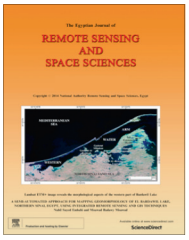
The Egyptian Journal of Remote Sensing and Space Sciences (2014) 17, 171–177

HOSTED BY
ELSEVIER

National Authority for Remote Sensing and Space Sciences
The Egyptian Journal of Remote Sensing and Space Sciences
www.elsevier.com/locate/ejrs
www.sciencedirect.com



RESEARCH PAPER

Verification of authentication protocols for mobile satellite communication systems

Reham Abdellatif Abouhogail

Electrical Quantities Metrology Dept., National Institute of Standards, Cairo, Egypt

Received 30 May 2013; revised 14 April 2014; accepted 7 July 2014
Available online 17 August 2014

KEYWORDS

Security;
Satellite communication systems;
Verification of protocols

Abstract In recent times, many protocols have been proposed to provide security for mobile satellite communication systems. Such protocols must be tested for their functional correctness before they are used in practice. Many security protocols for the mobile satellite communication system have been presented. This paper analyzes three of the most famous authentication protocols for mobile satellite communication system from the security viewpoint of data desynchronization attack. Based on strand spaces testing model, data desynchronization attacks on these protocols were tested and analyzed. Furthermore, improvements to overcome the security vulnerabilities of two protocols are mentioned.

1. Introduction

Nowadays, Mobile satellite communication systems have become one of the most important technologies. Security is a very important requirement in any system, especially in wireless communication systems. For a satellite user to communicate with other users, he must be authenticated first by the remote server. This paper is concerned with the authentication between mobile users and the remote server. Many mobile satellite communication systems have been proposed in recent years (Chang and Chang, 2005; Chen et al., 2009; Lasc et al., 2011; Eun-Jun et al., 2011; Lee et al., 2012; Cruickshank, 1996; Hwang et al., 2003). In the past, for more than 10 years the traditional satellite communication system that was used was the geostationary satellite. The geostationary satellite is located in geosynchronous equatorial orbit (GEO). Such a satellite returns to the same position in the sky after each sidereal day (Larson and Wertz, 1999). However, the quite far distance, exactly 22,300 miles, between the geostationary satellite and the earth resulted in a signal delay problem. Over the past 10 years, low-earth-orbit (LEO) satellite communication systems are used for establishing personal communication systems as shown in Fig. 1. This is due to their large broadcasting range and communication area, small attenuation of the signals and a shorter transmission delay (Chen et al., 2009). There have been many researches on the authentication protocols for the mobile satellite communication system. Some protocols are based on public key cryptosystems like Cruickshank (1996) which involves heavy computation costs. In 2003, Hwang et al. (2003) proposed an authentication protocol using symmetric encryption to reduce the complexity of computations. But both Cruichshank's protocol and Hwang et al.'s protocol

E-mail addresses: rehlatif@yahoo.co, rehlatif@gmail.com
Peer review under responsibility of National Authority for Remote Sensing and Space Sciences.

1110-9823 © 2014 Production and hosting by Elsevier B.V. on behalf of National Authority for Remote Sensing and Space Sciences.
http://dx.doi.org/10.1016/j.ejrs.2014.07.002

were not good for forward secrecy and efficiency. In 2005, Chang and Chang (2005) proposed a new protocol to solve the weakness found in previous protocols. They used the Diffie–Hellman key exchange (Diffie and Hellman, 1976). But from our analysis in Section 3 we found that Change et al.'s protocol (Chang and Chang, 2005) is susceptible to data desynchronization attack. In 2009, Chen et al. (2009) proposed a protocol based on discrete logarithm problem. It overcomes the complexity of public key infrastructure, reduces the hard computation from the mobile user and does not require sensitive verification table for the NCC. But this protocol is susceptible to data desynchronization attack as will be declared in Section 4.

In 2012, Lee et al. (2012) pointed out that Change et al.'s protocol (Chang and Chang, 2005) lacked user anonymity and impersonation attack. Lee et al. proposed a new protocol that has low computation cost. He claimed that his protocol avoids previous security flaws. But this protocol is also susceptible to data desynchronization attack as will be declared in Section 5.

In our paper we presented an analysis for the three most famous authentication protocols for mobile satellite communication systems (Chang and Chang, 2005; Chen et al., 2009; Lee et al., 2012). We are concerned with the data desynchronization attack in our analysis for these protocols. We chose this type of attack for our analysis because this attack depends on jamming which is considered the most dangerous enemy in space communication. The notations in Table 1 are used throughout this paper.

2. Definition of strand spaces model

A strand is a sequence of actions executed by a single principal in a single local session of a protocol (Guttman, 2011). We enrich strands to allow them to synchronize with the projection of the joint state that is local to the principal P executing the strand. The actions on a strand are defined into: message transmissions, message receptions, and state synchronization events. Strands are used for the protocol and communication behavior (Guttman, 2011). A strand is a (linearly ordered) sequence of nodes $n_1 \Rightarrow \ldots \Rightarrow n_j$, each of which represents either:

Transmission of some message $\mathrm{msg}(n_i) = t_i$, graphically $\bullet \xrightarrow{t_i}$;
Reception of some message $\mathrm{msg}(n_i) = t_i$, graphically $\xrightarrow{t_i} \bullet$;

A strand is a sequence of transmission and reception events local to a particular run of a principal. If this principal is honest, it is a regular strand. If it is dishonest, it is a penetrator strand (Guttman and Javier Thayer Fabrega, 2001). A bundle C is a causally well-founded collection of nodes and arrows of both kinds. In a bundle, when a strand receives a message m, there is a unique node transmitting m from which the message was immediately received. By contrast, when a strand transmits a message m, many strands (or none) may immediately receive m. The height of a strand in a bundle is the number of nodes on the strand that are in the bundle (Guttman and Javier Thayer Fabrega, 2001). In the following sections an analysis of the most three famous schemes for mobile satellite authentication protocols is presented. Suggestions are presented to improve these schemes. We assume in the three presented protocols that the LEO satellite is always a trust node. During the paper, transmission of messages from the *NCC* to the user U means transmission from the *NCC* to the LEO, then transmission from the LEO to U. Also the opposite is correct. Transmission of messages from the user U to the *NCC* means transmission from U to the LEO, then transmission from the LEO to the *NCC*.

Table 1 Notations.

Notation	Interpretation
U	The mobile user
LEO	Low earth orbit satellite
NCC	Network Control Centre
P_n	The penetrator
U_{ID}, LEO_{ID}	User/LEO permanent identity
T_{IDu}	User temporary identity
sk	Session key
$\mathrm{MAC}_k(.)$	Keyed one-way hash function using the key k
$(m)_k$	Symmetric key encryption function for a message m using the key k
$H(.)$	One-way hash function
K_s	User's long term secret key
$\oplus$	XOR function
$\|\|$	Concatenation operator
P	Authentication token
x	Long term private key
y	Long term public key
L	Large Prime number

3. Data desynchronization attack on the CC protocol

3.1. The CC protocol

Chang and Chang proposed a mutual authentication mechanism (Chang and Chang, 2005) hereafter referred to as the CC protocol. In the CC protocol, the authentication between the mobile user and the network control center (*NCC*) is within a LEO satellite communication system. Mobile users are interconnected directly through LEO satellite links, while communication between satellites and the *NCC* is managed by Gateways. The CC protocol is composed of three phases: registration, authentication and mobile update.

3.1.1. The registration phase

In this phase, U has to register at the system. U is assigned a permanent identity U_{ID}, the secret key K shared between U and *NCC*, a temporary identity T_{IDu} by the gateway, and the number of times (N) that the mobile user can access the service before an update phase is required.

