A Lightweight Authentication and Key Sharing Protocol for Satellite Communication

Abid Murtaza[1*], Tongge Xu[2], Syed Jahanzeb Hussain Pirzada[2], Liu Jianwei[2]

[1] School of Electronic and Information Engineering, Beihang University, Beijing, China.
[2] School of Cyber Science and Technology, Beihang University, Beijing, China.

* Corresponding author. Tel.: 13121917533; email: Xutg@buaa.edu.cn
Manuscript submitted May 31, 2019; accepted August 18, 2019.
doi: 10.17706/ijcce.2020.9.1.46-53

Abstract: Due to various critical aspects of information security, using Perfect Forward Secrecy (PFS) in many real-world communication applications is advantageous. Satellite communication is also among those applications where communication security is vital; hence, PFS is valuable. Security protocols are widely used to exchange session keys after authentication in many applications. However, the majority of existing security protocols use cryptographic algorithms (e.g., asymmetric/symmetric encryption and hash function to ensure the security of the protocol, which results in slow processing. Also, most of them require the exchange of several messages, which result in additional delay in communication and waste of bandwidth in terms of exchange of additional data. This paper presents a new authentication and key sharing protocol, which is much simpler and lighter hence faster than other protocols and still very secure because of having security comparable to that of theoretically secure One Time Pad.

Key words: Authentication, key-exchange, protocol, satellite communication.

1. Introduction

Satellites have been widely used around the world from decades to provide many useful services, such as communication, navigation, remote sensing, and weather monitoring, etc. for commercial, governmental, and military users. The communication of satellite operator/controller with satellite can be broadly divided into two main categories. First is the critical TT&C (Telemetry Tracking & Command) communication. Second is the payload data transfer communication such as data/images captured by earth observation satellite etc. For both of these communications, security is critical, and compromise in security could have serious consequences. Hence, different security requirements have been classified for space missions [1].

For communication security, user authentication & data confidentiality are among the two most critical required services. Authentication protocols are used in many applications, including satellites, to authenticate a user's authority to access system's resources. On the other hand, symmetric key encryption algorithms are proffered to provide data confidentiality in satellite communication due to their lower computational cost and faster speed. However, in cryptography, it is not recommended to use the same symmetric key for longer duration as it may reveal the useful information about the key being used. Therefore, the concept of use of a new key for every session called forward secrecy or perfect forward secrecy (PFS) is widely practiced in many applications. The major strength that PFS provides to communication security is that, in case of a compromise of one key for any reason, only one session is compromised (for which the key was used), while rest of communication remains secure.

Protocols are generally used to share these new session keys between sender and receiver for PFS as a secondary service after authentication in many modern applications. However, in most of available protocols, exchange of several messages is required for sharing of the new key, which consumes not only extra time due to the involvement of cryptographic algorithms (e.g., Public key algorithms, hash function, etc.), but also wastes critical bandwidth (sending extra data in these messages). In particular, the key sharing delay is vital in satellite communication, especially for LEO satellites, where visibility period lasts for few minutes only (5 to 10 minutes on average) and big data need to be transferred in this short duration generally (e.g., stored earth images). Hence we can see that existing security protocols may not be used efficiently for PFS in satellite communication.

We proposed a new and efficient authentication protocol for satellite communication where for the first time, a reliable mutual authentication mechanism is used, and after that, the first key is exchanged. Then for every next message, the same key is used for both authentication and sharing of new key. In this way, the protocol removes unnecessary computation to provide faster and secure communication with PFS.

The remaining paper is arranged as follow; in Section 2, related work is briefly reviewed. The protocol is proposed in Section 3. In Section 4, security analysis is presented. In Section 5, analyses against well-known attacks is discussed. Section 6 discusses the efficiencies of the protocol. Section 7 concludes the paper.

2. Related Work

A large number of authentication and key exchange protocols have been proposed to be used for different applications. Article [2] provides a brief survey of different types of well-known authentication protocols. For satellite application, authors in [3] first presented an authentication system for satellite networks in 1996 using a combination of Public key cryptography (PKC) and secret key cryptography, which is considered inefficient due to higher computational cost. Then an authentication protocol was proposed based on secret key cryptography by authors in [4]. However, their scheme later proved insecure and inefficient because of being vulnerable against the stolen-verifier attack and lacking perfect forward secrecy by [5]. Authors in [5] also proposed a hash- chain-based authentication which uses Diffie-Hellman key exchange for the new session key generation as an improvement. However, their scheme is suspected to impersonation attacks and also user's privacy is not kept confidential. A self-verification authentication protocol (CLC) was later proposed by [6], which claimed to eliminate PKI complexity. Based on CLC, later few more schemes were proposed [7]–[9]. The article [10] provides a survey of protocols proposed for satellite applications and highlights the pros & cons of them.

