An efficient and provably secure key agreement scheme for satellite communication systems

Yuanyuan Zhang*, Zhibo Zhai
School of Computer Science, Hubei University of Technology, Wuhan, China
* circle0519@hotmail.com

Abstract

Satellite communication has played an important part in many different industries because of its advantages of wide coverage, strong disaster tolerance and high flexibility. The security of satellite communication systems has always been the concern of many scholars. Without authentication, user should not obtain his/her required services. Beyond that, the anonymity also needs to be protected during communications. In this study, we design an efficient and provably secure key agreement scheme for satellite communication systems. In each session, we replace user’s true identity by a temporary identity, which will be updated for each session, to guarantee the anonymity. Because the only use of lightweight algorithms, our proposed scheme has high performance. Furthermore, the security of the proposed scheme is proved in the real-or-random model and the performance analysis shows that the proposed scheme is more efficient than some other schemes for satellite communication systems.

Introduction

With the development of communication technology, satellite communication has played an important part in many industries, such as economy, politics, culture, military affairs, etc [1–3]. Compared with traditional satellite communication systems, the low-earth-orbit (LEO) satellite communication systems have the advantages of shorter transmission delay, higher efficiency, higher availability and higher cost performance. As shown in Fig 1, an LEO communication system is made up of mobile devices/users (U), gateways, LEO and the network control center (NCC). The most of entities in this LEO satellite communication system are connected wirelessly, the natural defects of the wireless communication bring great security risks, which will cause the system to be easily attacked by attackers.

In order to solve these problems, some low-Earth-orbit satellite communication protocols [4–7] have been proposed. In 1996, the first security satellite communication scheme was proposed by Cruickshank [8], but it cost too much computation. In 2003, Hwang et al. proposed an authentication scheme for mobile satellite communication system [9]. However, Chang et al. pointed out that their scheme couldn’t proved perfect forward secrecy [10]. In 2009,
Chen et al. proposed a self-verification scheme and claimed there was no sensitive information stored in the verification [11]. But Lee et al. pointed out the user’s secret key is not secure in Chen et al.’s scheme when the hash value of user identity and another parameters are coprime numbers in 2011 [12]. In 2014, Zhang et al. proposed an improved authentication scheme [13]. But soon after, Qi et al. pointed out that Zhang et al.’s scheme was vulnerable to stolen-verifier attack and denial of service attack [14]. In 2017, Qi et al. proposed an enhanced authentication with key agreement scheme based on ECC(Elliptic Curves Cryptography), but the scheme has poor performance and some security defects. In this paper, we propose an enhanced scheme and prove that our scheme is secure under the real-or-random model.

In this context, an effective satellite communication scheme must possess the following characteristics to ensure the secrecy of normal operations in mobile satellite communication systems:

- Mutual authentication: NCC and user $U$ can authenticate each other and generate a session key without being illegally obtained by the attacker.
- Perfect forward secrecy: if a session key or sensitive data is leaked to an attacker, he/she cannot obtain previous session keys from the preceding interception.
- Anonymity: the user’s identity is securely hidden and an attacker cannot derive user identity in any way.
- Resistance to stolen-verifier attacks: an attacker may break into servers and steal verification table from trusted servers. However, he or she cannot use the data in the verification table to launch any attack.
- Resistance to smart card loss attacks: if an attacker gets a legal user’s smart card in some way, he or she cannot derive sensitive data from it.
- Resistance to denial of service attacks: prevent attackers from occupying server resources illegally to ensure that the legal users can access the authentication server normally.
- Resistance to impersonation attacks: an attacker cannot attempt to communicate with a trusted server as a legitimate user and also cannot attempt to communicate with a legitimate user as a server.

The rest of this paper is organized as follows: We describe the detail of our proposed scheme in Section 2 and section 3 shows the security analysis of our scheme. Section 4 compares the
security and the performance of our scheme with other related schemes. Finally, section 5 presents our conclusion.

**Our proposed scheme**

In this section, we propose an efficient and provably secure key agreement scheme for satellite communication systems. Some notions in our scheme are shown in Table 1. In the proposed scheme, we abandon the traditional temporary identity verification table, and adopt a dynamic temporary identity table [15]. Usually, the traditional temporary identity table only stores temporary identity of this time. However, when an attacker intercepts messages returned from NCC, the data in NCC’s database has been updated and the data in the smart card has not been updated, so data inconsistency occurs between database and smart card. To solve this problem, we adopt a dynamic temporary identity table which consists of hash value of user’s identity, shared hash value, dynamic temporary identity of last time and dynamic temporary identity of this time (shown in Table 2).

Our scheme contains the following phases: initialization phase, registration phase, login and authentication phase and password update phase. The details of our scheme are as follows.

**Initialization phase**

NCC chooses a large prime $x$ as long-term private key randomly and specifies a secure hash algorithm $h(\cdot)$. In the meantime, NCC creates a table in the databases. The table stores four data for each legitimate user. Two of these data are the hash values which used to authenticate the identity. The remaining two are used to store user’s dynamic ID. One $ID$ is for last time, the other one is for this time. If a user fails to authenticate with the this time $ID$, he will try to re-authenticate with the last time $ID$.

