A Novel Authentication Mechanism for Mobile Satellite Communication Systems

Anca Delia Jurcut*, Jinyong Chen†, Anshuman Kalla‡, Madhsanka Liyanage§‖, John Murphy¶
*†§¶School of Computer Science, University College Dublin, Ireland, ‡Manipal University Jaipur, India
‖Centre for Wireless Communications, University of Oulu, Finland
{anca.jurcut*, jinyong.chen†,madhusanka§, j.murphy¶}@ucd.ie, anshuman.kalla@jaipur.manipal.edu‡,
madhusanka.liyanage@oulu.fi‖

***Abstract*—The authentication protocols existing in the realm of mobile satellite communication networks, usually employ the one-time shared secret technique. Although the technique combats well against replay attacks, however, it is vulnerable to desynchronisation attacks. The later type of attacks, if framed and mounted against the crucial update mechanisms, which occur in mobile satellite communication systems, can lead to permanent Denial of Service (DoS) conditions. In this context, the authentication protocol initially proposed by Lee et al. [1] has emerged as the de-facto protocol and forms the basis for various other authentication protocols developed since then.**

In this paper our contribution is two-fold. The first part of the paper presents an analysis of the authentication protocol [1], which reveals that the protocol is fundamentally susceptible to two attacks: impersonation attack and desynchronisation attack. To overcome these susceptibilities, in the second part of the paper, a new authentication protocol is proposed which incorporates a resynchronization phase. The paper demonstrates that the proposed solution is robust to impersonation attacks as well as to permanent DoS conditions caused by desynchronisation attacks. Moreover, the proposed solution is expected to find its application to address the desynchronisation issue found in numerous other recently published enhanced authentication protocols.

***Index Terms*—Mobile Satellite Communication; Authentication; Network Security; Denial of Service;**

I. Introduction

With the rapid adaptation and development of wireless communication technologies, lately, mobile satellite communication has gained significant attention of users. The key advantages of using mobile satellite include ubiquitous (world-wide) coverage and high data-rate transmission capability. Essentially, mobile satellite communication eliminates the 5 constraint posed by cellular/terrestrial network on mobile equipment to be within the limited coverage area. Nevertheless, from a security perspective, as other communication systems, mobile satellite communication systems are also vulnerable to various security attacks such as Denial of Service (DoS), replay, impersonation, stolen-verifier and desynchronization attacks. To solve these security issues, different user authentication protocols have been proposed so far. A survey on various authentication protocols, attacks against these protocols and the reasoning for these attacks can be found in [2], [3], [4], [5], [6].

Within the realm of mobile satellite communication systems, some satellite phones use geosynchronous satellites situated in geostationary equatorial orbit. Such satellites apparently stay permanently in the same area in space and maintain near-continuous global coverage all-day long. However, the higher altitude of geostationary satellites results in signal delay issue, which affects the performance of real-time communication services such as telephone conversation. Compared to this, the mobile satellite communication system which makes use of Low Earth Orbit (LEO) satellite technology not just provides global wireless coverage with no gap but also incurs shorter transmission delay and thus has gained momentum. Particularly, in this context, numerous authentication protocols using one-time shared secrets have been proposed in the recent years [1], [3], [7], [8], [9], [10], [11]. Nonetheless, it turns out that many security issues such as vulnerability to impersonation attacks, DoS attacks and replay attacks, still need to be addressed in these systems.

In 2009, Chen et al. [7] proposed a self-verification authentication protocol for mobile satellite communication systems. The protocol was computationally complex as it makes use of exponent operations. Based on Chen et al.'s protocol, Yoon et al. [8] put-forward a new efficient and anonymous authentication protocol using a secure one-way hash function that gets rid of the sensitive verification table. In 2012, Lee et al. [1] pointed out that the protocol proposed by Chen et al. cannot withstand stolen-verifier attacks and proposed a light-weight protocol to knockout this attack. Further, the authors in [1] claimed that the proposed protocol was designed to withstand several kinds of attacks, such as impersonation attacks, denial-of-service attacks, replay attacks and stolen-verifier attacks. Since then, this protocol has become the de-facto base protocol for multiple authentication protocols i.e. [9], [10], [11].

Nevertheless, the protocol proposed by Lee et al. [1] is susceptible to two new attacks. First, an impersonation attack rolled-out by implementing a reduplicate registration. Second, a desynchronisation attack, where the attacker jams a message and replays a single message in the protocol in order to create the permanent DoS condition which consequently disrupts the counter value at the receiver. Inevitably, the discrepancy in the counter values leads to all the future communication to be literally out of synchronization.

