Vol. 1, Issue 4, pp. 1441-1447

A Novel Approach for Secured Symmetric Key Distribution in Dynamic Multicast Networks

Y.V.Srivani*, T.Sudha**
Associate Professor , C.S.E*, Assistant Professor , C.S.E**
Hyderabad Institute of Technology and Management[HITAM},
Hyderabad, A P, India.

ABSTRACT

Multicasting is an efficient means of distributing data in terms of resources usage. All the designated receivers or members in a multicast group share a session key. Session keys shall change dynamically to ensure both forward secrecy and backward secrecy of multicast sessions. The communication and storage complexity of multicast key distribution problem has been studied extensively. We implement 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, distribute multicast key dynamically. This scheme considerably reduces the computation load on each group member as compared to existing schemes employing traditional encryption algorithms. It easily combined with any key-tree-based schemes and provides much lower computation complexity while maintaining low and balanced communication complexity and storage complexity for secure dynamic multicast key distribution.

Keywords-**distribution, multicast, MDS codes, computation complexity, erasure decoding, GC.**

I. INTRODUCTION

Key Management is one of the security service required for many group oriented and distributed applications. In such applications data can be communicated using a secure group key which helps in key distribution techniques. Multicast is an essential mechanism to achieve scalable information, distribution for group-oriented applications. Multicast refers to communication where information is sent from one or more parties to a set of other parties in terms of resource (such as network bandwidth, server computation and I/O load) usage. In this case, the information is distributed from one or more senders to a set of receivers, but not to all users of the group. The advantage of multicast is that, it enables the desired applications to service many users without overloading a network and resources in the server. Security is provided when data is transmitting through an insecure network. Unicast security has several schemes to provide the issues which cannot be extended directly to the multicast environment.

As the transmission takes place over multiple network channels, multicasting is more vulnerable than unicasting. In many applications, the multicast group membership change dynamically i.e. some new members are authorized to join a new multicast session while some old members should be excluded. In order to ensure both forward secrecy and backward secrecy, session keys are dynamically changed the forward secrecy is maintained if an old member who has been excluded from the current session cannot access the communication of the current session. And backward secrecy is guaranteed if a new member of the current session cannot recover the communication of past sessions. This requires each session need a new key that is only known to the current session members, i.e., session keys need to be dynamically distributed to authorize session members. Group key management is the major issue in multicast security, which is the fundamental technology to secure group communication by generating and updating secret keys [4]. Access control and data confidentiality can be facilitated using key management by ensuring that the keys used to encrypt group communication are shared only among the legitimate group members and only those members can access the group communication [5] .The shared group key can be used for authentication and also for encrypting the message from a legitimate group member. In order to prevent the above said problems in the secured multicast communication environment, the following two security criteria are used. *Forward Secrecy*, is maintained if an old member who has been evicted should not be able to access the messages from the current and future sessions. *Backward secrecy*, is guaranteed if a new member of the current session cannot recover the communication data of past sessions. The process of changing the session key and communicating the same to only the legitimate group members is called as Re keying. Group key management schemes are of three types. *Centralized key management*: group members trust a centralized server, referred to as the key distribution center (KDC), which generates and distributes encryption keys. *Decentralized scheme*: the task of KDC is divided among subgroup managers. *Contributory key management schemes*: Group members are trusted equally and all participate in key establishment. In this paper, we study how a multicast group key can efficiently be distributed in computation. In this a

centralized key management model is used where session keys are issued and distributed by a central group controller (GC), as it has much less communication complexity, when compared to distributed key exchange protocols[4]. The group controller uses the communication, computation and storage resources for distributing the session key to the group members. The main problem here is how the resources can be used to distribute the session key, which is referred to as group key distribution problem. There are two approaches that are generally used for distributing the session key to the group of n members. The first approach is that the group controller GC shares an individual key with each group member. This key is used to encrypt a new group session key. The second approach is that the group controller shares an individual key with each subset of the group, which can then be used to multicast a session key to a designated and Subset of group members. This approach has less communication, computation and storage complexity when compared to the other approach.

