Available online at www.sciencedirect.com

ScienceDirect

Computer Standards & Interfaces 30 (2008) 95–99

COMPUTER STANDARDS & INTERFACES

www.elsevier.com/locate/csi

Security enhancement for a three-party encrypted key exchange protocol against undetectable on-line password guessing attacks

Hsing-Bai Chen [a], Tzung-Her Chen [b,*], Wei-Bin Lee [a], Chin-Chen Chang [a]

[a] Department of Information Engineering and Computer Science, Feng Chia University, Taiwan, ROC

[b] Department of Computer Science and Information Engineering, National Chiayi University, Taiwan, ROC

Received 2 September 2006; received in revised form 13 July 2007; accepted 10 August 2007
Available online 24 August 2007

Abstract

In 1995, a potential attack, called undetectable on-line password guessing attack, on three-party encrypted key exchange (3PEKE) protocol is highlighted by Ding and Horster. Since then, this attack has been one of the main concerns for developing a secure 3 PEKE protocol. Recently, Chang and Chang proposed a password-based three-party encrypted key exchange protocol that simultaneously possesses round and computation efficiencies. However, this paper shows that their protocol is potentially vulnerable toward undetectable on-line password guessing attacks. As their protocol is currently one of the most superior of all 3PEKE approaches; it seems worthwhile and valuable to remedy this potential security problem.

© 2007 Elsevier B.V. All rights reserved.

Keywords: Authentication; Three-party encrypted key exchange; Password guessing attack; Password

1. Introduction

Since passwords are able to be freely chosen and can be fairly easily memorized without any assistant storage device, password-based methods are widely used for user authentication. From the security perspective, protecting the low-entropy password from dictionary attacks is crucial for password-based authentication methods. For this reason, Bellovin and Merritt [2] showed a protocol *for password-based authentication and key agreement*, known as encrypted key exchange (EKE), which is capable of bootstrapping a high-entropy cryptographic key from a pre-shared but possibly weak password for any two communication participants *A* and *B* to protect the password against dictionary attacks. Since every two participants ought to share a password in advance, EKE is inconvenient in a large communication environment. Then, extending EKE, three-party EKE in which each participant only shares a password in advance with a trusted server that helps any two participants to establish a session key is proposed. Nowadays, Kerberos [13] and KryptoKnight [14] are two of the conventional three-party EKE services, as pointed in [9].

Recently, there are more and more efforts mounted to focus on key exchange protocols and furthermore standardize such all of the password-based EKE protocols in IEEE P1363.2 [6] and ISO/IEC FDIS 11770-4 [8]. In the standard IEEE Std1363a-2002 [7], such EKE protocols are defined and divided into two classes — balanced password-based authenticated key agreement schemes and augmented password-based authenticated key agreement schemes. The former is defined as that a participant and a server use a shared password to negotiate an ephemeral session key such that the key is established accordingly. In the latter, a participant, who has a password, and a server, which holds an image of the password, can negotiate an ephemeral session key such that the key is established if and only if the image corresponds to the password. This paper only focuses on the former — balanced password-based authenticated key agreement schemes.

The most distinguishable characteristic of the EKE scheme is that the security of EKE must incorporate protection against dictionary attacks. It is disappointed that Kerberos and KryptoKnight are susceptible to dictionary attacks with low-entropy password. In an attempt to bring about a more robust three-party EKE scheme, Ding and Horster [5] introduced three possible types of attacks. The detectable on-line guessing and the off-line

[*] Corresponding author. Department of Computer Science and Information Engineering, National Chiayi University 300 University Road, Chia-Yi City, Taiwan 600, ROC. Fax: +86 886 5 2717741.

E-mail address: thchen@mail.ncyu.edu.tw (T.-H. Chen).

0920-5489/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.csi.2007.08.010

guessing are two well known attacks on the password-based EKE. Additionally, an unusual attack, called undetectable on-line password guessing attack, was discovered to be a powerful and agile form of attack. The central concept of this attack is to assume the existence of a legal but malicious participant, who tries to guess another legal participant's password through continual polling of the server with the guessed passwords. If the response sent back from the server has a clue to checking the guess, this can be a rather effective form of attack as it is undetectable by the server, as its name implies. In the Steiner et al.'s three-party EKE (STW for short) scheme [15], the one-time message used to establish a session key is directly encrypted with the participant A/B's password as a request as well as a challenge to the server. If the participant A/B is authenticated by the server, a response relative to the one-time message is sent back to the participant A/B. This gives the legitimate but malicious participant, assuming B in this case, a chance to guess the password of another participant A. Eventually, the undetectable on-line password guessing attack was mounted on the STW scheme successfully in [5].

To resolve this problem, both Lin et al.'s and Sun et al.'s separately proposed new schemes in [10] and [16] (called LSH and SCH), respectively. In the LSH and the SCH schemes, since the communicating participant's passwords and one-time messages are encrypted with the server's public key as a request, only the server can obtain the passwords and one-time messages with its own private key. Authentication of participants can be confirmed by the server upon verifying the validation of the passwords decrypted from the request. Without the knowledge of the server's private key, the malicious insider attacker has no feasible way to verify his/her guesses even if he/she receives a response sent from the server. A security weakness of the SCH scheme was recently revealed by Nam et al. [12], however, as the focus of this paper is centered toward devising an undetectable attack resistant three-party EKE scheme, this issue is somewhat irrelevant and thus will not be discussed in this paper.