Moreover, after skimming through most of the protocols surfaced till date which have been built by inheriting the philosophy of Lee et al. [1] protocol, it could be concluded to a great extent that these are still vulnerable to attacks. This impel and paves the way for our current work. Thus the paper proposes a new authentication and key agreement protocol for mobile satellite communication systems, with the aim to solve the newly revealed exploitable weaknesses of the Lee et al.'s protocol. The new protocol includes an extra password requirement during the registration phase to block the chances of the same identity registration and also an additional resynchronisation challenge to prevent the possibility of the Network Control Centre (NCC) and the mobile user getting desynchronised on their shared secrets.

The remainder of this paper has the following structure. Section II analyses weaknesses of Lee et al.'s [1] protocol and manifests that it is prone to new attacks. Section III describes the details of our proposed protocol. Section IV delves to demonstrate the security analysis of our proposed protocol. Finally, Section V concludes our findings.

II. Security analysis of LLC protocol

The section intends to bring to limelight the fact that the Lee et al. [1]'s protocol is vulnerable to impersonation attack as well as to desynchronization attack leading to a permanent denial-of-service condition. For the rest of the paper, the authentication protocol proposed by Lee et al. [1] is dubbed as LLC protocol based on the names of the authors. Table I depicts the notations being used throughout this paper.

TABLE I: Summery of the notations

Notation	Description
U	Mobile user U
$I(U)$	Intruder impersonating U
U_{ID}	Identity of the mobile user
T_{ID}	Temporary identity of the mobile user
LEO_{ID}	Identity of the LEO satellite
K_{NCC}^{-}	A long-term private key generated by the NCC
SK	Session key
Q	User nonce parameter $Q = R \oplus h(U_{ID} \Vert N_k) \oplus Nu$
S	User nonce verifier $S = h(U_{ID} \Vert N_u)$
P	NCC private token $P = h(U_{ID} \Vert K_{NCC}^{-})$
R	User private token $R = P \oplus h(U_{ID} \Vert N_u)$
V_1	NCC nonce parameter $V_1 = P \oplus N_{NCC}$
V_2	NCC nonce verifier $V_2 = h(P \Vert N_u \Vert N_{NCC} \Vert V_{4)}$
V_3	TID update token $V_3 = h(N_u \Vert N_{NCC}) \oplus T_{IDnew}$
V_4	NCC generates $V_4 = V_3 \oplus T_{IDnew}$
N_u, N_{NCC}	Random value of the mobile user and NCC

A. Impersonation attack

Lee et al. [1] claimed that LLC protocol is proof against impersonation attacks, since it is impossible for an attacker to compute $P = h(U_{ID}||K_{NCC}^{-})$ without the long-term private key K_{NCC}^{-}; even if the attacker hacks the NCC and gets access to the verification table containing crucial identities.

On the contrary, there is another way to mount an impersonation attack. The rationale is, once the same IDs are registered in the NCC, it can still distinguish different users based on the temporary identities which were issued during the individual registration phase. Hence, the intruder can leverage this weakness to carry-out an impersonation attack as follows: Assume that an attacker just needs to find out the true identity by means of interception such as snooping, spoofing or guessing. Also, the true identity might be found with the help of collision attack [12], where the attacker tries to find two arbitrary messages with the same hash value. Since, a hash of n bits can be broken in $2^{n/2}$ time, it could be easy to infer the user ID from the login message $S = h(U_{ID}||N_u)$, if the random value N_u is not long enough to provide security strength. Once the true identity U_{ID} leaks out by any covetous means, the attacker can impersonate the victim in order to register the clashing IDs, which is accepted by the NCC. After successful reduplicate registration, the attacker can retrieve the servers private key $P = h(U_{IDab}||K_{NCC}^{-})$ by implementing the following calculations:

Step 1: Attacker registers its clashing ID, U_{IDab} at NCC.

Step 2: The NCC caches the intruder's unique temporary identity $T_{ID(I)}$, registers the user ID U_{IDab} into the database and loads ($T_{ID(I)}$, $R_{(I)}$, $N_{k(I)}$, $h(.)$) into the smart card. Next, the server determines the secret parameters by performing the following operations:

(i) Calculates the server's private key for the intruder $P_{(I)} = h(U_{IDab}||K_{NCC}^{-})$

(ii) Calculates the intruder's secret parameter $R_{(I)} = P_{(I)} \oplus h(U_{IDab}||N_{k(I)}) = h(U_{IDab}||K_{NCC}^{-}) \oplus h(U_{IDab}||N_{k(I)})$

Step 3: The attacker uses its secret parameter $R_{(I)}$ to maliciously retrieve the private key of server without knowing the long-term private key K_{NCC}^{-}. To do so, the attacker merely needs to login with the correct user ID and carries out following calculations: $P_{(I)} = R_{(I)} \oplus h(U_{IDab}||N_{k(I)})$ and $P'_{(I)} = P_{(I)} \oplus h(U_{IDab}||N_{k(I)}) \oplus h(U_{IDab}||N_{k(I)}) = P_{(I)} = h(U_{IDab}||K_{NCC}^{-})$

