International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 1– No.3, February 2012 – www.ijais.org

Analysis of Group Key Management Scheme using One-way Function Tree

Kiran R Khandarkar
Asst. Professor in CSE,
S. Y. College of Engineering,
Maharashtra.

Rahul B. Mapari
Asst. Professor in CSE, MIT,
Maharashtra.

ABSTRACT

Many real time network applications like teleconferences, online gaming, video-on-demand, Pay-per-view video streaming are based on group communications. These applications can be implemented by using secure multicast in which Group communication is secured by encrypting / decrypting data stream with a cryptographic key. Due to dynamic nature of group, the group key is needed to be changed dynamically to maintain backward secrecy and forward secrecy. In case of frequent join/leave operations, re-keying process becomes major issue. In this paper, a novel scheme for multicast group key establishment has been developed by using one-way function tree of degree three. The ternary OFT key tree is a particular type of ternary tree in which each interior node has maximum three children. Every leaf of the tree is associated with a group member, and the node secret of the root is the common group key. Group members can use this group key to communicate among themselves. This Scheme reduces overall runtime required for join/leave operations, lessens number of keys stored by group members, and requires minimum number of key-broadcasts to the group when new members are added or evicted. It also reduces computational cost of group manager in large dynamic multicast group.

Keywords

Multicast, dynamic group, forward secrecy, backward secrecy, logical key hierarchy, one-way function tree, secure group applications, cryptography, cryptographic protocols, group keying, rekeying, key establishment, key management.

1. INTRODUCTION

Multicast communications can greatly save bandwidth and resources of sender in delivering data to a group of recipients. Number of group-oriented applications, such as IPTV, Digital Video Broadcast, videoconferences, and interactive group games uses Multicast. Earlier uni-casting is the prominently used technique for such applications. Providing security in a group communication is a challenging issue with the dynamic nature of the group. The activity of key refreshment must be performed very often to maintain the forward and backward secrecy. The cost for key establishment and Key renewing is usually relevant to the group size and subsequently becomes a performance bottleneck in achieving scalability.

1.1 Cryptography

The art and science of keeping messages secure is cryptography. A message is plaintext or sometimes called as cleartext. The process of disguising a message in such a way as to hide its substance is encryption and encrypted message is called as ciphertext. The process of turning ciphertext back into plaintext is Decryption

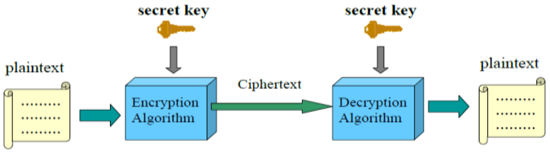


Figure 1: Encryption / Decryption with Key.

1.2 Unicast - Broadcast Multicast

Sending information to a single destination is called as **unicast**. Unicast is **one-to-one** approach. **Broadcast** is nothing but sending information from single source to all the destinations simultaneously. Broadcast is **one-to-many** approach.

Multicast is the delivery of information to a group of destinations simultaneously using the most efficient strategy to deliver the messages over each link of the network only once, creating copies only when the links to the multiple destinations split. Multicast is also **one-to-many (but selected entities)** approach. The multicast group can be identified with the class D IP address so that the members can enter or leave the group with the management of Internet group management protocol. The trusted model gives a scope between the entities in a multicast security system.

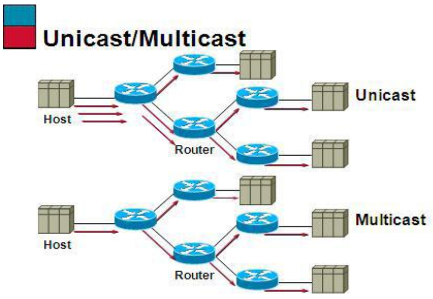


Figure 2 Unicast / Multicast Communication through routers

As Multicast applications have a vital role in enlarging and inflating of the Internet. The Internet has experienced explosive growth in last two decades. The number of the Internet users, hosts and networks triples approximately every two years. Also Internet traffic is doubling every three months partly because of the increased users, but also because of the introduction of new multicast applications in the real world such as video conferencing, games, ATM applications etc. broad casting such as www, multimedia conference and e-commerce, VOD (Video on Demand), Internet broadcasting and video conferencing require a flexible multicasting capability. Multicast is a relatively new form of communications where a single packet is transmitted to more than one receivers.

