Chapter 7
Scalable Rekeying Using Linked LKH Algorithm for Secure Multicast Communication

Priyanka Ahlawat
National Institute of Technology, Kurukshetra, India

Kanishka Tyagi
National Institute of Technology, Kurukshetra, India

ABSTRACT

In the real scenario, there is a large multicast group where nodes leave and join frequently, and also the number of nodes leaving and joining is also not proportionate. Hence, scalable rekeying process is an important issue that needs to be concerned for the secured group communication for dynamic groups. In basic rekeying scheme, which is based on the logical key hierarchy, the rekeying cost depends on the logarithm of the size of group for a join or depart request by the user. However, the memory efficiency of this group rekeying protocol (GREP) is a huge storage overhead over the system. The authors aim to provide a survey of various group key management schemes and then propose an efficient scalable solution based on linked LKH and the linked list data structure. Results have shown that the Linked LKH algorithm has a very low effective cost for rekeying the LKH as compared to the basic LKH algorithm (i.e., based on the number of new joined and departure requests).

INTRODUCTION

Group Communication among valid users is a powerful Message exchange Model. Group communication applications include content distribution over large-scale networks, smart wireless networks, software updates, Video Conferencing, and many more (He Niedermeier & Meer 2013). K**ey** management security should have basic requirements like authenticity, integrity, **and** confidentiality. The factors that affect any group key management process's performance are scalability, 1 affects n problem, delays bandwidth for efficient group key distribution. Scalability refers to the network's ability to handle large

DOI: 10.4018/978-1-7998-7789-9.ch007

dynamic groups, i.e., if the number of users becomes large, the key management scheme should handle without degrading its performance. 1-affects n problem relates with new key generation after a leave or joins processes during a rekeying. Quality of service should also be maintained during a rekeying process, such as control packets, storing keys, and delay induced during encryption and decryption. Key management is very important in group communication restricting access control (Duma, Shahmehri, lambrix 2003). Key management establishes and maintains the secret keying relationships between valid parties according to a policy. It includes member identification and authentication. in this regard, authentication plays a significant role. Once a new member joins, it has to be validated (Zhu, Jajodia 2003). It is also **essential** to change or update the group key at regular intervals to effectively maintain a communicated message's security. Also, key independence has to be properly maintained in which each key is independent of another key. It means the method of generating a new group key should be independent of the previous key generation. It also enhances the security of the overall system. Combinatorial optimization of group key is given in (Eltoweissy, Heydari, Morales, Sudborough, 2004). Any Participant can become a part of Group Communication by becoming a group member explicitly. A group member holds a secretly shared cryptographic Group key and used the same for exchanging the messages. When a participant becomes a group member, it is required to maintain backward secrecy, i.e., the participant cannot decipher the messages exchanged before it's joined.

Similarly, whenever the group member leaves the group, Group communication must maintain the forward secrecy. After the group leaves, it must prevent the node from accessing the messages. To achieve the forward secrecy and backward secrecy, group rekeying is performed, which ensures a new and different key among the group members when a node leaves or joins the group. However, group rekeying doesn't play an effective role if we could ensure the group's structure by defining the members through the pre-registration of members (Panda, Thool 2016). Group key management can be classified into three classes: centralized, decentralized, and distributed key management. In centralized schemes, a single entity generates, distributes, and management of the group key.

Hence a single entity controls **the** entire group. Minimization of storage, computational power, and bandwidth are the key challenges in this scheme. In decentralized schemes, the process of management of group keys is divided among different group members. Hence a single point of failure is not a problem in this scheme.

In distributed group key management, No key server is explicitly declared. In this scheme, group members perform the key generation function. It can be contributory or done by individual members. Maintaining security in every group communication protocol is a critical issue.

The safety goal in a group communication process is to guarantee access only to valid group members. The entry and leave of the group members or users are the main reason for modifying the group key and giving them greater confidence in secure communication, known as re-keying. Since it is a frequently performed activity during group communication, it is necessary to do the group key update in a scalable and efficient way. Previously, the client-server paradigm is the most commonly used technique for conferencing, chat groups, immersive video games, etc., that use the principle of unicast for data transmission. Present-day developments in Internet technology, with the increase in bandwidth, are encouraging new developments in the environment. Unlike the old network communication models, where packets are to be delivered in a unicast model, the multicasting technique provides an effective delivery service to a larger user community with efficient and effective network resources (Xu, Y., & Sun, Y 2005). The key tree approach is efficiency depends crucially on whether the key tree stays balanced. Rebalancing with fixed time intervals is used to balance the key tree if it becomes unbalanced.