S1.1 U → NCC : (Q, S, T_{ID})
S1.2 NCC → U : (V_1, V_2, V_4)
S2.1 I(U) → NCC : (Q, S, T_{ID})
S2.2 NCC → I(U) : (V_1, V_2, V_4)
S3.1 I(U) → NCC : (Q, S, T_{IDnew})
S3.2 NCC → I(U) : (V_1', V_2', V_4')

Fig. 1: The demonstration of the impersonation attack

Once the attacker manages to acquire the server's private key $P_{(I)} = h(U_{IDab}||K_{NCC}^{-})$ (by eavesdropping the messages exchanged between the victim mobile user and the NCC), s/he can impersonate the original user to gain access to the server. Figure 1 elucidates this new impersonation attack.

Upon receiving an authentication request from U, the intruder extracts (Q, S, T_{ID}) and then performs following tasks to capture the mobile user's secret information:

S2.1: After intercepting the login message, the intruder uses the previously captured private key $P_{(I)} = h(U_{IDab}||K_{NCC}^{-})$ to compute $N_u = Q \oplus P_{(I)}$ and $S' = h(U_{IDab}||N_u)$.

Next, the attacker verifies if the intercepted value of S' is the same as the computed value of S; if true then the attacker can confirm that the mobile user possesses the same user ID. In other case, there would be no interest for attacker to intercept the user further.

S2.2: Confirming that attacker possesses the same ID as that of mobile user, the attacker can compute the recorded replying message (V_1, V_2, V_4). Hence, the attacker can retrieve all the secret parameters by executing exactly the same steps from the authentication phase of the protocol [1]. Eventually, the attacker obtains the new temporary identity T_{IDnew} and the current session key. Henceforth, the attacker can impersonate the original user to communicate with the NCC.

S3.1: The intruder pretends to be the mobile user U and sends the login message (Q, S, T_{IDnew}) to LEO which relays it to the NCC with its own ID, LEO_{ID}.

S3.2: NCC authenticates the intruder $I(U)$ and updates a new $T_{ID'new}$ in the verification table for the next authentication phase. When the intruder receives the message (V_1', V_2', V_4'), it performs same tasks as described in S2.2. An attacker can impersonate U (and communicate with NCC) based on the assumptions of Lee et al.'s protocol [1]: NCC knows the old ID of a mobile user; intruder needs to establish connection with the NCC just one more time; NCC will replace T_{IDnew} with a new temporary identity for an intruder. As a result of impersonation, the actual user U will not be able to establish communication with NCC anymore, and the re-login system cannot work properly, since the temporary identity of U, T_{IDnew} stands invalid as per the NCC's database. This leads to a permanent DoS condition.

B. Desynchronisation Attack

The LLC protocol uses an online update mechanism to generate new instances of shared secrets (T_{IDnew}). All the involved communicating parties agree to accept this new temporary identity and current session key as the shared secrets. This update mechanism thus ensures that all entities will hold the same shared secrets at the end of each protocol run [4]. However, it is worth noticing that U and NCC find it difficult to update their shared secrets simultaneously during the roll-out of update mechanism. In fact, the NCC replaces the old secrets with new secrets after verifying the identity of the mobile user. Here, the update of the shared secrets happens at the mobile user asynchronously. Moreover, the authentication of the NCC relies on the successful message reception to ensure unanimity of shared secrets after a protocol run.

As presented in Figure 2, the authentication phase of the LLC protocol is prone to permanent DoS condition due to the dependence on successful message reception.

S1.1 U → NCC : (Q, S, T_{ID})
S1.2 NCC → U : (V_1, V_2, V_4)
A1.1 I(U) → NCC : (Q, S, T_{ID})
A1.2 NCC → I(U) : (V_1, V_2, V_4)
S2.1 U → NCC : (Q, S, T_{IDnew})
S2.2 NCC → U : Permanent DoS Condition

Fig. 2: The demonstration of desynchronisation attack

An attacker can mount a desynchronisation attack leading to a permanent DoS condition, by performing the following steps (as depicted in Figure 2):

S1.1: The attacker intercepts the login message and obtains the tuple (Q, S, T_{ID}).

S1.2: The attacker jams this message by using a low-power jammer and sends the recorded message. After receiving this message, U authenticates NCC. Following that, U changes its temporary identity to T_{IDnew}, for the next authentication request.

A1.1: The attacker replays the previous recorded message (Q, S, T_{ID}) to launch re-login request.

