A weakness in Sun-Chen-Hwang's three-party key agreement protocols using passwords ⋆

Junghyun Nam, Seungjoo Kim, Dongho Won,

School of Information and Communication Engineering, Sungkyunkwan University, 300 Cheoncheon-dong, Jangan-gu, Suwon-si, Gyeonggi-do 440-746, Republic of Korea

Abstract

Recently, Sun, Chen and Hwang [J. Syst. Software, 75 (2005), 63–68] have proposed two new three-party protocols, one for password-based authenticated key agreement and one for verifier-based authenticated key agreement. In this paper, we show that both of Sun-Chen-Hwang's protocols are insecure against an active adversary who can intercept messages, start multiple sessions of a protocol, or otherwise control the communication in the network. Also, we present a simple solution to the security problem with the protocols.

Key words: Three-party key agreement; Password; Verifier; Active adversary

1 Introduction

Through the research and analysis conducted over many decades, it has now become well known how to design secure protocols for authenticated key exchange using high-entropy cryptographic keys as underlying information. But, the possibility of secure password-authenticated key exchange was recognized in the relatively recent work of Bellovin and Merritt (1992), which shows how to bootstrap a high-entropy cryptographic key from a weak, low-entropy password. When it comes to designing password-based protocols, one must vigilantly ensure that protocols are immune to password guessing attacks in which an adversary simply enumerates all possible passwords until it gets a

⋆ Submitted to the Journal of Systems and Software

Email addresses: `jhnam@dosan.skku.ac.kr` (Junghyun Nam), `skim@ece.skku.ac.kr` (Seungjoo Kim), `dhwon@dosan.skku.ac.kr` (Dongho Won).

match. Indeed, there is a long history of protocols for this domain being proposed and subsequently broken by password guessing attacks (see the paper of Lee et al. (2005) and its related work for a typical example).

Due in large part to the practical significance of password-based authentication, the initial work of Bellovin and Merritt (1992) has been extended to a number of settings, including a three-party model where an authentication server exists to help two communicating parties establish a common session key. Roughly a decade ago, Steiner et al. (1995) proposed a three-party protocol for password-based key agreement which builds on the earlier protocol, known as encrypted key exchange, or EKE, proposed by Bellovin and Merritt (1992) in the two-party setting. However, Ding and Horster (1995) have pointed out that Steiner et al.'s three-party EKE is vulnerable to a new type of attack called "undetectable on-line password guessing attack".

For this reason, Sun et al. (2005) have recently presented an improved version of Steiner et al.'s three-party EKE and in addition, proposed a new verifier-based key agreement protocol. In order to provide resistance to undetectable on-line password guessing attacks, they consider a "hybrid" model in which the clients store the server's public key in addition to sharing a password (or a password verifier) with the server. However, the authors of this work, while they focus on undetectable on-line password guessing attacks, reproduce the same mistake that has been pointed out over the years. In this paper, we identify the vulnerability of Sun et al.'s protocols to an attack mounted by an active adversary who can intercept and inject messages, start multiple sessions of a protocol, or otherwise control communication flows in the network. Then we present a simple patch which fixes the security problem with the protocols.

2 Review of Sun et al.'s three-party key agreement protocols

There are three entities involved in the protocols: the authentication server S, and two clients A and B who wish to establish a session key between them. We assume that the public key pk of server S is known in advance to all parties in the network.

2.1 Improved three-party EKE

In the protocol, we assume that two passwords P_A and P_B respectively of A and B are known to S via a secure channel. A bird's-eye view of the protocol is given in Fig. 1 and a more detailed description is as follows:

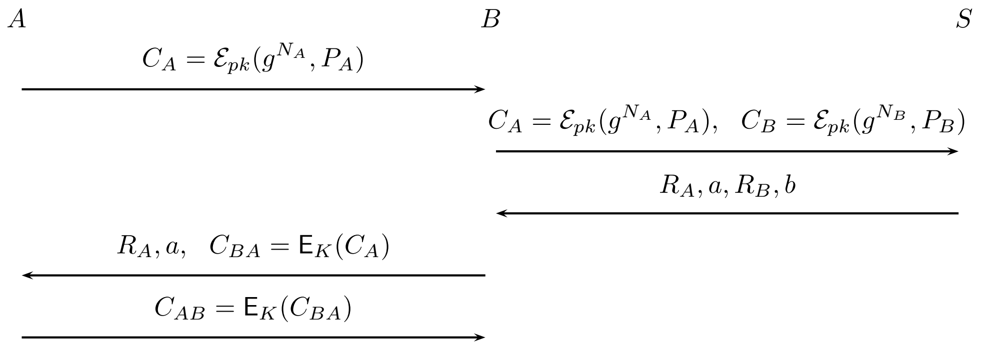

Fig. 1. Improved three-party EKE

(1) The client A chooses a random number N_A and computes g^{N_A}. Then A sends

$$C_A = \mathcal{E}_{pk}(g^{N_A}, P_A)$$

to the client B, where $\mathcal{E}_{pk}(g^{N_A}, P_A)$ denotes the ciphertext of the message (g^{N_A}, P_A) encrypted using the public-key encryption algorithm $\mathcal{E}$ under the key pk.

(2) The client B chooses a random number N_B, computes g^{N_B}, and sends to S the ciphertexts C_A and

$$C_B = \mathcal{E}_{pk}(g^{N_B}, P_B).$$

(3) The server S verifies the validity of the passwords by decrypting the ciphertexts C_A and C_B. If the verification succeeds, S chooses three random numbers a, b and N_S, and computes

$$R_A = (g^{N_B} \cdot g^b)^{N_S},$$
$$R_B = (g^{N_A} \cdot g^a)^{N_S}.$$

Here, a and b can be chosen to be such that $a, b \ll N_S$. Then S sends (R_A, a, R_B, b) to B.

(4) B computes the session key K as

$$\begin{aligned} K &= R_B^{N_B+b} \\ &= g^{(N_A+a)(N_S)(N_B+b)} \end{aligned}$$

and sends R_A, a, and $C_{BA} = \mathsf{E}_K(C_A)$ to A, where $\mathsf{E}_K(C_A)$ denotes the ciphertext of C_A encrypted using symmetric encryption algorithm E under the key K.

(5) Now, A computes the session key K as

$$\begin{aligned} K &= R_A^{N_A+a} \\ &= g^{(N_B+b)(N_S)(N_A+a)} \end{aligned}$$

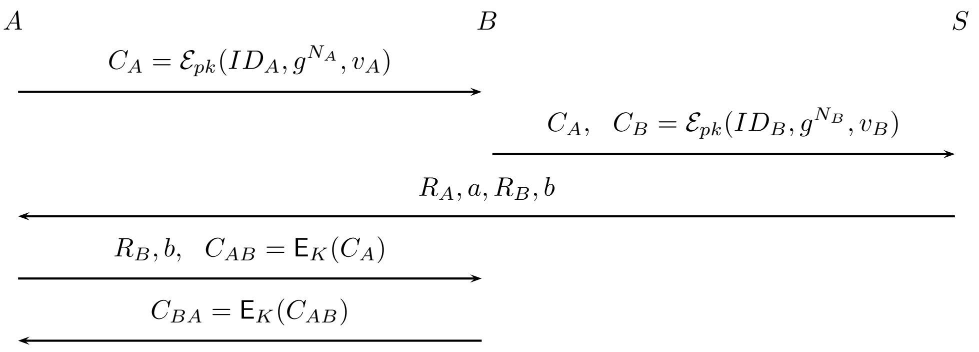

Fig. 2. Three-party verifier-based key agreement protocol

and verifies it through decryption of C_{BA}. Finally, A sends $C_{AB} = \mathsf{E}_K(C_{BA})$ to the client B, who will in turn decrypt it to check whether A has computed the correct session key K.

2.2 *Three-party verifier-based key agreement protocol*

We now review Sun et al.'s three-party verifier-based key agreement protocol (the basic scheme in Section 4.1 of Sun et al. (2005)). Let x_A and x_B be two private integers derived in any predetermined way respectively from P_A and P_B. Then we assume that the verifiers $v_A = g^{x_A}$ and $v_B = g^{x_B}$ respectively of P_A and P_B are known in advance to the server S via a secure channel. As pictured in Fig. 2, the protocol proceeds similarly as the improved three-party EKE described in the previous subsection, but using the verifiers in place of the passwords. The details of the protocol are as follows:

(1) The client A chooses a random number N_A, computes g^{N_A}, and sends

$$C_A = \mathcal{E}_{pk}(ID_A, g^{N_A}, v_A)$$

to the client B.