**Registration phase**

To be a legal user, he/she must submit his/her registration request to NCC (see in Fig 2).
step 1: With the permission of NCC, user U inputs ID_u and PW chosen by himself/herself. After that, user U generates a random number r_0 and computes

\[ hPW = h(PW \oplus r_0) \]
\[ hID = h(ID_u \oplus r_0). \]

Next, user U sends \(\{hID, hPW\}\) as registration request to NCC in a secure channel.

step 2: After receiving registration message, NCC generates a random number r_1 and computes

\[ P = h(hID||x) \]
\[ R = P \oplus h(hID||r_1) \]

with NCC’s private key x and registration message. Next, NCC generates two temporary identity \(T_{idb} \) \(T_{ido}\) and initializes \(T_{ido}\) to null. After that, NCC stores \(\{hID, T_{idb}, T_{ido}, hPW\}\) in the database. Then, NCC writes \(\{T_{idb}, R, r_1, h(\cdot)\}\) into a smart card.

step 3: NCC delivers the smart card to user U in a secure channel.

step 4: User U computes

\[ \text{Verify} = h(ID_u||r_0||PW||r_1) \]
\[ Z = r_0 \oplus h(PW||ID_u) \]

with the data stored in smart card. Then user U writes \(\{\text{Verify}, Z\}\) into the smart card.

### Login and authentication phase

Login and authentication phase are indispensable steps for user to get services from server. The following operations are intended to achieve the goal (see in Fig 3).

step 1: User U inserts his/her smart card into card reader and inputs ID_u and PW. Then, U computes

\[ r_0' = Z \oplus h(PW||ID_u) \]
\[ \text{Verify}' = h(ID_u||r_0'||PW||r_1) \]

and verifies whether \(\text{Verify} = \text{Verify}'\) is true. If not, the session will be broken off. Otherwise,
user $U$ generates a number $r_2$ and computes

$$
\begin{align*}
\text{hID} &= h(ID_u \oplus r_0') \\
P' &= R \oplus h(hID' || r_1) \\
Q &= P' \oplus r_2 \oplus h(PW \oplus r_0') \\
S &= h(hID' || r_2 || T_{id}) \\
\end{align*}
$$

Then, user $U$ sends $\{Q, S, T_{id}\}$ to $LEO$ in a public channel.

**step 2:** $LEO$ receives the login request from $U$ and forwards the login request $\{Q, S, T_{id}, LEO_{id}\}$ to $NCC$.

**step 3:** After receiving the login request from $LEO$, $NCC$ begins to match $T_{id}$ in the dynamic verification table (Table 2). Firstly, $NCC$ searches the column of dynamic identity of this time ($T_{id}$). If there is a value equals to $T_{id}$, $NCC$ extracts the corresponding $hID$, $hPW$ and chooses a new temporary identity $T_{idnew}$. Then $NCC$ replaces $(T_{id}, T_{idnew})$ with $\{T_{id}, T_{idnew}\}$. Else, $NCC$ keeps searching the column of dynamic identity of last time ($T_{id}$) in the verification table to see if there is a value equals to $T_{id}$. If so, $NCC$ extracts the corresponding $hID$, $hPW$, chooses a

---

**Fig 3. Login and authentication phase.**

https://doi.org/10.1371/journal.pone.0250205.g003
new temporary identity $T_{id_{new}}$ and replaces $(T_{id_{old}}, T_{id})$ with $(T_{id_{old}}, T_{id_{new}})$. If NCC cannot match $T_{id}$ in the dynamic verification table, it will reject the login request.

**step 4:** After completion of matching, NCC computes

$$\begin{align*}
P' &= h(hID|x) \\
r_2' &= Q \oplus P \oplus hPW \\
S' &= h(hID|r_2'||T_{id})
\end{align*}$$

and checks whether $S$ and $S'$ are equal. If not, NCC breaks off the session. Else, NCC generates a random number $r_3$ and computes

$$\begin{align*}
v_1 &= P \oplus r_3 \\
v_2 &= h(P || r_2' || r_3 || T_{id_{new}}) \\
v_3 &= h(v_1 || r_2') \oplus T_{id_{new}} \\
SK &= h(hID || r_2' || r_3 || P').
\end{align*}$$

$SK$ will be using as the session key between user $U$ and NCC. After that, NCC delivers $\{v_1, v_2, v_3, LEO_{id}\}$ to LEO.

**step 5:** LEO forwards the message $\{v_1, v_2, v_3\}$ from NCC to user $U$.

**step 6:** After receiving $\{v_1, v_2, v_3\}$ from LEO, user $U$ computes

$$\begin{align*}
r_3' &= v_1 \oplus P' \\
T_{id_{new}}' &= v_2 \oplus h(v_1 || r_3) \\
v_2' &= h(P || r_2' || r_3' || T_{id_{new}}')
\end{align*}$$

and compares whether $v_2$ and $v_2'$ are equal. If verification is failed, the session will be broken off. Else, user $U$ computes the session key $SK = h(hID' || r_2' || r_3' || P')$.

