Improved Methods and Principles for Designing and Analyzing Security Protocols

Ali Kartit[1], Hamza Kamal Idrissi[2], and Mohamed Belkhouraf[1]
(Corresponding author: Ali Kartit)
Laboratory of Technology and Information, Department of TRI/ENSAJ, Chouaib Doukkali University[1]
No. 2, Avenue Mohamed ben Larbi Alaoui, El Jadida, Morocco
Laboratory of RIT, Department of Physics/FSR, Mohammed V University[2]
B.P. 554 3 Rue Michlifen, Rabat, Morocco
(Email: alikartit@gmail.com)
(Received Dec. 06, 2014; revised and accepted July 4 & August 12, 2015)

Abstract

Security protocols are a critical element of the infrastructures needed for secure communication and processing information. Before designing and analyzing protocols, it is important to reduce avoidable work. In this article, we presented the methods to prevent replay attacks [11] and attacks of the type flaw attacks on the protocols. We studied two types of attacks already mentioned. We presented some principles for secure protocols. To meet these principles, we have presented some methods for the design of security protocols. Some security vulnerabilities in security protocols published could be found by the principles presented and then we try to improve these protocols with the methods presented. A number of examples in the literature show that the work done in the document is very important.

Keywords: Analyzing security protocol, designing security protocol, flaw attack, replay attack

1 Introduction

Most security protocols are extremely simple if only their length is considered. However, the properties they are supposed to ensure are extremely subtle, and therefore it is hard to get protocols correct just by informal reasoning and "eyeballing".

Designing a secure protocol is a very difficult task. A set of principles and methods have been proposed from various aspects for different purposes [4]. Although these principles are described informally and are neither sufficient nor necessary for the reliability of the protocols, many flaws security protocol can be avoided from the start and the security protocols are designed more reliable if the designers or manuals developers automatic tools are familiar with them [1]. After our detailed analysis of these principles, we have found some existing problems, namely, some are too general to be practical; some are ambiguous so that designers are hard to grasp; some speak only of thought, not to study how to build protocols and avoid mistakes. We put forward a set of principles and methods against replay attacks and type flaw attacks by analyzing the attack characteristics and the reasons for the attack. A large number of examples show that the set of principles and methods are simple, efficient and practical.

2 Related Work

Above the previous two decades, the design of trustworthy and unfailing security protocols has been lectured by several publications. Introduced firstly in [5], the two-way authentication protocol based on symmetric key cryptography was enhanced in [6] in order to avoid weaknesses in the design of these types of protocols. This publication announced a methodology to automatically form a family of cryptographic two-way authentication protocols that are unaffected by the majority of attacks. In order to protect protocol messages from being vulnerable to replay attacks, [8] and [9] presented the notion of fail-stop protocols over a restricting group of protocol design considerations that prevent from replay attacks under some circumstances.

In 1996, [2] featured a group of elementary basics for reinforcing the security protocols design. They lectured two main issues messages authentication and trust. They incorporate also asymmetric key encryption. Nevertheless, these sets of principles do not guarantee protocol correctness. [3] and [15] suggested recommendations to avoid replay attacks, by using type-tagging messages with unique cryptographic functions and unique session keys without supposing common trust between the participants.

[13] recognized desynchronization attacks in 2013 on a group of protocols that employ dynamic shared secrets mechanisms for wireless messages. The authors a formal

system to model up-date mechanisms for shared secrets.

The effort on designing a novel trustworthy security protocols is still active today, as is the identification and solving design weaknesses in existing protocols [7, 10].

3 Principles and Methods

With the study of a large number of examples of replay attack [12, 17] and type flaw attack examples [18], and to investigate the cause of the attacks leads us to say that to avoid both types of attacks, applicable to principals session key must satisfy the following conditions:

- It can correctly judge that the principals of the session key produced belongs to;
- It can correctly judge which protocol run received messages belongs to;
- It can correctly judge whether a received message is reassembled and is a whole message sent by other party;
- It can correctly distinguish between messages structured by other party and by myself.

