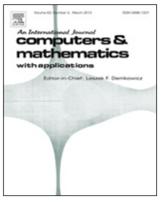
Computers and Mathematics with Applications 65 (2013) 1360–1368

Contents lists available at SciVerse ScienceDirect

Computers and Mathematics with Applications

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



Centralized key distribution protocol using the greatest common divisor method

P. Vijayakumar [a,*], S. Bose [a], A. Kannan [b]

[a] Department of Computer Science and Engineering, Anna University, Chennai-600 025, Tamilnadu, India

[b] Department of Information Science and Technology, Anna University, Chennai-600 025, Tamilnadu, India

ARTICLE INFO

Keywords:
Multicast communication
Key distribution
GCD
Computation complexity
Karatsuba multiplication
Extended Euclidian algorithm

ABSTRACT

Designing a key distribution protocol with minimal computation and storage complexity is a challenging issue in secure multimedia multicast. In most of the multimedia multicast applications, the group membership requires secured dynamic key generation and updation operations that usually consume much of the computation time. In this paper, we propose a new GCD (Greatest Common Divisor) based Key Distribution Protocol which focuses on two dimensions. The first dimension deals with the reduction of computation complexity which is achieved in our protocol by performing fewer multiplication operations during the key updation process. To optimize the number of multiplication operations, the existing Karatsuba divide and conquer approach for multiplication is used in this proposed work. The second dimension aims at reducing the amount of information stored in the Group Center and group members while performing the update operation in the key content. The proposed algorithm which focuses on these two dimensions has been implemented and tested using a Cluster tree based key management scheme and has been found to produce promising results. Comparative analysis to illustrate the performance of various key distribution protocols is shown in this paper and it has been observed that this proposed algorithm reduces the computation and storage complexity significantly.

© 2012 Elsevier Ltd. All rights reserved.

1. Introduction

Multimedia services, such as pay-per-view, videoconferences, some sporting events, audio and video broadcasting are based upon multicast communication where multimedia messages are sent to a group of members. In such a scenario, groups can be either opened or closed with regard to senders. In a closed group, only registered members can send messages to this closed group. In contrast, data from any sender is forwarded to the group members in open groups. Groups can be classified into static and dynamic groups. In static groups, membership of the group is predetermined and does not change during the communication. In dynamic groups, membership can change during multicast communication. Therefore, in a dynamic group communication, members either join or depart from the service at any time. When a new member joins into the service, it is the responsibility of the Group Center (GC) to disallow new members from having access to previous data. This provides backward secrecy in a secure multimedia communication. Similarly, when an existing group member leaves from any group, he/she should not have further access to data. This achieves forward secrecy. In order to provide forward and backward secrecy the keys are frequently updated whenever a member joins/leaves the multicast service. Furthermore, if a device lacks storage capabilities, it may be impossible within the receiving device to implement a group key management protocol based on a key tree structure. Hence the amount of information to be stored to find the updated key by GC and group

[*] Correspondence to: Department of CSE, University College of Engineering Tindivanam, Melpakkam, Tamilnadu, 604001, India. Tel.: +91 9940896665.
E-mail addresses: vijibond2000@gmail.com (P. Vijayakumar), sbs@cs.annauniv.edu (S. Bose), kannan@annauniv.edu (A. Kannan).

0898-1221/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2012.01.038

members should also be minimized. GC also takes care of the job of distributing the Secret key and Group key to the group members. In this paper, we propose a Key Distribution algorithm that reduces the computational complexity and at the same time, it decreases the number of keys to be stored by GC and group members. The remainder of this paper is organized as follows: Section 2 provides the features of some of the related works. Section 3 discusses the proposed key distribution protocol and a detailed explanation of the proposed work. Section 4 explains the Cluster tree based key management where the proposed key distribution is employed. Section 5 provides the optimization method based on Karatsuba multiplication approach for key updation process. Section 6 analyzes the comparative performance of our proposed algorithm with the other existing key distribution methods. Section 7 gives concluding remarks and suggests some future directions.