**Password update phase**

To enhance security, users are advised to update their passwords at set intervals. To complete the process, user $U$ will perform the following steps (see in Fig 4).

**step 1:** User $U$ inserts the smart card and inputs $ID_u, PW, PW_{new}$ where $PW_{new}$ is the new password chosen by himself/herself. Then user $U$ computes

$$\begin{align*}
r_0' &= Z \oplus h(PW' || ID_u) \\
Verify' &= h(ID_u || r_0' || PW' || r_1).
\end{align*}$$

After that, user $U$ checks whether $Verify'$ equals to $Verify$. If not, user $U$ breaks off the session. Else, user $U$ generates a random number $r_p$ and computes

$$\begin{align*}
hID' &= h(ID_u \oplus r_0') \\
P' &= R \oplus h(hID' || r_1) \\
Q &= P' \oplus r_p \oplus h(PW' \oplus r_0') \\
ChPW &= h(r_p \oplus T_{id}) \oplus h(PW_{new} \oplus r_0') \\
S &= h(hID' || r_3 || T_{id} || ChPW)
\end{align*}$$

After that, user $U$ sends $\{Q, S, T_{id}, ChPW\}$ to LEO.

**step 2:** LEO forwards the message received form user with $LEO_{id}$ to NCC.
step 3: Upon receiving the message, NCC executes the step 3 in Login and authentication phase to obtain $hID$ and $hPW$. Then NCC computes

$$\begin{align*}
P' &= h(hID|x) \\
r'_p &= Q \oplus P \oplus hPW \\
S' &= h(hID||r'_p||T_{id}||ChPW)
\end{align*}$$

and checks whether $S$ equals to $S'$. If not, NCC aborts the session. Else, NCC generates a new temporary identity $T_{idnew}$ and computes

$$\begin{align*}
hPW'_{new} &= h(r'_p \oplus T_{id}) \oplus ChPW \\
v_1 &= h(P||r'_p||T_{idnew}||hPW_{new}) \\
v_2 &= h(P||hPW) \oplus T_{idnew}
\end{align*}$$

Next, NCC replaces $(T_{id}, T_{id}, hPW)$ with $(T_{id}, T_{idnew}, hPW_{new})$ or $(T_{id}, T_{idnew}, hPW_{new})$. Then, NCC sends message $\{v_1, v_2, LEO_{id}\}$ to the LEO.

step 4: Upon receiving the message $\{v_1, v_2, LEO_{id}\}$, LEO forwards $\{v_1, v_2\}$ to $U$. 

Fig 4. Password update phase.

https://doi.org/10.1371/journal.pone.0250205.g004
step 5: After receiving the message, user $U$ computes

$$T_{\text{idnew}} = h(P^\prime || h(PW \oplus r_0^\prime)) \oplus v_2$$

$$v_1^\prime = h(P^\prime || r_p || T_{\text{idnew}}^\prime || h(PW_{\text{new}} \oplus r_0^\prime))$$

and checks whether $v_1$ equals to $v_1^\prime$. If holds, replaces $\{T_{\text{id}}\}$ with $\{T_{\text{idnew}}\}$ in the smart card.

Security analysis

In this section, we will show that our scheme is provably secure in the real-or-random model.

Security model

In 2005, Abdalla et al. introduced a new security model for two-party password-based authenticated key exchange scheme [16]. Based on their real-or-random model, we can prove the security of our scheme. In the model, there are two types of participants, user $U$ and network control center $NCC$, respectively. $U_i$ denotes the $i$th instance of $U$. The adversary $A$, which is abstracted as a probabilistic polynomial time Turing Machine, interacts with other participants through a bounded number of queries which model the capabilities of the adversary in an actual attack. The ability of adversary $A$ is defined by the following queries.

- **Execute**($U_i$, $NCC$): Return the messages transmitted between $U_i$ and $NCC$ in their last key agreement conversation. This query models the eavesdropping attack.

- **Send**($U_i/NCC$, $m$): After receiving message $m$ sent by $A$, $U_i/NCC$ generates a corresponding message for $m$ and outputs it as the result of this query. This query models the active attacks such as replay attack, impersonation attack and so on.

- **CorruptSC**($U_i$): Return the current data stored in $U_i$’s smart card. This query models the smart card lost attack.

- **CorruptDB**($NCC$): Return the current data stored in $NCC$’s database. This query models the insider attack and the stolen verifier attack.

- **Test**($U_i/NCC$): The semantic security of the session key is simulated by flipping an unbiased coin. The query returns a random binary of the same size of session key if $b = 0$ or the session key between $U_i$ and $NCC$ if $b = 1$. The adversary can ask only one time of **Test** query.

- **$H(x)$**: This is a hash query. If a record $(x, h)$ exists in the hash list, $h$ is returned. Otherwise, return a uniformly random string $h$ and store $(x, h)$ in the table.