To make the application of guiding the session key to achieve the objectives mentioned above, the server must meet the following conditions:

- It knows which principals are applying for a session key;
- It knows identities of protocol runs initiated by principals applying for session keys;
- A message must be structured as a whole, in addition to principals who know the decryption key, no entity can separate it.

In addition, the type flaw attack result from the cause that different principal might use same key to encrypt similar or anti-symmetric similar massages. Many solutions have studied how to build differentiable messages, but often their methods, as long as adding a viable hypothesis; they may not enter law attack. From another point of view, we find that principals send clear on the application server for a session key, which play the same role with the encrypted message. Thus, in the protocols, only the use of shared server key to encrypt a message, which makes the distinction, encrypted messages. With the method, attack type law would be avoided.

3.1 Principles

With above analysis, design principles of security protocols against replay attack or type flaw attack are as follows.

Principle 1. *Principals and server can distinguish between protocol runs, which is critical to make protocol avoid a wide variety of attacks.*

Principle 2. *The distributing session key message must be a whole, in addition to principals applying the session key, no one can separate them [16].*

Principle 3. *Principal must know which principals the obtained session key is distributed to and which protocols run it belongs to.*

Principle 4. *Principal can identify that received encrypted message is not structured by himself.*

Principle 5. *If a protocol run is interrupted or intercepted after some steps, it must be satisfied that the risk is as less as possible.*

3.2 Methods

In order to make generated messages in the protocol meet the above principles, we design security protocol with the following methods:

Method 1. *Generate SID (Session Identifier) of protocol run copy. SID often consists of identifiers of principals applying for session key, nonce produced by principals and so on. SID contains nonce or a time stamp. Different principal has different nonce, and different runs copy has different nonce. Every nonce is unique. Using the time stamp requests that all participants have a global time system, namely, their time must be consistent, but, because time stamp has a valid period, near runs are difficult to be distinguished.*

Method 2. *Message distributing session key should contain SID.*

Method 3. *Message distributing session key is encrypted with shared key between receiver and server as a whole, and, generally, is structured as follows.*

$$\{SID, SeK, \{SID, SeK\}_{ShK_1}\}_{ShK_2}$$

where SeK denotes session key, and ShK denotes a shared-key.

Method 4. *In protocol, message applying for session key is plaintext as possible as. Considerable evidences show that sending encrypted message applying for session key plays the same role as sending plaintext message.*

Method 5. *The order sending of messages is presented in the Figure 1.*

The order of sending messages is adopted mainly because protocols run is initiated firstly by principal who has secret information to send other party. If the principal believes that applying session key have been successful, he will encrypt secret message with the gained session key and then will send it. After, he thinks that the task has been completed. If other party thinks that applying session key have been successful, but the initiator doesn't know it, the initiator re-initiates protocol run after a period of time, which wouldn't bring out much damage.

