Computer Networks 58 (2014) 29–38

Contents lists available at ScienceDirect

Computer Networks

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

Provably secure three-party authenticated key agreement protocol using smart cards

Haomin Yang [a,*], Yaoxue Zhang [a], Yuezhi Zhou [a], Xiaoming Fu [b], Hao Liu [a], Athanasios V. Vasilakos [c]

[a] Department of Computer Science and Technology, Tsinghua University, Beijing, China
[b] Institute of Computer Science, University of Goettingen, Goettingen, Germany
[c] Department of Computer and Telecommunications Engineering, University of Western Macedonia, Kozani, Greece

ARTICLE INFO

Article history:
Received 9 February 2012
Received in revised form 9 July 2013
Accepted 26 August 2013
Available online 31 August 2013

Keywords:
Three-party authenticated key agreement
Smart card
Provable security
Random oracle model

ABSTRACT

Authenticated key agreement protocol is a useful cryptographic primitive, which can be used to protect the confidentiality, integrity and authenticity for transmitted data over insecure networks. From the point of view of the management of pre-shared secrets, one of the advantages of three-party authenticated key agreement (3PAKA) protocols is that they are more suitable for use in a network with large numbers of users compared with two-party authenticated key agreement protocols. Using smart cards is a practical, secure measure to protect the secret private keys of a user. Recently, some 3PAKA protocols using smart cards have been proposed. However, up to now, it is still a challenging problem to propose a 3PAKA protocol using smart cards with fewer rounds of messages and without using timestamp technique. Another important fact to be mentioned is that existing 3PAKA protocols using smart cards all lack provable-security guarantees. In this paper, we propose a new 3PAKA protocol using smart cards. The proposed protocol gains several advantages over existing related protocols: (1) The protocol is provably secure under the computational Diffie–Hellman assumption in the random oracle model, and hence can resist strong adversaries in network scenarios; (2) The protocol needs fewer rounds of messages, and can enable short communication latency and rapid response; and (3) The protocol is not based on timestamp technique, and does not need the complicated clock synchronization.

1. Introduction

To protect the confidentiality, integrity and authenticity for transmitted data over insecure networks, two communicating parties usually need to establish session keys via a key agreement protocol. In the seminal paper, Diffie and Hellman [1] proposed the first key agreement protocol, the so-called Diffie–Hellman (DH) protocol. However, the DH protocol does not use identity authentication mechanism, and hence inevitably suffers from the man-in-the-middle attack. Therefore authentication and key agreement are combined to yield the notion of authenticated key agreement. Authenticated key agreement attracts great interest in network security field, and many authenticated key agreement protocols have been proposed, e.g., [2–8].

According to whether an online trusted server is required or not when two communicating parties agree on a shared session key, authenticated key agreement protocols can be divided into two categories: two-party authenticated key agreement (2PAKA) protocols and three-party authenticated key agreement (3PAKA) protocols. 2PAKA protocols require no online trusted server, but they are unsuitable for use in a network with large numbers of users. The reason is that 2PAKA protocols require each pair of communicating parties to share a secret beforehand, and therefore the number of required pre-shared secrets

* Corresponding author. Tel.: +86 1051612650.
E-mail address: yanghm1976@gmail.com (H. Yang).

1389-1286/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comnet.2013.08.020

increases with the square of the number of network members. Thus, if a 2PAKA protocol is used in a network with large numbers of users, the management of pre-shared secrets will be a complex problem. In contrast, 3PAKA protocols only need each user share a pre-shared secret with an online trusted server, but do not need each pair of users to share a pre-shared secret. Thus, if a 3PAKA protocol is used in a network with large numbers of users, the management of pre-shared secrets will become relatively easy. Generallyspeaking, from the point of view of the management of pre-shared secrets, one of the advantages of 3PAKA protocols is that they are more suitable for use in a network with large numbers of users compared with 2PAKA protocols.

Two desirable characteristics of a 3PAKA protocol are fewer rounds of messages and no timestamps.

(1) Fewer rounds of messages

A round consists of all the messages that can be sent and received in parallel within one time unit [9]. In a network with large numbers of users, a particularly important factor in evaluating the performance of 3PAKA protocols is the number of rounds of messages, since fewer rounds can enable short communication latency and rapid response [13].

