Design Principles and Security of Authentication Protocols with Trusted Third Party

Xianxian Li, Jun Han

School of Computer Science, Beihang University, Beijing, China

<lixx@cscw.buaa.edu.cn> <jun_han@buaa.edu.cn>

Zhaohao Sun

School of Economics and Information Systems, University of Wollongong

<zsun@uow.edu.au>

ABSTRACT

Two-way identity authentication is the basis of secure communication in a distributed application environment. A trusted third party (TTP) is needed while PKI is not applicable, and the design of authentication protocols with TTP is a complicate and challenging task. This paper examines the characteristics of the security of authentication protocols with TTP, summarizes the essential factors of session key and illustrates the potential attacks while these essential factors are not well considered. It also proposes some design principles and a model of authentication protocol with TTP.

1. Introduction

How to identify the principal in the open internet environment, known as identity authentication, presents many interesting challenges in the network security area. Although the traditional approaches to identity authentication with user/password and one-way hash function seem simple, yet they often adapt only to apply on the cases involved in unilateral authentication. For example, an end user with his password makes effect to protect the access to some sensitive services, but such approach is prone to be cracked because the password can be acquired through the "dictionary attack". Therefore, some authentication protocols depending upon cryptography were specially designed to guarantee the security of communication.

Another key requirement related to identity authentication is to build secure tunnel through exchanging keys between two communicating entities. In fact, many security protocols, such as Kerberos authentication protocol[2] and Internet Key Exchange protocol (IKE)[3], were all designed to both of above two requirements. Generally, in the environment supported with PKI (Public Key Infrastructure) these objectives can be achieved by the Internet Key Exchange approach such as Diffie-Hellman. However, a trusted third party (shortly write as TTP) is required in the application environments without PKI.

Many authentication protocols have been proposed for the latter application environment. However it is very difficult to design protocols meeting above requirements due to that the potential security vulnerabilities are covert. As demonstrated by us, some research results also prove such opinions[4]. Formal methods have been widely used to analyze the security of authentication protocols in recent years. Many significant results have been achieved in the area since formal methods began to apply to cryptographic protocol security analysis in 1981. However, so far no universal approach can be efficient to all these problems. In fact, the security problem of protocol has proved undecidable [14].

For the sake of brevity, authentication protocol with TTP is named TAP. Discussions in this paper are under certain assumptions of the environment. A denotes the initiator, B denotes the responder, S denotes the TTP and P denotes the attacker. $P(X)$ denotes the attack when P pretends to be party X; P_X denotes the attack when P dominates over X.

In the remainder of this paper, we analyze the characteristics of the security of TAP and summarize the essential security factors. Lastly design principles and a model of authentication protocol with TTP are available to engineers and scientists on authentication protocol design and analysis.

2. Basic Conceptions and Assumptions

A TAP contains three roles: an *initiator*, a *responder* and a *TTP*. The initiator and the responder are in the equal position, and neither can get special knowledge outside the protocol. The TTP is just one server who is trusted by both initiator and responder and enable them achieve mutual authentication. The TTP also acts as the role of KDC (Key Distributed Centre) to generate session keys for them.

BASIC ASSUMPTIONS:

The protocol runs in an open network environment with some attackers existing. As the assumptions in Dolev-Yao model, the cryptographic algorithms are assumed to be perfect where no attacker can break. The attacker can control the whole network where he can arbitrarily read, intercept and modify any communicating messages. Especially, the attacker can pretend to be any a principal, but except for the TTP, and take part in the running protocol.

We now give an example of Needham-Schroeder Symmetric Key Authentication protocol[6] (Protocol 1) as following:

(1) $A \to S : A, B, N_a$

(2) $S \to A : \{N_a, B, K_{ab}, \{K_{ab}, A\}_{K_{bs}}\}_{K_{as}}$

(3) $A \to B : \{K_{ab}, A\}_{K_{bs}}$

(4) $B \to A : \{N_b\}_{K_{ab}}$

(5) $A \to B : \{N_b - 1\}_{K_{ab}}$

Protocol 1: Needham-Schroeder Symmetric Key Protocol

There have been many other protocols that have used a trusted third party to generate and distribute keys in a similar way.