Semantic security: Providing the above-mentioned queries, the adversary $A$ may interact with the participants to help him/her verify the value of $b$. If he/she can guess correctly, the scheme fails to provide semantic security. Let $\text{Succ}$ denotes the event that $A$ wins. $A$ has an advantage

$$\text{Adv}_{\text{sek}}(A) = |\Pr[\text{Succ}] - \frac{1}{2}|$$

in breaking the semantic security of the scheme. If $\text{Adv}_{\text{sek}}(A)$ is negligible, the scheme is secure under the real-or-random model.

Security proof

Theorem 1: Let $S_{\text{ID}}$ and $S_{\text{PW}}$ be uniformly distributed dictionary of user identity and password, respectively. $|S_{\text{ID}}|$ and $|S_{\text{PW}}|$ denoted the size of $S_{\text{ID}}$ and $S_{\text{PW}}$. $|H|$ denotes the range space of the hash function. Beyond that, we denote $q_h$ and $q_t$ to represent the number of $H(x)$ oracle queries
and the total number of queries executed by $A$. Then, we have

$$Adv_{\text{em}}(A) \leq \frac{8\kappa^2}{2^{|H|}} \max \left\{ \frac{4\kappa}{|\mathbb{Z}|^{\frac{1}{2}}}, \frac{\kappa}{2^{|H|}} \right\}$$

Proof: We define several attack games from game $G_{m_0}$ to game $G_{m_3}$. For each game $G_{m_0}$, $\text{Succ}_i$ denotes the event that $A$ has successfully guessed the bit $b$ in the test session. The games are listed as follows:

Game $G_{m_0}$: This game models the real attack by the adversary. We have

$$Adv_{\text{em}}(A) = |\text{Pr}[\text{Succ}_0] - \frac{1}{2}|$$ (1)

Game $G_{m_1}$: To increase the advantage of success, $A$ launches an eavesdropping attack by querying the $\text{Execute}(U, \text{NCC})$ oracle. Since the session key $SK$ is computed by $hID, r_2, r_3$ and $P$, $A$ tries to obtain these values from the messages transmitted in the public channel. We know that $r_2 = Q \oplus P \oplus hPW, r_3 = v_1 \oplus P$ and $P = h(hID)\left[ x \right] = R \oplus h(hID)\left[ r_1 \right]$. The $hID$ is concealed in the hash function. Thus, $A$ cannot get the values of $hID, r_2, r_3$ and $P$. In this game, the $\text{Execute}(U, \text{NCC})$ query does not provide any advantage compared to game $G_{m_0}$ and we have

$$\text{Pr}[\text{Succ}_0] = \text{Pr}[\text{Succ}_1]$$ (2)

Game $G_{m_2}$: In this game, we add the send query to simulate an active attack. In order to pass the authentication, $A$ must use $H(x)$ query to fabricate messages. No collisions will be found in the input while querying $H(x)$ oracle, because every message contains some different random numbers. By the birthday paradox, we can get

$$\text{Pr}[\text{Succ}_1] - \text{Pr}[\text{Succ}_2] \leq \frac{8\kappa^2}{2^{|H|}}$$ (3)

Game $G_{m_3}$: We transfer game $G_{m_2}$ to this game by adding the $\text{CorruptSC}(U)$ query or the $\text{CorruptDB}(U)$ query.

Case 1: The adversary asks the $\text{CorruptSC}(U)$ query. Then he/she can extract $\{T_{id}, R, r_1, \text{Verify}, Z\}$ stored in the user $U_i$’s smart card. For the adversary, $hID = h(ID_u \oplus r_0) = h(ID_u \oplus Z \oplus h(PW)\left[ ID_u \right]), r_2 = Q \oplus P \oplus hPW = Q \oplus R \oplus h(hID)\left[ r_1 \right] \oplus h(PW \oplus Z \oplus h(hID)\left[ ID_u \right]), r_3 = v_1 \oplus P = v_1 \oplus R \oplus h(hID)\left[ r_1 \right]$ and $P = h(hID)\left[ x \right] = R \oplus h(hID)\left[ r_1 \right]$. So $A$ tries a dictionary attack with the possible identity and password of the user in $S_{ID}$ and $S_{PW}$. Since the scale of the dictionary is $|S_{ID}|$ and $|S_{PW}|$, the adversary needs to guess the correct values of $U_i$’s identity and password simultaneously. In this case, the probability of a successful dictionary attack is negligible. So, we have

$$\text{Pr}[\text{Succ}_2] - \text{Pr}[\text{Succ}_3] \leq \frac{8\kappa^2}{|S_{ID}| \cdot |S_{PW}|}$$ (4)

Case 2: The adversary $A$ asks the $\text{CorruptDB}(\text{NCC})$ query. Then he/she can extract $\{hID, T_{ab}, T_{ah}, hPW\}$ stored in NCC’s database. For the adversary, $r_2 = Q \oplus P \oplus hPW = Q \oplus h(hID)\left[ x \right] \oplus hPW, r_3 = v_1 \oplus P = v_1 \oplus h(hID)\left[ x \right]$ and $P = h(hID)\left[ x \right]$. So $A$ tries a dictionary attack with the possible private key of NCC. So, we have

$$\text{Pr}[\text{Succ}_3] - \text{Pr}[\text{Succ}_4] \leq \frac{8\kappa^2}{2^k}$$ (5)

where $k$ is the security parameter.