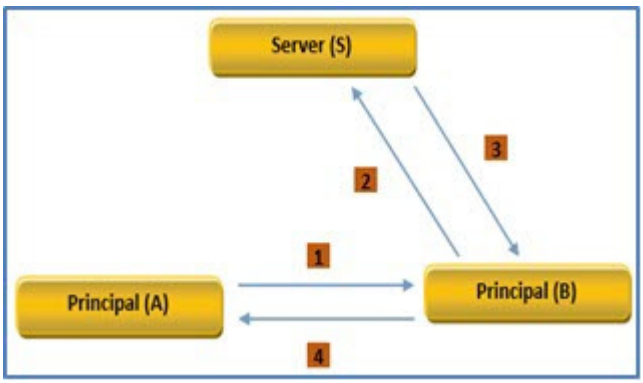

Figure 1: Order of sending messages

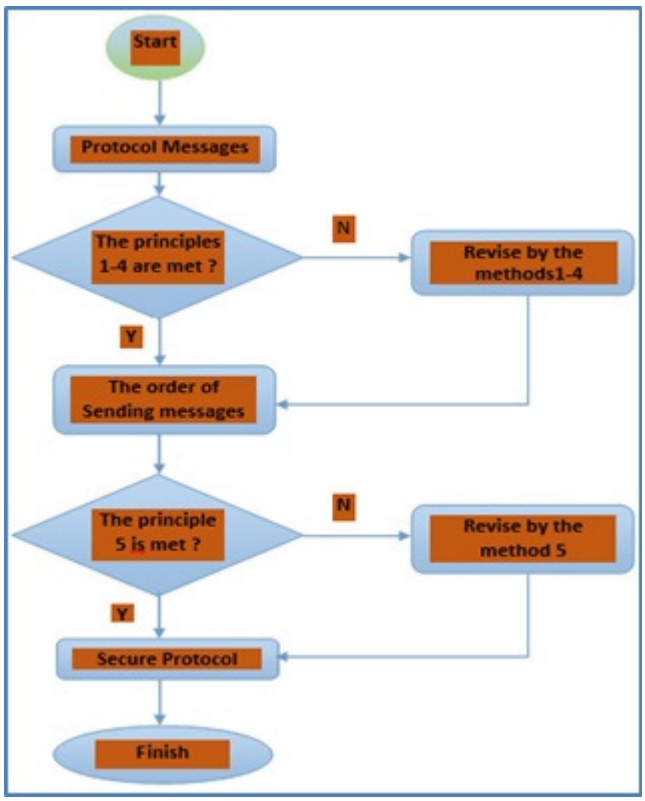

Figure 2: Process of analyzing and improving

4 Analysis and Improvement of the BAN-Yahalom Protocol

The process of using the above principles and methods to analyze and improve some security protocols is presented in the Figure 2.

By BAN logic analysis of Yahalom protocol, it is found that if A selects an old key to replay to B, B could not find it [14]. Therefore, BAN logic author improved Yahalom protocol. The improved Yahalom protocol (called BAN-Yahalom protocol) is as follows:

1) $A \longrightarrow B : A, N_a$;

2) $B \longrightarrow S : B, N_b, \{A, N_a\}_{Kbs}$;

3) $S \rightarrow A : N_b, \{B, K_{ab}, N_a\}_{Kas}, \{A, K_{ab}, N_b\}_{Kbs}$;

4) $A \rightarrow B : \{A, K_{ab}, N_b\}_{Kbs}, \{Nb\}_{Kab}$.

In this protocol, obviously, Principle 2, Principle 4 and Principle 5 are not met.

4.1 Principle 2 Destruction

To Principle 2 destruction, the protocol can be attacked as follows:

1) $A \longrightarrow P(B) : A, N_a$;

1') $P(B) \rightarrow A : B, N_a$;

2') $A \rightarrow P(S) : A, N'_a, \{B, N_a\}_{Kas}$;

2") $P(A) \rightarrow S : A, N_a, \{B, N_a\}_{Kas}$;

3') $S \rightarrow P(B) : N_a, \{A, K_{ab}, N_a\}_{Kbs}, \{B, K_{ab}, N_a\}_{Kas}$;

3) $P(S) \rightarrow A : N_p, \{A, K_{ab}, N_a\}_{Kbs}, \{B, K_{ab}, N_a\}_{Kas}$;

4) $A \rightarrow P(B) : \{A, K_{ab}, N_b\}_{Kbs}, \{N_p\}_{Kab}$.

