ELSEVIER

Information Processing Letters 53 (1995) 103–107

Information Processing Letters

Two attacks on Neuman–Stubblebine authentication protocols

Tzonelih Hwang *, Narn-Yih Lee, Chuan-Ming Li, Ming-Yung Ko, Yung-Hsiang Chen

Institute of Information Engineering, National Chen-Kung University, Tainan, Taiwan, ROC

Communicated by K. Ikeda; received 12 July 1993; revised 25 April 1994

Abstract

In 1993, Neuman and Stubblebin proposed a nonce-based mutual authentication protocol uing timestamps as nonces. In this paper, we show defects in their initial and subsequent authentication protocols. We give the reasons that cause these defects and modify their protocol to avoid these defects.

Keywords: Cryptanalysis; Authentication; Encryption/decryption

1. Introduction

The advance in the technology of computer networks has led to the development of many distributed computing systems. In these distributed systems, it is particularly important for any two principals to authenticate each other before communications. On the design of user authentication protocols (e.g. Kerberos [5]), it is quite often to use timestamps to guarantee the freshness of the message. Timestamps are also useful in protecting the system from replay attacks and supporting multiple accesses to an authentication server. But, the use of timestamps requires the assumption of synchronous clocks in the network, which suffers from several security risks [2].

Kehne, Schonwalder, and Langendorfer proposed a protocol for multiple authentications [3]. Their protocol has the same features as Kerberos and does not rely on the existence of synchronous clocks. Nonces are used to support multiple accesses as well as guarantee the freshness of the message in their protocol too.

Later, Neuman and Stubblebine proposed another nonce-based protocol for multiple authentications, which uses timestamps as nonces [4]. They claim that their protocol needs fewer messages than the approach of Kehne et al. [3] and has the same number of messages as required for mutual authentication in Kerberos. Their idea is to let the verifier issue timestamps as nonces, and the timestamps are also checked later by the issuer himself. Therefore, it only depends on the local clock.

In this paper, we aim to present two attacks on the Neuman–Stubblebine scheme [4] and propose methods to avoid these attacks. In Section 2, we review the scheme proposed by Neuman and Stubblebine [4]. In Section 3, two attacking methods are given. One, called the paradox attack, is used to show that the initial exchange protocol in [4] is insecure. The other, called the parallel

* Corresponding author.

SSDI 0020-0190(95)00177-4

session attack, is used to show the weakness of the subsequent authentication protocol in [4]. In Section 4, we propose a modified scheme to prevent the system from these attacks. Finally, a concluding remark will be given in Section 5.

2. Review of the Neuman-Stubblebine authentication protocol

In this section, we will review the Neuman–Stubblebine authentication protocol. Let us first describe the notation used in this paper as follows:

N_x:	a nonce generated by the principal X.
K_{xs}:	the secret key shared by the principal X and a trusted authentication server S.
K_{xy}:	the secret key shared by the principal X and Y.
$\{M\}K$:	the message M encrypted by the key K.
T_x:	the expiration time suggested by the principal X.

2.1. The initial authentication

Let us assume that the principal A wants to have a secure conversation with the principal B. A trusted authentication server (S), who shares a secret key with each principal, is used to generate session keys for the requesting principals. A starts the initial authentication (see Fig. 1) by sending to B a cleartext message in (1), which contains A's identity and a nonce N_a. Once B receives the message, he then sends to the authentication server S the message in (2), which contains B's identity, a nonce N_b and a ciphertext $\{A, N_a, T_b\}K_{bs}$.

Upon receiving the message from B, S decrypts the ciphertext with the key K_{bs} and generates a session key K_{ab} for A and B. S then sends to A a ciphertext $\{B, N_a, K_{ab}, T_b\}K_{as}$, a ticket $\{A, K_{ab}, T_b\}K_{bs}$ and the nonce N_b as the message in (3).

Upon receiving the message in (3), A decrypts the ciphertext and verifies whether the nonce N_a is the same as the one included in the message in (1). If yes, A will accept the session key K_{ab} as a valid one. In the message in (4), A forwards the ticket to B together with the nonce N_b encrypted by the session key K_{ab} to prove his/her identity to B. Fig. 1 shows the initial authentication of Neuman-Stubblebine protocol.

