Mathematical and Computer Modelling 57 (2013) 1175–1183

Contents lists available at SciVerse ScienceDirect

Mathematical and Computer Modelling

journal homepage: www.elsevier.com/locate/mcm

Re-attack on a three-party password-based authenticated key exchange protocol

Haiquan Liang, Jingtai Hu *, Shuhua Wu

Urban Mass Transit Railway Research Institute, Tongji University, Shanghai 201804, China
School of Transportation Engineering, Tongji University, China

ARTICLE INFO

Article history:
Received 27 October 2011
Received in revised form 11 October 2012
Accepted 14 October 2012

Keywords:
Password-based
Authenticated key exchange
Three-party
Dictionary attack

ABSTRACT

A password based authenticated key exchange protocol is of practical usefulness in the protection of sharing of urban rail train sensor monitoring data. However, many password-based protocols in the literature were not secure. Recently, Huang presented a simple and efficient three-party password-based authenticated key exchange protocol. However, Yoon et al. found it had some security weaknesses. In this paper, we further show it has another critical security weakness, which opens door to a partition attack (offline dictionary attack). Thereafter we propose an enhanced protocol that can defeat the attacks described (including Yoon et al.'s attacks) and yet is reasonably efficient. Furthermore, our protocol can resist against the stolen-verifier attacks and achieve the provable security.

1. Introduction

In order to improve the active prevention ability against hidden danger in an urban rail train power system, we often need to share the urban rail train sensor monitoring data. In order to protect the security of sharing data, user authentication is certainly needed, which brings an opportunity to study password-based authenticated key exchange protocols. The Password Authenticated Key Exchange (PAKE) is a protocol which allows one party authenticate the other party by a simple password known by the two parties (that is, password-based authentication), and to agree on a fresh symmetric key securely such that it is known only to these two parties (that is, key exchange). Humans directly benefit from this approach since they only need to remember a low-quality string chosen from a relatively small dictionary (e.g. 4 decimal digits). However, the intrinsic problem with password-based protocols is that the memorable password, associated with each user, has low entropy, so that it is not easy to protect the password information against so-called dictionary attacks.

The first PAKE protocol, known as Encrypted Key Exchange (EKE), was suggested by Bellovin and Merritt [1]. Subsequently, many other two-party PAKE protocols have been proposed (e.g. [2–7]). Because two-party PAKE protocols are only suitable for the client–server architecture, many researchers have recently begun to study the three-party PAKE (3PAKE) protocols (e.g. [8–12]), in which a trusted server (TS) exists to mediate between two communication parties to allow mutual authentication and each user only needs to share one password with the common server. Unfortunately, some of them are not efficient enough to be used in practice (e.g. [11,12]), the others are not secure (e.g. [8–10]). Later, two efficient three-party password-based key exchange protocol were proposed by Abdalla et al. [13] and Lu et al. [14] respectively. However, the two schemes were still found insecure in [15–22] respectively.

Recently, to the best of our knowledge, Huang [23] also proposed a simple three-party password-based key exchange (3PAKE) protocol, which is more efficient than previously proposed schemes. She claimed that her protocol could resist against various dictionary attacks and was suitable for some practical scenarios. Unfortunately, however, Yoon et al. [24]

* Corresponding author at: Urban Mass Transit Railway Research Institute, Tongji University, Shanghai 201804, China.
E-mail address: hujt@tongji.edu.cn (J. Hu).

0895-7177/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.mcm.2012.10.019

found it had some security weaknesses. Independently of and concurrently to Yoon et al.'s work, Wu et al. [25] also found some security weaknesses and proposed a fixed version. In this paper, we further show it (including Wu et al.'s enhanced version) has another critical security weakness, which opens the door to a partition attack (offline dictionary attack). Thereafter we propose an enhanced protocol that can defeat the attacks described (including Yoon et al.'s attacks) and yet is reasonably efficient. Furthermore, our protocol can resist against the stolen-verifier attacks and achieve the provable security. The fact that many previous cryptographic schemes (e.g. [16–22]), like her scheme [23], containing only informal arguments for security were subsequently shown to be insecure, illustrates that the importance of formal proofs of security should be emphasized to design cryptographic protocols.