A1.2: On receiving the re-login request, the NCC re-authenticates the attacker. Next, NCC replaces both the temporary identity T_{IDnew} and the session key SK_{new} with the new temporary identity T'_{IDnew} and the new session key SK'_{new} respectively. Further, it updates $(U_{ID}, T'_{IDnew}, T_{ID})$ in its verification table, however, the user U skips the updating task because the reply messages are jammed. Consequently, NCC and U are desynchronised based on the discrepancies in the values of temporary identity and the session key. This implies, while U attempts to exchange data with NCC using its old session key SK_{new}, the NCC expects U to utilize new session key SK'_{new}. This will lead to denial of access to the satellite services. If this scenario occurs, eventually the user's time-out timer will expires while waiting for a response from the NCC.

S2.1: As per LLC protocol, U's reaction to such a time-out is that it will initialize a re-login request using the updated identity T_{IDnew}, since U has already updated the value.

S2.2: On the contrary, NCC expects U to use either the new updated value T'_{IDnew} or the old value T_{ID}, rather than an invalid value T_{IDnew}. Ultimately, the mobile user will not be able to provide the accurate evidence of its legitimate identity to the NCC. This leads to, all the subsequent authentication requests from U to be permanently interpreted by the NCC as illegitimate requestsThis causes a permanent DoS condition.

III. Our Proposed Protocol

The two attacks presented in the section II, highlight the impact of the existing weaknesses in the design of LLC protocol. Addressing these issues, we propose a new security authentication protocol, as outlined in the flow diagram represented by Figure 3. The reduplicate registration problem of the LLC protocol is solved by verifying the unique user information token N during the registration phase, which in-turn prevents the weaknesses exploitable by the impersonation attack. Further, the weaknesses exploitable by the desynchronisation attack are fixed by extending the resynchronisation challenge.

A. Registration Phase

Assume K^-_{NCC} is the long-term private key owned by the NCC. During the registration phase, the mobile user U is free to register its identity U_{ID} with a password PW_U by sending this information to the service provider NCC via a secure channel. When NCC receives U's registration request,it performs the following operations:

$$P = h(U_{ID}||K^-_{NCC}) \text{ and } N = h(U_{ID}||PW_U)$$
$$R = P \oplus N = h(U_{ID}||K^-_{NCC}) \oplus h(U_{ID}||PW_U)$$

Every mobile user U with an identity U_{ID} holds a unique user information token N as well, which is used to prevent

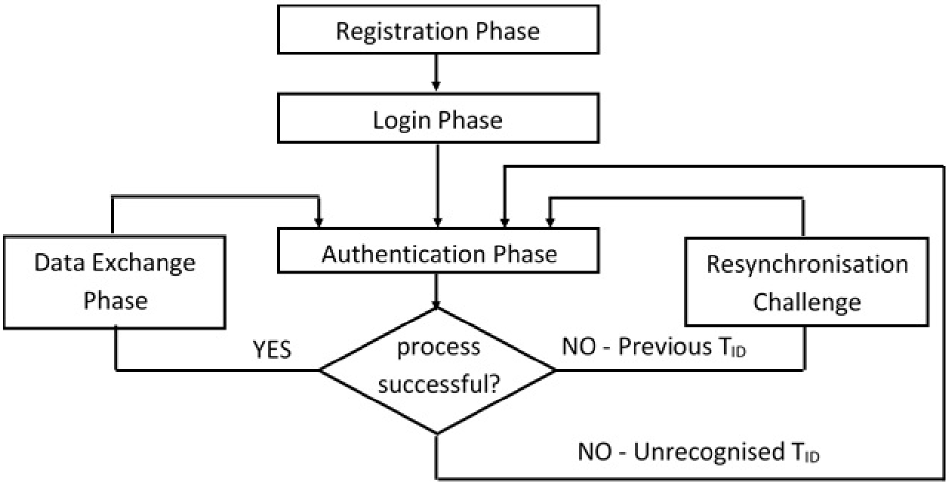

Fig. 3: Proposed protocol

reduplicate registration at the server. All the unique user information tokens are arranged to be stored in a standalone database. Here, NCC does not allow two different users to register with the same value N. When NCC receives a request from U, it compares the received value of N against all the stored user information values in the standalone database. If a match is found, U needs to pick a new username and password, and retries to submit the request to NCC. For a request with unique value of N, NCC generates corresponding temporary identity T_{ID}. For each successful authentication value of T_{ID} will be refreshed and accordingly NCC will update the values of U_{ID} and T_{ID} in the verification table. Then, the initial shared secret R and T_{ID} will be stored in a smart card or will be directly upload to the mobile device through a secure channel.