(2) The client B chooses a random number N_B, computes g^{N_B}, and sends to the server S the ciphertexts C_A and

$$C_B = \mathcal{E}_{pk}(ID_B, g^{N_B}, v_B).$$

(3) After decrypting C_A and C_B and checking the verifiers v_A and v_B, the server S chooses three random numbers a, b and N_S, and computes

$$R_A = (g^{bN_B} \cdot v_B)^{N_S},$$
$$R_B = (g^{aN_A} \cdot v_A)^{N_S}.$$

S then sends $(R_A,\ a,\ R_B,\ b)$ to A.

(4) A computes the session key K as

$$\begin{aligned} K &= R_A^{aN_A+x_A} \\ &= g^{(bN_B+x_B)(N_S)(aN_A+x_A)} \end{aligned}$$

and sends $(R_B,\, b,\, C_{AB} = \mathsf{E}_K(C_A))$ to B.

(5) Now, B computes the session key K as

$$\begin{aligned} K &= R_B^{bN_B+x_B} \\ &= g^{(aN_A+x_A)(N_S)(bN_B+x_B)} \end{aligned}$$

and verifies it through decryption of C_{AB}. Finally, $C_{BA} = \mathsf{E}_K(C_{AB})$ is sent to the client A who can then confirm that B has computed the same session key K.

3 Security Analysis

In this section we show that both of Sun et al.'s three-party key agreement protocols are insecure in the presence of an active adversary.

3.1 Attack on the improved three-party EKE

A high-level depiction of the attack is shown in Fig. 3, where M denotes an adversary, and a dashed line indicates that the corresponding flow is blocked by M from reaching the destination. In this attack, we assume that the adversary M is a legitimate user who is registered with the authentication server S. The goal of adversary M is to share a session key with A by masquerading as B and to share another session key with B by masquerading as A. To achieve this goal, the adversary faces the server S with her true identity, while sitting in between the clients and the server to intercept and inject messages for her own sake. The detailed attack scenario is as follows:

(1) As a preliminary step, M chooses two random numbers N_M and N'_M and computes g^{N_M}, $g^{N'_M}$, $C_M = \mathcal{E}_{pk}(g^{N_M},\, P_M)$ and $C'_M = \mathcal{E}_{pk}(g^{N'_M},\, P_M)$.
(2) The adversary M launches the attack by intercepting the message going to the server (i.e., the message $(C_A,\, C_B)$ sent from B to S). After intercepting this message, M immediately sends two separate messages $(C_A,\, C_M)$ and $(C_B,\, C'_M)$ to S alleging that she wants to establish two concurrent sessions, each with A and B.
(3) Since $(C_A,\, C_M)$ and $(C_B,\, C'_M)$ are both valid, the server S constructs two response messages, one for the session between A and M and the