In the above description the format "$X \to Y : M$" means that a principal X sends a message M to another principal Y. The preceding sequence number indicates the step in the protocol. M is called the message body which consists of some messages such as random nonces N_a and N_b, encrypted message $\{m\}_K$ where K is a key and m is a plain message, etc.

3. The Basic Security Requirements of TAP

In this section we mainly analyze and summarize the basic security requirements of TAPs, and then propose the design principles of this kind of protocols.

The fundamental objectives of TAP include the following:

- Achieve mutual identity authentication between two parties A and B.
- Share a session key K_{ab} between two parties A and B.

Generally, both the security objectives are achieved mainly by using the session keys produced by TTP in the environments with TTP. These protocols usually include two basic steps: the first, two parties of protocols (or one of them) send a request to the TTP, then the TTP creates and distributes a session key for both; the second, the two parties mutually prove to the other party that they hold the fresh session key distributed by the TTP.

According to the characteristics of TAPs, their basic security requirements can reduce to the following factors:

- **R1: Secrecy.** None of other principals except for the participants of protocols and TTP can obtain the session key K_{ab}.
- **R2: Authenticity.** The session key K_{ab} was originally produced by the TTP.
- **R3: Freshness.** The session key K_{ab} is just produced in the corresponding session.
- **R4: Consistency.** Both the keys built for A and B in a session are same.

Above four factors are essential to the security of TAPs. The security of these protocols will often fail whenever any of them is not well considered. In what follows, we'll analyze and show that all the four factors may induce the corresponding potential attacks for these protocols.

3.1 Secrecy

The requirement for "secrecy" is obviously necessary to the identity authentication since attackers can pretend the participants A and B if they know the session key. One way to meet this security requirement is to encrypt K_{ab} with the sharing long-term key K_{as} or K_{bs}, then send it to the corresponding participant A (or B), such as the message in step (2) of Protocol 1.

3.2 Authenticity

The secrecy is not enough to the security of the protocols, because in some cases the attackers may achieve the attack objective by using a fake key to replace the session key produced by the TTP. Therefore, the session key has to be authenticated, which is the requirement of authenticity. This requirement is mainly to prevent from using a fake session key. Since the fake key can be easily learnt by the attackers, the protocol that does not meet this requirement fails to achieve its security objectives.

In the most cases, the approach to encrypt the session key with the long-term keys K_{as} and K_{bs} is not sufficient to ensure the authenticity, for example, the following protocol (BAN-Yahalom).

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

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

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

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

BAN-Yahalom Protocol

The BAN-Yahalom Protocol is a typical authentication protocol with TTP. The message containing the session key is encrypted for transmission with the long-term keys K_{as} and K_{bs} in the protocol, however, the following can be an attack to the BAN-Yahalom protocol.

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

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

(1′) $P(A) \to B : A, (N_a, N_b)$

(2′) $B \to P(S) : B, N'_b, \{A, N_a, N_b\}_{K_{bs}}$

(3) Omitted

(4) $P(A) \to B : \{A, N_a, N_b\}_{K_{bs}}, \{N_b\}_{N_a}$

A reason to reduce the attack to the protocol is that B is permitted to generate a message term encrypted with K_{bs} which is similar to the message $\{A, K_{ab}, N_b\}_{K_{bs}}$ containing session key created by S. Hence this factor should be well considered in the design for this type of protocols.

3.3 Freshness

The session key shared by A and B is still necessary to be freshness, since the old key that is used in the former sessions may be learnt by attackers due to some reasons such as the storage problem. In fact, frequently updating session keys is one of fundamental requirements for the secure communication. And it is also one of design objectives of some protocols.

The following is an attack[8] to the Needham-Shroder protocol when freshness is not satisfied.

(1) $P_A \to S : A, B, N_a$

(2′) $S \to P_A : \{N'_a, B, K'_{ab}, \{K'_{ab}, A\}_{K_{bs}}\}_{K_{as}}$

(2) $S \to P_A : \{N_a, B, K_{ab}, \{K_{ab}, A\}_{K_{bs}}\}_{K_{as}}$

(3) $P_A \to B : \{K'_{ab}, A\}_{K_{bs}}$

(4) $B \to P_A : \{N_b\}_{K'_{ab}}$

(5) $P_A \to B : \{N_b - 1\}_{K'_{ab}}$

This flaw of the protocol is due to that the principal A can not verify the freshness of the received message $\{K_{ab}, A\}_{K_{bs}}$ in the Needham-Shroder protocol.

The "freshness" is mainly achieved by using nonces (generated randomly) or time-stamps, such as, N_a in the Needham-Shroder protocol and N_b in the BAN-Yahalom Protocol.

3.4 Consistency

In some cases the attacker can lead that the two parties can not make an agreement about the session key if the consistency has not been well considered. In a protocol if the message with the session key material does not contain the content indicating the owner of the key, then the attacker may replace this message by another message in other sessions. For example, if the two message terms produced by S in the third step of the BAN-Yahalom Protocol did not contain the *ID* of the corresponding principal, that is, $N_b, \{K_{ab}, N_a\}_{K_{as}}, \{K_{ab}, N_b\}_{K_{bs}}$, the attacker could use the message for $\{K_{ac}, N_a\}_{K_{as}}$, then form an attack to this protocol. Hence it is essential that the message term with the session key material contains the *ID* of the corresponding participant. However, this is not sufficient. For example, even the well-modified Otway-Rees protocol can still be attacked[10, 12].

The following is the modified Otway-Rees protocol in [10].

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

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

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

(4) $B \to A : \{N_a, A, B, K_{ab}\}_{K_{as}}$

Modified Otway-Rees Protocol

This protocol has corrected several flaws in the Otway-Rees Protocol. The following is an attack to this protocol that induces disagreement between the two parties[12].

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

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

(2′) $P(B) \to S: A,B,N_a,N'_b$

(2″) $P(B) \to S: A,B,N'_a,N_b$

(3′) $S \to P(B): \{N_a,A,B,K_{ab}\}_{K_{as}},$
$\{N'_b,A,B,K_{ab}\}_{K_{bs}}$

(3″) $S \to P(B): \{N'_a,A,B,K'_{ab}\}_{K_{as}},$
$\{N_b,A,B,K'_{ab}\}_{K_{bs}}$

(3) $P(S) \to B: \{N_a,A,B,K_{ab}\}_{K_{as}}$
$\{N_b,A,B,K'_{ab}\}_{K_{bs}}$

(4) $B \to A: \{N_a,A,B,K_{ab}\}_{K_{as}}$

The fourth step of the modified Otway-Rees protocol can be further corrected as the following:

(4) $B \to A: \{N_a,A,B,K_{ab}\}_{K_{as}},\{N_a\}_{K_{ab}},N'_b$

(5) $A \to b: \{N'_b\}_{K_{ab}}$

Then B can prove to A that itself holds the session key K_{ab}, and require that A should prove to hold the key K_{ab} as well as. So it has prevented from the above attack.

4. Design principles of TAP

Based on the analysis of the basic objectives and security requirements of TAP in the previous section, the following design principles of TAP are proposed.

Principle 1 (P1): TTP is responsible for session keys, and applies sharing secret key K_{as} or K_{bs}.

Principle 2 (P2): The identity of key owner and the nonce should be included in the session key message published by TTP; and changing of the structure of encrypted message and adding of abundant information should be considered to avoid similar structural message received by the first two parties.

Principle 3 (P3): If a party other than TTP has to send message encrypted with K_{as} or K_{bs}, reordering of sub-messages and adding of redundant information should be considered.

Principle 4 (P4): After the first two parties have received the session key, shaking hands should be proceeded to guarantee the consistency of the session key.

These principles correspond to the basic security requirements mentioned in the previous section: P1 corresponds to R1; P2 considers the requirements of both R2 and R3; P3 comes from the requirement of authenticity (R2) and P4 corresponds to R4.

Principle 5 (P5): The balance of security and efficiency should be considered.

In the actual design of security protocols, runtime efficiency should also be considered besides security issues. The runtime efficiency of a security protocol is influenced by the following factors:

1. Complexity of computation
2. Bandwidth used
3. Number of time for message transfer

Basic computation tasks involved in TAP are encryption and decryption of messages, and the generation of nonce and time-stamps. The usage of bandwidth is mainly influenced by the length of messages and the number of time that messages are passed.

