International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 4 Issue 1 –DECEMBER 2013.

Secure Multicast Key Distribution Systems In Large Dynamic Groups

R.Venkatesan[1], G.Saravanan[2] and J.Santhosh kumar[3]

Department of Computer Science and Engineering, Excel Engineering College

[1]venkatesan.phd2012@gmail.com, [2]gsaravanan.xl@gmail.com and [3]commail4sant@gmail.com

***Abstract*- Efficient key distribution is an important problem for secure group communications. The communication and storage complexity of multicast key distribution problem has been studied extensively. In this paper, we propose a new multicast key distribution scheme whose computation complexity is significantly reduced. Instead of using conventional encryption algorithms, the scheme employs MDS codes, a class of error control codes, to distribute multicast key dynamically. This scheme drastically reduces the computation load of each group member compared to existing schemes employing traditional encryption algorithms. Such a scheme is desirable for many wireless applications where portable devices or sensors need to reduce their computation as much as possible due to battery power limitations. Easily combined with any key-tree-based schemes, this scheme provides much lower. Computation complexity while maintaining low and balanced communication complexity and storage complexity for secure dynamic multicast key distribution.**

I. INTRODUCTION

The objectives of the project are to generate a group key among the group. The communication with other group members is done with the help of the group key. The communication such as sharing the resources like sending the files to other group members is done with the group key. The dynamic nature of the system allows the existing members to leave the group while new members can join, Instead of performing individual re keying operations we are going to re key for a batch of join operations. The system uses Queue-batch algorithm for re-keying.

The problem with the centralized key server is that all the members depend on centralized key server for key generation. All the members depend on the centralized key server for key generation. Re keying is performed whenever there is any groupmembership change including any new member joining or any existing member leaving the group. More resources have been utilized by the server in case of multiple join and leave of members in the group. So we are going for collaborative key agreement in which all nodes become a part of the group key. The communication in the group is done with the help of the group key. Instead of performing individual re keying operations, i.e., re computing the group key after every join or leave request, we are going to re key for a batch of join operations.

II. MODELLING

Central Group Controller

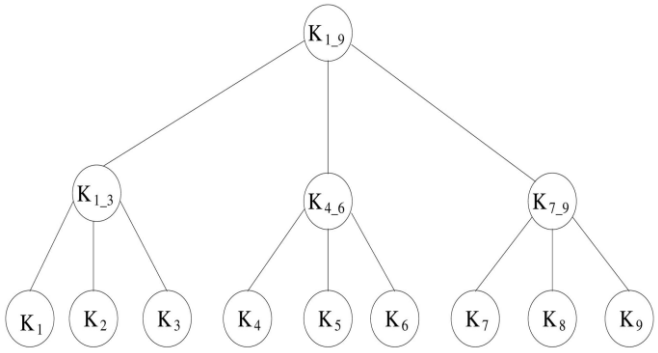
Each session thus needs anew key that is only known to the current session members,i.e., session keys need to be dynamically distributed toauthorized session members.In this paper, we study how a multicast group key canefficiently be distributed in computation. We adopt acommon model where session keys are issued and distributedby a central group controller (GC), as it has much lesscommunication complexity, as compared to distributed keyexchange protocols, which is a very desired property inmost wireless. Theresources needed for the GC to distribute session keys togroup members include communication, storage, andcomputation resources. The communication complexity isusually measured by the number of data bits that need tobe transmitted from the GC to group members to conveyinformation of session keys, whereas the storage complexityis measured by the number of data bits that the GC andgroup members need to store to obtain session keys.Another similarly important but usually undernoticed, ifnot ignored, factor is the computation complexity, which canbe measured by the number of computation operations (orthe computation time on a given computing platform) thatthe GC and group members need to distribute and extractsession keys. Hereafter, the problem of how resources caneffectively be used to distribute session keys is referred to asthe group key distribution problem.