Space Information Network (SIN) is a concept of networking of satellites to provide global availability of services for everyone [11]. There are some authentication protocols proposed for SIN, such as [12]–[14]. More recently, few more authentication and key exchange protocols have been proposed for SIN [15]–[17]. However, majority of existing protocols may not be used for PFS in satellite communication because they are inefficient either due to complex computations, as well as several messages need to be exchanged for session key establishment, or otherwise they are not secure or vulnerable to different attacks.

3. Proposed Protocol

Based on the background discussed in Sections 1 & 2, the design goals of our protocol are following,

1. Strong mutual authentication is required between satellite and user/operator.
2. Should have minimum use of the complex cryptographic algorithms for being faster.
3. The number of messages exchanged and data to be sent should be least to be bandwidth efficient.
4. The protocol should be very secure.
5. PFS is desired.

There are two phases of the proposed protocol. Phase 1 is for the first time strong mutual authentication and key establishment, while phase 2 is for sharing of new key together with encrypted data. We will use the common terminologies of Alice, Bob, and Trudy instead of a satellite, controller, and attacker.

3.1. Phase 1

Phase 1 is shown in Fig. 1 below.

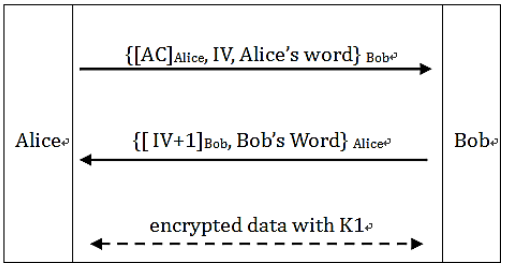

Fig. 1. Phase 1 of the proposed protocol.

Here,

AC = Alice certificate, IV = Initialization vector of 128 bits

Alice's word = 128 bits Alice's word for the key

Bob's word = 128 bits Bob's word for the key

$$K1 = \text{Alice's word} \oplus \text{Bob's word} \quad (1)$$

Here $\oplus$ is Exclusive-OR operation (XOR). When Alice wants to communicate with Bob, she will send his signed certificate with, a random IV and Alice's word by encrypting it with the public key of Bob as shown in message 1 of Fig. 1. Upon receiving this message, Bob can decrypt this message using his private key and can extract the data from Alice, after verification of Alice's certificate through the public key of Alice. Bob can send his word together with adding 1 to IV and signed with his private key to Alice using Alice's public key, as shown in message 2 of Fig. 1. If Bob fails to verify Alice through his certificate, he will not reply and discard the message. After verification of certificate correctly, Bob will store this message from Alice for the next time use of this phase (if needed to be used). Next time if Alice wants to execute phase 1 with Bob, he will send the 1st message of protocol again to Bob, this first message should be different from the last message stored by Bob, because Bob will discard the new request if it is exactly same as previous to avoid a replay attack or denial of service attack.

Upon receiving the message from Bob, Alice can decrypt the message contents using his private key and verify that the message is from Bob (IV+1) because only Bob knows correct IV and also IV+1 will only be decrypted by Bob's Public key. From Alice's and Bob's word's, both Alice and Bob can drive Key K1, as shown in equation 1. Now the first phase is completed, and Alice/Bob has authenticated each other, and the first key K1 is exchanged.

3.2. Phase 2

Phase 2 is for session/message 2 to N and shown in Fig. 2

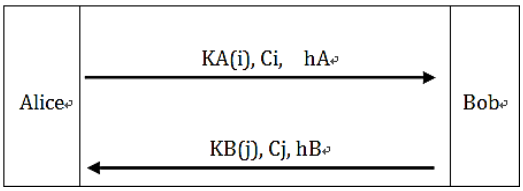

Fig. 2. Phase 2 of the proposed protocol.

