Computer Networks 129 (2017) 159–177

Contents lists available at ScienceDirect

Computer Networks

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

A key agreement protocol with partial backward confidentiality

Orhan Ermiş[a,*], Şerif Bahtiyar[b], Emin Anarım[c], M. Ufuk Çağlayan[a]

[a] Computer Networks Research Laboratory-NETLAB, Department of Computer Engineering, Boğaziçi University, 34342, Bebek, Istanbul, Turkey
[b] Istanbul Technical University, Faculty of Computer and Informatics, Ayazaga Campus, Maslak, Istanbul, 34469, Turkey
[c] Boğaziçi University, Department of Electrical and Electronic Engineering, 80815 Bebek, Istanbul, Turkey

ARTICLE INFO

Article history:
Received 24 February 2017
Revised 11 July 2017
Accepted 16 September 2017
Available online 19 September 2017

Keywords:
Public key cryptography
Partial backward confidentiality
Key agreement protocols
Secure file sharing systems

ABSTRACT

The essence of dynamic group key agreement protocols is to help compute a secure key for a group communication with a dynamic set of participants in distributed systems. In dynamic group key agreement protocols, the number of participants may change over time because of participants leaving or joining the group. The security of such join and leave operations are affected by the existence of backward confidentiality and forward confidentiality, respectively. Dynamic group key agreement protocols are expected to be used in applications such as file sharing systems. However, there are a number of problems in the use of existing dynamic group key agreement protocols in file sharing systems such as lack of privacy, violation of availability and dependency for key escrow. In this study, we propose a new security property called partial backward confidentiality. Partial backward confidentiality is the property, in which a new participant can compute the last valid group key just before joining the group but the new participant cannot compute former group keys. Moreover, we propose a key agreement protocol to show the provision of partial backward confidentiality that helps to solve file sharing system problems above. Furthermore, we have analyzed the security of the proposed protocol with respect to impersonation attacks under the difficulty in discrete logarithm problem and eavesdropping under the Decisional Difie-Hellman Problem. We present a proof of concept case study called Private File Sharing System in order to show the applicability of partial backward confidentiality property.

1. Introduction

Group key agreement protocols are among the best candidate for establishing a secure communication in a distributed network. The early designs of group key agreement protocols focus on static groups, in which the set of participants do not change until the end of a communication session. However, with the growth of technology, dynamic group structures replaced the static groups in multi-party communications. Group key agreement protocols that were designed for static groups became outdated since the handling of updating a group key has a challenging overhead. Therefore, group key agreement protocols are evolved to overcome this overhead by providing dynamic group operations. Dynamic group key agreement protocols such as [13,20,30,31] update the group key by performing less effort than static group based key agreement protocols. Moreover, group key agreement protocols with dynamic group capability are achieving the basic security properties with the static ones.

In general, group key agreement protocols are based on Diffie-Hellman protocol [1], which enables only two participants to agree on a common key. Later, the first multi-participant group key agreement protocol was proposed in [2]. In addition, there have been many studies about group key agreement protocols with different security properties. One well-known property of group key agreement protocols is the authentication, which is used for confirming the identities of participants in the group communication [3]. Two important group key agreement protocols with and without authentication were proposed by Burmester and Desmedt in [4]. Both of these protocols are for static groups not for dynamic groups. Authentication with anonymity may be used as a security property in various application areas, such as IoT-enabled devices in a distributed cloud computing environment [5], patient monitoring system using wireless medical sensors [6] and payment systems [7].

The fault-tolerance property, which is introduced by Tzeng in [8], is necessary for detecting and correcting the malicious behavior of participants during key computations. Other group key agreement protocols with fault tolerance property are in [9–12,16]. For instance, protocols in [9,10,16] improved the group key computation performance with respect to Tzeng's protocol. The protocol

[*] Corresponding author.
E-mail addresses: orhan.ermis@boun.edu.tr, orhanermis@gmail.com (O. Ermiş).

https://doi.org/10.1016/j.comnet.2017.09.008
1389-1286/© 2017 Elsevier B.V. All rights reserved.

in [11] provides non-interactive approach for fault-tolerance property to identify and remove malicious participants from the group key computation. In addition, the forward secrecy property is also crucial for providing security against compromising group keys if the long-term private key of any participant is compromised [12].

There are two additional security properties for dynamic group key agreement protocols, namely backward confidentiality and forward confidentiality [13,14]. In backward confidentiality, participants who joined the group cannot compute former group keys. In forward confidentiality, participants who left the group cannot compute subsequent group keys. Dynamic group key agreement protocols are expected to be used in applications such as teleconferences, instant communications, file sharing systems, etc. On the other hand, there are a number of problems on the use of existing dynamic group key agreement protocols in File Sharing Systems (FSS) such as lack of privacy [38], violation of availability [37] and dependency for key escrow [42]. The most important reason of problems in FSS is the existence of backward confidentiality property. Since joining participants cannot compute the previous group key just before joining the group, FSS must provide a mechanism to grant access permissions for joining participants. Trusted third parties (TTPs) or dedicated participants in the group (for instance Group Managers) are used to overcome this problem. However, if TTPs are involved in file sharing, the privacy of the file is endangered. If dedicated participants distribute group key, there is a possibility of the violation of availability due to the single-point-of-failure. Moreover, if TTPs and dedicated participants exist in file sharing systems, the key escrow mechanism provides data recovery keys for encrypted files [15]. Since files are shared by the participants of a communication group, there is no need for such backup mechanism. In this study, our main motivation is to solve these problems with the provision of partial backward confidentiality.