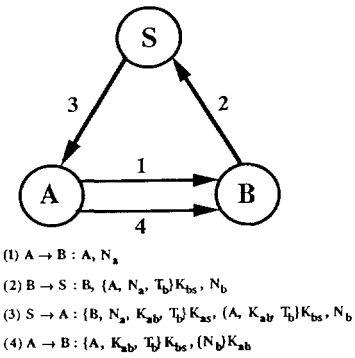


Fig. 1. The initial exchange of Neuman's protocol.

2.2. The subsequent authentication

After the initial authentication, A posseses a ticket and a session key that may be used for the subsequent communications with B without asking the authentication server again for key distribution. The subsequent authentication of [1] is described as follows:

(1) $A \rightarrow B$: $N'_a, \{A, K_{ab}, T_b\}K_{bs}$

(2) $B \rightarrow A$: $N'_b, \{N'_a\}K_{ab}$

(3) $A \rightarrow B$: $\{N'_b\}K_{ab}$

If A tries to communicate with B, A directly sends the ticket and a new nonce, N'_a, to B. Upon receiving the ticket, B decrypts the ticket and obtains the session key K_{ab}. B then sends back the $\{N'_a\}K_{ab}$ and a new nonce, N'_b, to A. Finally, A authenticates the identity of B and also sends the $\{N'_b\}K_{ab}$ back to B to prove his/her identity.

3. Attacks on the Neuman–Stubblebine protocols

In this section, we propose two attacks on the initial and subsequent authentication protocols respectively.

3.1. The paradox attack

Assume that anyone can intercept messages from the network. Then, in the initial authentication protocol, an attacker X can pretend to be A by sending A's identity and a nonce N_x to B. While B sends message (2) to S, X intercepts the ciphertext $\{A, N_x, T_b\}K_{bs}$ and the nonce N_b generated by B from the communication channel. X ignores the message (3) (bypasses Step (2) and Step (3)) and sends $\{A, N_x, T_b\}K_{bs}$ together with $\{N_b\}N_x$ as the message (4) to B. Because both $\{A, N_x, T_b\}K_{bs}$ and $\{A, K_{ab}, T_b\}K_{bs}$ have the same format in the Neuman–Stubblebine protocol, B cannot distinguish one from the other and will mistake the N_x as the session key K_{ab} issued by S. B then decrypts $\{N_b\}N_x$ with the key N_x and verifies the result is the nonce, N_b, originally generated by himself. Finally, B accepts N_x as the session key shared by X and B. This is called the *paradox attack* [6]. By running this paradox attack, X can impersonate A to possess a ticket and a session key shared with B. Fig. 2 shows the paradox attack to the initial exchange of Neuman-Stubblebine protocol.

3.2. The parallel session attack

The *parallel session attack* [2] on the subsequent authentication protocol is shown in Fig. 3.

When A tries to communicate with B after obtaining a ticket from S by the initial exchange protocol, A performs the subsequent authentication protocol by sending the ticket and a newly generated nonce, N'_a, to B. Assume that an attacker X tries to impersonate A. X first intercepts the ticket. Although X cannot decrypt the cipher, X may send the intercepted ticket and a forged nonce, N'_X, to B. Upon receiving the message, B verifies the validity of the ticket. If it is valid, B responds $\{N'_x\}K_{ab}$ and a new nonce, N'_b, to A. Once X intercepts this message, he/she uses B as an oracle and starts a new session with B. X sends the nonce, N'_b, he just received from B and the same ticket to B. Upon receiving the message, B sends back $\{N'_b\}K_{ab}$ and a new nonce, N''_b, to A. Now, X can intercept it and get the encrypted nonce $\{N'_b\}K_{ab}$. Finally, X successfully passes the first authentication session of B by sending the $\{N'_b\}K_{ab}$ back to B.

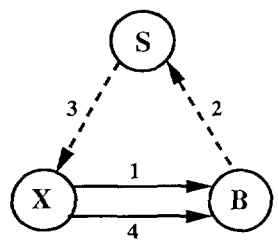


Fig. 2. The paradox attack to the initial exchange.

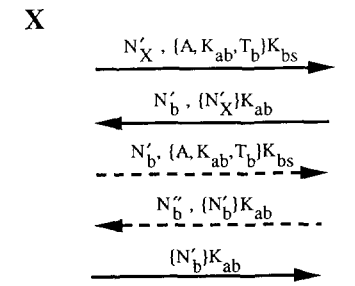


Fig. 3. Parallel session attack.