B. Login Phase

To launch a login request, a mobile user U needs to input both its identity U_{ID} and password PW_U into the mobile phone. Then, the smart card selects a secret random number N_U so as to calculate the following secret values:

$N' = h(U_{ID}||PW_U)$
$P' = R \oplus N'$ and $Q = P' \oplus N_U$
$S = h(U_{ID}||T_{ID}||N_U)$

Subsequently, U sends the login message (Q, S, T_{ID}) to LEO and then LEO forwards it along with its LEO_{ID} to NCC.

C. Authentication Phase

Figure 4 delineates the authentication phase of our proposed protocol.

The authentication phase is enabled to distinguish between legitimate authentication responses (1.3a and 1.4a) and fraudulent authentication responses (1.3b and 1.4b). Upon receiving the first authentication request from U, LEO responds with (Q, S, T_{ID}, LEO_{ID}). Next, NCC checks the legitimacy of LEO's identity and then lookup T_{ID} to retrieve (U_{ID}, T_{ID}). Subsequently, NCC computes the following secret values:

1.1 U → LEO : (Q, S, T_{ID})
1.2 LEO → NCC : (Q, S, T_{ID}, LEO_{ID})
1.3a NCC → LEO : $(Grant, V_1, V_2, V_4, LEO_{ID})$
1.4a LEO → NCC : $(Grant, V_1, V_2, V_4)$
1.3b NCC → LEO : $(Deny, V_1, V_2, V_4, LEO_{ID})$
1.4b LEO → NCC : $(Deny, V_1, V_2, V_4)$

Fig. 4: Proposed authentication phase

$P = h(U_{ID}||K^{-}_{NCC})$
$N'_u = Q \oplus P$
$S' = h(U_{ID}||T_{ID}||N'_u)$

NCC then compares the computed value S' with the received value S and if found same, the mobile user is authenticated. Now, NCC chooses a secret random number N_{NCC} to compute the following values:

$V_1 = P \oplus N_{NCC}$ and $V_3 = h(N'_U||N_{NCC})$

NCC generates the new temporary identity T_{IDnew} and calculates the following:

$V_4 = V_3 \oplus T_{IDnew}$
$V_2 = h(Grant||P||N'_U||N_{NCC}||V_4)$
$SK = h(U_{ID}||N'_U||N_{NCC}||P)$

Next, NCC updates the lookup table with the corresponding entries i.e. $(U_{ID}, T_{IDnew}, T_{ID})$ and sends the message including Grant flag, V_1, V_2, V_4, LEO_{ID} to LEO.

On receiving the message $(Grant, V_1, V_2, V_4)$ from LEO, U computes the following values:

$N'_{NCC} = P' \oplus V_1$
$V'_2 = h(Grant||P'||N_U||N'_{NCC}||V_4)$

U verifies the validity of the equation $V'_2 = V_2$. If this holds, U accepts the authenticity of responding NCC and updates the stored secret data to $(T_{IDnew}, R, h(.))$.

D. Resynchronisation Phase

If NCC detects an illegitimate authentication request generated using the previous value T_{ID}, it concludes that a desynchronisation situation has occurred. Otherwise, the NCC simply discards that authentication request thinking it as a replay attack. In the former condition, NCC responds with a resynchronisation challenge for an incoming Us request with an old value T_{ID}.

In this resynchronisation phase (outlined in Figure 5), after ensuring the previous value T_{ID} from U, NCC needs to re-authenticate the user under the regular authentication process. Once the authentication is completed, it generates the secret values based on the received parameters, V_2. The validation seed of resynchronisation phase must be calculated in a different manner, with V_2 instead of the normal authentication phase:

$$V_2 = h(Grant||P||N'_U||N_{NCC}||V_4)$$

Next, NCC sends back to user the deny message including the $DENY$ flag, V_1, V_2 and V_4. Then, U has to authenticate NCC, by using the T'_{ID} provided within V_4.

1.1 U → LEO : (Q, S, T_{ID})
1.2 LEO → NCC : (Q, S, T_{ID}, LEO_{ID})
1.3a NCC → LEO : $(Deny, V_1, V_2, V_4, LEO_{ID})$
1.4a LEO → NCC : $(Deny, V_1, V_2, V_4)$
2.1 U → LEO : (Q', S', T'_{ID})
2.2 LEO → NCC : $(Q', S', T'_{ID}, LEO_{ID})$

Fig. 5: Proposed resynchronisation phase

On receiving the deny message $(Deny, V_1, V_2, V_4)$, the user establishes the validation seed $V'_2 = h(Grant||P'||N_U||N'_{NCC}||V_4)$. This message contains the expected random value N_U. U verifies the authenticity of the NCC by ensuring that the computed value V'_2 is same as the value of received replying parameter V_2. Further if values are equal, U accepts the resynchronisation challenge as a valid challenge. Then, U retrieves T'_{ID} from the re-challenge message and uses it to re-compute the login parameter $S' = h(U_{ID}||T_{ID}||N'_u)$ and the secret value $Q' = P' \oplus N'_U$. Next, U resends the authentication request (Q', S', T'_{ID}). If the value of S' is same as that of S, NCC accepts the re-login request as legitimate and repeats the steps in the authentication phase. If the expected T'_{ID} is not included in the re-login request or the re-login message is not received by NCC within a time-out period, the resynchronisation challenge is deemed bogus.