Our contributions in this study are as follows:

(i) We propose a new security property called Partial Backward Confidentiality (PBC). In PBC, a new participant can compute the last valid group key just before joining the group but the new participant cannot compute former group keys.
(ii) Moreover, we propose a Key Agreement Protocol with Partial Backward Confidentiality (KAP-PBC). KAP-PBC design is based on the protocol in [16] to provide operations for dynamic groups while preserving the basic security properties.
(iii) We also present a proof of concept case study called Private File Sharing System (PFSS) to demonstrate the applicability of the partial backward confidentiality property to solve the lack of privacy, the violation of availability and the dependency for key-escrow problems.

The rest of the paper is organized as follows. In the next section, general definitions and properties of group key agreement protocols are given. Section 3 overviews the dynamic group key agreement protocols. KAP-PBC is proposed in Section 4. Performance analysis and security analysis are given in Section 5 and 6, respectively. In Section 7, we propose a proof of concept case study called PFSS as an application of KAP-PBC. Finally, Section 8 concludes the paper.

2. Comparison of dynamic group key agreement protocols

In this section, we compare KAP-PBC and previously proposed dynamic group key agreement protocols as shown in Table 1. The Criteria used for comparing protocol properties are listed as follows:

(i) **Dynamic Group Operations (DGO):** Dynamic group operations can be listed as join, leave, mass join (merge) and mass leave (divide).
(ii) **Security Properties for Group Key Agreement Protocols (SPGKAP):** The basic parameters to assess the security level of a group key agreement protocols are authentication, fault-tolerance and forward secrecy.
(iii) **Security Properties for Dynamic Group Key Agreement Protocols (SPDGKAP):** In group key agreement protocols, security of the resulting group key after dynamic group operations can be assessed by the existence of Backward Confidentiality and Forward Confidentiality properties.
(iv) **Partial Backward Confidentiality (PBC):** The last criterion for dynamic group key agreement protocols is the Partial Backward Confidentiality property. With this property, a new participant can compute the last valid group key just before joining the group but the new participant cannot compute former group keys.

As in Table 1, we have compared the proposed protocol with other protocols in the literature regarding the dynamic group operations, security properties of group key agreement protocols, dynamic security properties of group key agreement protocols and partial backward confidentiality property. Since it is initially proposed in this study, the only protocol that provides partial backward confidentiality is KAP-PBC. Protocols in [13,30] and KAP-PBC satisfy all of the criteria in a dynamic group key agreement protocol. Moreover, [13,30] and KAP-PBC have extra operations for efficiently handling of mass join and mass leave operations. Specifically, if a protocol provides mass join and mass leave operations, then a protocol can accomplish join or leave operation at one execution instead of executing separate operations for each join or leave. Therefore, the performance of protocols that provide mass join and mass leave operations is better than protocols that provide single join and leave operations.

In terms of security criteria, we have compared protocols with respect to SPGKAPs and SPDGKAPs. Protocol in [28] has the worst protocol among other protocols since it does not satisfy any of the security criteria. In addition, protocols in [25–27] do not provide neither fault-tolerance nor forward secrecy. Therefore, these protocols are vulnerable against security threats such as malicious attempts to compute a wrong group key or compromise of group keys. On the other hand, protocol in [22] does not provide forward and backward confidentiality properties, which causes the protocol to expose former or subsequent group keys after the set of participant is updated. When forward confidentiality property or backward confidentiality property is not provided, joining participants or leaving participants can view former or subsequent communications in the group.

3. General definitions and properties of group key agreement protocols

This section gives the general definitions and properties of group key agreement protocols.

Definition (Participants and Group)**.** Participants, participant list and their public properties are defined as follows:

- Each participant is an entity and is denoted as U_i.
- The participant list is represented as $\langle U_1, U_2, \ldots, U_n \rangle$.
- The list is circular so that $U_{n+i} = U_i$ for some positive $1 \le i \le n$. The order of the participants is known by each participant.
- Let $\mathcal{U} = \{U_1, U_2, \ldots, U_n\}$ be the set of participants, during the execution of the protocol, participant U_j, which has at least one verification matrix entry $V_{i,j} =$ "*failure*", for $1 \le i \le n$ and $i \ne j$, is defined as potential malicious participant until its malicious behavior is proved. Otherwise, the participant is defined as trusted participant.