SECRET KEYS MULTIPLICATION FOR SCALABLE GROUP RE-KEYING

Yaser Mohamed and Atsushi Kara
Aizu University
Aizu Wakamatsu, Fukushima 965-8580, Japan
Tel: 0242-37-2500
Email: d8041103@u-aizu.ac.jp

ABSTRACT
In this paper we introduce a simple and secure method to solve the multicast group rekeying problem. Our rekeying method, which is called secret keys multiplication "SKM", is based on multiplying secret keys that are held by the group members to construct the rekeying message. The secret keys are organized in a logical hierarchy to avoid the multiplication overflow. Using our method, no encryptions or decryptions are needed to distribute the group key to the group members, small number of rekeying messages is needed to rekey the group, and frequent membership change can be handled efficiently.

Keywords
Secure multicasting, multicast group rekeying, logical key hierarchy, secret keys multiplication.

1. INTRODUCTION
The rapid growing of the World Wide Web stimulated the appearance of new types of group communications such as multimedia streaming, multi-party conferencing, real time distribution of market data, and pay per view distribution of digital media. These types of group communications use the multicast service that is supported by the Internet today. Those applications require securing the transmitted data to restrict the service to only the members that subscribed to the multicast session or to only the legitimate users that have the right to access the multicast information. This involves preventing any new member from accessing previous multicast session data, which is known as perfect forward secrecy (PFS), and preventing any previous member from accessing any following multicast session data, which is called perfect backward secrecy (PBS).

To add security services to multicast services, each secure multicast group can be associated with one or more trusted servers responsible for managing group membership (new member joining and old member leaving). This trusted server is called the Group Controller (GC) in this paper. In case of new member joining the group, the GC changes the current session key (SK) (to guarantee PFS). The SK is used for encrypting the data traffic between group members. The GC sends the new SK and auxiliary keys (depending on the used keying technique) to the new member encrypted with its individual key and multicasts the new SK to the existing group members encrypted with the previous SK. Only the group members and the GC know the SK. In case of one or more members leave the group, the SK and all auxiliary keys known to the leaving members and shared with the remaining members have to be changed (to guarantee PBS). Because the old SK cannot be used any more, sending the new SK to the remaining members in a scalable and secure way is still a critical and difficult problem.

Many group rekeying approaches have been proposed to address this problem [21][3][4][6][7][8][9]. In this paper we introduce a very simple rekeying method that eliminates the encryption overhead of the group controller. The logical hierarchy of keys which is proposed in [2] and [3] is used to avoid the multiplication overflow problem by multiplying only "d" private keys where "d" is the logical key tree degree. Our method is scalable to be used with large groups and with highly dynamic membership groups.

2. RELATED WORK
Many authors have been addressed the scalability problem of multicast group rekeying. In this section we will briefly review some related work that have addressed the problem of distributing the session key to all group members in a secure way, especially, by using the key tree hierarchy.

In [2] the logical key tree approach, which is called "key graph", has been proposed to improve the scalability problem of the rekeying process. Besides the session key and its individual key, each user is given some auxiliary keys to facilitate rekeying. Also, user-oriented, key-oriented, and group-oriented rekeying strategies have been proposed besides join/leave protocols based on these rekeying strategies. In user-oriented rekeying, the server constructs a re-key message that contains precisely the new keys needed by the user and encrypts them using a key held by the user. In key-oriented rekeying, each new key is encrypted individually. While in group-oriented rekeying, the server constructs a single rekeying message containing all new keys. For member leave request, the key graph approach reduces the processing cost to O(log N). More details about the key tree structure will be provided in section 3, and a brief comparison between the key graph approach and SKM will be presented in section 4.

The idea of key hierarchy was independently discovered in some other work such as [3], and [5].

In [6] a different scheme using the key tree hierarchy is described. This method has modified the key tree hierarchy that is used in [2]. A different distribution of keys in the key tree has been considered that used a dynamic key hierarchy instead of the fixed hierarchy of keys. Also, Boolean function minimization has been used to minimize the number of messages needed for rekeying. [6] minimized the number of keys maintained by the GC to O(N), and it is efficient to be used in case of bulk membership changes, however, it suffers from collusion attack where some previous members can collude together to decrypt the new session key.

In [8], the notion of secure distribution tree is used but in a different way, instead of keys in the tree levels, the distribution tree is composed of a number of smaller subgroups, and each subgroup has its own subgroup key, there is no global group