Here,

$$KA(i) = K1 \oplus (IV + i) \oplus Ki \text{ (128 bits New Key)} \quad (2)$$

$$KB(j) = K1 \oplus (IV + j) \oplus Kj \text{ (128 bits New Key)} \quad (3)$$

$$Ci = E((\text{plaintext})i, Ki) \quad (4)$$

$$Cj = E((\text{plaintext})j, Kj) \quad (5)$$

$$hA = \text{hash}(Ci, Ki, IV + i) \quad (6)$$

$$hB = \text{hash}(Cj, Kj, IV + j) \quad (7)$$

In the second phase, for messages i/j=2 to N, both Alice and Bob can encrypt their plaintext data using the new key Ki/Kj respectively and send ciphertext Ci/Cj (equations 4 & 5) together with KA(i)/KB(j) (equations 2 and 3) as shown in Fig. 2. Ki and Kj must always be new (no repetition). Upon receiving the message, Alice/Bob can XOR KA (i)/KB (j) with K1 and (IV+i) or (IV+j) to extract Ki/Kj as shown in Eq. 8 & 9

$$Ki = KA(i) \oplus K1 \oplus (IV + i) \quad (8)$$

$$Kj = KB(j) \oplus K1 \oplus (IV + j) \quad (9)$$

After extracting Ki/Kj, the receiver will verify the hash to check the integrity of the message, as shown in equations 6 and 7. If the integrity is not compromised, the receiver can decrypt ciphertext Ci/Cj with Ki/Kj to get (plaintext)i/(plaintext)j as shown below.

$$(\text{Plaintext})i = D(Ci, Ki)$$

$$(\text{Plaintext})j = D(Cj, Kj)$$

Hash is used for verification of the integrity of message contents. If the generated hash is not the same as received hash, the receiver can discard the message by considering it a forged message. Also, the receiver will request the sender to resend the previous message by identifying to sender about the counter (IV+i/IV+j) of the last message received correctly. This will prevent desynchronization of the counter (IV+i/IV+j) at both ends.

4. Security Analysis of the Protocol

Security of the protocol can also be divided into two phases as the protocol itself, as the attacker has the options to attack the first phase or the second phase of protocol. As in the first phase, data is sent by encrypting with the public key of receiver, hence for an attacker; it is not possible to get the key K1 without either obtaining the private key of the receiver or by breaking the asymmetric crypto. Hence, the security of this phase of the protocol is as equal to the security of public key cryptographic algorithm used (e.g., RCA or ECC). It is important to note here that this first phase of the protocol is supposed to be used either only once or less frequently at least. Therefore, once Alice and Bob execute this phase of protocol securely, then the security of protocol or in other words, the security of all the future keys shared between Alice and Bob will only depend on the security of the second phase of the protocol.

The second phase of the protocol is extremely secure despite being simple. To understand the security of this phase, it is reasonable to quickly recall the operation of theoretically provably secure encryption algorithm OTP. In OTP, plaintext bits are XORed with the secret key bits of the same size to produce the ciphertext. The secret key in OTP is used only once and hence there is no way for an attacker to be sure about plaintext from the ciphertext because the same ciphertext can be achieved by the 2^n possible unique permutations of Plaintext and Key, where n is the number of bits in plaintext/key.

Our protocol uses XOR of three numbers (K1, Ki or Kj and (IV+i) or (IV+j)). The number of unique permutations of three numbers of n bits each that can produce the same ciphertext is 2^{2n}. In proposed

protocol, all three numbers of 128 bits each are XORed to produce $KA_{(i)}$ or $KB_{(j)}$. Therefore, there exist, $2^{2*128} = 2^{256}$ possible unique permutations of these three numbers. This is itself a huge number. But more importantly, K1 and (IV+i)/(IV+j) is secret (shared in 1st phase); therefore, guessing Ki/Kj correctly from $KA_{(i)}$ or $KB_{(j)}$ is impossible for an attacker. Because 2^{256} wrong combinations of K1, Ki/Kj and (IV+i)/(IV+j) can produce the same $KA_{(i)}$ or $KB_{(j)}$, and the attacker has no further information to be sure about K1 or Ki/Kj or (IV+i)/(IV+j), hence this second phase of the protocol is extremely secure.

Although in our protocol Alice and Bob will use different counter number (i and j) according to the number of messages sent by themselves, however, it is possible that the value of 'i' and 'j' will be same at any particular instant of communication. Such as if both are sending their 2nd message then in that case, the value IV+i and IV+j will become same (IV+2), in addition to the same K1, hence an attacker may try to XOR $KA_{(i)}$ and $KB_{(j)}$ to get some key information. However, from equations 2 and 3 we can see that in this case,

$$KA(i) \oplus KB(j) = Ki \oplus Kj$$

where Ki and Kj are unknown to the attacker and this is similar to OTP. So we can say that the minimum security of our protocol at the second phase will be equivalent to OTP.