3.1.2. The authentication phase

The authentication phase is performed by U and *NCC* before any communication. Note that *NCC* stores (U_{ID}, T_{IDu}, LEO_{ID}, $H^{N+1-(j-1)}(K||U_{ID}||T_{IDu})$, $(N-(j-1))$) for U, and U keeps (U_{ID}, T_{IDu}, K, $(N-(j-1))$) at this stage. The details are described as follows:

Step 1: the LEO sends the authentication request to U.

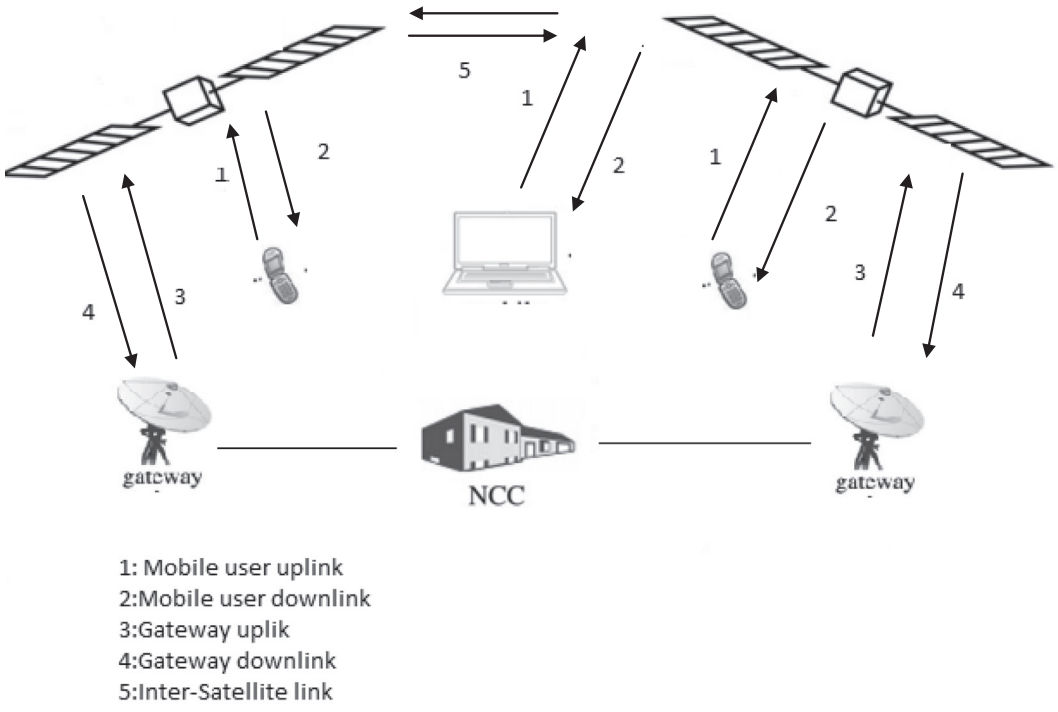


Figure 1 A LEO communication system.

Step 2: After getting the authentication request, U computes $H(P) \oplus R$ and $P \oplus H(R)$, where $P = H^{N-(j-1)}(K||U_{ID}||T_{IDu})$ and R is a one-time random number. Then U sends the computation results and T_{IDu} to LEO.
Step 3: Upon getting $H(P) \oplus R$, $P \oplus H(R)$, T_{IDu}, and LEO_{ID}, NCC.
Step 4: after getting $H(P) \oplus R$, $P \oplus H(R)$, T_{IDu}, and LEO_{ID}, NCC uses T_{IDu} to obtain the corresponding secret data using the lookup table. The authentication process between the NCC and U in this protocol is based on proving possession of the current value of the authentication token P. The NCC computes $R' = H^{N+1-(j-1)}(K||U_{ID}||T_{IDu}) \oplus (H(P) \oplus R)$, $P' = H(R') \oplus (P \oplus H(R))$, $H(P')$. If $H(P') = H^{N+1-(j-1)}(K||U_{ID}||T_{IDu})$, NCC first updates U_{ID}, T_{IDu}, LEO_{ID}, $H^{N+1-(j-1)}(K||U_{ID}||T_{IDu})$, $(N-j+1)$ to $(U_{ID}, T_{IDu}, LEO_{ID}, P', (N-j))$ and computes $H(R'||P')$. Then NCC sends $H(R'||P')$, LEO_{ID}, T_{IDu} and the grant message to LEO. Otherwise, NCC terminates the protocol.
Step 5: After receiving the messages sent from NCC, LEO forwards them to U.
Step 6: Upon getting $H(R'||P')$ and T_{IDu} forwarded by LEO, U checks whether $H(R||P)$ is equal to $H(R'||P')$. If it holds, U is convinced that NCC is legal and updates U_{ID}, T_{IDu}, K, $(N-(j-1)))$ to U_{ID}, T_{IDu}, K, $(N-j)$).

