. Next, it calculates $W_s = n_s \oplus h(ID_i \| T_2)$, $SK = h(ID_i \| B_i \| X_i \| n_s \| T_2)$, and $Auth_s = h(ID_i \| SK \| X_i \| n_s \| V_i^* \| T_2)$ to send $\langle W_s, Auth_s, T_2, ID_{LEOS} \rangle$ to LEOS$_{q}$ finally.

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: $(0,1)^l \rightarrow (0,1)^n$ |
| $\oplus$ | Bitwise XOR operation                           |
| $\|$   | Concatenate operation                           |
Mathematical Problems in Engineering

Step 4: Upon receiving response message, LEOS$_q$ forwards $⟨W_s, \text{Auth}_s, T_2⟩$ to $U_i$.

Step 5: After receiving the message, $U_i$ checks the freshness of timestamp by verifying $(T_2 - T_s) \leq \Delta T$. If $(T_2 - T_s) \leq \Delta T$, $U_i$ authenticates NCC$_q$ and computes $n_i = W_s \oplus h(\text{ID}_i \parallel T_2)$, $\text{SK}* = h(\text{ID}_i \parallel \text{mSk} \parallel n_s \parallel T_2)$, and $\text{Auth}^*_i = h(\text{ID}_i \parallel \text{SK*} \parallel X_i \parallel n_i \parallel V_i \parallel T_2)$. If $\text{Auth}^*_i = \text{Auth}_i$, $U_i$ and NCC$_q$ 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 $\overline{P_i} = h(\text{ID}_i \parallel P_i \parallel n_s)$ and the description in Step AP1 of $\overline{P_i} = h(\text{ID}_i \parallel P_i \parallel n_i)$. The second is that NCC$_q$ should append LEOS$_q$’s identity information when sending this message $⟨W_s, \text{Auth}_s, T_2, \text{ID}_{\text{LEOS}^q}⟩$ 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$_q$ 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 $\text{Adv}$ has the following abilities when attacking the efficiency and security of their scheme:

1. $\text{Adv}$ can access the full public communication channel to modify, replay, amend, and intercept the confidential information.
2. $\text{Adv}$ can extract the secret information stored in user’s smart card with the help of power analysis.
3. $\text{Adv}$ can cheat the user by making the legitimate identity to $U_i$, because LEOS$_q$ only needs to have identity confirmation with NCC$_q$ in the LSC system.
3.2.1. Offline Password-Guessing Attack. This section describes how a malicious attacker can obtain the private key msk of NCC, 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 $P_{Adv} = h(ID_{Adv} \parallel P_{Adv})^{n_{Adv}}$ with arbitrary selection of identity ID$_{Adv}$, password $P_{Adv}$, and a random number $n_{Adv}$. Then, Adv records the values $\langle P_{Adv}, ID_{Adv} \rangle$ and sends $\langle ID_{Adv}, P_{Adv} \rangle$ to NCC. 

Step 2: After receiving the login message from Adv, NCC computes $X_{Adv} = h(ID_{Adv} \parallel \text{msk})$, $Y_{Adv} = X_{Adv} \oplus P_{Adv}$, and $M_{Adv} = h(P_{Adv} \parallel X_{Adv} \parallel ID_{Adv})$. Then, it will issue a smart card SC$_{Adv}$ recording 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 $X_{Adv} = Y_{Adv} \oplus P_{Adv}$ and records the value $X_{Adv}$. 

Step 4: Since $X_{Adv} = h(ID_{Adv} \parallel \text{msk})$, 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 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 $\langle \text{PID}, \text{Auth}, Q, T, \text{ID}_{LEOS} \rangle$ from $U_i$ is first verified by checking the condition $(T - T_s) \leq \Delta T$, where $T_1$ is the
timestamp transmitted from \( U_i \), \( T_i \) is the current timestamp of CNN\(_i\), and \( \Delta T \) 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 delays between gateway and NCC, because of 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 cryptography (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\(_i\) 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\(_i\) selects a random number \( k \in [1, n - 1] \) as its private key and computes the corresponding public key \( K = kP \).

Step 3: NCC\(_i\) picks a one-way key derivation function \( kdf: (0, 1)^l \rightarrow (0, 1)^m \), which is mainly used to generate shared session password.