(2) No timestamps

In a key agreement protocol, some measures must be taken to ensure that old messages cannot be reused. Timestamps can be used to provide this kind of guarantee. A timestamp is the time at which a message is created. However, as pointed out in [2,10–12], the use of timestamps introduces a number of practical problems, such as time synchronization problem, and hence it is desirable in practice to avoid relying on the use of timestamps. Especially, it is highly undesirable in a network with large numbers of users, since it is rather difficult to keep the clocks synchronized between large numbers of users. As pointed in [14], timestamps in protocols can be replaced by a random number challenge plus a return message, the so-called random challenge-response technique. Random challenge-response technique does not need time synchronization, and therefore the protocols based on random challenge-response technique are more suitable for use in a network with large numbers of users than those timestamp-based protocols.

In a run of key agreement protocol, the secret private keys of users, including the long-term private keys and the ephemeral private keys, must be protected carefully. If an adversary has obtained the private keys of a legal user, he is able to impersonate the legal user to agree on session keys with other legal users. Using smart cards is a practical, secure measure to protect the secret private keys of a user. A smart card is a credit card size plastic card with an embedded microcontroller, a memory chip and an embedded operating system. The card connects to a reader with direct physical contact or with a remote contactless radio frequency interface. With an embedded microcontroller, smart cards have the ability to store data, carry out their own on-card functions (e.g., encryption and mutual authentication) and interact intelligently with a smart card reader [15].

Recently, some 3PAKA protocols using smart cards have been proposed. For example, Juang [16] proposed a 3PAKA protocol using smart cards. However, Juang's protocol requires five rounds of messages and hence cannot support rapid response. Yoon and Yoo [17] proposed a token-based 3PAKA protocol with four rounds of messages. Chang et al. [18] proposed a 3PAKA protocol using smart cards without any modular exponentiation operation; however, this scheme needs five rounds of messages. Later, Yoon and Yoo [19] proposed an improved smart card-based 3PAKA protocol without modular exponentiation, which needs four rounds of messages. Kwon et al. [20] proposed a smart card-based 3PAKA protocol which needs only three rounds of messages. Unfortunately, as pointed out by Yoon and Yoo [21], Kwon et al.'s protocol [20] is vulnerable to impersonation attacks. Yoon and Yoo [21] also proposed an improved 3PAKA protocol with three rounds of messages. However, like Kwon et al.'s protocol [20], Yoon and Yoo's protocol [21] is based on timestamp technique and is not suitable for use in a network with large numbers of users.

Up to now, it is still a challenging problem to propose a 3PAKA protocol using smart cards with three rounds and without using timestamp technique. Another important fact to be mentioned is that the above-mentioned protocols all lack provable-security guarantees. In cryptography, a system has provable security if its security requirements can be stated formally in an adversarial model, as opposed to heuristically, with clear assumptions that the adversary has access to the system as well as enough computational resources. The proof of security is that these security requirements are met provided the assumptions about the adversary's access to the system are satisfied and some clearly stated assumptions about the hardness of certain computational tasks hold. Since many cryptographic protocols without provable security have been broken, provable security has been viewed as an important criterion in designing cryptographic protocols.

In this paper, we achieve the following contributions:

(1) We propose a new 3PAKA protocol using smart cards. The proposed protocol enjoys a number of desirable features: (a) it needs only three rounds of messages, and hence can enable short communication latency and rapid response; (b) it is based on random challenge-response technique instead of timestamps, and hence does not need the complicated clock synchronization; and (c) the smart cards in our protocol is protected with passwords, and the owners of the smart cards can freely choose and change the passwords.
(2) We prove the security for the proposed protocol in the Abdalla, Fouque and Pointcheval security model. The proof results show that the proposed protocol is secure under the computational Diffie–Hellman (CDH) assumption in the random oracle model.

The rest of this paper is organized as follows. In Section 2, some preliminaries, including the CDH problem and assumption, random oracle model, and the Abdalla, Fouque and Pointcheval security model, are reviewed. Section 3 presents the proposed protocol. In Section 4, we simply