The Internet does not manage the multicast group membership tightly. A multicast message is sent from a source to a group of destination hosts. A source sends a packet to a multicast group specifying as the multicast group address. The packet is automatically duplicated at intermediate routers and any hosts that joined the group can receive a copy of the packet. Because a host can receive transmitted data of any multicast groups, secure communications is more important in multicasting than in uni-casting

1.2 Secure Multicast

For securing multicast, data stream in the communication is encrypted by the sender and decrypted by the group members. A symmetric key is used for encryption/decryption purpose. This key is referred as **group key** or **traffic encryption key.** The special Characteristics of a secure system include: Confidentiality, Integrity, Authentication, and Access control, Non-repudiation

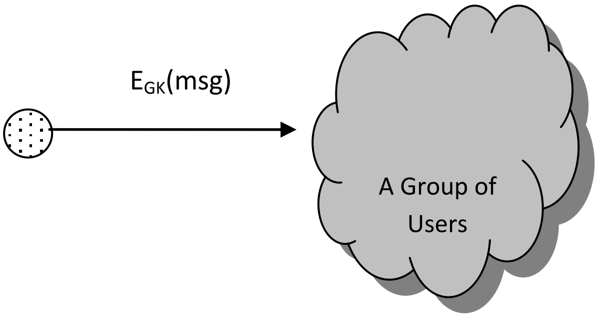


Figure 3 Secure Multicasts

Secure group communication refers to a setting in which a group of participants can send and receive messages to group members. In which outsiders are unable to glean any information even when they are able to intercept the messages. SGC is more important research area because many applications require it including the teleconferencing, telemedicine, real time information services, and collaborative application.

2. LITERATURE SURVEY

A variety of group keying solutions have been proposed, including methods based on a Simple Key Distribution Center (SKDC), information theoretic approaches, Group Diffie-Hellman (GDH), hybrid approaches that trade off information theoretic security against storage requirements, and recent centralized hierarchical methods including the Logical Key Hierarchy (LKH). In this section, brief review of some methods is taken, including ones that were developed before or after OFT.

2.1 Linear, Information-Theoretic Methods

The simplest solution is Simple Key Distribution Center (SKDC), in which a group manager shares a secret key with each group member and sequentially uses each member's key to communicate the secret group key to that member. Each time that a member is added to (or evicted from) a group with n members, the group manager must perform n encryptions and transmit n keys. This approach is attractive for its simplicity—at least for relatively small groups. A limitation of SKDC is apparent in the Spread System [2], which cannot handle groups of more than a few thousand members. Information theoretic approaches must satisfy the memory lower bounds of Blundo et al. [6] and use storage exponential in-group size.

2.2 Centralized Hierarchical Methods

For large groups that permit a centralized approach, the leading candidates are the centralized hierarchical methods, which scale logarithmically in group size. Recently, three such hierarchical methods have been proposed that do not require trusted internal nodes: the Logical Key Hierarchy (LKH) of Wallner [24] and Harney and Harder [16], the One-Way Function Tree (OFT) method of Sherman and McGrew [22], and Balenson et al. [4] and the One- Way Function Chain (OFC) of Canetti [8]. LKH is a top-down method in that it "pushes" new group keys down the key tree; OFT and OFC are bottom-up methods in that they derive new group keys from the leaves up to the root. In comparison with LKH, the bottom-up OFC and OFT methods broadcast approximately 50 percent less bits during a single member evict operation. Only OFT, however, has the option of being member contributory.

OFT was developed at Network Associates Laboratories (formerly, Trusted Information Systems) as part of the DARPA-funded Dynamic Cryptographic Context Management (DCCM) Project [10], [3]. In November 1997, Sherman [13] made the first presentation on OFT. The OFT algorithm is the subject of this document.

2.3 Common Terminology for Hierarchical Methods

To describe the three hierarchical methods OFT, LKH, and OFC in a common framework, it is convenient to introduce the following terminology. Each node v of the key tree has a node secret x_v and a node key k_v (in LKH each node key is equal to its node secret). The group key is the root node secret (i.e., the node secret of the root node). The shared keys are completely separate from the leaf node keys. The node keys are used to protect communications from the manager to certain members; the node secrets are used in OFT and OFC to derive the group key. It is sound security engineering to separate these two functions. In each method, each member must know its own shared secret and all node secrets along the path from the member to the root. The methods differ primarily in how they compute the node secrets and in what information must be broadcast. The most interesting operation is the evict operation. In each method, during an evict, all of node secrets known by the evicted member are changed; these node secrets are the node secrets for all nodes from the (leaf of the) evicted member to the root. The manager broadcasts Information sufficient to enable all remaining group members to compute the newly changed node secrets, including the new group key. The OFT and OFC methods require fewer broadcast bits than does LKH by exploiting a functional Relationship among certain node secrets. The logarithmic broadcast size stems from the fact that, for balanced trees, the length of the path from the evicted member to the root is logarithmic in the group size. The OFT, LKH, and OFC hierarchical methods can be described using the following common set of cryptographic primitives. To wrap broadcast messages for confidentiality in transmission, each of the hierarchical methods uses a symmetric encryption function E. Let $E_k($x). Denote the encryption of message x under key k.

To compute the node secrets and node keys, the OFT and OFC methods use two special one-way functions *f* and g. These functions are defined as the left and right halves of a pseudorandom function; they are not simply any one-way functions. The function *f* is used to compute functional relationships among node secrets; the function g is used to compute each node key from its corresponding node secret. Both *f* and g compute "blinded" values from the node secrets in such a way that protects the confidentiality of the node secrets; for any node secret χ , however, only *f* (χ) is referred as the "blinded node secret." Adapting an approach of Canetti [9] to OFT, we shall define the functions *f* and g in terms of a single length-doubling pseudorandom function H. Specifically, given H, define *f* and g by *f* || g = H, where || denotes concatenation. For example, in practice, one might define H based on a cryptographically secure hash function such as SHA-1 [12]. There are some subtle security advantages in using *f* and g as defined here, at least in terms of proving the security of the methods [23]. Required conditions on the component functions E, *f*, and g are discussed in detail in Sherman. In short, sufficient conditions for the security of OFT are that E is a secure encryption function and that H is pseudorandom (with *f* and g defined from H as explained above). It is also necessary for the security of OFT that E be secure.

For some applications, it is convenient to be able, for any node, to use its node key as a "subgroup key" for the subgroup of members that are descendants of the node (leaves of the subtree rooted at the node). This optional functionality is an attraction of the hierarchical model. The terminology and detailed functional relationships of OFT as described in this paper differ from those given in earlier drafts [21], [5] of this paper. For example, the *f* function in this paper corresponds to the "blinding function" g in the earlier drafts; the g function in this paper is new; and the *f* function in the earlier drafts is now simply bitwise exclusive-or. Our current formulation provides a cleaner system for which it is easier to prove security properties.

2.4 Computing Node Keys

In LKH, the manger chooses all node keys independently. In OFT each node key is computed by applying the function g to the node secret: For each node v, the node key k_v is computed by k_v. g. x_v. The node keys are used to protect communications from the manager. It is convenient to derive node keys from node secrets and doing so in this way enjoys certain security advantages.

2.5 Eviction Broadcasts

During an eviction, the node secrets (and, hence, node keys) change for each node along the path in the key tree from the evicted member to the root. The manager broadcasts certain encrypted information to enable the group members to learn the new node keys they need to know. Let v be any interior node in the key tree, and let L and R be, respectively, the left child and right child of v. In LKH, the manager chooses the new node keys. If node v is along this path, then the manager's broadcast would include the components $E_{KL}\ (k_v)$ and $E_{KR}\ (k_v)$. In OFT, the manager chooses a new node secret for the evictee's sibling and recomputes the function tree to the root. Assume without loss of generality that nodes L and v are along the evictee's path to the root. Then, the manager's broadcast would include the component E_{KR} [$f_{(xL)}$]. In OFC; the manager chooses a new secret for the evictee's parent and recomputes the function chain to the root. Assume without loss of generality that node v is along the evictee's path to the root and that the chain passes through nodes R and v but not L. Then, the manager's broadcast would include the component $E_{KL}\ (k_v)$.

3. TERNARY OFT SCHEME

The prototype of OFT Developed by Alan T Sherman uses a static tree, implemented in java language with in-order node numbering. The Key tree is constructed with all leaves always on the lowest level and initially the key tree is allocated for a certain maximum size. To grow beyond the size, the tree could be easily doubled in size by making the current tree as the left sub-tree of a new tree and broadcasting the new root node number without renumbering any existing nodes.

The OFT prototype uses 128 bit node key and SHA1 as one-way function. It uses blowfish encryption algorithm with no padding. In this prototype member names are equated with node numbers and node number never change. The prototype assigns numbers to all nodes in key-tree, with an in-order numbering scheme. Broadcast messages are sent by a preorder traversal of key-tree.

3.1 Extending concept of one-way function tree

OFT algorithm can be implemented by trees with different degrees of nodes. The OFT algorithm with *k-ary trees* is studied by Canetti, where *k* is degree of Tree. The optimal choice of *k* depends on cost metric which is to be minimized. The choice k = 2 minimizes broadcast size for single member eviction. To minimize total manager computation Wong et al. shown experimentally that k = 4 is best.

In this paper, trees with different degrees of nodes are considered such as k = 2, k = 3, k = 4, k = 5 etc. These trees are analyzed on the basis of manager computation and storage requirement of each member. The manager computation cost is calculated using formula given by A.T. Sherman as, $h\left(C_E + 2C_f\right) + C_r$, where h is height of the key-tree , C_E is computational cost of one evolution of the encryption function E, C_r is generating one key from a cryptographically secure random source, C_f is one evolution of the one-way function *f*. The member storage requirement is calculated using formula$(h * K$), where *h* is height of the tree and K is key-size in bits.

In this paper, a tree with degree three of nodes is used after analyzing above factors. In this paper dynamic ternary key-tree is implemented in C++ and linked with Network Simulator 2.34.

With dynamic nature of key-tree, height of the tree is maintained at minimum level. Due to this, new members can be added as close to the root as possible. Some part of tree is reconstructed after each join / leave operation. In this paper MD5 is used as one-way hash function. MD5 requires length of input as 64 bit less than exact multiple of 512. In this proposed scheme, input to MD5 is of 56 bits. As padding, single 1 and three hundred ninety one 0's are added. For encrypting group communication traffic DES with key size 56 bit is used.

Basic structure of proposed one-way function tree is as shown in figure 4

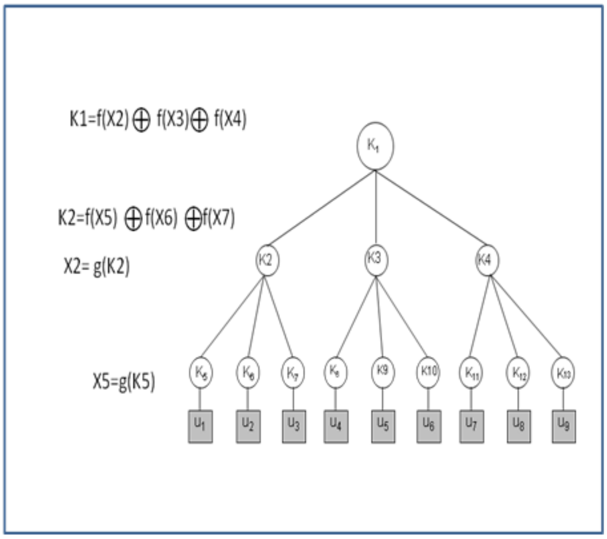


Figure 4 Basic structure of proposed one-way function tree

In the above figure, *K* is a secret of a node, *X* is key of a node, *f ( )* is one-way function for blinding the secret *g ( )* is one-way function for generating key from the secret. U1, U2, U3 up- to U9 are the group members which are attached to nodes K5, K6, K7 to K13 respectively. Member U1 knows the Key K5 and blinded keys of node K6, K7, K3, K4 by property of One-way function tree. Similarly other Group members knows the node key of the node to which they are attached and blinded node keys of the nodes which are siblings towards its path to the root. All group members can then compute the group key K1 locally. All group members can then communicate securely by using DES encryption algorithm and group key K1.