Step 4: NCC\(_i\) publishes \( \{ E_p(a, b), n, P, K, h(\cdot), kdf \} \) 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 \( v_i = h(\text{ID}_i \parallel P_i \parallel B_i) \). Then, the device sends the message \( \langle \text{ID}_i, v_i \rangle \) to NCC\(_i\) 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\(_i\) calculates the following operations:

\[
\begin{align*}
S_i &= (k + h(\text{ID}_i))^{-1}, \\
P &= (x_r, y_r), \\
l_i &= x_i \oplus v_i, \\
m_i &= h(v_i \| x_i \| \text{ID}_i).
\end{align*}
\]

After that, NCC\(_i\) 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\(_i\) 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

$$v_i^* = h(ID_i^+ || P_i || B_i^+),$$

$$x_i^* = l_i \oplus v_i^*,$$

$$m_i^* = h(v_i^* || x_i^* || ID_i^+),$$

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:

$$Q_i = b_iP = (x_i, y_i),$$

$$V_i = b_iK = (x_i, y_i),$$

$$PID_i = x_i \oplus ID_i,$$

$$a_i = h(ID_i || x_i || PID_i || N_0).$$

Then, the device sends $M_1 = \langle PID_i, a_i, Q_i, N_0 \rangle$ to LEOS$_q$.

Step 2: After getting the message, LEOS$_q$ forwards $\langle PID_i, a_i, Q_i, N_0, ID_{LEOS_q} \rangle$ to NCC$_s$.

Step 3: When obtaining the message, NCC$_s$ calculates

$$V_i^* = kQ_i = (x_i^*, y_i^*),$$

$$ID_i^* = x_i^* \oplus PID_i.$$  

Next, NCC$_s$ uses $N_0$ to find the matching $h(ID_i)$ in the verifier table. If it is not found, NCC$_s$ refuses the request of $U_i$; otherwise, it computes

$$S_i^* = (k + h(ID_i))^{-1},$$

$$P = (x_i^*, y_i^*),$$

$$a_i^* = h(ID_i || x_i^* || PID_i || N_0),$$

and it checks $a_i^* = ?a_i$. If $a_i^* = ?a_i$, NCC$_s$ authenticates CS and generates a nonce $N_1$ to compute the operations

$$SK = kdf(ID_i || x_i || N_0 || N_1),$$

$$\beta_i = h(ID_i || SK || x_i || N_1).$$

Then, NCC$_s$ records nonce $N_1$ next to $N_0$ in the verifier table and sends $\langle \beta_i, N_1, ID_{LEOS_q} \rangle$ to LEOS$_q$.

Step 4: After getting the message, LEOS$_q$ forwards $M_2 = \langle \beta_i, N_1 \rangle$ to $U_i$.

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

$$SK_i^* = kdf(ID_i || x_i || N_0 || N_1),$$

$$\beta_i^* = h(ID_i || SK_i^* || x_i || N_1).$$

Then, $U_i$ checks $\beta_i^* = ?\beta_i$. If $\beta_i^* = \beta_i$, $U_i$ confirms that NCC$_s$ is 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:

$$SK = kdf(ID_i || x_i || N_0 || N_1).$$

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

$$V_i^* = kQ_i = k(b_iP) = b_i(kP) = b_iK = V_i.$$

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 $M_1$ 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 $v_i^* = h(ID_i^+ || P_i || B_i^+)$, $x_i^* = l_i \oplus v_i^*$, and $m_i^* = h(v_i^* || x_i^* || ID_i^+)$. 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 $P_i^{new}$. 
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\(_i\). The first symmetric secret key is \( x_i \), which is established by calculating \( S_i = (k + h(ID_i))^{-1}P = (x_i, y_i) \) during the user registration process. In addition to the authentication of both parties, \( x_i \) is also used to verify the integrity of the messages \( M_{0i}, M_{1i}, \) 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 \( x_i \), it can only be obtained through brute-force calculation, because \( x_i \) is protected by \( U_i \)’s real identity ID\(_i\) and NCC\(_i\)’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 \( x_s \), which is secured by calculating ECDLP and CDHP on ECC. On the one hand, Adv obtains \( Q_s \) by intercepting the message \( M_1 \) through the public channel. According to ECDLP, Adv cannot obtain the value of \( b \) by calculating \( Q_i = bP \), and it is also impossible to obtain the server’s master key \( k \) by operating \( K = kP \). On the other hand, if Adv obtains \( Q_s \) and \( K_i \), it is also impossible to obtain \( V_s \), because, according to \( Q_s = bP \) and \( K = kP \) to find \( Q_s = b \cdot (kP) \), this is equivalent to calculating CDHP on ECC. The key \( x_s \) 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\(_i\) has a private key \( k \). Next, 3 roles in the scheme are defined: “role I” simulates \( U_i \); “role R” presents NCC\(_i\); “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\(_i\) and checks \( m_1^* = ?m_1 \) 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\(_i\)’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 \( m_1^* = ?m_1 \), indicated on line 18 and line 19, respectively. Apart from that, the 28th line adds the matching of the verification \( \beta_i \), 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.
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; “$$" means encrypted channel, "-" means open channel, and "\~" represents the inverse function; for example, "Ks\~" is the private key of $NCC_s$ while "ks" is the public key here. In lines 19 to 21, each line represents the parameters $U_i$, LEOS, and $NCC_s$ known during the protocol execution process. The 28th line defines the knowledge of intruder when attacking the security of our scheme. After generating the HLPSL format file from the CAS+ file, we manually add the verification target “$secret(KD-F(ECC(inv(Ks).H(ID_i)).N0.N1),sec1,J,S)$$” 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 $\alpha_i$ and $\beta_s$, 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 ID $LEOS$, 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 identity $ID_i$ by verifying $h(ID_i) = h(ID^*_i)$ and then verifies the validity of $U_i$ by checking $a^*_i = \alpha_i$. The conditions for $h(ID_i) = h(ID^*_i)$ and
\(\alpha_i = \alpha_i\) are all guaranteed by the symmetric keys \(x_i\) 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 \(x_i\), 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

---

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

**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.
secret key $x_v$ through the operation $PID_i = x_v\oplus ID_i$; SK is bound with ID_i and the symmetric key $x_v$ in one-way key derivation operation $SK = kdf(ID_i, x_v, N_0, N_1)$. 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 $m_0^* = ?m_i$; $M_1$ and $M_2$ are verified by checking $a_i^* = ?a_i$ and $b_i^* = ?b_i$, respectively. Moreover, the values $m_i$, $a_i$, and $b_i$ are all bound to the symmetric key $x_v$ 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, and $U_i$. There are only $\langle N_0, h(ID_i) \rangle$ in NCC,’s verifier tables and $\langle N_0, I_0, m_0 \rangle$ stored in $U_i$’s device in our scheme. $N_0$ is just a nonce and is refreshed every session. As for $h(ID_i)$, Adv may obtain the user’s identity ID_i through offline password-guessing attacks after penetrating attack to the NCC,’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 master key. For data stored in $U_i$’s mobile device, $I_0$ and $m_0$ 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, and $U_i$ in our scheme.

5.3.4. Perfect Session Key Secrecy. The shared session key is derived from $SK_i = kdf(ID_i, x_v, N_0, N_1)$, which is enclosed by the nonce $N_0$ and the nonce $N_1$, which are refreshed every session. Therefore, even if jth session key $SK_j$ is leaked this time, it will not cause the previous $SK_j^{-1}$ or next session key $SK_{j+1}$ 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 $\langle ID_i, v_i \rangle$, where $v_i = h(ID_i, P_i, \| TB_i)$, in order to keep the secret password $P_i$ and biometric $B_i$ of $U_i$ from NCC. Then, during the login and authentication phase, the pseudo-identity $PID_i$ which is derived from $PID_i = x_v\oplus ID_i$, 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.

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 asynchrony situation.

As we all know, NCC 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/nounce 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 LEO8 receives $M_1 = \langle PID_i, a_i, Q_i, N_0 \rangle$ from $U_i$’s login request message and sends $M_2 = \langle \beta_i, N_1 \rangle$ 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 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. |
|----------------------|----------|-----------------------------|
| 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 |
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 related 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 Networks*, 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 Networks*, 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,”
[18] L. Yan, Y. Chang, and S. Zhang, "Comments on "An improved authentication scheme for mobile satellite communications 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 multicell 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 P. 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.