3.1.3. The mobile update phase

In the Nth authentication, U and NCC enter the mobile update phase. The NCC generates and issues a new session key K and a new temporary identity T_{IDu} for the user to be used for the next N authentications.

3.2. Data desynchronization attack on the CC protocol

In the authentication phase of the CC protocol, NCC and U update their shared secret information shown in Fig. 2 as follows: the NCC updates its shared secret before sending message 2. U updates its shared secret after receiving message 2. If message 2 of the authentication phase of the protocol does not reach the mobile user U because of the penetrator P_n for example while the NCC has already updated the secret shared data and U has not updated yet the NCC and U will operate at different levels in the hash chain. Thus, they operate asynchronously on the hash value that is used to authenticate each other.

3.3. Improvement of the CC Protocol

For the above flaw, the following improvements to the CC protocol to resist data desynchronization attacks are presented in Lasc et al. (2011). The mobile authentication phase is

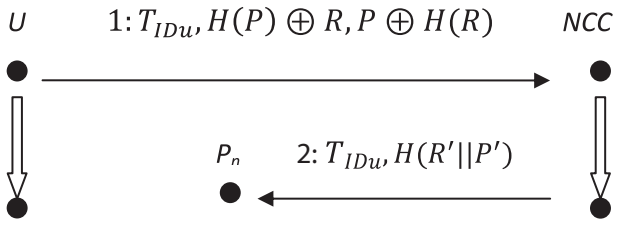


Figure 2 Data desynchronization attack on CC protocol.

changed into two modes. The selection between the two modes is according to whether there are legitimate authentication requests (messages 1, 2a) or illegitimate authentication requests (1, 2b) as shown in Fig. 3. After the mobile sends message 1 as in Fig. 3 the NCC computes R', P', and $H(P')$. The NCC then checks if $H(P') = H^{N+1-(j-1)}(K|U_{ID}|T_{IDu})$. If the check holds the NCC updates the corresponding entry in the lookup table to (U_{ID}, T_{IDu}, LEO_{ID}, P', $(N-j)$). In message 2a NCC sends T_{IDu}, $H(R'||P')$ to U by the LEO. U verifies the authenticity of NCC as before then if the NCC is authentic U updates the stored data. On the other hand, if the NCC receives an incorrect authentication request, it responds with a resynchronization challenge (messages 1, 2b). The message 2b contains $H(P_{NCC}||T_{IDu})$, $H(P_{NCC}) \oplus R_{NCC}$; where P_{NCC} is NCC's currently stored authentication token, R_{NCC} is a newly generated random number. Once receiving message 2b, the mobile user compares the received P_{NCC} with the remaining P values in the chain. If a match is found the mobile user advances to the resynchronization phase. If no match is found the resynchronization request is considered illegitimate and the same authentication request is re-sent.

4. Data desynchronization attack on the CLC protocol

4.1. The CLC protocol

Chen et al. proposed an authentication mechanism for mobile satellite communication systems in Chen et al. (2009). The CLC protocol is as follows:

4.1.1. The initialization phase

The NCC chooses a large prime L and a generator g of the multiplicative group Z_L^* with order q (q is a large prime factor of $L-1$). The NCC selects a long-term private key x, $1 \leqslant x < q$, and the corresponding public-key is $y = g^x \, mod \, L$.