The group key distribution problem has been studiedextensively in the larger context of key management forsecure group communications mainly on balancingthe storage complexity and the communicationcomplexity. There are two trivial schemes for distributinga session key to a group of n members. The first one is thatthe GC shares an individual key with each group member,which can be used to encrypt a new group session key. Inthis scheme, the communication complexity is O(n),whereas the GC needs to store O(n)key information, eachmember stores O(1)key information, and O(n)encryptionand decryption operations are needed. In the secondscheme, the GC shares an individual key with each subsetof the group, which can then be used to multicast a sessionkey to a designated subset of group members. Now, boththe communication complexity and the computation complexityreduce to O(1), but at the cost of increasing thestorage complexity to O2(n)for both the GC and each groupmember. It is easy to see that neither scheme works forpractical applications with a reasonable group size n. Thus,research efforts have been made to achieve low communicationand storage complexity for group key distribution. However, this threshold-based scheme canonly distribute a session key to a designated group ofmembers for one-time use. Once a session key is distributedto the group, any member can calculate the secretinformation that other members in the same group hold.

Itsprohibitively high communication complexity and computationcomplexity make it only practical for a very smallgroup with limited number of members. Various theoreticalmeasures and schemes for group key distribution wereintroduced. Along the same line, many researchefforts have been made on balancing communicationcomplexity and storage complexity of the group keydistribution problems.For a multicast group with a large number of members,key-tree-based schemes were introduced to decompose a largegroup into multiple layers of subgroups with smaller sizes. Using these schemes, a group membershipchange can be effectively handled in the correspondingsubgroups without affecting other ones. Thus, the communicationcomplexity is reduced, but at the cost of increase instorage and computation complexity together with extracommunication delays. For a group of n members, key-treebasedschemes have a communication complexity of O(logn) and a storage complexity of O(n)for the GC and O(logn)foreach group member. It has been shown that if a groupmember can store at mostO(logn)keys, then the lower boundof the communication complexity is O(logn)if a structurepreservingprotocol is used for group key distribution.

The Basic Scheme To Key Trees

The basic key distribution scheme reduces computation complexity by replacing computationally expensive encryption and decryption operations with more efficient erasure decoding operations of MDS codes. This basic scheme has the same communication complexity as conventional key distribution schemes using secure unicasts. Thus, the basic scheme can be readily used as a building block to replace encrypted unicasts in any key distribution schemes, particularly schemes with low communication complexity. To reduce the communication complexity of rekeying operations, a key-tree-based scheme and many of its variations have been proposed. This scheme reduces the communication complexity of rekeying operations to OðlognÞ, whereas each member needs to store OðlognÞ keys, and the GC needs to store



Maximum Distance Separable Codes

Maximum Distance Separable (MDS) codes are a class of error control codes that meet the Singleton bound. Letting GF(q) be a finite field with q elements, an (n. k) (block) error control code is then a mapping from GF(q)k to GF(q)n : E(m) = c, where m = m1m2 …. mk is the original message block, c = c1c2……. cn is its code word block, and E(.) is an encoding function, with k - n. If a decoding function D(.) exists such that D(ci1ci2 …… cik, i1, i2 …… ik) = m for 1 <ij< n and 1< j < k, then this code is called an (n, k) MDS code. For an (n, k) MDS code, the k original message symbols can be recovered from any k symbols of its code word block. The process of recovering the k message symbols is called erasure decoding. All the symbols are defined over GF(q), and usually, q = 2m. The well-known Reed-Solomon (RS) codes [28] are a class of widely used MDS codes. Notably, the RS codes and other MDS codes can be used to construct secret-sharing and threshold schemes. Description of the Basic Scheme For a dynamic multicast group, a session key is issued by a GC. Using this session key, the GC can establish a secure multicast channel with the authorized group members. Every time group memberships change because of the join or leave of some group members, the GC reissues a new session key, which is independent of all the old session keys. This rekeying procedure ensures the security of the current session and that of the old sessions, i.e., the newly joined members cannot recover the communications of the old sessions, and those old members who left the group cannot access the current session. Thus, both the backward secrecy and the forward secrecy of group communication are maintained. The complexity of the rekeying operation is asymmetric between a new member's join and an old member's leave. When a new member joins, the GC can easily multicast the

new session key encrypted by the current session key to all the current members, followed by a unicast to the new member to send the new session key encrypted by a predetermined encryption key shared between the GC and the new member. Thus, join is easy, with low communication and computation cost. However, when an old member leaves, the current session key cannot be used to convey the new session key information securely, since it is also known to the old member. Thus, hereafter, we will focus on the rekeying operation for a single member leave. The same idea can easily be extended to other rekeying operations such as batch rekeying