In the above description, P(A), P(B) and P(S) represent that attacker P personate identity of A, B and S respectively. During the attack, the attacker P personate B to intercept the message (1) $A \longrightarrow P(B) : A, N_a$ and change the label of entity's name from A to B (1') $P(B) \rightarrow A : B, N_a$, by it A initiates a new run of distributing session key. The entity A thinks that B want to apply a session key with him, selects the nonce N'_a and encrypts received message in (1') to send them to S. However, the attacker P intercept the message (2') $A \rightarrow P(S) : A, N'_a, \{B, N_a\}_{Kas}$ In (2"), N'_a will be replaced with Na by the attacker P, by which P personate A to send message to S. When S receive the applying session key message, he think that B initiate a protocol's run round of applying session key to A and then generates a session key and encrypt it with the shared key K_{bs} to send B. The attacker personate B to intercept it, changes the inside plaintext N_a as N_p and personate S to send the obtained message to A in (3). When A receives the message (3), he can prove that protocol run applying session key initiated by oneself has successfully completed and gets the session key K_{ab}. Finally, A encrypt the nonce N_p with K_{ab} and send encrypted message and the message that S send to B to B, but the messages are intercept by the attacker P. As the result, A believe that protocol run of applying session key with B is successful and obtained session key is K_{ab}. Nevertheless, in the whole process, B does not participate in at all. To avoid the attack, we modify the above message (3) by Method 3 as follows:

3) $S \rightarrow A : \{B, K_{ab}, N_a, \{A, K_{ab}, N_b\}_{Kbs}\}_{Kas}$.

4.2 Principle 4 Destruction

Because Principle 4 is not satisfied, we can carry out the following attacks in the above protocol:

1) $P(A) \rightarrow B : A, N_a$;

2) $B \rightarrow P(S) : B, N_b, \{A, N_a\}_{Kbs}$;

1') $P(A) \rightarrow B : A, N'_a$;

2') $B \rightarrow P(S) : B, N'_b, \{A, N'_a\}_{Kbs}$;

4) $P(A) \rightarrow B : \{A, N'_b\}_{Kbs}, \{N_b\}_{Kab}$.

In above expression, P(A) and P(S) stand for that attacker P personate identity of A and S respectively. Assume that message $N'_a = K_{ab} + N_b$ and message K_{ab} are any strings that attacker know. In the process, entity A and entity S don't participate in the run of protocol, but the result is that attack P personate identity of A to share the key K_{ab} with B and that attack P know the key K_{ab}, which is very dangerous. To this defect, we use Method 4 to modify message 2 as follows:

2) $B \rightarrow S : A, B, N_a, N_b$.

4.3 Principle 5 Destruction

In this protocol, exchanging message sequence is not perfect and violates Principle 5. The attacker only need to intercept the message in the fourth step to make A believe that the application is successful and make B believe that the application is failed. In order to reduce harm that this kind of simple attacks brought about, the exchanging message order should be adjusted according to the design Method 5. Therefore, to avoid attack of BAN-Yahalom protocol, we modify the protocol by our principles and methods as follows:

1) $A \rightarrow B : A, N_a$;

2) $B \rightarrow S : A, B, N_a, N_b$;

3) $S \rightarrow B : \{A, K_{ab}, N_b\{B, K_{ab}, N_a\}_{Kas}\}_{Kbs}$;

4) $B \rightarrow A : \{B, K_{ab}, N_a\}K_{as}$.

5 Analysis and Modification of Abadi and Needhan Improved Otway-Rees Protocol

The Otway-Rees protocol is a simple security protocol put forward by 1987. On the help of server, both parties of communication securely get the session key. The author of BAN logic formally analyzed the Otway-Rees protocol and the result is that the protocol is secure, but there are redundant messages in it. Therefore, he modified the Otway-Rees protocol. Later, Boyd and Mao found the improved protocol to have security flaws. Since then, Adadi and Needham noted this defect and improved it. The improved protocol is as follows:

1) $A \rightarrow B : A, B, N_a$;

2) $B \rightarrow S : A, B, N_a, N_b$;

3) $S \rightarrow B : \{N_a, A, B, K_{ab}\}_{Kas}, \{N_b, A, B, K_{ab}\}_{Kbs}$;

4) $B \rightarrow A : \{N_a, A, B, K_{ab}\}_{Kas}$.

The above protocol is correct and efficient by BNA logic verification. but we can easily see that it doesn't meet Principle 2. There are a replay attack defect in the protocol because the message that server sends to entity B doesn't meet the atomicity principle. The attack process is as follows:

1) $A \rightarrow B : A, B, N_a$;

2) $B \rightarrow S : A, B, N_a, N_b$;

3') $S \rightarrow P(B) : \{N_a, A, B, K_{ab}\}_{Kas}, \{N_b, A, B, K_{ab}\}_{Kbs}$;

2") $P(B) \rightarrow S : A, B, N_a, N_b$;

3") $S \rightarrow P(B) : \{N_a, A, B, K'_{ab}\}_{Kas}, \{N_b, A, B, K'_{ab}\}_{Kbs}$;

3) $P(S) \rightarrow B : \{N_a, A, B, K'_{ab}\}_{Kas}, \{N_b, A, B, K_{ab}\}_{Kbs}$.

P(B) stands for that attacker P personate identity of B. The attacker intercepts the message in Step 3) and personate B to initiate a new run of protocol. S think that A and B apply for a new session key and distribute a session key K'_{ab} to B. The attacker intercepts it. At the time, the attacker has two distributed session keys K_{ab} and K'_{ab} to A and B and in Step 3) combine them to personate S to send it to B. When B receive the combined messages, he doesn't know that the message has been reassembled, and he believes that applying the session key is successful and forwards message to A. When A receives the message, he verify it to be his application session key. As the result, both believe that this application is successful, but their obtained the session keys are inconsistent. The attacker reach his deliberate destruction goal. To such attack, the protocol could be modified by above Method 3. The revised protocol is as follows:

1) $A \rightarrow B : A, B, N_a$;

2) $B \rightarrow S : A, B, N_a, N_b$;

3) $S \rightarrow B : \{N_a, A, B, K_{ab}, \{N_a, A, B, K_{ab}\}_{Kas}\}_{Kbs}$;

4) $B \rightarrow A : \{N_a, A, B, K_{ab}\}_{Kas}$.

The revised protocol meet the above principles, which can avoid various kinds of attacks. Here the exchanging message sequence is of vital importance. We exchange Steps 3) and 4) as follows.

3) $S \rightarrow A : \{N_a, A, B, K_{ab}, \{N_b, A, B, K_{ab}\}_{Kas}\}_{Kbs}$;

4) $A \rightarrow B : \{N_b, A, B, K_{ab}\}_{Kbs}$.

There is no much effect on the attack, but their security goal is not the same. When A receives message from server, he verify that the session key is correct and then forwards the corresponding message to B. However, he was not sure whether B receives the message. Therefore, he cannot decide that whether send his secret message encrypt by the session key to B or initiate a new run of protocol for applying session key. It can be easily seen that exchanging messages sequence is very important and that designing security protocol is difficult, in which subtle difference will bring about different effect.

6 Conclusion

In this article, the theory of examples of the replay attack and the type flaw attack are analyzed and a set of principles and methods are put forward.

In addition, we illustrated their simplicity and efficiency through analyzing and improving some classic protocols. The result shows that understanding the set of principles and methods make us avoid errors of replay or type-flaw attack in designing and analyzing security protocols. We hope that the work has a good guiding role in protocol analysis and design.

Before using formal tool to analyzing security protocols, defects of replay and type flaw attack can be found and avoided as much as possible by informal ways.

In future work, we intend to put into practice the principles and methods mentioned above to secure such a protocol.

References

[1] M. Abadi and R. Needham, "Prudent engineering practice for cryptographic protocols," *IEEE Transactions on Software Engineering*, no. 1, pp. 6–15, 1996.

[2] R. Anderson and R. Needham, "Robustness principles for public key protocols," in *Advances in Cryptology (CRYPTO'95)*, pp. 236–247, Springer, 1995.

[3] T. Aura, "Strategies against replay attacks," in *Proceedings of 10th IEEE Workshop on Computer Security Foundations*, pp. 59–68, 1997.

[4] G. Bella, "The principle of guarantee availability for security protocol analysis," *International Journal of Information Security*, vol. 9, no. 2, pp. 83–97, 2010.

