AN EFFICIENT KEY MANAGEMENT SCHEME FOR MULTICAST COMMUNICATION

Ja-Long Chen, Department of Information Management, National Chung Cheng University, Chia-Yi County, Taiwan, ROC, 886-5-2721500, jalongchen@mis.ccu.edu.tw
Cheng-Yuan Ku, Department of Information Management, National Chung Cheng University, Chia-Yi County, Taiwan, ROC, 886-5-2721500, cooperku@mis.ccu.edu.tw
David C. Yen, Department of DSC and MIS, Miami University, Oxford, OH 45056, USA, 513-529-4827, yendc@muohio.edu

ABSTRACT

The proliferation of e-commerce has inspired the development of numerous online applications. Many of these applications are related to the information content delivery, such as news services and teleconference. For these services, multicast is the major delivery method. In order to protecting the information content and keeping the security requirements of forward and backward secrecy, the multicast transmission process needs to maintain the group key infrastructure and implement the rekeying procedure.

The Network Working Group organizing RFC proposed a key management protocol named logical key hierarchy (LKH) that is highly suitable for managing keys in the multicast network [3]. Sherman and McGrew [2] suggested that the key distribution center (KDC) sends a revision number to the related nodes when a node joins or leaves a group or switches between groups. After receiving this revision number, the nodes use it and a hash function to calculate the middle keys. Zhang and Wang tried to reduce the total group number for the purpose of decreasing middle nodes [4]. Pour et al. grouped the bottom nodes to reduce the number of hierarchical levels [1]. Compared to the LKH scheme, this scheme needs less rekeying overheads. Meanwhile, Zhang et al. [5] focused on improving key management efficiency by using a greedy algorithm to trace the membership group and a resource group directed acyclic graph (resource group DAG) to establish a key graph with fewer middle nodes. However, the above-mentioned rekeying action or key management requires lots of computing capacity and network bandwidth, and thus how to reduce rekeying time and effort is an important research topic for multicast communication.

In this research, we thus aim to improve the key management scheme. During the phase of key distribution, our key management scheme generates an access control polynomial (ACP) string to replace the traditional rekeying steps. Furthermore, for the purpose of improving the processing efficiency of the server and client, this study also revises the key selection process when generating polynomial string.

The simulation was implemented using Java J2SE. It was on Window's XP using Intel Celeron CPU 2.53 GHz and 504 MB RAM. The simulation program contains three parts: key server, event trigger and client nodes. The key server manages the LKH tree, distributes the updated key to clients and runs the ACP and multi-group key management scheme. The event trigger generates the event messages randomly and the event messages include join, leave and switch. The client node decrypts the encrypted keys that are sent by key server, calculates the revision number from the ACP string and uses a hash function to update the keys. After comparing with the traditional multi-group, we observe that this scheme not only needs fewer instances of rekeying but also demonstrates superior performance. The analytical results show that the ACP key management scheme is more efficient than the multi-group scheme and also allows a higher switching rate. From the simulation results, we also find that the rekeying time for the Multi-group scheme increases more quickly than that for the ACP scheme. This

study also compares the processing time and concludes that our scheme requires less processing time at the server than the multi-group scheme. The only trade-off to achieving this reduced time is that the ACP key management scheme increases processing time a little bit at the client.
This abstract proposes the concept of ACP key management, which is suitable for group communication within a network. When a node needs to switch group, the key server inserts the revision number in the polynomial string. After receiving this string, the node uses its key as polynomial input parameters to calculate the revision number. This node thus can automatically generate the middle key through the one-way hash function. Furthermore, to optimize the ACP scheme, the key server picks the same keys of the received nodes to generate the ACP string. This procedure markedly reduces the consuming power and helps save processing time at both the server and client. Until now, the proposed scheme is only designed to process the switching action of a single node. We expect to consider batch operations in the near future.

REFERENCES

[1] Pour, A. N., Kumekawa, K., Kato, T., & Itoh, S., A hierarchical group key management scheme for secure multicast increasing efficiency of key distribution in leave operation, *Computer Networks*, 2007, 51 (17), 4727-4743.

[2] Sherman, A., & McGrew, D., Key establishment in large dynamic groups using one-way function trees, *IEEE Transactions on Software Engineering*, 2003, 29 (5), 444-458.

[3] Wallner, D. M., Harder E. J., & Agee, R. C., Key management for multicast: issues and architectures, *IETF Network Working Group, RFC 2627*, 1999.

[4] Zhang, Q., and Wang, Y., A centralized key management scheme for hierarchical access control, *Proceedings of IEEE GLOBECOM'04*, 2004, 4, 2067-2071.

[5] Zhang, Q., Wang, Y., & Jue, J. P., A key management scheme for hierarchical access control in group communication. *International Journal of Network Security*, 2008, 7 (3), 323-334.