4.1.2. The registration phase

$NCC \rightarrow U: \; U_{ID}, T_{IDu}, K_s$

The NCC assigns to each mobile user U a permanent identity U_{ID}, an initial temporary identity T_{ID}, and selects a random number K; $1 \leqslant K < q$, and computes:

$r = g^k \, mod \, L$.
$s = H(U_{ID})x + Kr^{-1}$ mod q, and generates the user secret key K_s.
$K_s = H(U_{ID}, K)$.

The NCC stores U_{ID}, T_{IDu}, r, s into NCC's verification table. The NCC stores U_{ID}, T_{IDu}, K_s in user's smart card.

1. U-> NCC:$T_{IDu}, H(P) \oplus R, P \oplus H(R)$
2. a. NCC->U: $T_{IDu}, H(R^{'}||P^{'})$.
2. b. NCC-> U:$T_{IDu}, H(P_{NCC}||T_{IDu}), H(P_{NCC}) \oplus R_{NCC}$.

Figure 3 Improvement of CC protocol.

4.1.3. The authentication phase

During this phase, the two parties U and NCC use the shared secrets established during the initialization and registration phases to prove their identity to each other.

$U \rightarrow LEO : T_{IDu}, c$

U calculates the session key $sk = \mathrm{H}(K_s, T_{IDu})$; T_{IDu} is refreshed after one successful login, and calculates $c = \mathrm{MAC}_{Ks}(U_{ID}, T_{IDu}, sk)$.

$LEO \rightarrow NCC$:T_{IDu}, c, LEO_{ID}; the LEO appends its identity LEO_{ID} upon receiving the authentication message.

$NCC \rightarrow LEO : (T_{IDu}, T_{IDunew})_{sk}, LEO_{ID}$.

The NCC checks the identity of the LEO upon receiving the authentication message and does the following operations:

a) Find the corresponding information $\{U_{ID}, r, s\}$ associated with T_{ID} by looking up the verification table.
b) Validates K using the equation $s = h(U_{ID})x + Kr^{-1}$ mod q.
c) Computes the possible user secret key $K_s' = h(U_{ID}, K)$, and the possible session key $sk' = h(K_u, T_{IDu})$.
d) Compute $c' = MAC_{K_s'}(U_{ID}, T_{IDu}, sk')$, then check if $c' = c$. If they match the user is authenticated and the session key is confirmed otherwise the authentication request is rejected.
e) Generate T_{IDnew} and update the verification table, then return $\{(T_{IDu}, T_{IDunew})_{sk}, LEO_{ID}\}$ to the LEO.

$LEO \rightarrow U : (T_{IDu}, T_{IDunew})_{sk}$

Once the user receives the data, he can verify whether the decrypted T_{IDu} is identical to the stored T_{IDu}. If they are identical, the mobile user U will authenticate the NCC and will update its temporary identity to T_{IDunew} for the next authentication request. After authentication, the two parties will encrypt their data with the session key sk. T_{IDunew} will be used for the next authentication request.

4.2. Data desynchronization attack on the CLC protocol

In (Chen et al., 2009), Chen et al. deemed that their proposed authentication protocol could guarantee data confidentiality, mutual authentication, and user's privacy. However, the CLC protocol cannot offer any protection against data synchronization attack: a penetrator P_n can easily force an honest tag to fall out of synchronization with the reader so that it can no longer authenticate itself successfully. The data desynchronization attack on the CLC protocol can be described in Fig. 4. In the data desynchronization attack, the penetrator P_n easily destroys the synchronization of T_{IDu} between the NCC and the mobile user U. P_n can intercept the message $(T_{IDu}, T_{IDunew})_{sk}$ from NCC to U. Therefore, NCC will refresh the user temporary identity T_{IDu} while the user U will not do it. Thus, the shared temporary identity between U and NCC is not identical. After a successful data desynchronization attack, U and NCC will share different secrets and U and NCC will not authorize each other mutually. Thus, the availability of the CLC protocol is destroyed.

4.3. Improvement of the CLC protocol

To solve the data desynchronization attack problem described above, the *NCC* must keep the old temporary identity T_{IDu} from the last successful authentication. When *U* and *NCC* fail to authenticate because of data desynchronization attack, it will use the previous stored temporary identity T_{IDu} as shown in Fig. 5:

Two extra messages must be added to solve the problem of data desynchronization attack for the CLC protocol. When the user *U* does not receive any messages from the *NCC* because of jamming, he will resend message No. 2 with a new *c*, which is called *c*′ as shown in step 4 in Fig. 5. The *NCC* must compare the received T_{IDU} with the new one and the previous one. If the received T_{IDU} is equal to the last stored one, the *NCC* will send message No. 3 again with the last generated T_{IDnew} to save time and computation overhead.

5. Data desynchronization attack on the Lee et al. protocol

5.1. The Lee et al. protocol

Lee et al. proposed an authentication scheme for satellite communication systems (Lee et al., 2012). Their scheme consists of three phases: Registration phase, Login phase, and authentication phase.

5.1.1. The registration phase

The steps of registration are as follows:

Step R1. $U \Rightarrow NCC{:}U_{ID}$

Step R2. $NCC \Rightarrow U{:}T_{IDu}, R, k$

$$P = h(U_{ID}||x)$$

$$R = P \oplus h(U_{ID}||k)$$

where *k* is a secret random number and *x* is a long-term private key generated by the NCC. Then, the *NCC* creates the temporary identity T_{IDu} and stores U_{ID} and T_{IDu} in the verification table. After that, the *NCC* issues a smart card and sends it to *U*. The smart card contents are: $\{T_{IDu}, R, k, H()\}$.

5.1.2. Login phase

The steps of login phase are as follows:

Step L1. *U* inserts his smart card into a smart card reader and inputs his identity U_{ID}.

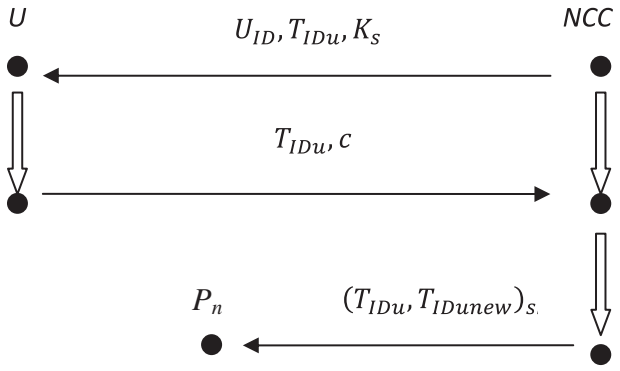


Figure 4 Data desynchronization attack on CLC protocol.

1. $NCC \rightarrow U: U_{ID}, T_{IDu}, K_s$
2. $U \rightarrow NCC: T_{IDu}, c$
3. $NCC \rightarrow U: (T_{IDu}, T_{IDunew})_{sk}$
4. $U \rightarrow NCC: T_{IDu}, c'$
5. $NCC \rightarrow U: (T_{IDu}, T_{IDunew})_{sk}$

Figure 5 Improvement of CLC protocol.

Then the smart card chooses a secret random number *r* to calculate

$$P' = R \oplus h(U_{ID}||k)$$

$$Q = P' \oplus r, S = H(U_{ID}||r)$$

Step L2. $U \rightarrow NCC{:}Q, S, T_{ID}$

Finally, *U* sends the login message $\{Q, S, T_{ID}\}$ to the *NCC*.

5.1.3. Authentication phase

Step A1. After receiving the authentication request from *U*, the *NCC* obtains U_{ID} according to T_{ID} and computes

$$P = h(U_{ID}||x)$$

$$r' = Q \oplus P$$

$$S' = h(U_{ID}||r')$$

a) The *NCC* checks if $S' = S$. If they match the user is authenticated. Otherwise, the authentication request is rejected.

