Enabling Secure Multicast Using a New Java LKH Rekeying Tool

Josep Pegueroles and Francisco Rico-Novella

Departamento de Ingeniería Telemática. Universitat Politècnica de Catalunya.
Jordi Girona 1 y 3. Campus Nord, Mod C3, UPC. 08034 Barcelona. Spain
{josep.pegueroles, telfrn}@entel.upc.es

Abstract. The most promising protocols for multicast key management are those based on multilevel logical binary trees of Key Encryption Keys (KEKs). One of these protocols, Logical Key Hierarchy (LKH) has become a standard de facto today. This work presents an open implementation of LKH protocol for Java applications that enables the use of Multicast Security in web applications.

1 Introduction

When multicast applications pretend to be commercial, security services become essential. Confidentiality in such environment need a common secret shared by all members. This provides group secrecy and source authentication. The shared key must be updated every time a member joins or leaves the group [1]. In this scenario, where session keys are continuously changing, Key Management (KM) has an important role. The most promising multicast KM protocols are those based on binary trees of Key Encryption Keys (KEK) [2,3]. This approach results in a quite efficient key update mechanism in which the number of multicast messages needed for membership update is not proportional to the number of members in the group but to the tree depth. If KEK tree is balanced, depth of the tree is logarithmic in the size of the secure multicast group $O(\log N)$, same behavior shows the number of multicast messages needed for key update. Logical Key Hierarchy algorithm (LKH) has become the most accepted one. Neither applications nor libraries allowing secure multicast rekeying are known prior to this work.

2 JavaLKH Rekeying Tool

According to that, the Information Security Group [4] of the Technical University of Catalonia has developed a Java tool enabling secure multicast rekeying for multicast networks by means of LKH algorithm. All the efforts were focused on the reuse of code for future implementations of other algorithms. The tool takes advantage of the encryption utilities of JCE [5] and uses the provider-architecture in order to have algorithm and implementation independence.

JavaLKH has 4 main parts. The core classes for the easy management of key tree. The proper classes in order to implement the LKH itself. The class tree programmed for the generation of valid packet formats in GDOI (Group Domain of Interpretation) IETF standard proposal [6]. And two standalone applications (one for the client, and one for the key server) that help the testing of the programmed classes. The developed modules can be downloaded for test purposes from [7]. In this site can also be found the javaDoc class documentation, an easy HOWTO for newbies and a page with snapshots and examples of use.

The core package can be included in java applications in order to add key management functionality to new programs. Apart from that, the standalone Server and Client programs are very useful for educational and demonstrative purposes. These allow a Key Server to establish secure communications within a multicast group, maintaining the logical key tree and sending valid rekeying messages to the remaining group members every time a joining or a leaving is requested. The same way the client can easily join or leave the group and update the Key Session when needed.

3 Future Work

As future works for our research and development are pointed out the improvement of performance by the programming of more efficient algorithms for managing trees. After that, the extension of the capabilities of package to other rekeying algorithms such as OFC [3] or Balanced Batch Rekeying [8] and inclusion of them in a provider structure are also expected to be achieved in the near future.

Acknowledgments. This work is supported by the Spanish research council under projects DISQET (CICYT TIC2002-818) and CREDO (CICYT - TIC2002-00249)

References

1. Wallner, Harder, Agee. Key Management for Multicast: Issues and Architectures. RFC2627
2. Harney, Harder. Logical Key Hierarchy Protocol (LKH). Mar 99. I-D Harney-sparta-lkhp-sec-00. Work in progress
3. Canetti, Malkin, Nissim. Efficient Communication Storage Tradeoffs for Multicast Encryption. Eurocrypt99 pp 456–470 1999
4. UPC's Information Security Group web page <http://isg.upc.es>
5. Pistoia. Java 2. Network Security. Prentice Hall PTR. 1999.
6. Baugher, Hardjono, Harney, Weis. The Group Domain of Interpretation. I-D draft-ietf-msec-gdoi-07. Work in progress
7. Download Site for Secure multicast application. <http://isg.upc.es/gsec/download>.
8. Pegueroles, Rico-Novella. Balanced Batch LKH: New Proposal, Implementation and Performance Evaluation IEEE Symposium on Computers and Communications – 2003