IV. Security analysis of our proposed protocol

Security analysis and verification of the required goals in the protocols designed for communication is an imperative step as similar to [13], [14], [15]. This section analyzes the security features of the proposed protocol and demonstrates that it fulfills all the required security goals for a mobile satellite communication network.

A. Key Security Criteria

Preferably, a mobile satellite communication protocol should satisfy following security criteria for reliable and efficient operation.

1) Free selection of identity: The mobile user should be able to register its identity freely, without caring of the possibility of an impersonation attack where an identical user ID can be registered by an attacker. In the proposed systems, NCC prevents any attacker to register with already used user information token value.

2) Mutual authentication: Since both mobile user and the NCC are able to verify each others identity and share common session key, mutual authentication is essentially achieved. Server, on receiving the login request (Q, S, T_{ID}) by a user, authenticates the user by verifying the login parameter S. It then sends the message (V_1, V_2, V_4) to the user. Hence, only a legitimate user can obtain the secret random T to authenticate the NCC, by means of checking the validity of the response V_2.

3) Confidential communication: Confidentiality between mobile user and the NCC is ensured by utilizing the shared session key to encrypt the exchanged messages. Employing a random secret number N_U and N_{NCC} in each session, the mobile user and the NCC establish a session key $SK = h(U_{ID}||N_U||N_{NCC}||P)$ during every authentication phase. Thus, the session keys are independently generated for a particular session and are simultaneously confirmed by both participants.

4) User's privacy: It is vital to keep the identity of the user private. To do so the proposed authentication protocol prevents the transmission of the user's identity U_{ID} over the network, instead, a temporary identity is adopted in each session.

5) Minimum trust establishing parties: No extra trust party except the NCC (where the mobile users register themselves) is needed. The NCC is assumed to be trustworthy because the mobile users have to register with their personal details and their password to access required services.

6) Perfect forward secrecy: No secret information of the user is leaked during the transmission, since all the secret parameters are computed by a one-way hash function and are transmitted using Exclusive-OR operations with random numbers. Even if an attacker intercepts the login request (Q, S, T_{ID}) from the mobile user and records the replying messages (V_1, V_2, V_4) from the NCC to deploy replay attacks, the attacker will be unable to compute the useful secrets without knowing the random number. A demonstration of the protocol being resistant to replay attacks will be presented in the subsection IV-B.

7) Reliable Key management: A simplified key management is yet another forte of our proposed protocol. In this protocol, use of public key is replaced with long-term private key K^-_{NCC} of the NCC. Moreover, the last session key plays no role in determining the new session key. Furthermore, no correlation exists between the session keys since a new random number is used for every session.

8) Resynchronisation process: The resynchronisation process is invoked whenever NCC receives an illegitimate authentication request. Instead of granting the access, NCC responds with a resynchronisation challenge. Moreover, NCC expects to repeat authentication using the updated T'_{ID}.

9) Low computational Cost: The proposed protocol dislodges the use of public and symmetric keys. Instead, it makes use of a few one-way hash functions and some Exclusive-OR operations in order to carry out the computations required by the protocol. Since these are not computationally intensive operations, it is efficient to implement on the mobile devices.

Table II reflects a performance comparison of the proposed protocol with the other relevant protocols in terms of computational complexity. As observable, the proposed protocol incurs an extra resynchronize phase which accounts for 17 hash and 14 XOR operations. Nevertheless, this overhead shows-up only when the resynchronization phase is triggered. Otherwise the protocol cost only 10 hash and 8 XOR operations which is the same as that of the LLC protocol. It is worth noting that even without extra resynchronize phase the proposed protocol is more secure than LLC protocol.

TABLE II: Computational complexity in authentication phase

	Chen et al.[7]	**Lee et al.[1]**	**Our proposal**
Hash	2	10	10(17)*
XOR	0	8	8(14)*
MAC	2	0	0
Symmetric (en/de)cryption	2	0	0
Computational cost	23.52 μJ[16]	7.60μJ	7.60(12.92)* μJ

* () shows the resynchronisation phase involved

B. Efficiency of the Proposed Protocol

The efficiency of the proposed protocol is checked against the three existing relevant protocols, i.e. [7], [8] and [1]. The results are shown in Table III. Scanning through the results, one can easily conclude that the proposed protocol is able to satisfy all the nine criteria required to design a secure and efficient mobile satellite authentication mechanism.

TABLE III: Comparison of Various Protocols

Key Criteria	**Chen et al.[7]**	**Yoon et al.[8]**	**Lee et al.[1]**	**Our proposal**
Free selection of ID	NO	YES	YES	YES
Mutual authentication	YES	YES	YES	YES
Confidentiality	YES	YES	YES	YES
User's Privacy	YES	YES	YES	YES
Minimum trust building parties	YES	YES	YES	YES
Low computational cost	NO	YES	YES	YES
Perfect forward secrecy	YES	YES	YES	YES
Reliable Key management*	NO	YES	YES	YES
Resynchronisation	NO	NO	NO	YES

* No complex of PKI or SKI involved

C. Discussion on Ability to Resist Various Attacks

In this section, we discuss the ability of the proposed protocol to withstand various kind of attacks.

1) Impersonation attacks: Provided that server allows the reduplicated registration during the registration phase, the LLC protocol is susceptible to impersonation attacks, as showed in section II-A. To prevent this attack, one of the simple ways is to set up policies on the servers that do not allow certain kinds of reduplicated registration during the registration phase. A better way, which the proposed protocol advocates, is to make use of the information token N stored on the database to prevent reduplicated registration.

Even if an attacker hacks the database, it can retrieve all the hash values but cannot find the specific user information for user U. Also, the attacker is unable to compute $N' = h(U_{ID}||PW_U)$ without knowing the U's ID and password. Therefore, the attacker cannot determine the server's private key $P = (U_{ID}||K^{-}_{NCC})$, just by using XOR operation, as proved in Section II-A. In addition, the attacker cannot discover P without knowing the long-term private key K^{-}_{NCC}. Thus, the attacker cannot forge valid login request or valid message based on the limited parameters. Another possible way the attacker can impersonate authentic subscribers is by deploying a reply attack which is discussed next.

2) Replay attacks: During the authentication phase, an attacker can merely delay the authentication process by using accidental or malicious interference [17], [18], [19]. The attacker cannot breach the security of the protocol by replaying the exchanged messages of the principals. Same is illustrated as below:

1) NCC simply discards any authentication request of U which is older than the most recent request. Thus intruder replaying such authentication requests gets easily knocked-out. However, when U's most recent authentication request is replayed, NCC will respond with the resynchronisation challenge using the updated T'_{ID} . If it was not U who sent the authentication request to NCC then U will ignore the resynchronisation challenge. Else, U repeats authentication with new updated T'_{ID}.
2) Next consider the case, when U requested authentication and is synchronised with NCC, however, U received a replayed resynchronisation challenge based upon old random value r instead of the replying message $(Grant, V_1, V_2, V_4)$. In this case, U will do that least by ignoring the challenge message. Subsequently, U will witness a time-out and re-sends an authentication request to NCC. Consequently, such replay attacks are incapable of producing any security threat against the proposed protocol.
3) According to previous analysis, the moment replay attack is stopped, NCC and U will authenticate each other. Therefore, the proposed protocol is resistant to both replaying authentication requests of U and replaying of NCC's responses.

3) Desynchronisation attacks: As elaborated in Section II-B, the LLC protocol is vulnerable to permanent DoS condition, when a replay attack is involved during the authentication phase. This weakness has been counterbalanced by introducing a resynchronisation phase. Once a desynchronisation is detected with previous T_{ID} by NCC, the new phase will ensure that both principals are resynchronised.

Moreover, if an attacker jams the replying message destined to U, using low power jamming technology [3], it will result in NCC and U getting temporarily desynchronised. Implying, NCC gets updated with T_{IDnew} while U remains at the old T_{ID}. On the very next authentication attempt by U, the NCC detects the desynchronisation condition and issues a resynchronisation challenge to achieve synchronisation over shared secrecy .

4) Stolen-verifier attacks: Let us turn our attention to the case of stolen-verifier attack where attacker somehow gets access to the verification table at the server and is thus able to steal the identities, passwords and the temporary identity. In spite of that, the attacker cannot obtain $P = (U_{ID}||K_{NCC}^{-})$ (which is used to generate a valid login request) without knowing the long-term private key K_{NCC}^{-} of the server. Thus storing no sensitive information in the verification table, enables our proposed protocol to withstand stolen-verifier attacks.

5) Smart card loss attacks: The LLC protocol is potentially susceptible to smart card loss attack, which occurs when an attacker steals the user's smart card and impersonates that user in order to login to the NCC just by guessing user's identity. However, the proposed protocol solves this issue by introducing an extra password. The weakness of the limited length of U_{ID} gets solved by combining U_{ID} with the extra password. Even if an attacker steals the U's identity, it cannot retrieve the server's private key without knowing the password. Thus, an attacker cannot impersonate a legal mobile user to login at the NCC.

V. Conclusion

The paper analysed the authentication and key agreement protocol proposed by Lee et al. (LLC protocol[1]) for mobile satellite communications. Our security analysis reveals that LLC protocol is inherently susceptible to both desynchronisation and impersonation attacks. Simple mechanism of triggering the re-login phase is not enough to make the LLC protocol immune to such attacks. In this direction, a detailed demonstration of these attacks was presented.

To reconcile security issues found in the LLC protocol, we proposed a new authentication and key agreement protocol. The crux of the proposed protocol is smart use of an extra password to improve the security strength and introduction of the resynchronisation phase to eliminate the desynchronisation condition. Security analysis of the new protocol verifies its effectiveness.

Acknowledgements

This work has been performed under European Union RESPONSE 5G (Grant No: 789658) and the Academy of Finland 6Genesis Flagship (Grant 318927) projects. The authors would also like to acknowledge the contribution of the COST Action CA15127 (RECODIS) and CA16226 (SHELD-ON) projects.

References

[1] C.-C. Lee, C.-T. Li, and R.-X. Chang, "A simple and efficient authentication scheme for mobile satellite communication systems," *International Journal of Satellite Communications and Networking*, vol. 30, no. 1, pp. 29–38, 2012.

[2] T. Saroj, G. S. Gaba, and S. K. Arora, "A survey on authentication schemes for satellite communications," pp. 431–435, 2016.

[3] 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.

[4] ——, "A mutual authentication protocol with resynchronisation capability for mobile satellite communications," *International Journal of Information Security and Privacy (IJISP)*, vol. 5, no. 1, pp. 33–49, 2011.

[5] T. Saroj and G. Gaba, "A lightweight authentication protocol based on ecc for satellite communication," *Pertanika Journal of Science & Technology*, vol. 25, no. 4, 2017.

[6] A. Jurcut, T. Coffey, and R. Dojen, "A novel security protocol attack detection logic with unique fault discovery capability for freshness attacks and interleaving session attacks," *IEEE Transactions on Dependable and Secure Computing*, 2017.

[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] E.-J. Yoon, K.-Y. Yoo, J.-W. Hong, S.-Y. Yoon, D.-I. Park, and M.-J. Choi, "An efficient and secure anonymous authentication scheme for mobile satellite communication systems," *EURASIP Journal on Wireless Communications and Networking*, no. 1, p. 86, 2011.

[9] 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.

[10] X. Wu, A. Zhang, J. Li, W. Zhao, and Y. Liu, "A lightweight authentication and key agreement scheme for mobile satellite communication systems," in *International Conference on Information Security and Cryptology*. Springer, 2016, pp. 187–204.

[11] 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, 2018.

[12] V. Pasca, A. Jurcut, T. Coffey, and R. Dojen, "Determining a parallel session attack on a key distribution protocol using a model checker," in *ACM Proceedings of the 6th International Conference on Advances in Mobile Computing and Multimedia (MoMM 08)*. ACM, 2008, pp. 150–155.

[13] A. Jurcut, T. Coffey, and R. Dojen, "On the prevention and detection of replay attacks using a logic-based verification tool," in *International Conference on Computer Networks, Computer Networks, Series: Communications in Computer and Information Science*. Springer International Publishing Switzerland, 2014, pp. 128–137.

[14] ——, "Symmetry in security protocol cryptographic messages a serious weakness exploitable by parallel session attacks," in *7th IEEE International Conference on Availability, Reliability and Security (ARES12)*. IEEE, 2012.

[15] A. Jurcut, M. Liyanage, J. Chen, C. Gyorodi, and J. He, "On the security verification of a short message service protocol," in *16th IEEE Wireless Communications and Networking Conference WCNC 2018*. IEEE, 2018.

[16] N. R. Potlapally, S. Ravi, A. Raghunathan, and N. K. Jha, "A Study of The Energy Consumption Characteristics of Cryptographic Algorithms and Security Protocols," *IEEE Transactions on mobile computing*, vol. 5, no. 2, pp. 128–143, 2006.

[17] A. Jurcut, T. Coffey, and R. Dojen, "Design requirements to counter parallel session attacks in security protocols," in *12th IEEE Annual Conference on Privacy, Security and Trust (PST14)*. IEEE, 2017, pp. 298–305.

[18] ——, "Design guidelines for security protocols to prevent replay parallel session attacks," *Computers & Security*, vol. 45, pp. 255–273, 2014.

[19] A. Jurcut, T. Coffey, R. Dojen, and R. Gyorodi, "Analysis of a key-establishment security protocol," *Journal of Computer Science and Control Systems*, pp. 42–47, 2008.