In any key distribution schemes, a basic operation is needed to distribute a piece of secret data to a small group of n members, where each member shares a different individual key with the GC. In all current existing schemes, this operation is fulfilled by the GC using n encryptions, followed by n unicasts. Now, we describe a new scheme that realizes this operation by using one erasure decoding of certain MDS code, followed by one multicast to all the n members. We call this scheme the basic scheme of key distribution. We will then show that this basic scheme can be easily integrated into any key distribution scheme, especially the schemes based on key trees, to reduce computation cost.

The basic scheme consists of three phases: 1) the initialization of the GC, 2) the join of a new member, and 3) the rekeying procedure whenever a group member leaves. Here again, a targeted multicast group has n members.

Groupcontroller Initialization

Initially, the GC constructs a nonsystematic(L,n). MDS code C overGF(q) and a secure one-way hash function H(.) whose codomain is GF(q). The domain of H(.) can be an arbitrary space F that is large enough so that H(.) has a secure one-way property: given any arbitrary

Y belongs GF(q), it is impossible or computationally hard to derive

X belongs F such that H(x) = y. Since other strong properties such as second-preimage resistance are not necessary, the hash function H can be implemented more efficiently. The GC then makes both the MDS code C and the one-way hash function H public.

Member Initial Join

Whenever a new member i is authorized to join the multicast group for the first time, the GC sends it (using a secure unicast) a pair (ji; si), where si is a random element in H(.)'s domain F, and ji is a positive integer satisfying ji not equal to jk for all k's, where k is a current member of the multicast group. The pair (ji; si) will be used as member i's seed key (denoted as Si) and is kept in the GC's local database, as long as member i remains a potential member of the multicast group.

Rekeying

Whenever some new members join or some old members leave a multicast group, the GC needs to distribute a new session key to all the current members. As already discussed, we will focus on the rekeying operation when an old member leaves. After an old member leaves, the GC needs to distribute a new key to n remaining members to achieve both forward and backward secrecy of the session key.

The GC executes the rekeying process in the following steps:

1. The GC randomly chooses a fresh element r in F, which has not been used to generate previous keys.
2. In the remaining group of n members, for each member i of the current group with its seed key (ji; si), the GC constructs an element cji in GF(q) : cji =H(si + r), where + is a simple combining operation in F, for example, string concatenation.
3. Using all the cji 's in the above step, the GC constructs a code word c of the (L; n) MDS code C: set the (ji)th symbol of the code word c to be cji . Since C is an (L, n) MDS code, the code word c is uniquely determined by its n symbols. Using an efficient erasure decoding algorithm for C, the GC can easily calculate the n corresponding message symbols m1, m2 ….mn.
4. The GC sets the new session key k to be the first message symbol m1 : k = m1.
5. The GC multicasts r and m2 …… mn.

III. Conclusion and Future Work

In this project we have presented a dynamic multicast key distribution scheme using MDS codes. The computation complexity of key distribution is greatly reduced by employing only erasure decoding of MDS codes instead of more expensive encryption and decryption computations. Easily combined with key trees or other rekeying protocols that need encryption and decryption operations, this scheme provides much lower computation complexity while maintaining low and balanced communication complexity and storage complexity for dynamic group key distribution. This scheme is thus practical for many applications in various broadcast capable networks such as Internet and wireless networks.

IV. References

[1] M. Blaum, J. Bruck, and A. Vardy, "MDS Array Codes with Independent Parity Symbols," IEEE Trans. Information Theory, vol. 42, no. 2, pp. 529-542, Mar. 1996.

[2] A. Fiat and M. Naor, "Broadcast Encryption," Advances in Cryptology—Proc. 13th Ann. Int'l Cryptology Conf. (CRYPTO '94), pp. 480-491, 1994.