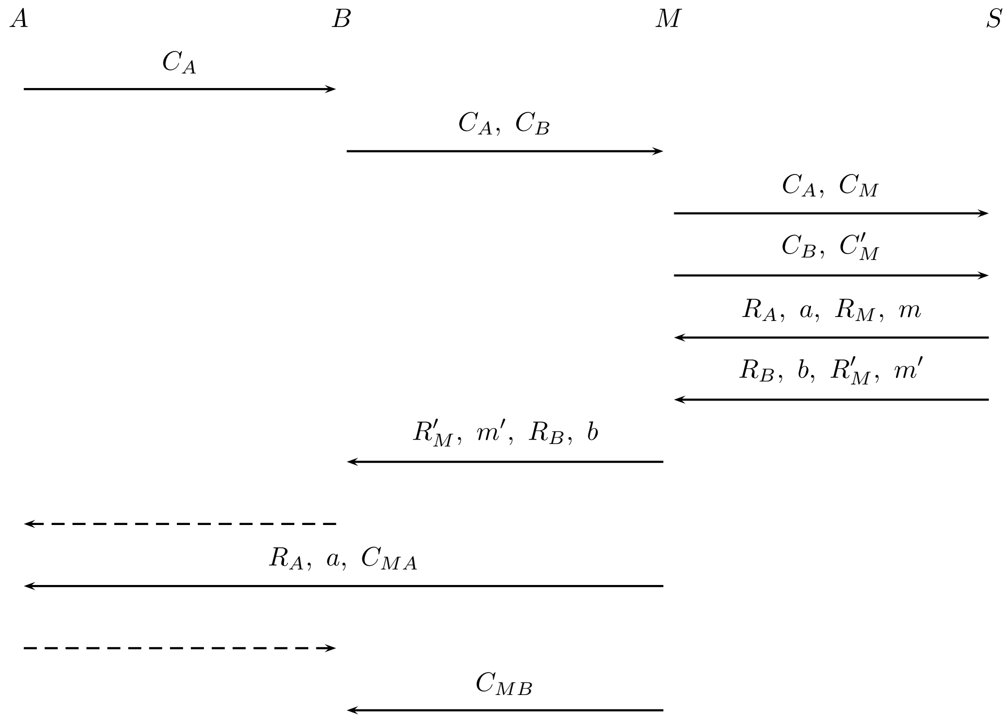

Fig. 3. An attack on the improved three-party EKE

other for the session between B and M, as specified in the protocol; it chooses two pairs of triple random numbers $(a,\ m,\ N_S)$ and $(b,\ m',\ N'_S)$ and computes

$$\begin{aligned}
R_A &= (g^{N_M} \cdot g^{m})^{N_S},\\
R_M &= (g^{N_A} \cdot g^{a})^{N_S},\\
R_B &= (g^{N'_M} \cdot g^{m'})^{N'_S},\\
R'_M &= (g^{N_B} \cdot g^{b})^{N'_S}.
\end{aligned}$$

S then sends back two messages $(R_A,\ a,\ R_M,\ m)$ and $(R_B,\ b,\ R'_M,\ m')$ to the adversary M.

(4) After receiving the two messages from S, the adversary M computes two session keys K and K' to be shared respectively with A and B as follows:

$$\begin{aligned}
K &= g^{(N_A+a)(N_S)(N_M+m)} = R_M^{N_M+m},\\
K' &= g^{(N_B+b)(N'_S)(N'_M+m')} = {R'}_M^{N'_M+m'}.
\end{aligned}$$

M then sends the message $(R'_M,\ m',\ R_B,\ b)$ to B alleging that it comes from S.

(5) The client B thinks that the message $(R'_M,\ m',\ R_B,\ b)$ is the response to $(C_A,\ C_B)$ from S. Hence, as per the protocol specification, B computes its session key as

$$K' = g^{(N'_M+m')(N'_S)(N_B+b)} = R_B^{N_B+b}$$

and sends the message (R'_M, m', $C_{BA} = \mathsf{E}_{K'}(C_A)$) to A. But this message is replaced with (R_A, a, $C_{MA} = \mathsf{E}_K(C_A)$) by the adversary.

(6) A thinks that the message (R_A, a, C_{MA}) is from B. A hence computes its session key as

$$K = g^{(N_M+m)(N_S)(N_A+a)} = R_A^{N_A+a},$$

verifies that the decryption of C_{MA} under K is equal to C_A, and then sends $C_{AB} = \mathsf{E}_K(C_{MA})$ to B. But this message is also replaced with $C_{MB} = \mathsf{E}_{K'}(C_{BA})$ by the adversary.

Through the attack, the authentication mechanism of the protocol is completely compromised. At the end of this scenario, the client A believes that he has established a secure session with B sharing a secret key K, while in fact he has shared the key with the adversary M. Similarly, B thinks that he has shared with A a session key K' which indeed is shared with M. As a result, the adversary M can not only access and relay any confidential communications between A and B, but can also send arbitrary messages for her own benefit impersonating one of them to the other.

