Journal of Computer and System Sciences 80 (2014) 195–206

Contents lists available at SciVerse ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture

Kaiping Xue *, Peilin Hong, Changsha Ma

The Information Network Lab of EEIS Department, USTC, Hefei, 230027, China

ARTICLE INFO

Article history:
Received 12 November 2012
Received in revised form 25 June 2013
Accepted 12 July 2013
Available online 25 July 2013

Keywords:
Authentication and key agreement
Identity protection
Multi-server architecture
Traceability
Smart card

ABSTRACT

Traditional password based authentication schemes are mostly considered in single-server environments. They are unfit for the multi-server environments from two aspects. Recently, base on Sood et al.'s protocol (2011), Li et al. proposed an improved dynamic identity based authentication and key agreement protocol for multi-server architecture (2012). Li et al. claim that the proposed scheme can make up the security weaknesses of Sood et al.'s protocol. Unfortunately, our further research shows that Li et al.'s protocol contains several drawbacks and cannot resist some types of known attacks. In this paper, we further propose a lightweight dynamic pseudonym identity based authentication and key agreement protocol for multi-server architecture. In our scheme, service providing servers don't need to maintain verification tables for users. The proposed protocol provides not only the declared security features in Li et al.'s paper, but also some other security features, such as traceability and identity protection.

1. Introduction

With the rapid growth of modern computer networks, increasing numbers of systems contain a certain quantity of service providing servers around the world and provide services via the Internet. It's important to verify the legitimacy of a remote user in a public environment before he/she can access the service. But traditional password based authentication schemes are mostly considered in single-server environments. They are unfit for the multi-server environments from two aspects. On the one hand, users need to register in each server and to store large sets of data, including identities and passwords. On the other hand, servers are required to store a verification table containing user identities and passwords. [1] firstly proposed a remote authentication scheme using smart card based on Elgamal's public key cryptosystem [2], which doesn't need to maintain verification tables. After that, numerous smart card based single-server authentication schemes using one-way hash functions had been proposed [3–11]. However, it is still hard for a user to use different smart cards to login and access different remote servers. This is because users still need to remember numerous sets of identities and passwords. In order to resolve this problem, several schemes have been proposed to the study of authentication and key agreement in the multi-server environment [12,13,20–25], all of which claim not to store verification tables. Most of these schemes can be divided into three categories: hash-based [12,13,25], symmetric cryptosystem based [24] and public key cryptosystem based [20–23]. Hash-based protocols are considered to be the most efficient. Among these schemes designed

* Corresponding author.
E-mail address: kpxue@ustc.edu.cn (K. Xue).

0022-0000/$ – see front matter
http://dx.doi.org/10.1016/j.jcss.2013.07.004

Table 1
Notations used in Li et al.'s paper.

U_i	a user
S_j	a service providing server
CS	the control server
ID_i	the identity of U_i
SID_j	the identity of S_j
x	the master secret key
y	the secret number
b	a random number chosen by the user for registration
CID_i	the dynamic identity generated by U_i for authentication
SK	session key shared among the user, the server and CS
N_{i1}, N_{i2}, N_{i3}	random numbers chosen by U_i, S_j and CS
$h(\cdot)$	a one-way hash function
$\oplus$	the bitwise XOR operation
$\|$	the bitwise concatenation operation

for the multi-server environment, in [12], Hsiang and Shih proposed a dynamic identity and one-way hash-based remote user authentication protocol for multi-server architecture without a verification table. However, in 2011, Sood et al. [13] pointed that Hsiang and Shih's protocol cannot resist many types of security attacks, such as replay attack, impersonation attack and stolen smart card attack. Then Sood et al. proposed an improved scheme which is claimed to achieve user anonymity and resist different types of common security attacks. Recently, in [25], Li et al. found that Sood et al.'s protocol is still vulnerable to some types of known attacks, such as replay attack, stolen smart card attack and so on. Also the mutual authentication and key agreement phase of Sood et al.'s protocol cannot be successfully finished within some specific scenes. Furthermore, in [25], they proposed an improved dynamic identity based authentication and key agreement protocol for multi-server architecture, which is claimed to remove the aforementioned weaknesses of Sood et al.'s protocol. Unfortunately, our further research shows that Li et al.'s protocol contains several drawbacks and cannot resist some types of known attacks, such as leak-of-verifier attack, stolen smart card attack, eavesdropping attack, replay attack, Denial-of-Service attack and forgery attack and so on.

Meanwhile, identity protection is considered to be important for authentication and key agreement protocol design in single-server and multi-server architectures. Some existed researches adopt pseudonym [14] and dynamic identity [15,16] technologies to hide users' real identities. [12,13,25] also use dynamic identity technology to provide user anonymity, but the above discussion reveals that there are some security flaws of these schemes. Furthermore, they haven't provided traceability while providing user anonymity. Usually, there are conflicts between the anonymity and traceability objectives [17,18], which need to be well addressed. [12,13,25] don't provide the function of traceability while providing user anonymity. [19] proposes a scheme and claims that the scheme can provide the functions of traceability and anonymity simultaneously. But the pseudonym used in this scheme is fixed and can be considered as a user's another identity.

The rest of this paper is organized as follows: Section 2 gives the overview of Li et al.'s protocol; Section 3 points out the security weaknesses of the protocol in details. Section 4 gives our proposed protocol. Security and performance analysis of our proposed protocol is given in Section 5 and Section 6. At last, Section 7 presents the overall conclusion.

2. Overview of Li et al.'s protocol

In this section, we give the overview of Li et al.'s proposed protocol, which is an enhanced scheme based on Sood et al.'s protocol. We firstly summarize the notations used throughout Li et al.'s paper in Table 1. Li et al.'s protocol involves 3 kinds of participants: users (taking U_i for example), service providing servers (taking S_j for example), and the control server (CS). CS is a trusted third party (TTP) responsible for the registration and authentication of the users and the service providing servers. CS firstly chooses two security elements x and y. In the registration phase, S_j obtains $h(SID_j\|y)$ and $h(x\|y)$ from CS via a secure channel. U_i randomly selects a number b, and computes $A_i = h(b\|P_i)$, where P_i is U_i's password. After the initialization and the registration phases, U_i can get a smart card from CS via a secure channel. The following elements, $h(\cdot)$, $h(y)$ and b are stored in the smart card for the user U_i:

$$\begin{aligned} C_i &= h\big(ID_i\|h(y)\|A_i\big) \\ D_i &= B_i \oplus h(ID_i\|A_i) = h(ID_i\|x) \oplus h(ID_i\|A_i) \\ E_i &= B_i \oplus h(y\|x) = h(ID_i\|x) \oplus h(y\|x) \end{aligned} \tag{1}$$

In U_i's login phase, U_i inserts his smart card into a terminal and inputs his identity ID_i and password P_i, then computes $A_i^* = h(b\|P_i)$ and $C_i^* = h(ID_i\|h(y)\|A_i^*)$. If C_i^* is equal to the stored C_i, U_i is considered as a legitimate user. Else, the terminal rejects U_i's login request. After the verification, the authentication and key agreement phase takes place among U_i, S_j and CS, as depicted in Fig. 1. We introduce them as follows: