ELSEVIER

Information Processing Letters 69 (1999) 297–302

Information Processing Letters

Inter-protocol interleaving attacks on some authentication and key distribution protocols [☆]

Wen-Guey Tzeng [*], Chi-Ming Hu

Department of Computer and Information Science, National Chiao Tung University, Hsinchu 30050, Taiwan

Received 1 October 1998
Communicated by S.G. Akl

Abstract

We present a new attack, called the inter-protocol interleaving attack, on authentication and key distribution protocols. The attack enlightens us two things. The first is that when considering attacks against a protocol, we should consider not only the protocol itself, but also the interaction with other protocols. The second is about a warning of "not using the shared secret keys between a server and its clients in any other places" that appears in many security-related communication standards, such as CCITT X.509 and ISO 9798. Our attack provides a concrete example for showing that this warning is necessary. © 1999 Elsevier Science B.V. All rights reserved.

Keywords: Cryptographic attack; Inter-protocol interleaving attack; Authentication; Key distribution

1. Introduction

The authentication and key distribution problem has been a focus research topic in cryptography since Needham and Schroeder's pioneer paper. Many protocols for authentication and key distribution have been proposed in the literature [10–12,15]. Some of them are shown weak against cryptographic attacks, such as the modification, the man-in-the-middle, the chosen plaintext, the impersonation, the oracle session, the parallel session attacks etc., while some of them are shown secure by means of cryptographic analysis or logic proof methods [10,15]. In showing the security of a protocol by cryptographic analysis against cryptographic attacks, almost all of them focus on interaction between the attacker and one or more sessions of the protocol. However, interaction between the attacker and sessions of two or more different protocols is largely ignored. In this paper we show an attack, called the *inter-protocol interleaving* attack, on using a slightly modified Neuman–Stubblebine protocol [10] against the Kehne–Schonwalder–Langendorfer protocol [9] and using the Otway–Rees protocol [14] against the Kerberos protocol [13].

The inter-protocol interleaving attack is a variation of the interleaving attack [2,3,5,8]. The interleaving attack is usually applied to two parallel sessions of a protocol. The inter-protocol interleaving attack is applied to two sessions of two different protocols that are activated simultaneously. In the attack, the attacker intercepts the messages in the session of a protocol and replays them for the messages in the session of another protocol for masquerading as a legal principal in cheating another legal principal. To our

[☆] Research supported in part by grant NCS-88-2219-E-009-058, National Science Council, Taiwan.

[*] Corresponding author. Email: tzeng@cis.nctu.edu.tw.

0020-0190/99/$ – see front matter © 1999 Elsevier Science B.V. All rights reserved.
PII: S0020-0190(99)00023-X

best knowledge, there is no known published concrete example of the inter-protocol interleaving attack up to date.

The attack enlightens two points. The first is that when considering attacks against a protocol, we should consider not only the protocol itself, but also the interaction with other protocols. The second is about a warning of "not using the shared secret keys between a server and its clients in any other places" that appears in many security-related communication standards, such as CCITT X.509 [4] and ISO/IEC 9798 [6]. The main reason for this warning is due to the intuition that if not doing so, it may increase the possibility of key leakage. Our attack provides a concrete example for showing that this warning is necessary.

Some may argue that it is not possible to implement two different protocols for the same purpose in a server practically. We feel that the real world is very heterogeneous so that we cannot expect that all principals use the same protocol. For example, the server may be a printer server as well as a file server simultaneously. Different providers may supply different protocols for authentication and key distribution. Therefore, this situation is possible. Some may also argue that the attack is very implementation-dependent and needs quite a lot of assumptions. Indeed, since it is implementation-dependent, we can not expect what the implementors would do. They might be careless, imprudent, or even lazy. Anything becomes possible in the real world.

The paper is organized as follows. We first briefly introduce the Kerberos, Kehne–Schonwalder–Langendorfer, Neuman–Stubblebine and Otway–Rees protocols. In Section 3, we present two examples of the inter-protocol interleaving attack. In Section 4, we discuss possible solutions for protocols to prevent from such attack. Finally, we conclude the paper.

2. The protocols

In this section we outline the protocols that are used in our attacks. These protocols are the Kerberos protocol [13], the protocol presented by Kehne, Schonwalder and Langendorfer [9] (called the KSL protocol hereinafter), the protocol presented by Neuman and Stubblebine [10] (called the NS protocol hereinafter) and the protocol presented by Otway and Rees [14] (called the OR protocol hereinafter). These four protocols are all trusted server and secret-key cryptosystem based authentication and key distribution protocols. In each protocol, there are two participating principals A and B who wish to establish a session key K for secure communication. There is also a server S who generates the session key and is trusted by both principals. Each principal X shares a secret key K_{XS} with the server S so that they can communicate secretly with the secret key. The message M encrypted using the key K_{XS} is denoted by $E_X(M)$, in which the used secret-key cryptosystem is the same. Similarly, the message encrypted using the key K is denoted by $E_K(M)$. We shall use X to denote the identity name of the principal X. The nonce (random number) issued by X is denoted by N_X and the timestamp issued by X is denoted by T_X. For each message pass, we use "$(i)\ X \to Y$: M" to denote that the message M is sent from X to Y in message pass i.

For convenience in explaining our attacks later, we interchange two principals A and B in the OR and NS protocols described in the original papers. We also discard the messages for repeated authentication in the Kerberos and NS protocols since they are irrelevant to our attacks.

2.1. The Kerberos protocol

There are four message passes in the Kerberos protocol shown below, in which L denotes the lifetime. For its detailed description, see [13].

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

(2) $S \to A$: $E_A(T_S, L, K, B, E_B(T_s, L, K, A)),$

(3) $A \to B$: $E_B(T_S, L, K, A), E_K(A, T_A),$

(4) $B \to A$: $E_K(T_A + 1).$

2.2. The KSL protocol

There are five message passes in the KSL protocol shown below, in which N'_B is the second nonce generated by the principal B. For its detailed description, see [9].

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

(2) $B \to S$: $N_B, A, N_A, B,$

(3) $S \to B$: $E_B(N_B, A, K), E_A(N_A, B, K)$,

(4) $B \to A$: $E_A(N_A, B, K), N'_B, E_K(N_A)$,

(5) $A \to B$: $E_K(N'_B)$.

2.3. The NS protocol

There are four message passes in the NS protocol shown below. For its detailed description, see [10].

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

(2) $A \to S$: $A, E_A(B, N_B, T_A), N_A$,

(3) $S \to B$: $E_B(A, K, N_B, T_A), E_A(B, K, T_A), N_A$,

(4) $B \to A$: $E_A(B, K, T_A), E_K(N_A)$.

For our attack, we consider a slightly modified NS protocol, called the NS′ protocol hereinafter, which is almost the same as the NS protocol except that the positions of the identity name B and the nonce N_B in message pass 2 are interchanged, which is shown below.

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

(2) $A \to S$: $A, E_A(N_B, B, T_A), N_A$,

(3) $S \to B$: $E_B(A, K, N_B, T_A), E_A(B, K, T_A), N_A$,

(4) $B \to A$: $E_A(B, K, T_A), E_K(N_A)$.

This interchange does not affect the security of the protocol. In implementing the NS protocol, this might happen due to preference of the implementors.

2.4. The OR protocol

There are four message passes in the OR protocol shown below, in which I denotes the identifier of the session. For its detailed description, see [14].

(1) $B \to A$: $I, B, A, E_B(N_B, I, B, A)$,

(2) $A \to S$: $I, B, A, E_B(N_B, I, B, A)$,
$E_A(N_A, I, B, A)$,

(3) $S \to A$: $I, E_B(N_B, K), E_A(N_A, K)$,

(4) $A \to B$: $I, E_B(N_B, K)$.

3. The inter-protocol interleaving attacks

In this section we present two inter-protocol interleaving attacks. One uses the intercepted messages of the NS′ protocol against the KSL protocol and the other uses the intercepted messages of the OR protocol against the Kerberos protocol.

In the attacks, we shall use C_X to denote the attacker who masquerades as the legal principal X. Since all attacks on cryptographic protocols depends on some assumptions about implementation details [15], for each such attack we make some implementation assumptions first. Although the last two assumptions for the attack of using the OR protocol against the Kerberos protocol are not quite reasonable, it serves as another example to demonstrate that such attack is possible.

3.1. Use the NS′ protocol against the KSL protocol

The implementation assumptions for this attack are as follows.

(i) The shared secret key K_{XS} between the server S and the principal X is the same for both of the NS′ and KSL protocols, which also use the same secret-key cryptosystem. Since there are standards for secret-key cryptosystems, it is very likely that two protocols use the same secret-key cryptosystem.

(ii) The timestamp T_A is not distinguishable from a session key K, that is, when the timestamp T_A appears in the position of the session key K, the principal will treat the timestamp T_A as the session key K.

The attack of using the NS′ protocol against the KSL protocol is shown in Fig. 1 and described as follows. In the end, the principal A of the KSL protocol mistreats the timestamp T_A as the session key and is not aware that C_B is an attacker.

Step 1. When the principal A of the KSL protocol initiates a session, the attacker C_B intercepts the message N_A, A.

Step 2. The attacker C_B initiates a session of the NS′ protocol and sends N_A, B to the principal A of the NS′ protocol, in which N_A is treated as the nonce N_B from B.

Step 3. The principal A of the NS′ protocol then sends A, $E_A(N_A, B, T_A)$, N'_A to the server S. The

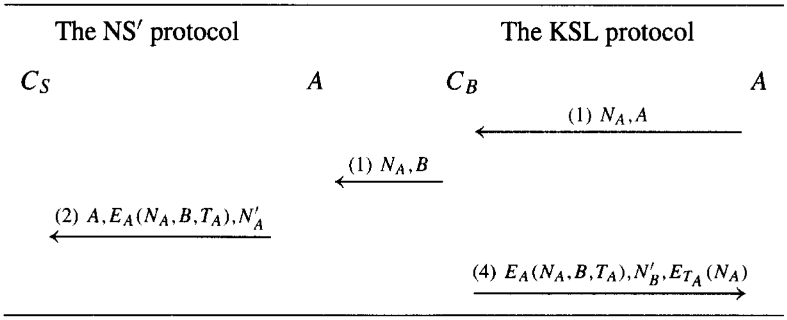

Fig. 1.

message is intercepted by C_S, who is also the attacker C_B. The nonce N'_A is issued by the principal A for the NS′ protocol.

Step 4. The attacker C_B bypasses the second and third message passes of the KSL protocol.

Step 5. The attacker C_B then sends $E_A(N'_A, B, T_A)$, N'_B, $E_{T_A}(N_A)$ to the principal A of the KSL protocol as the fourth message pass. After receiving the message from C_B, the principal A shall misinterpret the timestamp T_A as the session key of the session of the KSL protocol.

Step 6. After Step 5, the attacker C_B simply ignores the reply $E_{T_A}(N'_B)$ from the principal A of the KSL protocol and aborts the session of the NS′ protocol.

Step 7. In the end, the attacker C_B impersonates the principal B of the KSL protocol and uses the key T_A to communicate with the principal A in the KSL protocol.

We make a remark here. Although the timestamp T_A is not known to the attacker C_B, there is a constant probability that C_B can derive it, for example, by observing the system time of the principal A. By a common accepted concept about the security of cryptographic protocols [12], the attack succeeds. Furthermore, there is an attack against the Kerberos protocol using the weakness of timestamps [7]. Thus, encrypted timestamps should not be considered as secure.

3.2. *Using the OR protocol against the Kerberos protocol*

The implementation assumptions for this attack are as follows.

(1) The shared secret key K_{XS} between the server S and the principal X is the same for both the OR and Kerberos protocols, which also use the same secret-key cryptosystem.

(2) An identity name B is not distinguishable from a session key K, that is, when the identity name B appears in the position of the session key K, the principal will treat the identity name B as the session key K.

(3) The nonce N_B of the OR protocol is not detected and is treated as the timestamp T_S issued by the server S by the principal B of the Kerberos protocol. In [14], the field for N_B is actually a challenge issued by the principal B. An implementer may use a timestamp for it. Therefore, this assumption is possible.

(4) The identifier I of the OR protocol is not detected and is treated as the lifetime L by the principal B of the Kerberos protocol.

The attack of using the OR protocol against the Kerberos protocol is shown in Fig. 2 and described as follows. In the end, the principal B of the Kerberos protocol mistreats the identity name B as the session key K (i.e., $B = K$) and is not aware that C_A is an attacker to impersonate the principal A.

Step 1. When the principal B of the OR protocol initiates a session, the attacker C_A intercepts the message.

Step 2. The attacker C_A initiates a session of the Kerberos protocol while ignoring the first two message passes of the session.

Step 3. The attacker C_A treats B as the session key K and generates the message $E_K(A, T_A)$. C_A then sends $E_K(A, T_A)$, together with the intercepted message in Step 1, as the third message pass of the Kerberos protocol to the principal B.