4. How to avoid the foregoing attacks

In this section, we modify the initial and subsequent authentication protocols of Neuman and Stubblebine to avoid the foregoing attacks.

4.1. Modified initial exchange protocol

From the paradox attack of the initial exchange protocol, we find the problem of the protocol is in that the ciphertext of message (2) and the ticket of message (3) have the same format in the plaintext. Thus, the ciphertext of the message (2) can be used by the intruder as a ticket in message (4). Therefore, one way to avoid this attack is to modify the structure of the mes-

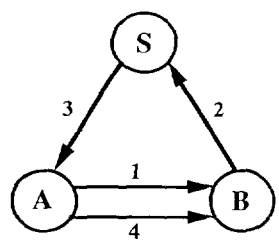
sage (2) into another format such that it cannot be used as a ticket anymore.

In Fig. 4, the nonce N_b is contained in the ciphertext of the message (2) for the purpose of discriminating the formats between the ciphertext of message (2) and the ticket of the message (4). It should be noted here that many other methods can also be used to achieve this purpose as long as the principal B can distinguish the ciphertext from the ticket easily. For example, simply exchange the enciphering order of the plaintext will make the ciphertext in (2) distinguishable from the ticket in the message (4).

4.2. Modified subsequent authentication protocol

From the Section 3.2, the parallel session attack can succeed because $\{N_b'\}K_{ab}$ can be extracted easily from the message in (2). Therefore, the impersonator X can use B as an oracle to get the answer of B's challenge from B himself. In order to avoid such attack, the message in (2) is modified in such a way that $\{N_b'\}K_{ab}$ cannot be obtained easily. The new subsequent authentication protocol is shown in Fig. 5.

In the new subsequent authentication protocol, we modify message (2) of the subsequent authentication protocol into $\{N_a', N_b'\}K_{ab}$. Suppose that an intruder wants to impersonate the principal A using the parallel session attack, he can only obtain a message containing two nonces encrypted by the session key K_{ab}. If the intruder sends the message to the principal B to prove his identity, then B will find the mismatch in the received message. Thus the attack cannot be successful.



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

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

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

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

Fig. 4. The new initial authentication protocol.

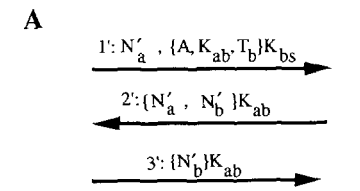


Fig. 5. The new subsequent authentication protocol.

5. Conclusions

In this paper, we have proposed the paradox attack and parallel session attack to show the weakness of the initial and subsequent authentication protocols of Neuman and Stubblebine respectively. We modify the original schemes to avoid these attacks. What we do not consider here is the attacks in which the attacker acts as a "relay" between the two parties and cuts the communication lines after mutual authentication and then takes the role of the legal party. However, this kind of attacks can be easily avoided by requesting that all communications have to be encrypted by the session key once both parties are mutually authenticated. Since the attacker does not have the session key, he/she cannot perform any further attacks. The readers are encouraged to challenge the security of the modified protocols.

Acknowledgement

The authors wish to thank the referees for their useful comments. This paper is supported by the National Science Council of R.O.C. under the contract number NSC 82-0408-E006-419.

References

[1] R. Bird, I. Gopal, A. Herzberg, P. Janson, S. Kutten, R. Molva and M. Yung, Systematic design of two-party authentication protocols in: *Proc. Crypto* '91 (1991) 44–61.

[2] Li Gong, A security risk of depending on synchronized clocks, *Operating Systems Rev.* **26** (1) (1992) 49–53.
[3] A. Kehne, J. Schonwalder and H. Langendorfer, A nonce-based protocol for multiple authentication, *Operating Systems Rev.* **26** (4) (1992) 84–89.
[4] B.C. Neuman and S.G. Stubblebine, A note on the use of timestamps as Nonces, *Operating Systems Rev.* **27** (2) (1993) 10–14.
[5] J.G. Steiner, G. Neuman and J.I. Schiller, Kerberos: An authentication service for open network systems, in: *USENIX Winter Conf.* (1988) 191–202.
[6] E. Snekkenes, Roles in cryptographic protocols, in: *Proc. IEEE Symp. on Security and Privacy* (1992) 105–119.