The adversary $A$ can choose case 1 or case 2 as the last game $G_{m_3}$. From game $G_{m_0}$ to game $G_{m_3}$, all the oracles are simulated and $A$ has no choice but querying the $\text{Test}$ query and
guessing the bit \( b \) in the last game. Therefore,

\[
Pr[Succ_A] = \frac{1}{2}
\]

Combining (1)-(6), we can derive

\[
Adv_{\text{auth}}(A) \leq \frac{\xi^2}{4}\mathcal{H} + \max\left\{\frac{\xi^4}{|\mathcal{S}_{ID}|^2}, \frac{\xi^3}{2^k}\right\}
\]

The advantage for an adversary to guess the correct session key is negligible since \(|\mathcal{H}|, |\mathcal{S}_{ID}|, |\mathcal{S}_{PW}|\) and \(2^k\) are beyond the polynomial time. So our scheme can provide semantic security in the real-or-random model.

**Other security features**

In this section, we will analyze the security and practicability of our scheme [17–20].

*Provide credible mutual authentication*: In login and authentication phase, NCC can authenticate user \( U \) only when user \( U \) submits correct identity \( ID_\alpha \), password \( PW \) which can satisfy the equation \( S' = h(hID||r_2)||T_{id} = S \). An attacker cannot falsify valid \( S \) and \( Q \) to pass NCC's authentication without the correct identity and password. User \( U \) can authenticate NCC only when the user receives the correct message \( \{v_1, v_2, v_3\} \) which can satisfy the equation \( v_2' = h(P||r_3)||T_{schw}) = v_2 \). But an attacker cannot counterfeit the correct message \( \{v_1, v_2, v_3\} \) without \( P \). Thus, our scheme provides credible mutual authentication.

*Provide perfect forward secrecy*: Forward secrecy means an attacker cannot obtain the past session keys even he/she has got NCC’s private key, users’ password and users’ identity. In our scheme, the session key \( SK = h(hID||r_2)||r_3||P' \) is established with the user’s identity, the random number \( r_2, r_3 \) and the hash value \( P = h(hID||x) \). Even if an attacker obtains those sensitive data, he/she cannot obtain other session keys because he/she cannot derive \( r_2 = P \oplus hPW \oplus Q \) without user \( U \)’s smart card and extracts the identity from it and the attacker also intercepts the communication message between user \( U \) and NCC. Thus, our scheme can resist the stolen-verifier attack.

*Resist stolen-verifier attack*: Stolen-verifier attack means an inside attacker may steal the data in the database, and he can derive users’ password or impersonate legal users to send legitimate login requests. In our scheme, the database stores \( hID, T_{id}, T_{id}, hPW \), if the insider attacker steals these data, he/she cannot derive users’ password \( PW \) without the random number \( r_0 \). In addition to this, the insider attacker cannot impersonate legal users to send legitimate login requests \( Q = P' \oplus r_2 \oplus h(PW \oplus r'_0) \) and \( S = h(hID||r_2)||T_{id} \) to NCC without the password \( PW \), the secret data \( P \) and the random number \( r_0 \). Therefore, our scheme can resist the stolen-verifier attack.

*Resist smart card loss attack*: Assume an attacker gets user \( U \)’s smart card and extracts the parameters \( \{T_{id}, R, r_1, h(\cdot), Verify, Z\} \) from it and the attacker also intercepts the communication message between user \( U \) and NCC. But these parameters do not help him/her perform any attacks without the user’s password \( PW \) and identity \( ID_\alpha \). Thus, our scheme can resist smart loss attack.

*Resist denial-of-service attack*: In login and authentication phase, after NCC authenticating user \( U \), an attacker may intercept and modify the message \( \{v_1, v_2, v_3\} \) which is forwarded from
NCC to U. Obviously, the modified message cannot pass the authentication of user U and the updating of temporary identity will be inconsistent between NCC and user U which must lead to the denial-of-service attack. To avoid that, our scheme adopts dynamic temporary identity (see in Table 2), and NCC stores the temporary identity both of this time and last time. Even the update is inconsistent, the legal user also can login successfully next time with the old temporary identity. Thus, our scheme can resist denial-of-service attack.

**Resist impersonation attack**: Impersonation attack means that an attacker impersonates a legal user to login NCC or impersonates NCC to communicate with a legal user. In our scheme, if the attacker intends to impersonate a legal user U, he/she must computes \( Q = P \oplus r_2 \oplus h(PW \oplus r_0) \) and \( S = h(hID||r_2||T_{id}) \). However, Q and S are established with the secret data \( P \), the temporary identity \( T_{id} \), the password \( PW \) and the identity \( ID_u \). All of those data are secret, so the attacker cannot impersonate a legal user. On the other hand, even if the attacker obtains the temporary identity of the user, he/she cannot know \( P' = h(hID||x) \) without NCC’s private key \( x \). So the attacker cannot masquerade as NCC without \( P \) to compute the correct message \( \{v_1, v_2, v_3\} \). Therefore, our scheme can resist impersonation attack.

**Security and performance comparison**

In this section, we evaluate the performance of our proposed scheme with other three related schemes [14, 21, 22]. The computational cost in the login and authentication phase is compared in detail and the security features of these schemes are also analyzed.

In the proposed scheme, the traditional verification table is improved to form a dynamic one which can resist the denial-of-service attack. In addition, the value stored in the smart card is useless for the attacker to launch any attack. Even if the smart card is lost, user information will not be leaked out. As for the transmitting messages, we transmit user’s temporary identity \( T_{id} \) instead of user’s true identity \( ID_u \). Thus, anonymity has been achieved.

As shown in Table 3, the related schemes [14, 21, 22] have some design flaws and cannot satisfy all the security features. Qi et al.’s scheme [14] suffers from smart card loss attack, denial-of-service attack, off-line guessing attack, replay attack. In addition to this, their scheme also cannot provide forward secrecy and anonymity. Lin et al.’s scheme [21] satisfies most of the security requirements, where as they ignore the smart card loss attack. In Mo et al.’s scheme [22], they cannot resist stolen-verifier attack, smart card loss attack, impersonation attack and cannot provide forward secrecy. Compared with the related works, our scheme can resist most of the known attacks and provide password update service for users to choose.

Table 3. Comparison of security features between our scheme and others.

| Feature                          | Ours | Qi et al.’s [14] | Lin et al.’s [21] | Mo et al.’s [22] |
|----------------------------------|------|-----------------|-----------------|-----------------|
| Provide credible mutual authentication | Y    | Y               | Y               | Y               |
| Provide anonymity                | Y    | N               | Y               | Y               |
| Provide forward secrecy           | Y    | N               | Y               | N               |
| Provide password update           | Y    | Y               | N               | N               |
| Resist stolen-verifier attack     | Y    | -               | Y               | N               |
| Resist smart card loss attack     | Y    | N               | N               | N               |
| Resist denial-of-service attack   | Y    | Y               | Y               | Y               |
| Resist impersonation attack       | Y    | N               | Y               | N               |
| Resist off-line guessing attack   | Y    | N               | Y               | Y               |
| Resist replay attack              | Y    | N               | Y               | Y               |

https://doi.org/10.1371/journal.pone.0250205.t003
Since the registration phase just needs to be executed only once for a certain user, the password update phase isn’t always executed and the login and authentication phase is executed each time, we focus on the performance of login and authentication phase.

The running time of a hash function is $T_h$, the running time of a large number addition is $T_{LA}$, the running time of an elliptic curve scalar point multiplication is $T_{PM}$ and the running time of a XOR operation is $T_x$. Table 4 shows theoretical computational costs comparisons between the proposed scheme and other related schemes in the login and authentication phases. As we know, point multiplication based on elliptic curves is quite time-consuming operation [23–26], our proposed scheme and Lin et al.’s scheme have a great advantage on computational costs, because only hash, XOR and string connection operations are adopted in our scheme and Lin et al.’s scheme.

We carried out a simulation of these schemes with Miracl Library. The hardware platform for user and sever is given an Inter(R) Core(TM) i7-6700 HQ CPU @ 2.60GHz and 8.00GB memory. The length of the random number we used in our simulation is 1000 bits, each simulation was performed for 100 times and the logarithm of average execute time(ms) in the login and authentication phase are shown in Fig 5. The bottom part of the Fig 5 shows real average execute time(ms) of the login and authentication phase. Because the value of time is too small, we show the logarithm of the average execute time on the vertical axis of the graph.

| scheme                  | User Computation | Server Computation | Total(ms)          |
|-------------------------|------------------|--------------------|-------------------|
| Ours                    | $7T_x + 8T_h$    | $3T_x + 4T_h$      | $10T_x + 12T_h$   |
| Qi et al.’s [14]        | $5T_x + 6T_h + 2T_{PM}$ | $3T_x + 2T_h + T_{PM}$ | $8T_x + 10T_h + 3T_{PM}$ |
| Lin et al.’s [21]       | $5T_x + 4T_h$    | $7T_x + 5T_h + 2T_{LA}$ | $12T_x + 9T_h + 2T_{LA}$ |
| Mo et al.’s [22]        | $T_x + 3T_h + 2T_{PM}$ | $T_x + 4T_h + 2T_{PM}$ | $2T_x + 7T_h + 4T_{PM}$ |

https://doi.org/10.1371/journal.pone.0250205.t004

Since the registration phase just needs to be executed only once for a certain user, the password update phase isn’t always executed and the login and authentication phase is executed each time, we focus on the performance of login and authentication phase.