[5] R. Bird, I. Gopal, A. Herzberg, P. Janson, S. Kutten, R. Molva, and M. Yung, "Systematic design of two-party authentication protocols," in *Advances in Cryptology (CRYPTO'91)*, pp. 44–61, Springer, 1992.

[6] R. Bird, I. Gopal, A. Herzberg, P. Janson, S. Kutten, R. Molva, M. Yung, et al., "Systematic design of a family of attack-resistant authentication protocols," *IEEE Journal on Selected Areas in Communications*, vol. 11, no. 5, pp. 679–693, 1993.

[7] B. Blanchet, "Automatic verification of security protocols in the symbolic model: The verifier proverif," in *Foundations of Security Analysis and Design*, pp. 54–87, Springer, 2014.

[8] U. Carlsen, "Cryptographic protocol flaws: know your enemy," in *Proceedings of IEEE Computer Security Foundations Workshop (CSFW'94)*, pp. 192–200, 1994.

[9] Li Gong and P. Syverson, "Fail-stop protocols: An approach to designing secure protocols (preprint)," in *Dependable Computing for Critical Applications*, pp. 44–55, IEEE, 1994.

[10] A Jurcut, T. Coffey, and R. Dojen, "Establishing and fixing security protocols weaknesses using a logic-based verification tool," *Journal of Communication*, vol. 8, no. 11, pp. 795–806, 2013.

[11] A. D. Jurcut, T. Coffey, and R. Dojen, "Design guidelines for security protocols to prevent replay & parallel session attacks," *computers & Security*, vol. 45, pp. 255–273, 2014.

[12] R. Khera and R. Sethi, "Enhancement in alarm protocol to prevent replay attack in MANET," in *International Journal of Engineering Research and Technology*, vol. 2. no. 5, pp. 2115–2119, 2013.

[13] I. Lasc, R. Dojen, and T. Coffey, "On the detection of desynchronisation attacks against security protocols that use dynamic shared secrets," *Computers & Security*, vol. 32, pp. 115–129, 2013.

[14] G. M. Li, "Secure analysis and improvement of yahalom protocol," *Microcomputer Development*, vol. 4, pp. 34, 2005.

[15] S. Malladi, J. Alves-foss, and R. B. Heckendorn, "On preventing replay attacks on security protocols," in *Proceedings of the International Conference on Security and Management*, pp. 77–83, CSREA Press, 2002.

[16] T. S. Sobh, A. Elgohary, and M. Zaki, "Performance improvements on the network security protocols," *International Journal of Network Security*, vol. 6, no. 1, pp. 103–115, 2008.

[17] L. Sun, Z. Luo, Y. Wu, and Y. Wang, "A technique for preventing replay attack in road networks," in *7th IEEE International Conference on Computer Science & Education (ICCSE'12)*, pp. 807–810, 2012.

[18] J. Wang, J. Zhang, and H. Zhang, "Type flaw attacks and prevention in security protocols," in *Ninth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD'08)*, pp. 340–343, 2008.

Ali Kartit received the PhD degree in Computer Science (November 2011) Specialty Security of Computer Networks. He graduated from the University Mohamed V Faculty of Rabat. The author has developed a rich and diverse experience of over 12 years in the computer world, including 8 years in technical and vocational education as a computer network trainer and manager module "computer network security notions" and 4 years

in the corporate world as Administrator of computer networks and Head of the park. The author is a certified Cisco and Microsoft Exchange Server 2003. His research area covers security policies of firewalls, the Intrusion detection systems and cloud security.

Hamza Kamal Idrissi graduated in 2010 at ENSIAS, Mohammed V University Rabat, as an IT Engineer. After a three years experience as a formative teaching Computer Networks and IT development for post baccalaureate students, he is currently working on a PhD, his area of research covers security aspects in the Cloud Computing, IDS and security policies.

Mohamed Belkhouraf graduated in 2010 at ENSIAS, Mohammed V University Rabat, as an IT Engineer. After a three years experience as an Information System Project Engineer in the private sector, he is currently working on a PhD, his area of research covers security aspects in the Cloud Computing.