Step A2. Once the mobile user is authenticated, the *NCC* chooses a secret random *t* to compute

$$V_1 = P \oplus t$$

$$V_3 = h(r'||t)$$

Then, the NCC generates the new temporary identity T_{IDnew} to calculate $V_4 = V_3 \oplus T_{IDnew}$ and replaces T_{ID} with T_{IDnew} in the verification table. The NCC computes $V_2 = h(P||r'||t||V_4)$.

Step A3. $NCC \rightarrow U{:}V_1, V_2, V_4$

After receiving these messages, *U* computes

$$t' = V_1 \oplus P'$$

$$V_2' = h(P'||r||t'||V_4)$$

U checks if V_2' is the same as V_2. If they are, then *U* authenticates the NCC successfully. Then *U* computes

$$V_3' = h(r||t')$$

$$T_{IDnew} = V_3' \oplus V_4$$

U replaces T_{ID} with T_{IDnew} in the user's smart card and computes the session key *sk*.

$$sk = h(U_{ID}||r||t'||P')$$

sk is used for secret communication.

5.2. The data desynchronization attack on the Lee et al. protocol

From Fig. 6, it is evident that the Lee et al. Protocol cannot resist the data desynchronization attack. The penetrator P_n can intercept the message V_1, V_2, V_4 from the *NCC* to the LEO. Therefore the *NCC* has refreshed the temporary identity T_{IDnew} before *U* received the updated message.

5.3. Improvement of the Lee protocol

The *NCC* must store the old T_{ID}. If the replying message from the *NCC* is lost. *U* will re-login and the *NCC* uses the old T_{ID} of *U*. The desynchronization problem of the Lee protocol is solved by extending NCC's storage time for T_{ID} until *U* enters the next authentication phase using the expected value T_{IDnew} which proves that *U* has successfully updated its secrets. Storing the value T_{ID} used in the previous authentication session allows *NCC* to get the correct U_{ID} when *U* makes re-login. So Fig. 6 should be modified into Fig. 7.

As shown in Fig. 7, two extra steps are added. After the user *U* fails to receive V_1, V_2, V_4 due to the effect of the penetrator, he must re-login with a new *Q* and *S* as shown in step 5 (Fig. 7).

6. Performance and security analysis

The three presented protocols satisfy mutual authentication. But in the case of jamming or data desynchronization attack the mutual authentication will not be achieved. The improvements to the three presented protocols make the mutual authentication return after finishing of the jamming. So the connection between the user *U* and the *NCC* will not be lost forever. Two extra messages are needed for the presented three protocols and storing of the previous authentication token is required in the case of the CC's protocol. In the case of the CLC's and Lee et al.'s protocols, storing of the previous user's identity is required.

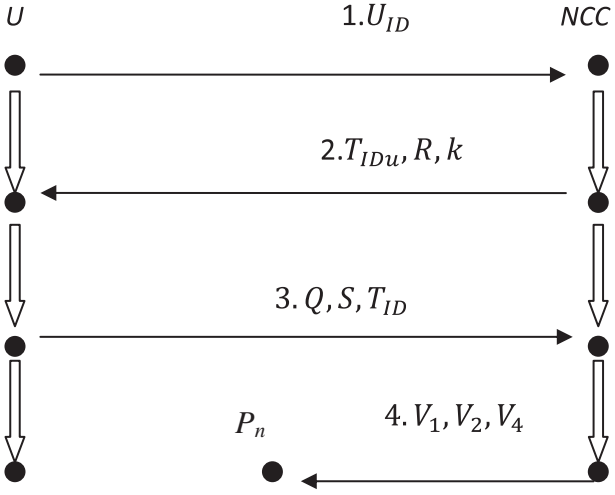


Figure 6 Data desynchronization attack on Lee protocol.