The running time of a hash function is $T_h$, the running time of a large number addition is $T_{LA}$, the running time of an elliptic curve scalar point multiplication is $T_{PM}$ and the running time of a XOR operation is $T_x$. Table 4 shows theoretical computational costs comparisons between the proposed scheme and other related schemes in the login and authentication phases. As we know, point multiplication based on elliptic curves is quite time-consuming operation [23–26], our proposed scheme and Lin et al.’s scheme have a great advantage on computational costs, because only hash, XOR and string connection operations are adopted in our scheme and Lin et al.’s scheme.

We carried out a simulation of these schemes with Miracl Library. The hardware platform for user and sever is given an Inter(R) Core(TM) i7-6700 HQ CPU @ 2.60GHz and 8.00GB memory. The length of the random number we used in our simulation is 1000 bits, each simulation was performed for 100 times and the logarithm of average execute time(ms) in the login and authentication phase are shown in Fig 5. The bottom part of the Fig 5 shows real average execute time(ms) of the login and authentication phase. Because the value of time is too small,
in the vertical axis of the histogram, we use $lg(t)$ to represent the result, where $t$ is real average execute time(ms). The computational costs in Qi et al.’s [14] and Mo et al.’s [22] schemes are much higher than our proposed scheme and Lin et al.’s [21] scheme. The reason is that those schemes involve point operations of an elliptic curve [27–30]. Thus, they are not suitable for satellite communication systems due to the limited computational capability of the devices. Compared with our scheme, the computational overhead of Lin et al.’s scheme costs a little more and we offer more security features such as anonymity, which is not provided in Lin et al.’s scheme. Therefore, our proposed scheme provides an efficient and secure authentication for satellite communication systems.

Conclusions
In this paper, we proposed an efficient and provably secure key agreement scheme for satellite communication systems to provide credible mutual authentication. A dynamic temporary identity mechanism was adopted to ensure users’ anonymity. Besides, the traditional verification table was replaced by a dynamic verification table to resist denial-of-service attack caused by inconsistent data updating between NCC and user $U$. In addition, our scheme only adopted lightweight hash and string operations, which reduced the computational cost in comparison with other related works. We also proved the proposed scheme is provably secure in the real-or-random model. Therefore, the proposed scheme can meet the efficiency demands and security needs of communication satellite systems successfully.

Supporting information
S1 File.
(DOCX)

Author Contributions
Conceptualization: Yuanyuan Zhang.
Data curation: Yuanyuan Zhang, Zhibo Zhai.
Formal analysis: Yuanyuan Zhang.
Funding acquisition: Yuanyuan Zhang.
Investigation: Yuanyuan Zhang.
Methodology: Yuanyuan Zhang.
Project administration: Yuanyuan Zhang.
Resources: Yuanyuan Zhang, Zhibo Zhai.
Software: Zhibo Zhai.
Supervision: Yuanyuan Zhang.
Validation: Yuanyuan Zhang.
Visualization: Yuanyuan Zhang.
Writing – original draft: Yuanyuan Zhang.
Writing – review & editing: Yuanyuan Zhang, Zhibo Zhai.
References

1. Fossa CE, Raines RA, Gunsch GH, Temple MA. An overview of the IRIDIUM (R) low Earth orbit (LEO) satellite system. In Proceedings of the IEEE 1998 National Aerospace and Electronics Conference, NAECON'98, Dayton, U.S.A. 1998; 152-159.

2. Yiltas D, Halim Zaim A. Evaluation of call blocking probabilities in LEO satellite networks. Int J Satell Commun Netw 2006; 27(2): 103–115. https://doi.org/10.1002/sat.928

3. Zhou Y, Sun F, Zhang B. A novel QoS routing protocol for LEO and MEO satellite networks. Int J Satell Commun Netw 2007; 25(6): 603–617. https://doi.org/10.1002/sat.892

4. Lasc L, Dojen R, Coffey T. Countering jamming attacks against an authentication and key agreement protocol for mobile satellite communications. Comput Electr Eng 2011; 37(2): 160–168. https://doi.org/10.1016/j.compeleceng.2011.01.011

5. Ercetin O, Ball MO, Tassiulas L. Next generation satellite systems for aeronautical communications. Int J Satell Commun Netw 2004; 22(2): 157–179. https://doi.org/10.1002/sat.770

6. Liu Y, Zhang A, Li S, Tang J, Li J. A lightweight authentication scheme based on self updating strategy for space information network. Int J Satell Commun Netw 2017; 35(3): 231–248. https://doi.org/10.1002/sat.1179

7. Chang CC, Cheng TF, Wu HL. An authentication and key agreement protocol for satellite communications. Int J Commun Syst. 2012; 27(10): 1994–2006. https://doi.org/10.1002/dacs.2448

8. Cruickshank HS. A security system for satellite networks. IEEE Satellite System Mobile Communication Navigation. UK, 1996; 187–190. https://doi.org/10.1049/cp:19960437