However, both the LSH and SCH schemes make use of public-key cryptosystems which involve time-consuming computation cost. Consequently Lin et al. proposed an efficient non-public-key scheme [11] (called LSSH scheme) to counter this problem. In the LSSH scheme, the server can authenticate the participant in a way such that only those who possess the exact password can generate a valid response to server's challenge. At the same time, this scheme is also immune against the undetectable on-line password guessing attack, because the attacker has no clue to verifying his/her guesses.

The above three schemes have been illustrated to be secure enough to resist the undetectable on-line password guessing attacks. Taking computation cost into account, the LSH and SCH schemes impose a heavy burden on every participant compared with the LSSH scheme due to the public-key cryptographic computation. Taking communication cost into account, the LSSH scheme excels over the LSH and the SCH schemes in terms of the rounds required for complete the protocol. Accordingly, no one possesses computation and round efficiencies, simultaneously.

Recently, Chang and Chang [3] proposed another solution (the CC protocol for short) using super-poly-to-one trapdoor function which requires no certificate and can be efficiently constructed from one-way hash functions [1]. It is more superior than the above three approaches in terms of not only round and computation efficiencies but also its mutual authentication property and practicality. The authors claimed that the protocol is secure against various password guessing attacks. Unfortunately, as we will point out in detail, the CC protocol suffers from undetectable on-line password guessing attacks. It is thus our aim to propose an improved solution to this problem while at the same time preserving the original merits of the CC protocol.

2. Review of the CC protocol

In the CC protocol, P_A and P_B denote two participants A's and B's passwords securely shared with the server S, respectively. N_A, N_B, and N_S denote one-time and secret exponents separately chosen and secretly held by A, B, and S, respectively. Let $R_A \equiv g^{N_A} (\bmod p)$ and $R_B \equiv g^{N_B} (\bmod p)$, where p denotes a public large prime and g a generator of order $\phi(p)$. In addition, "f_K ()" denotes a pseudo-random function (PRF) indexed by key K. "$\langle\rangle_P$" denotes a symmetric encryption scheme using password P as the encryption key. "F_S()" represents a super-poly-to-one trapdoor function (TDF) constructed from one-way hash functions where only S knows the trapdoor. "$A \rightarrow B$: M" denotes a message M sent from A to B.

Note, since $f_{K_{AS}}\left(A, B, K_{AS}, R_B^{N_S}\right)$ was incorrectly stated as $f_{K_{BS}}\left(A, B, K_{BS}, R_A^{N_S}\right)$ in *Step* 4 of the original paper [3], the typographical error has been corrected below. The protocol performs the following five rounds:

1. $A \rightarrow B$: A, B, S, $\langle R_A\rangle_{P_A}$, $F_S(r_A)$, $f_{K_{AS}}(R_A)$
   A generates a random value r_A and computes $K_{AS} \equiv R_A^{r_A}$ (mod p) as a one-time key with S.
2. $B \rightarrow S$: A, B, S, $\langle R_A\rangle_{P_A}$, $F_S(r_A)$, $f_{K_{AS}}(R_A)$, $\langle R_B\rangle_{P_B}$, $F_S(r_B)$, $f_{K_{BS}}(R_B)$
   B generates a random value r_B and computes $K_{BS} \equiv R_B^{r_B}$ (mod p) as a one-time key with S.
3. $S \rightarrow B$: $R_B^{N_S}$, $f_{K_{AS}}(A, B, K_{AS}, R_B^{N_S})$, $R_A^{N_S}$, $f_{K_{BS}}(A, B, K_{BS}, R_A^{N_S})$
   S uses P_A/P_B and a trapdoor to derive R_A/R_B and r_A/r_B from $\langle R_A\rangle_{P_A}/\langle R_B\rangle_{P_B}$ and $F_S(r_A)/F_S(r_B)$, respectively. Then, S can compute K_{AS}/K_{BS} to authenticate A/B by verifying $f_{K_{AS}}(R_A)/f_{K_{BS}}(R_B)$. If successful, S proceeds by sending a response including $R_B^{N_S}/R_A^{N_S}$ and the corresponding hashed credential with K_{AS}/K_{BS} to A/B.
4. $B \rightarrow A$: $R_B^{N_S}$, $f_{K_{AS}}(A, B, K_{AS}, R_B^{N_S})$, $f_K(B, K)$
   B authenticates S by checking the validation of $f_{K_{BS}}(A, B, K_{BS}, R_A^{N_S})$. If it holds, B believes that the received $R_A^{N_S}$ is valid and then computes the session key $K \equiv (R_A^{N_S})^{N_B}$(mod p) and a challenge $f_K(B, K)$ for A.
5. $A \rightarrow B$: $f_K(A, K)$
   A authenticates S by verifying the validation of $f_{K_{AS}}(A, B, K_{AS}, R_B^{N_S})$. If it holds, A computes $K \equiv (R_B^{N_S})^{N_A}$ (mod p) to verify $f_K(B,K)$ to authenticate B. If successful, A sends $f_K(A,K)$ to B. B can authenticate A by checking the validation of $f_K(A,K)$. After mutual authentication between A and B, the session key agreement between A and B is established and confirmed.

Download English Version:

https://daneshyari.com/en/article/453438

Download Persian Version:

https://daneshyari.com/article/453438

Daneshyari.com