The remainder of this paper is organized as follows. Section 2 briefly reviews Huang's three-party password-based authenticated protocol. Section 3 then reveals a new weakness existing in Huang's protocol. Section 4 presents an enhanced 3PAKE protocol along with its performance analysis. Section 5 provides the rigorous proof of the security for our protocol. Finally, a conclusion is presented in Section 6.

2. Review of Huang's protocol

This section describes the 3PAKE protocol proposed by Huang [23], starting with some notations.

2.1. Notations

The notations used in their protocol are described as in the following:

- A, B: identity of two clients (users).
- TS: a TS (remote server).
- pw_A (pw_B): the password shared between user A (resp. B) and TS.
- p: a large prime number such that $p-1$ has a large prime factor q ($q \geq 2^{256}$).
- g: a generator with order q in $GF(p)$.
- G: the cyclic group generated by g;
- $\oplus$: an exclusive-or operator.
- $h(\,)$: a public one-way hash function.

2.2. Protocol description

There are three entities involved in the protocol: the authentication server TS, and two users A (initiator) and B (responder) who wish to establish a session key between them. Each user's password is assumed to be shared with the server TS via a secure channel. As illustrated in Fig. 1, A and B authenticate each other with TS's help, then A and B can share a common session key K. The details will be described in the following steps. Here, we just follow the description in [23].

Step 1. User A chooses a random number x and computes $R_A = (g^x \bmod p) \oplus h(pw_A, A, B)$, then sends (A, R_A) to user B.

Step 2. User B also selects a random number y and computes $R_B = (g^y \bmod p) \oplus h(pw_B, A, B)$, then forwards (A, R_A, B, R_B) to TS.

Step 3. Upon receiving (A, R_A, B, R_B), the TS first uses pw_A and pw_B to compute $g^x = R_A \oplus h(pw_A, A, B)$ and $g^y = R_B \oplus h(pw_B, A, B)$, respectively. Then, TS chooses another random number z and computes $a = g^{xz} \bmod p$, $b = g^{yz} \bmod p$. Finally, TS sends (Z_A, Z_B) to user B, where $Z_A = b \oplus h(pw_A, g^x)$ and $Z_B = a \oplus h(pw_B, g^y)$.

Step 4. When B receives (Z_A, Z_B), it uses its password pw_B and g^y to obtain $a = Z_B \oplus h(pw_B, g^y)$, and uses the random number y to compute the common session key $K = a^y = (g^{xz})^y = g^{xyz} \bmod p$ and $S_B = h(K, B)$. Next, user B forwards (Z_A, S_B) to user A.

Step 5. After receiving (Z_A, S_B), user A also uses its password pw_A and g^x to derive $b = Z_A \oplus h(pw_A, g^x)$, and uses the random number x to obtain the common key $K = b^x = (g^{yz})^x = g^{xyz} \bmod p$. Then, A checks whether $S_B = h(K, B)$ holds or not. If it does not hold, A terminates the protocol. Otherwise, A is convinced that $K = g^{xyz}$ is a valid session key. Then, A computes $S_A = h(K, A)$ and sends it to user B.

Step 6. Upon receiving S_A, user B verifies whether $S_A = h(K, A)$ holds or not. If it does not hold, B terminates the protocol. Otherwise, K is a valid session key. Both the users A and B can use this session key K for secure communication. Here, K is only used for one session.

3. Weakness of Huang's protocol

Unfortunately, it was found insecure against some off-line password guessing attacks and undetectable online password guessing attacks. However, their off-line password guessing attacks can only work in a special scenario and one critical security weakness was still not addressed in [24]. In this section, we show one can explore this weakness to mount a partition attack (an offline dictionary attack) on Huang's protocol. That is, the adversary just needs to wiretap a valid session and she is able to use the gathered information to partition the password space (the dictionary) into feasible and infeasible passwords. Finally the correct password will be recovered after a number of valid sessions have been observed from the intersection of the feasible partition of the passwords for each session. The similar weakness was first found in [7].