9. Hwang MS, Yang CC, Shiu CY. An authentication scheme for mobile satellite communication systems. ACM SIGOPS Operating Systems Review 2003; 37(4): 42–47. https://doi.org/10.1145/958965.958970

10. Chang YF, Chang CC. An efficient authentication protocol for mobile satellite communication systems. ACM SIGOPS Operating Systems Review 2005; 39(1): 70–84. https://doi.org/10.1145/1044552.1044560

11. Chen TH, LEE WB, Chen HB. A self-verification authentication mechanism for mobile satellite communication systems. Commput Electr Eng 2009; 35(1): 41–48. https://doi.org/10.1016/j.compeleceng.2008.05.003

12. Lee CC, Li CT, Chang RX. A simple and efficient authentication scheme for mobile satellite communication systems. Int. J. Satell. Commun. Network. 2012; 30: 29–38. https://doi.org/10.1002/sat.993

13. Zhang Y, Chen J, Huang B. An improved authentication scheme for mobile satellite communication systems. Int J Sat Comm Network. 2015; 33(2): 135–146. https://doi.org/10.1002/sat.1079

14. Qi M, Chen J. An enhanced authentication scheme for satellite communication systems. Int J Satell Commun Network. 2018; 36(3): 296–304. https://doi.org/10.1002/sat.1218

15. Zhang LP, Zhang YX, Tang SY. Privacy Protection for E-Health Systems by Means of Dynamic Authentication and Three-Factor Key Agreement. IEEE Transactions on Industrial Electronics 2018; 65(3): 2795–2808. https://doi.org/10.1109/TIE.2017.2739683

16. Abdalla M., Fouque P.A. and Pointcheval D. Password-based authenticated key exchange in the three-party setting. public Key Cryptography-PKC 2005; 3386: 65–84.

17. Jiang Q, Zhang N, Ni JB, Ma JP, Choo RK. Unified Biometric Privacy Preserving Three-factor Authentication and Key Agreement for Cloud-assisted Autonomous Vehicles. IEEE Transactions on Vehicular Technology 2020; 69(9): 9390–9401. https://doi.org/10.1109/TVT.2020.2971254

18. Dabra V, Bala A, Kumari S. Reconciliation based Key Exchange Schemes using Lattices: A Review. Telecommunication Systems 2021. https://doi.org/10.1007/s11235-021-00759-0

19. Dabra V, Bala A, Kumari S. LBA-PAKE: Lattice-based Anonymous Password Authenticated Key Exchange for mobile devices. IEEE Systems 2020. https://doi.org/10.1109/JSYST.2020.3023808

20. Kumari S, Renuka Km. A Provably Secure Biometrics and ECC Based Authentication and Key Agreement Scheme for WSNs. International Journal of Communication Systems 2019; 33(3).

21. Lin HY. Efficient dynamic authentication for mobile satellite communications systems without verification table. Int. J. Satell. Commun. Network 2016; 34: 3–10. https://doi.org/10.1002/sat.1088

22. Mo JQ, Hu ZW, Lin YH. Remote user authentication and key agreement for mobile client-server environments on elliptic curve cryptography. The Journal of Supercomputing 2018; 1–17.

23. Abichar PE, Mhamed A, Elhassan B. A fast and secure elliptic curve based authenticated key agreement protocol for low power mobile communications. In: International Conference on Next Generation Mobile Applications, Services and Technologies 2017; 235–240.

24. Yang JH, Chang CC. An efficient three-party authenticated key exchange protocol using elliptic curve cryptography for mobile-commerce environments. J Syst Softw 2009; 82: 1497–1502. https://doi.org/10.1016/j.jss.2009.03.075
25. Liu T, Zhu H. An ID-based multi-server authentication with key agreement scheme without verification table on elliptic curve cryptosystem. In: International Conference on Computational Aspects of Social Networks 2010; 61–64.

26. Reddy AG, Das AK, Yoon EJ, Yoo KY. A secure anonymous authentication protocol for mobile services on elliptic curve cryptography. IEEE Access 2016; 4: 4394–4407. https://doi.org/10.1109/ACCESS.2016.2596292

27. Ammayappan K, Saxena A, Negi A. Mutual authentication and key agreement based on elliptic curve cryptography for GSM. In: International Conference on Advanced Computing and Communications, 2006; 183–186.

28. Lee CI, Chien HY. An elliptic curve cryptography-based RFID authentication securing e-health system. Int J Distrib Sens Netw 2015; 11(12): 642425. https://doi.org/10.1155/2015/642425

29. Chien HY. Elliptic curve cryptography-based RFID authentication resisting active tracking. Wirel Pers Commun 2017; 94: 2925–2936. https://doi.org/10.1007/s11277-016-3756-0

30. Li X, Niu J, Bhuiyan MZA, Wu F, Karuppiah M, Kumari S. A robust ECC based provable secure authentication protocol with privacy protection for industrial internet of things. IEEE Transactions on Industrial Informatics 2018; 14(8): 3599–3609. https://doi.org/10.1109/TII.2017.2773666