Information Sciences 180 (2010) 4069–4074

Contents lists available at ScienceDirect

Information Sciences

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

Secure group key agreement protocol based on chaotic Hash

Xianfeng Guo [a,b], Jiashu Zhang [a,*]

[a] Sichuan Key Lab of Signal and Information Processing, Southwest Jiaotong University, Chengdu 610031, PR China
[b] College of Computer Science and Technology, Southwest University for Nationalities, Chengdu 610041, PR China

ARTICLE INFO

Article history:
Received 30 August 2007
Received in revised form 27 February 2010
Accepted 13 June 2010

Keywords:
Chaos
Hash function
Key agreement
Chebyshev

ABSTRACT

Recently, Xiao et al. proposed an improved key agreement protocol based on chaotic maps, in which only a predetermined long-term key is utilized to ensure its security. This paper demonstrates that none of these schemes can satisfy the contributory nature of key agreement. To fill the gaps, we present a secure key agreement protocol based on chaotic Hash. The proposed scheme utilizes the chaotic Hash function to achieve the contributory nature and enhance its security. Cryptanalysis demonstrates that our chaotic Hash-based scheme can overcome all the current deficiencies.

1. Introduction

Group key establishment protocol allows participants to construct a common conference key for secure communication over an open channel. The proposed group key establishment protocols generally fall into two categories: group key distribution [11,15,17] and group key agreement [3,7,9,10,13,23]. In group key distribution protocols, there is a chairman who is responsible for generating a common key and then securely distributing the key to the other participants. The group key agreement protocols [3,7,9,10,13,23] involve all participants cooperatively to establish a group key by using the contribution of every group member. Compared with the key distribution protocol, one advantage of the key agreement protocol is that no participant can predetermine the group key. In many scenarios, the group key distribution protocol is not appropriate because the elected entity might activate a single point of failure for the group's security, and the group key agreement is a promising solution to achieving access control in collaborative and dynamic group applications [7,10,13,23]. The drawback of most group key agreements, however, is that they have not achieved the performance lower bound in terms of time, communication, and computation cost.

To overcome the deficiencies of group key agreement, Xiao et al. [21] proposed a chaos-based key agreement protocol, which utilized efficient chaotic public-key cryptosystem (CPKC) [8] to reduce computation costs. Unfortunately, Bergamo et al. [2] have pointed out that CPKC is insecure, and Alvarez [1] has demonstrated that the CPKC-based group key agreement scheme [21] is vulnerable to a man-in-the-middle attack. To enhance the security of group key agreement, Xiao et al. [18] proposed an improved key agreement by assuming that all participants have a shared long-term secret key. However, Song [4] points out that the improved scheme [18] can not resist replaying attacks. Recently, Song and Chang [5], Xiao et al. [19] used time-stamps or nonces to enhance the security of scheme [18], respectively.

In this paper, we illustrate that none of [5,18,19] can satisfy the contributory nature of key agreement, that is, the malicious server can predetermine the shared secret key. This weak spot of the protocol lies in the fact that there are several

* Corresponding author.
E-mail addresses: guoxianf@126.com (X. Guo), jszhang@home.swjtu.edu.cn (J. Zhang).

0020-0255/$ - see front matter © 2010 Published by Elsevier Inc.
doi:10.1016/j.ins.2010.06.013

Chebyshev polynomials passing through the same point. In addition, some related attacks on scheme [18] are briefly reviewed. To surmount the aforementioned flaws, we propose a secure group key agreement based on the chaotic Hash function. The proposed scheme performs mutual authentication to withstand server spoofing attacks and conquers denial of service attacks only by means of some efficient chaotic Hash computations, and utilizes the one-way property of the Hash function to guarantee the contributory nature of key agreement. Compared with the schemes in [5,19], our proposed scheme utilizes the chaotic Hash function rather than time-stamp or nonce to achieve the contributory nature of key agreement and avoid the security flaws.

The rest of this paper is organized as follows. Section 2 briefly reviews Xiao et al.'s key agreement protocol [18] and analyzes its security weaknesses. Section 3 presents an enhanced secure group key agreement protocol based on chaotic Hash. Section 4 demonstrates the performance analysis of the improved protocol. Concluding remarks are given in Section 5.

2. Original key agreement protocol and its security analysis

2.1. Review of the original protocol

Xiao et al.'s Scheme [18] assumes that User A and Server B share the Hash value $h_{pw} = H(ID_A, PW_A)$ of User A's password PW_A and identification ID_A, where $H(\cdot)$ denotes the chaotic Hash function of paper [20]. It can be briefly reviewed as follows:

2.1.1 Authentication phase

(1) A sends random number $ra \in [-1,1]$ and ID_A to B.
(2) B sends a random number rb to A.
(3) A computes Hash value $AU = H(h_{pw}, ra, rb)$ as the authenticated message and sends to B.
(4) B takes out his own copies of h_{pw}, ra, rb and calculates $AU' = H(h_{pw}, ra, rb)$. Then B checks whether $AU = AU'$ to verify A's validity.

2.1.2. Key agreement phase

(5) A choose a random integer r and sends $X = h_{pw} \oplus T_r(ra)$ to B, where "$\oplus$" denotes XOR operation and $T_r(ra)$ denotes the Chebyshev polynomial [8] of degree r i.e. $T_r(x) = \cos(r \cdot \arccos x)\ (-1 \leqslant x \leqslant 1)$.
(6) Similar with A, B chooses a random integer s and sends $Y = h_{pw} \oplus T_s(ra)$ to A.
(7) A and B compute $T_s(ra) = h_{pw} \oplus Y$ and $T_r(ra) = h_{pw} \oplus X$, respectively, and then calculate the shared secret key: $k = T_r(T_s(ra)) = T_s(T_r(ra)) = T_{rs}(ra)$.

2.2. Cryptanalysis of Xiao et al.'s Scheme [18]

Compared with scheme [21], scheme [18] utilizes a shared Hash value $h_{pw} = H(ID_A, PW_A)$ to protect security transmission. However, it is also vulnerable and can be easily attacked. In the following subsection, we demonstrate that scheme [18] can't satisfy the contributory nature of key agreement. Furthermore, several different kinds of attacks are briefly reviewed.

2.2.1. Contributory property

The contributory key agreement [13] is that no participant can predetermine the group key. It means that a group key K is generated as $f(N_1, \ldots, N_n)$, where $f()$ is some one-way function and N_i is an input (or key share) randomly chosen by the ith party. The method of computing group keys must guarantee that:

(i) each party contributing one N_i can calculate K;
(ii) no information about K can be extracted from a protocol run without the knowledge of at least one of the N_i;
(iii) all inputs N_i are kept secret, i.e. if party i is honest then even a collusion of all other parties cannot extract any information about N_i from their combined view of the protocol.

According to scheme [18], a malicious Server B can predetermine the communication key as follows:

(1) Once Server B receives message $X = h_{pw} \oplus T_r(ra)$ in Step (5) of Section 2.1, he can get $T_r(ra)$ by computing $T_r(ra) = h_{pw} \oplus X$.
(2) By utilizing several Chebyshev polynomials passing through the same point [1,2], Server B, who knows ra and $T_r(ra)$, is able to efficiently compute an integer solution r' to the equation $T_r(ra) = T_{r'}(ra)$:

$$r' = \left. \frac{\arccos(T_r(ra)) + 2k\pi}{\arccos(ra)} \right| k \in Z.$$

The reader is referred to [2] for the details on how to solve the previous equation, using a system of two linear equations.