A multicast group with large number of members uses the key-tree-based approach. In this approach it decomposes a large group into multiple layers of subgroups with smaller sizes. Using this approach the communication complexity is much reduced, but the storage and computation complexity is increased. In this paper, the main aim is to reduce the rekeying cost. A new novel approach for computation efficient rekeying for multicast key distribution is introduced[4][5][14], which reduces the rekeying cost by employing a hybrid group key management scheme .It also maintains the same security level without increasing the communication and storage complexity. In this scheme, session keys are encoded using error control codes. In general encoding and decoding uses error control code to reduce the computation complexity. Thus, the computational complexity of key distribution can be significantly reduced.

1.1 Proposed System

For a time sensitive applications membership change occurs very frequently. In such environment the group controller only communicates the new session keys to the only existing members of the group. 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. The scheme employs MDS (Maximum Distance Separable) codes [3], 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. This ensures easy combination with any key-tree-based schemes and also provides much lower computation complexity while maintaining low and balanced communication complexity and storage complexity for secure dynamic multicast key distribution [10].

Advantages of the Proposed System:

The group controller responsibilities are shared by the Group control intermediate such as Re keying process and scalability of the group process. Use the Identity tree based structure:

1. The group members are not affected by the key generation process when they are willing to communicate with any other group members.
2. The Centralized key server used for key generation process and the KGC is also act as a Router for group to group communication.
3. The Re keying process is done only to the particular group members not to the entire group members.

II. BASIC SCHEME (KDC with MDS CODES)

MDS codes

Block codes that achieve equality in Singleton bound are called **MDS (maximum distance separable) codes**. Examples of such codes include codes that have only one codeword (minimum distance n), codes that use the whole of $(F_q)^n$(minimum distance 1). Maximum Distance Separable (MDS) codes are a class of error control codes. that meet the Singleton bound [13, chapter 11]. Letting GF(q) be a finite field with q elements [13], an (n; k) (block) error control code is then a mapping from GF(q)k to GF(q)n : E(m) ¼ c, where m ¼ m1m2 _ _ _mk is the original message block, c ¼ 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) ¼ m for 1 _ ij _ n and 1 _ j _ k, then this code is called an (n; k) MDS code[3]. 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 symbol is called erasure decoding. All the symbols are defined over GF(q), and usually, q ¼ 2m. The well-known Reed-Solomon (RS) codes [4] are a class of widely usedMDScodes. Notably, the RS codes and other MDS codes [8] can be used to construct secret-sharing and threshold schemes [5] [7].

2.1 MDS Code Algorithm

MDS consists of three phases:

1) The initialization of the GC
2) The join of a new member
3) The re-keying procedure whenever a group member leaves.

Step 1: The initialization of the GC, Initially GC constructs a codeword C using MDS, C is over a finite space GF(q)

Step 2: One way Hash function H (.)[12], domain of H(.) is GF.
Step 3: Property of H(.) is H(x)=y, it is impossible to derive X from y.
Step 4: The joining of a new member: For a member I, Then GC sends (ji; si) as a unicast message.
Step 5: Ji= +ve integer ji!=jk.
Step 6: Si= random seed element.
Step 7: The re-keying procedure whenever a group member leaves Select r from space f, r should not be used already to generate the group key**.** For every member j GC constructs an element Cj in GF(q).
Step 8: Cj=H(Si+r).
Step 9: Member j every 'n' members in the group calculate these own codeword C1 C2……..Cn.

III. KEY GENERATION AND DISTRIBUTION FRAMEWORK

3.1 Key Generation

- **Private Key**

The Private Key is generated using MDS code. The GC sends his number of group members to the KGC (Key Generation Center). The keys are generated by the KGC and submitted to the GC.

- **Session Key**

In session key generation, initially sixteen decimal digits are generated by using random number generation method .Then each decimal digit is splited and compared with pre determined binary format. In DES algorithm the 64 bits session key is considered as a message file and generated user's private key is considered as a key file. DES algorithm encrypts the session key by using user's private key and transmitted to the appropriate users.

- **Join operation**

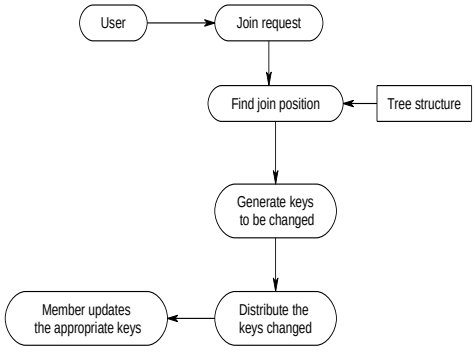

Fig 1: join operation

- **Join Request**

A Network node issues a request to GC to join the group. The GC checks whether the request is from an

authenticated member or not. If yes the GC accepts the request. Then the node communicates its session key through some secure channel as shown in figure 1.

- **Find join position**

The GC group controller maintains a tree structure and the tree structure is the logical arrangement of members. The GC (Group controller) traverses the tree structure and finds a position for the new member. The GC inserts the member details in this new position, which is a leaf node.

- **Generate keys**

From the new position onwards the GC generates the new key(s) along the path to root. The new keys are used to replace the old keys of the auxiliary nodes. Update tree structure Old keys are replaced by their corresponding new keys. Hence forth newly generated keys are used for future communication. This operation provides backward secrecy i.e. it prevents the newly joined member from accessing the previously communicated data.

- **Distribute keys**

A packet is constructed, which consists of newly generated key(s). This packet is encrypted using the old key known by a member or sub-group of members.

- **User-oriented re-keying**

In the user-oriented re keying, the GC constructs each re keying message. Rekey message contains the encrypted form of session key. So that they contain exactly all the messages that some user or a group of users need.

- **Key-oriented re-keying**

Key-oriented strategy emphasizes that each new key should be packed into a separate message and distributed to the holders

- **Leave operation**

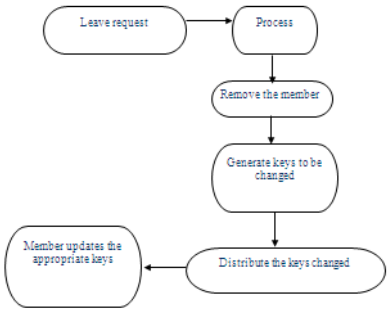

Figure 2: leave operation

- **Leave Request**

The member issues a request to leave the group as shown in figure 2 .

- **Process Request**

The GC checks whether the request is from an existing member, if so the GC accepts the request as shown in figure 2.

- **Find leave position**

The GC traverses the tree structure and finds the leaving position of the member. The GC then deletes the member details and removes the node from tree structure.

- **Generate keys**

From the leaving position onwards the GC generates the new key(s) along the path to root. Old keys are replaced by their corresponding new keys. Hence forth newly generated keys are used for future communication. This operation provides forward secrecy, i.e. it prevents the left member from accessing the data sent in future communication.

- **Distribute keys**

A packet is constructed, which consists of newly generated key(s). This packet is encrypted using the old key known by a member or sub-group of members. These new keys help the members to decrypt the messages sent in future communication.

- **Member updates keys**

After receiving the message, the member updates the appropriate set of keys.

- **User-oriented re-keying**

In the user-oriented re keying, the group controller constructs each re keying message [14]. Rekey message contains the encrypted form of session key, So that they contain exactly all the messages that some user or a group of users need.

- **Key-oriented re-keying**

Key-oriented strategy emphasizes that each new key should be packed into a separate message and distributed to the holders.

3.2 Message Transmission

Multicasting is a process of sending a message to a selected group. Internet applications, such as online games, newscast, stock quotes, multiparty conferences, and military communications can benefit from secure multicast communications [10]. In most of these applications, users typically receive identical information from a single or multiple senders. Hence, grouping these users into a single multicast group and providing a common session encryption key to all of them will reduce the number of message units to be encrypted by the senders. Various types of data communication are broadcast, Multicast, group communication.

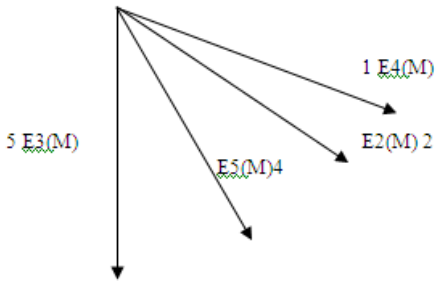

Figure 3: Transmission of the message M through 4 point-to-point connections.

Above figure 3 shows the transmission of message m to four point to point connections. Here node number 1 is the service provider. Nodes 2,3,4,5 are the receiving nodes. 2,3,4,5 Nodes are receiving the same message.

- **Group communication**

For group communications, the server distributes to each member a group key to be shared by all members [10]**.** The GC distributing the group key securely to all members requires messages encrypted with individual keys as shown in [figure 4]. Each such message may be sent separately via unicast. Alternatively, the messages may be sent as a combined message to all group members via multicast. Either way, there is a communication cost proportional to group size (measured in terms of the number of messages or the size of the combined message). Observe that for a point-to-point session, the costs of session establishment and key distribution are incurred just once, at the beginning of the session. On the other hand, group session may persist for a relatively long time with members joining and leaving the session. Consequently, the group key should be changed frequently. To achieve a high level of security, the group key should be changed after every join and leave so that a former group member has no access to current communications and a new member has no access to previous communications.

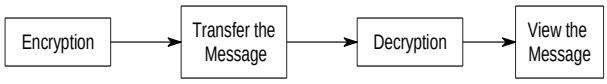

Figure 4: Group Communication

3.3 Cryptography

Cryptography is the process of protecting information by transforming it into an unreadable format called cipher text. Only those who possess a secret key can decrypt the message into text. Encryption is the process of conversion of original data (called plain text) into unintelligible form by means of reversible translation i.e. based on translation table or algorithm, which is also called enciphering. Decryption is the process of translation of encrypted text

(called cipher text) into original data (called plain text), which is also called deciphering. Cryptography plays a

major role in the security aspects of multicasting. For example, consider stock data distribution group, which distributes stock information to a set of users around the world. It is obvious that only those who have subscribed to the service should get the stock data information. But the set of users is not static. New customers joining the group should receive information immediately but should not receive the information that was released prior to their joining. Similarly, if customers leave the group, they should not receive any further information.

3.4 Authentication

The Login Module is used for the newly joined users to send a request to the Group Controller and it is used for to retrieve the Private keys after the Group Controller assign keys to the new users as shown in [figure 5]. The user login the group to enter the user Id and Private Key. If the user Id and private key is correct means the user view the inbox and outbox message otherwise to display the message box "Enter the correct Password".

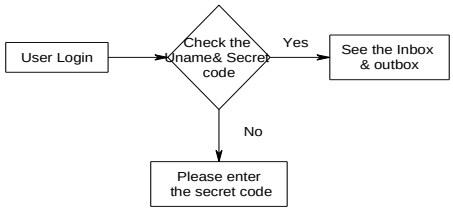

Figure 5: Authentication

- **System Flow Diagram**

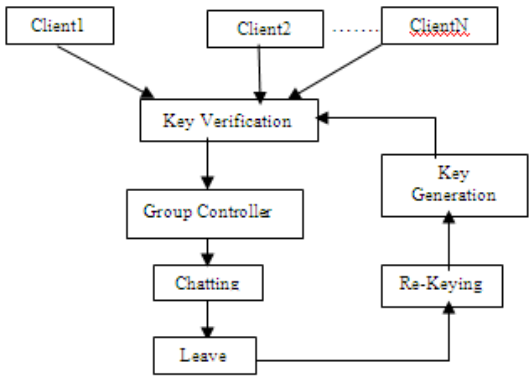

Figure 6: system Flow Diagram

IV. COMPARISON WITH TRADITIONAL CRYPTOGRAPHIC SCHEMES

To evaluate the proposed scheme, a multicast key distribution scheme is implemented to disseminate 128-bit session keys among a 3-ary balanced key tree. The pro-posed schemes compared with traditional cryptographic schemes. As the communication and storage complexity are the same among all the schemes, it suffices to simply

compare the computation complexity. The comparison considers the following scenario, where each three-member group has one member that departs. These departures are not constrained to happen at the same time, but un practice, they might tend to be close. For example at the end of one movie broadcast etc.This makes a batch proceed possible, which means that all remaining members could be rekeyed at once. Before reporting the experimental result, it is worth pointing out that any one-way hash function [112] used in the proposed scheme can be simplified from general-sense hash function implementations. For instance, we use the MD5 algorithm [9] as an exemplary hash function in our evaluation. Which produces a 128-bit hash output from any arbitrary length input? A general MDS input input consists of three components:1)input data 2)padding bits, and 3)a final 64-bit field for length. In our case, as the input is always 128 bits, we can present the final length field to represent 128. Moreover, all the rest bits can be set to 0 and removed from MDS logic. This tends to make MDS algorithm more efficient. Obviously, the same method can be readily applied to other hash algorithms. For example, SHA-1 and SHA 256[9].The MDS algorithm is considered insufficient or using longer session key becomes necessary Experiments are conducted to compare the computation complexity of the proposed scheme with the conventional cryptographic schemes. The particular cryptographic algorithms compares in the experiments are CAST-128[14], IDEA [8], AES [1], and RC4 [9], All keys are set to be 128 bits. To make the comparison as fair as possible, we use the widely adopted and highly optimized software-based cryptography implementation, i.e., the Cryptography [6] package .We use the same optimization flags to compile both the cryptography package and our own optimized RS code implementation. We disable the hardware acceleration (SSE/SSE2) and tend not to compare under that scenario, simply because it is not ubiquitous over all platforms yet (for example, not available on mobile devices).

Table-1Computational Time Comparing To the RC4 (Multicast Group Size of 59,049)

TIME(MS)	MDS	AES	IDEA	CAST-128	RC4
GC	23	81	71	99	227
MEMBER	5	23	82	23	61

The computation time of the key distribution is compared to conventional stream ciphers, as shown in Table 1, for a selected multicast group size. Notice that the computation times of both the GC and the member using the RC4 cipher are significantly larger than using other schemes. Even

though RC4 itself is a fast stream cipher, its key scheduling process has dominant effect in this particular scenario,

where only 128-bit data is encrypted/decrypted using any given key. Results under other multicast group sizes are similar, which are thus not duplicated here. Finally it is worth nothing that our basic scheme simply reduces computation complexity by replacing crypto-graphic encryption and decryption operations with more efficient encoding and decoding operations. It is orthogonal to any other schemes that use different rekeying protocols and procedures. This basic scheme can always be combined with any re-keying schemes that use cryptographic encryption and decryption operations. For example, this basic scheme can be readily adapted incorporate so the called one-way function tree scheme [6], where a different rekeying protocol on a key tree is used other than the traditional scheme, as described in Section 4, to further reduce the computation complexity. We leave this simple exercise to interested readers.

V. IMPLEMENTAION AND RESULTS

A member can register under a particular group controller by selecting one of the groups as shown in [figure 5.1]. When the member selects the group new session key is created for that group. This session key is used to send a request to the group controller where in the group controller maintains all the details regarding all the members of that particular group in a database as shown in the [figure 5.2]. Whenever the request goes to the group controller the session key is going to the compare with the exercising session key available if the session key matches then the member joins the group and active rekeying process starts as shown in [figure 5.3].

Figure 5.1 Group Login Window

Figure 5.2 Send Request Window

Fig 5.3 Group Controller member join Window

Whenever the member leave the group the existing data base is going to be updated automatically updated to the system a new session key is generated so that a new member can join the group. The enter data base is maintained by the group controller. The GC is responsible for generating the session key every time whenever the group member joins or leaves the group as shown in [figure 5.4 and 5.5].