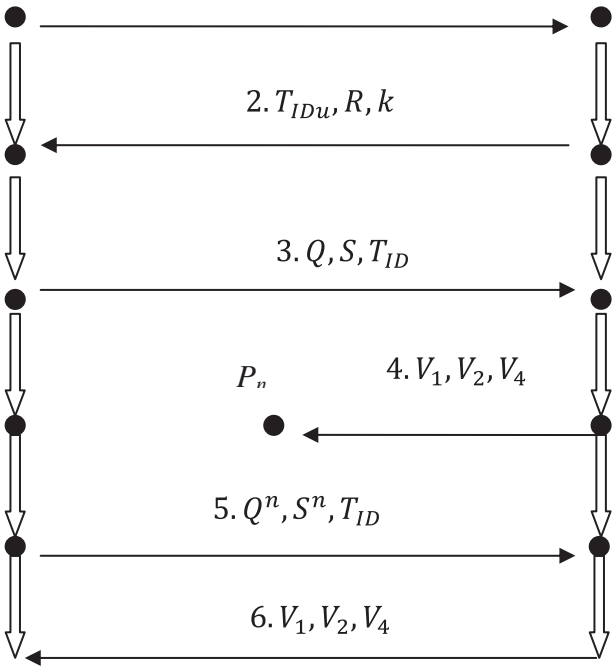


Figure 7 Improvement of CLC protocol.

7. Future study

Our paper makes a security analysis to authentication protocols for mobile satellite communication systems. The paper concentrates on special type of attack. It is a data desynchronization attack. As a future work we can investigate further attacks and try to give other suitable solutions to these other types of attacks. In our paper, we used Strand Spaces verification tool, which is considered the most suitable verification method for data desynchronization attack. We can test the other attacks using other formal analysis methods which are more suitable to other types of attacks.

8. Conclusion

This paper discusses data desynchronization attacks on some authentication protocol for mobile satellite communication proposed in Chang and Chang (2005), Chen et al. (2009), and Lee et al. (2012). In the strand spaces model, we found that the three presented protocols were vulnerable to data desynchronization attacks. In addition, improvements to overcome the vulnerabilities of two protocols were given.

References

Chang, Y.F., Chang, C.C., 2005. An efficient authentication protocol for mobile satellite communication systems. ACM SIGOPS Oper. Syst. Rev. 39 (1), 70–84.

Chen, T.Z., Lee, W.B., Chen, H.B., 2009. A self-verification authentication mechanism for mobile satellite communication systems. Comput. Elect. Eng. 35 (1), 41–48.

Cruickshank, H.S., 1996. A security system for satellite networks. IEEE Satellite System Mobile Communication Navigation, UK, 187–190.

Diffie, W., Hellman, M., 1976. New directions in cryptography. IEEE Trans. Inf. Theory 22 (6), 644–654.
Eun-Jun, Y., Kee-Young, Y., Jeong-Woo, H., Sang-Yoon, Y., Dong, I., Myung-Jin, C., 2011. An efficient and secure anonymous authentication scheme for mobile satellite communication systems. Wireless Communications and Networking, 68.
Guttman, J.D., 2011. State and Progress in Strand Spaces: Proving Fair Exchange. National Science Foundation.
Guttman, J., Javier Thayer Fabrega, F., 2001. Authentication tests and the structure of bundles. Theoret. Comput. Sci.
Hwang, M.S., Yang, C.C., Shiu, C.Y., 2003. An authentication scheme for mobile satellite communication systems. ACM SIGOPS Oper. Syst. Rev. 37 (4), 42–47.
Larson, W., Wertz, J.R., 1999. Space Mission Analysis and Design. Microcosm Press and Kluwer Academic Publisher.
Lasc, I., Dojen, R., Coffey, T., 2011. A mutual authentication protocol with resynchronisation capability for mobile satellite communications. Int. J. Inf. Sec. Priv. 5 (1), 33–49.
Lasc, I., Dojen, R., Coffey, T., 2011. Countering jamming attacks against an authentication and key agreement protocol for mobile satellite communications. Comput. Electr. Eng., 160–168.
Lee, C.-C., Li, C.-T., Chang, R.-X., 2012. A simple and efficient authentication scheme for mobile satellite communication systems. Int. J. Satell. Commun. Netw., 29–38.