5. Protection against Well-Known Attacks

In this section, we will consider the security of our protocol against two well-known attacks.

5.1. Man in the Middle Attack

In the first phase of protocol, all the messages are encrypted with the public keys, hence if Trudy sits in the middle of Alice and Bob for man in the middle attack in this phase, he can neither read nor modify the contents of the messages (i.e., IV, Alice's word or Bob's word, etc.) due to PKC. Now suppose If Trudy blocks the message from Alice and instead sends a message to Bob with his certificate instead of Alice certificate, then it will not be verified at Bob's end. Similarly, suppose Trudy blocks the message from Bob to Alice and instead wants to send his message to Alice it will not be verified because IV+1 will not be according to IV send by Alice to Bob. So, in either case, Trudy cannot perform a man in the middle attack in phase 1 of the protocol. Similarly in the second phase, if Trudy sends his $K_{T(i)}$ to Bob instead of Alice's $K_{A(i)}$, then firstly ciphertext Ci will not be decrypted correctly (eq. 8 & 9), secondly, Bob will not use Trudy's key, $K_{T(i)}$, for sending his message, instead he will use K1, and his new key Kj. Therefore Trudy cannot get any information about Kj or K1 and therefore about plaintext as he does not have any information.

5.2. Replay Attack

Suppose Trudy stores the message of Alice during the first phase of the protocol to later replay that, then Trudy will not get any benefit because Bob will discard this message by considering it a replay message as mentioned in the description of protocol phase 1 in section 3. So, there is no effect/benefit of this replay for Trudy. Similarly for phase 2 replaying will not provide any benefit to Trudy because the hash is generated by ciphertext (Ci/Cj), new session key (Ki/K_j) and the counter (IV+i/IV+j). So replaying will not benefit Trudy as the hash generated by Alice/Bob will not match with the hash received. Hence receiver will discard the message by considering it a forged message. Therefore replay attack is ineffective on our proposed protocol.

6. Computational Efficiency of Protocol

As the design objective of the protocol was to make it simple and fast despite being secure, hence we can see that firstly only two messages are required for mutual authentication & establishment of key between the satellite and a legitimate user in the first phase of the protocol. While in second phase no separate

message is required for session key exchange like many existing protocols, instead the new session key can be sent together with ciphertext, which shows that the protocol uses the least number of messages exchanged for PFS hence the delay is also minimum. Also, protocol is bandwidth efficient because, only the data equal to the length of key, i.e., 128 bits and hash need to be sent additionally to ciphertext in 2nd phase.

The first phase of protocol uses public key cryptography so its time will be comparable to that of other protocols using PKC. However, the advantage of the proposed protocol is that this first phase is required to be executed either only once or less frequently. Hence in the overall communication, processing delay will be much lesser than that of other protocols for PFS. Secondly, for the second phase of protocol only one XOR operation is used for computation of $KA_{(i)}$ or $KB_{(j)}$, (equations 2 & 3)which generally consumes only one clock cycle on many processors (e.g., FPGA). Therefore, this phase of the protocol is faster than all other existing protocols.

7. Conclusion

In this paper, we have presented a novel authentication and key sharing protocol for satellite communication for forward secrecy. The proposed protocol is not only efficient in terms of computational complexity, speed, and bandwidth, but at the same time, the security of the protocol is almost similar to that of the one-time pad. Hence the proposed protocol is suitable to be used in satellite application for PFS.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Abid Murtaza conducted the research and wrote the paper; Tongge Xu has reviewed the research work and the paper; Syed Jahanzeb Hussain Pirzada provided assistance in research and editing of the paper; Liu Jianwei provided the idea of the work and overall supervision; all authors had approved the final version

References

[1] CCSDS. (2019). The application of security to CCSDS protocols. *Informational Rep. (Green Book)*, 350.

[2] Prakash, A., & Kumar, U. (2018, June). Authentication protocols and techniques: A survey. *Int. J. Comput. Sci. Eng., 6(6)*, 1014–1020.

[3] Cruickshank, H. S. (1996). A security system for satellite networks. *Proceedings of 1996 IET International Conference on Satellite Systems for Mobile Communications and Navigation* (pp. 187–190).

[4] Hwang, M., Yang, C., & Shiu, C. (2003). An authentication scheme for mobile satellite communication systems. *ACM SIGOPS Oper. Syst. Rev.*, 42–47.

[5] Chang, Y.-F., & Chang, C.-C. (2005). An efficient authentication protocol for mobile communication system. *ACM SIGOPS Oper. Syst. Rev.*, 70–81.

[6] Chen, T. H., Lee, W. B., & Chen, H. B. (2009). A self-verification authentication mechanism for mobile satellite communication systems. *Comput. Electr. Eng.*, *35(1),* 41–48.

[7] Lasc, I., Dojen, R., & Coffey, T. (2011). Countering jamming attacks against an authentication and key agreement protocol for mobile satellite communications. *Comput. Electr. Eng., 37(2)*, 160–168.

[8] Chang, R.-X., Lee, C.-C., & Li, C.-T. (2012). A simple and efficient authentication scheme for mobile satellite communication system. *Int. J. Satell. Commun. Netw., 30*, 29–38.

[9] Yoon, E.-J., Yoo, K.-Y., Hong, J.-W., Yoon, S.-Y., Park, D.-I., & Choi, M.-J. (2011). An efficient and secure anonymous authentication scheme for mobile satellite communication systems. *EURASIP J. Wirel. Commun. Netw.*, *2011(1)*, 86–95.

[10] Saroj, T., Gaba, G. S., & Arora, S. K. (2016). A survey on authentication schemes for satellite communications. *Int. J. Comput. Technol. Appl.*, *9(38)*, 431–435.

[11] Murtaza, A., & Jianwei, L. (2019). Multipurpose IP-based space air-ground information network. *J. Phys. Conf. Ser.*, *1187(4042044)*.

[12] Chang, C. C., Cheng, T. F., & Wu, H. L. (2014). An authentication and key agreement protocol for satellite communications. *Int. J. Commun. Syst.*, *27(10)*, 1994–2006.

[13] Li, D., Liu, J., & Liu, W. (2016). Secure and anonymous data transmission system for cluster organised space Information Network. *Proceedings of IEEE Int. Conf. Smart Cloud, SmartCloud 2016* (pp. 228–233).

[14] Murtaza, A., & Jianwei, L. (2018). A simple, secure and efficient authentication protocol for real-time earth observation through satellite. *Proceedings of IEEE 15th International Bhurban Conference on Applied Sciences and Technology (IBCAST)* (pp. 822–830).

[15] Chen, C., Wang, F., Chen, C., Wu, T., & Wang, E. (2019). Reconsidering a lightweight anonymous authentication protocol. *J. Chinese Inst. Eng., 42(1),* 9–14.

[16] Yang, Q., Xue, K., Xu, J., Wang, J., Li, F., & Yu, N. (2019). AnFRA : Anonymous and fast roaming authentication for space information network. *IEEE Trans. Inf. Forensics Secur., 14(2)*, 486–497.

[17] Xue, K., Meng, W., Li, S., Wei, D. S. L., Zhou, H., & Yu, N. (2019). A secure and efficient access and handover authentication protocol for internet of things in space information networks. *IEEE Internet Things J.*

Abid Murtaza was born in Karachi, Pakistan. In 2010, he has received the M.Sc. electronics degree from the University of Karachi, Karachi, Pakistan. He is working with Pakistan's National space agency SUPARCO since 2010. He is currently working towards the Ph.D. degree in space technology applications at Beihang University, Beijing, China. His research interests include space information network, information security, cryptography, security protocols, and satellite communication

Tongge Xu graduated from Beijing University of Aeronautics and Astronautics in 1993 with a master's degree in engineering. He is now an associate professor of School of Cyber Science and Technology at Beihang University, Beijing, China. His research areas are network management and flow / protocol analysis technology, UNIX / Linux system development, large information system design and development technology, public opinion big data analysis and mining

Syed Jahanzeb Hussain Pirzada was born in Attock, Pakistan. In 2007, he received his BE degree in electronics engineering from NED University of Engineering and Technology, Karachi, Pakistan. In 2012, he received MS degree in electrical, electronics, control and instrumentation engineering from Hanyang University, Seoul, South Korea. Since 2018 he is enrolled for PhD degree in School of Cyber Science and Technology at Beihang University, Beijing, China. His research interest is cryptography for satellite applications.

Liu Jianwei was born in Shandong, China. He received BS and MS degrees in electronics and information engineering from Shandong University, Shandong, China in 1985 and 1988 respectively. He received his Ph.D. degree in electronics and communication systems from Xidian University, Shaanxi, China in 1998. He is now the dean of School of Cyber Science and Technology at Beihang University, Beijing, China. His current research interests include wireless communication networks, cryptography, and information and network security.