Figure 5.4 Group Controller member Leave Window

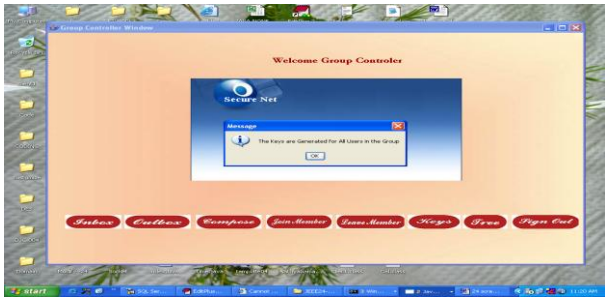

Figure 5.5 Keys Generated Window

VI. CONCLSION

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 in-stead 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.

VII. REFERENCE

[1] AES Algorithm (Rijndael) Information, http://csrc.nist.gov/Cryptotoolkit/aes/rijindael/, 2007.
[2]M.Abdalla y.shavitt,and A.wool,"Towards making broad cast encryption practical,"IEEE/ACM trans.networking,vol.8,no.4,pp.443-454,aug.2000
[3] M. Blaum, J. Bruck, and A. Vardy, "MDS Array Codes withIndependent Parity Symbols," IEEE Trans. Information Theory,vol. 42, no. 2, pp. 529-542, Mar. 1996.
[4] R. Blom, "An Optimal Class of Symmetric Key Generation Systems," Advances in Cryptology—Proc. Workshop Theory and Application of Cryptographic Techniques (EUROCRYPT '84).
[5] J. Snoeyink, S. Suri, and G. Varghese, "A Lower Bound for Multicast Key Distribution," Proc. IEEE INFOCOM '01, Apr. 2001.
[6]W.Dai,Crypto++Library,http://www.eskimo.com/~weidai/ cryptlib.html, 2007.
[7]J.Blomer,M.Kalfare,M.Karpinaki,R.Karp,M.Luby,and D.Zuckeman,"An XOR –Based Erasure-Resilient coding scheme", technical report tr-95-048,int'1 computer science inst,aug.
[8] L. Xu and J. Bruck, "X-Code: MDS Array Codes with Optimal Encoding," IEEE Trans. Information Theory, vol. 45, no. 1, pp. 272- 276, Jan. 1999.
[9] B. Schneier, Applied Cryptography, second ed. John Wiley & Sons,1996
[10] Jack Snoeyink, Subhash Suri, George Varghese "A Lower Bound for Multicast Key Distribution" IEEE PAPER,2001.
[11] Xiaozhou Steve Li, Yang Richard Yang, Mohamed G. Gouda, Simon S. Lam "Batch Rekeying for Secure Group Communications " IEEE PAPER,2009.
[12]A.T. Sherman and D.A. McGrew, "Key Establishment in Large dynamic group on way hash function trees",IEEE trans,software engg,vol 20,po 5,pp444-458,may 2003.
[13]F.J.macwilliams and N.J.A Sloane,The thory of error correcting codes,north-holland math.library,1977.
[14] Sanjeev Setia_ Samir Koussih_,Sushil Jajodia "A Scalable Group Re-Keying Approach for Secure Multicast" IEEE PAPER,2000.

Authors

Graduated in AM.I.E.T.E. from I.E.T.E, New Delhi, India, in 1997 and M.Tech in Computer science from Osmania University,Hyderabad, A.P.,India in 2003. Currently working in Hyderabad Institute of Technology and Management as Associate professor in CSE department (HITAM) R.R.Dist, A.P, and India. She has 8 years of experience. Her research interests include Data mining, Distributed Systems, Information Retrieval systems.

Sudha tadepu received Bachelor's degree in Computer science and Engineering from JNTUH, Pursuing **M.Tech** in Computer Science and Engineering from Hyderabad Institute of Technology and Management. She is a research scholar in field of Information Security. She can be reached at E-Mail: tadepus77@gmail.com.