The following protocol was first proposed by Carlson[12], and no attack has been reported on this protocol.

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

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

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

(4) $B \to A: \{N_a,B,K_{ab}\}_{K_{as}},\{N_a\}_{K_{ab}},N'_b$

(5) $A \to B: \{N'_b\}_{K_{ab}}$

We now apply the proposed design principles to evaluate the protocol. First, P1 is satisfied since session key K_{ab} is not an explicit term. The two encryptions of K_{ab} are $\{K_{ab},N_b,A\}_{K_{bs}}$ and $\{N_a,B,K_{ab}\}_{K_{as}}$. Next, considering $\{K_{ab},N_b,A\}_{K_{bs}}$, the receiver is B; A,which is the identity of the owner of the secret session key, and N_b, which is the freshness value shown to B are both included. $\{N_a,B,K_{ab}\}_{K_{as}}$ has similar property. Moreover, these two encryptions are not in a symmetric structure and therefore, P2 is satisfied.

Encryptions using K_{as} or K_{bs} can be only generated by a TTP S and P3 is satisfied accordingly. $\{N_a\}_{K_{ab}},N'_b$ in message (4) and $\{N'_b\}_{K_{ab}}$ in message (5) achieve the handshaking process for determining the new session key, therefore P4 is also satisfied.

It should be noted that, in message (4), B uses a new nonce N'_b, which can be replaced by an old nonce in our view. Because the purpose of using the nonce is to verify if session key K_{ab} has been got, and since the session key is new, even if N'_b is and old nonce that has been used, an attacker can not cheat B by reply.

5. Conclusion

This work analyzed the fundamental objectives, security requirements and problems of authentication protocol with TTP. Some basic security constraints and design principles have been proposed. These principles have been shown to be simple, effective, efficient, easy to be implemented and practical in actual design.

References

[1] Schneier B. *Applied Cryptography: Protocols, Algorithm, and Source Code in C (2nd Edn)*, John Wiley & Sons, Inc., 1996.

[2] Kohl JT, Neuman BC. The Kerberos Network Authentication Service. RFC1510, 1993.

[3] Harkins D, Carrel D. The Internet Key Exchange(IKE). RFC2409, 1998.

[4] Clark J, Jacob J. A survey of authentication protocol literature. http:// www.cs.york.ac.uk/jac/ papers/ drareviewps.ps, 1997.

[5] Dolev D, Yao A. On the security of public key protocols. IEEE Transactions on Information Theory, 1983, 29(2):198~208.

[6] Needham R, Schroeder M. Using encryption for authentication in large networks of computers. Communications of the ACM, 1978, 21(12):993~999.

[7] Syverson P. A taxonomy of replay attacks. In: Proc. of 7th IEEE Computer Security Foundations Workshop - CSFW'94, Franconia, New Hampshire, USA: IEEE Computer Society, 1994,187~191.

[8] Denning D, Sacco G. Timestamps in Key Distribution Protocols. Communications of the ACM, 1981,24(8):533~536.

[9] Gong L. Variations on the themes of Message Freshness and Replay. In: Proc. of the Computer Security Foundations Workshop VI, Franconia, New-Hampshire, 1993.

[10] Abadi M, Needham R. Prudent Engineering Practice for Cryptographic protocols. IEEE Transactions on Software Engineering, 1996, 22(1):6~15.

[11] Otway D, Rees O. Efficient and Timely Mutual Authentication. ACM Operating Systems Review, 1987, 21(1):8~10.

[12] Carlsen U. Optimal privacy and authentication on a portable communications system. ACM SIGOPS Operating Systems Review, 1994, 28(3):16~23.

[13] Donovan B, Norris P, Lowe G. Analyzing a library of security protocols using Casper and FDR. In:Proc. of Workshop on Formal Methods and Security Protocols, Trento, 1999.

[14] N.A. Durgin, P.D. Lincoln, J.C. Mitchell and A. Scedrov. Undecidability of bounded security protocols. In Electronic Proceedings of the Workshop on Formal Methods and Security Protocols, 1999. http://www.cs.bell-labs.com/who/nch/ fmsp99/program.html.