2. Literature survey

There are many works that are present in the literature on key management and key distribution [1–13]. In most of the existing Key Management schemes, different types of group users obtain a new distributed multicast Group key which is used for encrypting and decrypting multimedia data for every session update. Among the various works on key distribution, Maximum Distance Separable (MDS) [14] method focuses on error control coding techniques for distributing re-keying information. In MDS, the key is obtained based on the use of Erasure decoding functions [15] to compute session keys by the GC/group members. Moreover, the Group center generates *n* message symbols by sending the code words into an Erasure decoding function. Out of the *n* message symbols, the first message symbol is considered as a session key and the group members are not provided with this particular key alone by the GC. Group members are given the $(n-1)$ message symbols and they compute a code word for each of them. Each of the group members uses this code word and the remaining $(n-1)$ message symbols to compute the session key. The main limitation of this scheme is that it increases both computation and storage complexity. The computational complexity is obtained by formulating $l_r + (n-1)m$ where l_r is the size of *r* bit random number used in the scheme and *m* is the number of message symbols to be sent from the group center to group members. If $l_r = m = l$, computation complexity is *nl*. The storage complexity is given by $\lceil \log_2 L \rceil + t$ bits for each member. *L* is number of levels of the Key tree. Hence Group Center has to store $n\,(\lceil \log_2 L \rceil + t)$ bits.

Secure communication using the extended Euclidean algorithm [16] was proposed for centralized secure multicast environments. The main advantage of this algorithm is that only one message is generated per rekeying operation and only one key is stored in each user's memory. In this algorithm, two values (δ, L) are computed in the intermediate steps of GC. The main limitation of the Euclidean algorithm is that the two computed values must be relatively prime. If this is not the case, then the algorithm fails in which the user cannot recover the secret information sent by GC. Also, the time taken for defining a new multiplicative group is high, whenever a new member joins or leave the multicast operation. This approach is only suitable for a star based key management scheme.

The Data Embedding Scheme proposed in [17] is used to transmit a rekeying message by embedding the rekeying information in multimedia data. In this scheme, the computation complexity is $O(\log n)$. The storage complexity also increases to the value of $O(n)$ for the server machine and $O(\log n)$ for group members. This technique is used to update and maintain keys in a secure multimedia multicast via a media dependent channel. One of the limitations of this scheme is that a new key called an embedding key has to be provided to the group members in addition to the original keys, which causes a lot of overheads. A level homogeneous key tree [18] based key management scheme was proposed in [19] to reduce computation and storage complexity. A Key management scheme using key graphs has been proposed by Wong Gouda [20] which consists of the creation of secure group and basic key management graphs scheme using a Star and Tree based method. The limitation of this approach is that scalability is not achieved. A new group keying method that uses one-way functions [21] to compute a tree of keys, called the One-way Function Tree (OFT) algorithm has been proposed by David and Alan. In this method, the keys are computed up the tree, from the leaves to the root. This approach reduces re-keying broadcasts to only about $\log n$ keys. The major limitation of this approach is that it consumes more space. However, the time complexity is more important than space complexity. The storage complexity of GC is $2nK$ and group member is LK, where K is the key size in bits. In our work, we focused on reduction of computation time complexity.

Wade Trappe and Jie Song proposed a Parametric One Way Function (POWF) [22] based binary tree Key Management. Each node in the tree is assigned a Key Encrypting Key (KEK) and each user is assigned to a leaf and given the IKs of the nodes from the leaf to the root node in addition to the session key. These keys must be updated and distributed using top down or bottom up approach. The storage complexity is given by $(\log_\tau n)+2$ keys for a group center. The amount of storage needed by the individual user is given as $\frac{(\tau^{L+1}-1)}{\tau-1}$ keys. Computation time is represented in terms of amount of multiplication required. The amount of multiplication needed to update the KEKs using the bottom up approach *is* $\tau \log_\tau n - 1$. Multiplication needed to update the KEKs using the top down approach *is* $\frac{(\tau-1)\log_\tau n\left(\log_\tau n+1\right)}{2}$. This complexity can be reduced substantially if the numbers of multiplications are reduced. Therefore, in this paper we propose a new cluster tree based Key Management Scheme that reduces computation time by reducing the number of multiplications required in the existing approaches. We also use the Karatsuba fast multiplication algorithm to optimize the multiplication operations used in the key distribution protocol in the GC. The proposed method also reduces the amount of information that needs to be stored for updating the keys when there is a change in the group membership. Our proposed algorithm is suitable for single join/leave operation (Single Rekeying operation). When the number of joining or leaving operations is more, batch join and leaving operations can be integrated for a group of users in our proposed key distribution protocol. To perform batch joining and leaving operation

Download English Version:

https://daneshyari.com/en/article/470513

Download Persian Version:

https://daneshyari.com/article/470513

Daneshyari.com