The weakness of Sun et al.'s three-party EKE against the attack above is mainly because that the messages sent to the server by the clients in one run of the protocol can be replayed in another run even with a different set of clients. Thus, fortunately, the patch is simple. It suffices to modify the computations of C_A and C_B to the following:

$$C_A = \mathcal{E}_{pk}(ID_A, ID_B, g^{N_A}, P_A),$$
$$C_B = \mathcal{E}_{pk}(ID_B, ID_A, g^{N_B}, P_B).$$

With this modification, the messages sent to the server in each run of the protocol become bounded to the identities of the clients participating in that run, and therefore, such an attack presented above would be impossible.

3.2 Attack on the three-party verifier-based key agreement protocol

We now show that Sun et al.'s three-party verifier-based protocol suffers from an attack depicted in Fig. 4. This attack is essentially same as the one illustrated in the previous subsection. The same attack works because there is still no way for the server to verify whether the two incoming ciphertexts are paired honestly. Note that as in $\mathcal{E}_{pk}(ID_A,\ g^{N_A},\ v_A)$, the inclusion of the sender's identity as part of the plaintext does not play any role to prevent a client's ciphertext from being paired with an unintended ciphertext generated by the adversary. Again, the goal of adversary M is to share a session key with

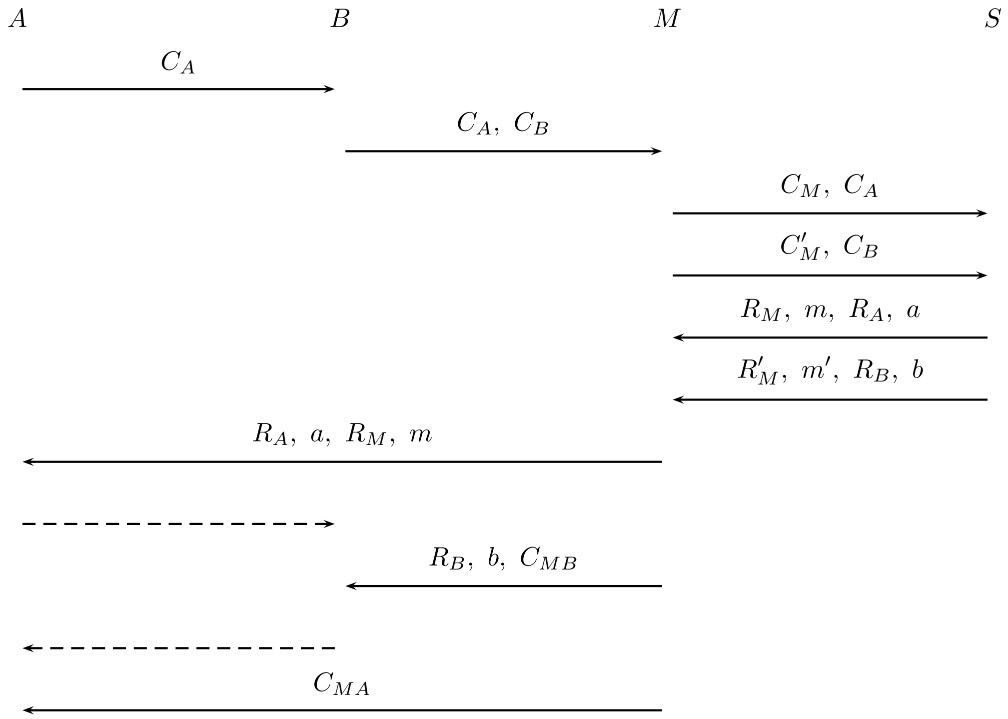

Fig. 4. An attack on the three-party verifier-based key agreement protocol

each client separately, while deluding the clients into believing that they have established a secure session between them. The attack scenario is as follows:

(1) To begin with, the adversary M chooses two random numbers N_M and N'_M and computes g^{N_M}, $g^{N'_M}$, $C_M = \mathcal{E}_{pk}(ID_M,\ g^{N_M},\ v_M)$ and $C'_M = \mathcal{E}_{pk}(ID_M,\ g^{N'_M},\ v_M)$.
(2) M constructs two message pairs $(C_M,\ C_A)$ and $(C'_M,\ C_B)$ by intercepting $(C_A,\ C_B)$ sent to S by B. Then M sends to S the message $(C_M,\ C_A)$ alleging that it comes from A, and the message $(C'_M,\ C_B)$ alleging that it comes from B.
(3) Since S thinks that both A and B want to establish a session with M, it sends to M the two messages $(R_M,\ m,\ R_A,\ a)$ and $(R'_M,\ m',\ R_B,\ b)$ in response, respectively, to $(C_M,\ C_A)$ and $(C'_M,\ C_B)$ such that

$$\begin{aligned}
R_M &= (g^{aN_A} \cdot v_A)^{N_S},\\
R_A &= (g^{mN_M} \cdot v_M)^{N_S},\\
R'_M &= (g^{bN_B} \cdot v_B)^{N'_S},\\
R_B &= (g^{m'N'_M} \cdot v_M)^{N'_S}.
\end{aligned}$$

(4) Upon receiving the two messages from S, the adversary M computes two session keys K and K' to be shared respectively with A and B as follows:

$$K = g^{(aN_A+x_A)(N_S)(mN_M+x_M)} = R_M^{mN_M+x_M},$$
$$K' = g^{(bN_B+x_B)(N'_S)(m'N'_M+x_M)} = R'^{m'N'_M+x_M}_M.$$

M then sends the message (R_A, a, R_M, m) to A alleging that it comes from S.

(5) After receiving (R_A, a, R_M, m), the client A computes its session key K as

$$K = g^{(mN_M+x_M)(N_S)(aN_A+x_A)} = R_A^{aN_A+x_A}$$

and sends the message (R_M, m, $C_{AB} = \mathsf{E}_K(C_A)$) to B. But, M intercepts this message and instead sends (R_B, b, $C_{MB} = \mathsf{E}_{K'}(C_A)$) to B.

(6) Upon receiving the message (R_B, b, C_{MB}), B computes its session key as

$$K' = g^{(m'N'_M+x_M)(N_S)(bN_B+x_B)} = R_B^{bN_B+x_B}$$

and verifies that the decryption of C_{MB} under K' is equal to C_A. B then sends the response $C_{BA} = \mathsf{E}_{K'}(C_{MB})$ to A. But, M intercepts this message and instead sends $C_{MA} = \mathsf{E}_K(C_{AB})$ to A.

This scenario leads to the same consequence as stated at the end of the attack scenario for the improved three-party EKE, and the exact same solution as given there can be applied to this case. That is, as a countermeasure to our attack, the computations of C_A and C_B are modified slightly as follows:

$$C_A = \mathcal{E}_{pk}(ID_A, ID_B, g^{N_A}, v_A),$$
$$C_B = \mathcal{E}_{pk}(ID_B, ID_A, g^{N_B}, v_B).$$

Finally, we remark that the four-round verifier-based protocol (the optimal round scheme) described in Section 4.2 of Sun et al. (2005) also suffers from a similar problem as the one studied above. We do not detail it in this paper due to the similarity.

4 Conclusion

We have shown that the three-party key agreement protocols proposed by Sun et al. (2005) are susceptible to an attack mounted by an active adversary. But fortunately, the security hole identified here can be easily patched by integrating participants' identities as part of the message being encrypted by each client.

References

Bellovin, S.M., Merritt, M., 1992. Encrypted key exchange: password-based protocols secure against dictionary attacks. In: Proceedings of the 1992 IEEE Computer Society Conference on Research in Security and Privacy, pp. 72–84.

Ding, Y., Horster, P., 1995. Undetectable on-line password guessing attacks. ACM SIGOPS Operating Systems Review 29 (4), 77–86.

Lee, S.-W., Kim, H.-S., Yoo, K.-Y., 2005. Improvement of Lee and Lee's authenticated key agreement scheme. Applied Mathematics and Computation, in press. Available, with subscription, at http://www.sciencedirect.com/science/journal/00963003/.

Steiner, M., Tsudik, G., Waidner, M., 1995. Refinement and extension of encrypted key exchange. ACM SIGOPS Operating Systems Review 29 (3), 22–30.

Sun, H.-M., Chen, B.-C., Hwang, T., 2005. Secure key agreement protocols for three-party against guessing attacks. The Journal of Systems and Software 75, 63–68.