International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 23 (2015) pp 43692-43696
© Research India Publications. http://www.ripublication.com

An application of Methods and Principles for Designing and Analyzing Security Protocols

H. Kamal Idrissi,

Laboratory of Research in Informatics and Telecommunication (LRIT) Department of Computer physics/FSR, MOHAMMED V UNIVERSIY, RABAT, MOROCCO h.kamalidrissi@gmail.com

A. Kartit,

Laboratory of Technology and Information (LTI), Department of TRI/ENSAJ CHOUAIB DOUKKALI UNIVERSITY, EL JADIDA, MOROCCO alikartit@gmail.com

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 [1] 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: Designing Security Protocol, Analyzing Security Protocol, Security Protocols, Secure Communication, Replay Attack, Flaw Attack, Process of Analyzing and Improving a Protocol Security, Security Attacks Characteristics, Protocols Messages, Security Logic Verification.

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 [2]. In [1], Anderson and Needham propose a number of influential principles for designing security protocols [3]. Often has it been remarked that those principles are not meant to totally ensure the protocol goals, but that it merely is prudent to conform to them, as the title of an influential paper confirms [4]. In other words, no design principle should be taken as biblic. For example, one of the most popular principles states that each protocol message should be explicit about its meaning, that is to say that nothing should be taken for granted. However, Syverson warns us that this principle too has limitations [5]. Protocols such as Bellovin and Meritt's Encrypted Key Exchange (EKE) [6] do not conform to it, and indeed meet their goals by not conforming to it. 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 [7]. 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.

Principles and methods

With the study of a large number of examples of replay attack [8]-[9] and type flaw attack examples [10], 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:

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

- Knows which principals are applying for a session key;
- 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. 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.

A. Principles

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

i. Principle 1

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

ii. Principle 2

The distributing session key message must be a whole, in addition to principals applying the session key, no one can separate them. [11]

iii. Principle 3

Principal must know which principals the obtained session key is distributed to and which protocol's run it belongs to.

iv. Principle 4

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

v. Principles 5

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

B. Methods

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

i. 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 run's 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.

ii. Method 2

Message distributing session key should contain SID.

iii. ***Method 3***

Message distributing session key is encrypted with
Shared key between receiver and server as a whole,
And, generally, is structured as follows:

$\{SID,\ session\ key,\ \ SID,\ session\ key\ _{Shared-key_1}\}_{Shared-key_2}$

iv. ***Methods 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.

v. ***Methods 5***

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

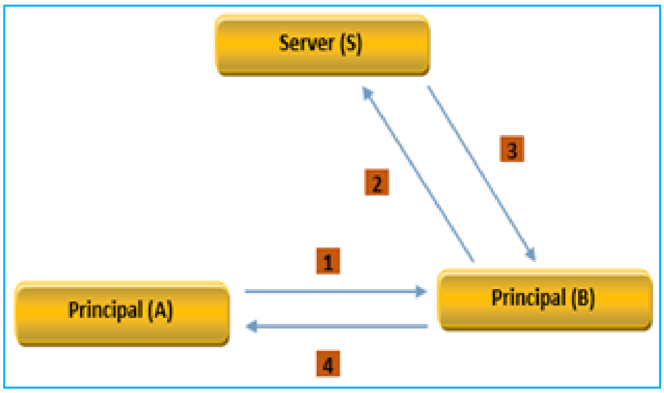

Fig.1. Architecture of Sensor Node

The order of sending messages is adopted mainly because protocol's 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.

Analysis of Security Protocols

A. Analysis and Improvement of Security Protocol

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

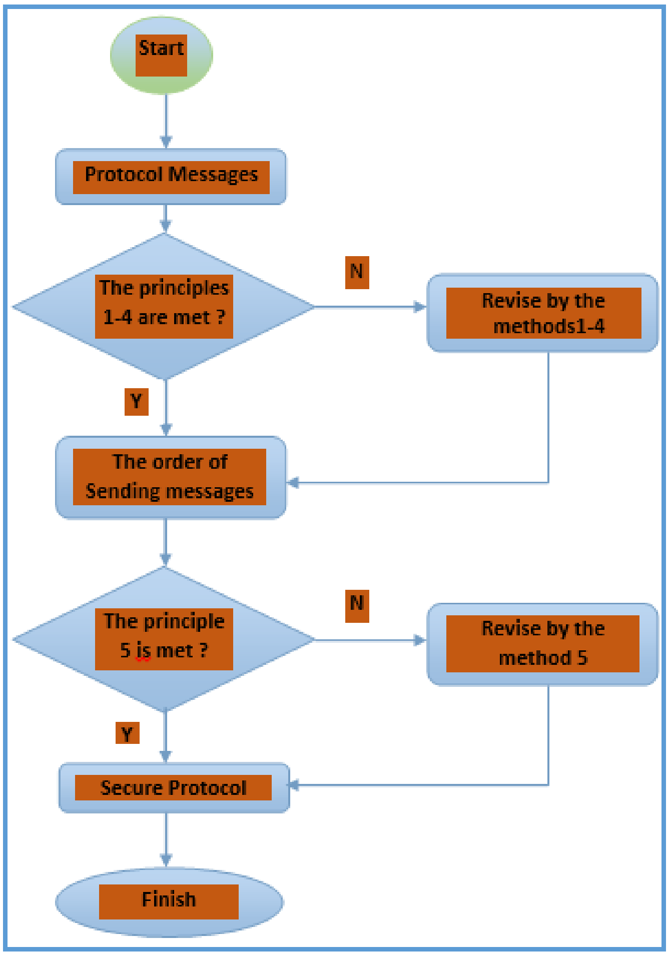

Fig. 2. Process of analyzing and improving

B. Analysis and Improvement of The BAN-Yahalom protocol

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 [12]-[13]. Therefore, BAN logic author improved Yahalom protocol. The improved Yahalom protocol (called BAN-Yahalom protocol) is as follows:

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

$(2) B \rightarrow S : B , N_b , \{A , N_a\}_{K_{bs}}$

$(3) S \rightarrow A : N_b \{B , K_{ab} ,N_a\}_{K_{as}}$

$\{A , K_{ab} ,N_b\}_{K_{bs}}$

$(4) A \rightarrow B : \{A , K_{ab} ,N_b\}_{K_{bs}} , \{N_b\}_{K_{ab}}$

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

i. Principle 2 destruction

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

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

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

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

$(2'') P(A) \rightarrow S : A , N_a , \{B , N_a\}_{K_{as}}$

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

$\{B , K_{ab} ,N_a\}_{K_{as}}$

$(3) P(S) \rightarrow A : N_p ,\{A , K_{ab} ,N_a\}_{K_{bs}},$

$\{B , K_{ab} ,N_a\}_{K_{as}}$

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

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→P (B): A, Na and change the label of entity's name from A to B (1') P (B)→ A: B, Na, 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→P (S): A, N'a {B, Na} 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 Kbs to send B. The attacker personate B to intercept it, changes the inside plaintext Na as NP 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 Kab. Finally, A encrypt the nonce NP with Kab 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 Kab. 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:

$(3) S \rightarrow A : \{B , K_{ab} ,N_a \{A , K_{ab} ,N_b\}_{K_{bs}}\}_{K_{as}}$

ii. Principle 4 destruction

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

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

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

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

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

$(4) P(A) \rightarrow B : \{A , N'_b\}_{K_{bs}} , \{N_b\}_{K_{ab}}$

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= Kab +Nb and message Kab 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 Kab with B and that attack P know the key Kab, which is very dangerous. To this defect, we use the method 4 to modify message 2 as follows:

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

iii. Principle 5 destruction

In this protocol, exchanging message sequence is not perfect and violates the 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) $B \rightarrow A : \{B , K_{ab}, N_a\}_{Kas}$

Our Contribution: Analysis and Modification of Abadi and Needhan [14] 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, Abadi 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) $S \rightarrow B : \{N_a, A, B, K_{ab}\}_{Kas}$ $,\{N_b, A, B, K_{ab}\}_{Kbs}$

(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. However, we can easily see that it does not meet the 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 the 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 the 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) $S \rightarrow B :$ $\{N_b , A , B , K_{ab}$ $\{N_a , A ,$
$B , K_{ab}\}_{Kas}\}_{Kbs}$

(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 the steps (3) and (4) as follows:

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

(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 can't 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.

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] D.Jurcut, T.Coffey, R.Dojen, *Design Guidelines for Security Protocols to Prevent Replay & Parallel Session Attacks*. Computers & Security, Vol. 45, pp. 255-273, 2014.

[2] G. Bella, The Principle of Guarantee Availability for Security Protocol Analysis, *International Journal of Information Security*, Vol. 9, n. 2, pp. 83-97, 2010.

[3] Anderson, R., Needham, R., *Robustness Principles for Public Key Protocols*, IN Advances in Cryptology—CRYPT0'95, Springer Berlin Heidelberg (Page: 236-247 Year of Publication: 1995).

[4] M.Abadi, R.Needham, Prudent Engineering Practice for Cryptographie Protocols, *IEEE Transactions on Software Engineering*, Vol. 22, n. 1, pp. 6-15, 1996.

[5] Syverson, P, *Limitations on Design Principles for Public Key Protocols*, Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy (PAGE: 62 Year of Publication: 1996).

[6] Bellovin, S.M., Merritt, M., *Augmented Encrypted Key Exchange: a Password-Based Protocol Secure Against Dictionary Attacks and Password File Compromise*, 1ST ACM Conference on Computer and Communications Security (Page: 244 Year of Publication: 1993).

[7] Abadi, M.,Needham, R., *Prudent Engineering Practice for Cryptographic Protocols*, Proceedings of the IEEE Computer Society Symposium on Research in Security and Privacy (PAGE: 122 Year of Publication: 1994).

[8] Sun, L., Luo, Z., Wu, Y., Wang, Y., *A Technique for Preventing Replay Attack in Road Networks*, ICCSE 2012-Proceedings of 2012 7th International Conference on Computer Science and Education (PAGE: 807 Year of Publication: 2012).

[9] R.Khera, R.Sethi, *Enhancement in Alarm Protocol to Prevent Replay Attack in MANET*. International Journal of Engineering Research & Technology, Vol. 2, pp 2115-2119, 2013.

[10] Wang, J., Zhang, J., Zhang, H., *Type Flaw Attacks and Prevention in Security Protocols*, Proc. 9th ACIS Int. Conf. Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, SNPD 2008 and 2nd Int. Workshop on Advanced Internet Technology and Applications (PAGE: 340 Year of Publication: 2008).

[11] I.Lasc, R.Dojen, T.Coffey, *on the Detection of Desynchronisation Attacks Against Security Protocols that use Dynamic Shared Secrets*, Computers & Security, Vol. 32, pp. 115-129, 2013.

[12] L. I. Guo-min, Secure Analysis AND Improvement Of Yahalom Protocol, *Microcomputer Development*, Vol. 4, pp. 034, 2005.

[13] L.C.Paulson, Relations between secrets: Two formal analyses of the Yahalom protocol, *Journal of Computer Security*, Vol. 9, n. 3, pp. 197-216, 2001.

[14] M.Burrows, M.Abadi, R.Needham, Logic of Authentication, *Operating Systems Review (ACM)*, Vol. 23, n. 5, pp. 1-13, 1989.

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. 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