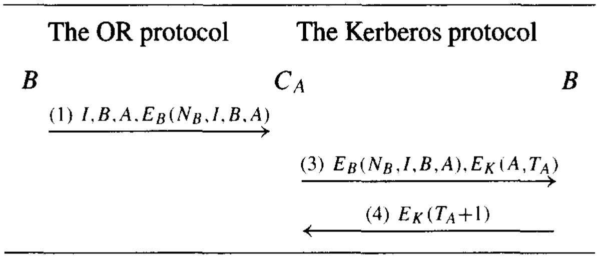

Fig. 2.

Step 4. When the principal B of the Kerberos protocol receives the message from C_A, he misinterprets the nonce N_B as the timestamp T_S, the identifier I as the lifetime L and the identity name B as the session key K without careful checking.

Step 5. B sends the message $E_K(T_A+1)$ to C_A as the fourth message pass of the Kerberos protocol.

Step 6. The attacker C_A simply ignores the message in Step 5 and aborts the OR protocol with the principal B.

Step 7. In the end, the attacker C_A can cheat the principal B of the Kerberos protocol by pretending the identity name B as the session key K in the session of the Kerberos protocol. Thus, the attacker C_A impersonates the principal A of the Kerberos protocol and uses the key B to communicate with the principal B in the Kerberos protocol.

4. Conclusion

We have presented the inter-protocol interleaving attack on some well-known trusted server and secret-key cryptosystem based authentication and key distribution protocols. The methods that can prevent the interleaving attack can usually be used to strengthen the protocol against the inter-protocol interleaving attack. As discussed in [1,15], adding direction bits to the message passes and associating messages with types can effectively deter the attack. We would suggest supplementally that each message pass of the protocol is associated not only with the direction bits but also with the protocol name. Of course, as discussed in the beginning, not using shared keys in two or more protocols can prevent the attack.

References

[1] M. Abadi, R. Needham, Prudent engineering practice for cryptographic protocols, IEEE Trans. Software Engrg. 22 (1) (1996) 6–15.

[2] R. Bird, I. Gopal, A. Herzberg, P. Janson, S. Kutten, R. Molva, M. Yung, Systematic design of a family of attack-resistant authentication protocols, IEEE J. Selected Areas Comm. 11 (5) (1993) 679–693.

[3] M. Burrows, M. Abadi, R. Needham, A logic of authentication, ACM Trans. Computer Systems 18 (1) (1990) 18–36.

[4] CCITT Recommendation X.509, The directory-authentication framework, Consultation Committee, International Telephone and Telegraph, International Telecommunications Union, Geneva, Switzerland, 1993.

[5] Carlsen, Cryptographic protocol flaws, in: Proc. Computer Security Foundations Workshop VII, IEEE Computer Society Press, 1994.

[6] ISO/IEC 9798, Entity authentication mechanisms, International Organization for Standardization, Geneva, Switzerland, 1993.

[7] L. Gong, A security risk of depending on synchronized clocks, Operat. Systems Review 26 (1) (1992) 49–53.

[8] T. Hwang, N.-Y. Lee, C.-M. Li, M.-Y. Ko, Y.-H. Chen, Two attacks on Neuman–Stubblebine authentication protocols, Inform. Process. Lett. 53 (1995) 103–107.

[9] A. Kehne, J. Schonwalder, H. Langendorfer, A nonce-based protocol for multiple authentications, Operat. Systems Review 26 (4) (1992) 84–89.

[10] B.C. Neuman, S.G. Stubblebine, A note on the use of timestamp as nonces, Operat. Systems Review 27 (2) (1993) 10–14.

[11] R.M. Needham, M.D. Schroeder, Using encryption for authentication in large networks of computers, Comm. ACM 21 (12) (1978) 993–999.

[12] B. Schneier, Applied Cryptography: Products, Algorithms, and Source Code in C, 2nd edn, John Wiley & Sons, New York, 1996.

[13] J.G. Steiner, B.C. Neuman, J.I. Schiller, Kerberos: An authentication service for open network systems, in: USENIX Conference Proceedings, 1988, pp. 191–202.

[14] D. Otway, O. Rees, Efficient and timely mutual authentication, Operat. Systems Review 21 (1) (1987) 8–10.

[15] P. Syverson, On key distribution protocols for repeated authentication, Operat. Systems Review 27 (4) (1993) 24–30.

[16] P. Syverson, A taxonomy of replay attacks, in: Proc. Computer Security Foundations Workshop VII, IEEE Computer Society Press, 1994.