The exact savings in broadcast size of OFT over LKH depends on the parameter choices in the OFT and LKH implementations, including the key lengths and the output sizes of the one-way function and encryption function. The prototype of A.T. Sherman uses 128-bit group keys and the 160-bit SHA-1 one-way function, our initial prototype OFT achieved a savings in broadcast size over LKH of a factor of 2 x 128/160 = 1.6 times, which is slightly less than two. In this paper 56 bit group key is used with 64 bit Block cipher DES algorithm is used. Broadcast savings in this proposed scheme is of a factor of 3 x 56 /64 = 2.625 which is more than two.

As Manager Computation is minimum when d=3, the tree used for implementation of one-way function tree is a *ternary tree*. Each node of a ternary tree has 0, 1, 2 or 3 children.

Key management is done in bottom-up fashion. Group members are attached to the leaf nodes of the tree. Parent leaf node's key is generated by using special one-way function *g ( )* on the secret of group member. Secret of internal node is generated by combining blinded keys of all the child nodes and its key is generated by using special one-way function on its secret.

On every membership change in the multicast group, re-keying process is invoked. As the approach used in the proposed scheme for key management is bottom-up, the re-keying process is to be invoked at each internal node is from, the parent node of the subgroup where memberships change occurred to the root node. The key at root node is nothing but the group encryption key.

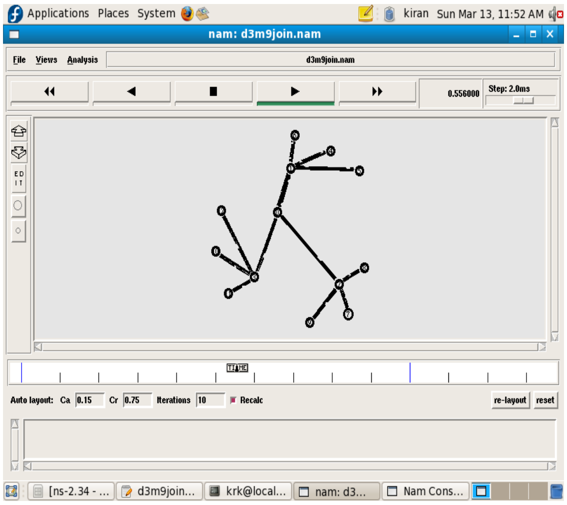
As the aim of paper is performance evaluation of key generation during membership change, it has been assumed that the member wishing to join has been authenticated previously.

4. SIMULATION DETAILS

The simulations were carried out in NS2 software, which is an event driven network simulator, developed at UC Berkeley that simulates variety of IP networks. It implements network protocols such as TCP and UPD, traffic source behavior such as FTP, Telnet, Web, CBR and VBR, router queue management mechanism such as DropTail, RED and CBQ, and more. NS also implements multicasting and some of the MAC layer protocols for LAN simulations. Currently, NS (version 2) written in C++ and OTcl (Tcl script language with Object-oriented extensions developed at MIT) is available.

Ns2 can be built and run both under Unix and Windows. An analysis of all the schemes discussed in this paper report is carried out using Network Simulator *ns* version *2.34.* In *Ns 2.34* we can define user defined protocols. For defining protocol, classes and their functionalities should be defined using C++. Parallel classes are needed to be defined using OTcl (Object-oriented Tool Command Language) and linking is done between the OTcl classes and C++ classes.

Ns have a built in centralized multicast protocol named CtrMcast. In this paper CtrMcast protocol is used as base Protocol for defining new user defined protocols.



0.556000 Step: 2.0ms

Auto layout: Ca 0.15 Cr 0.75 Iterations 10 Recalc

Figure 5 Simulation of join operation for group size of 9

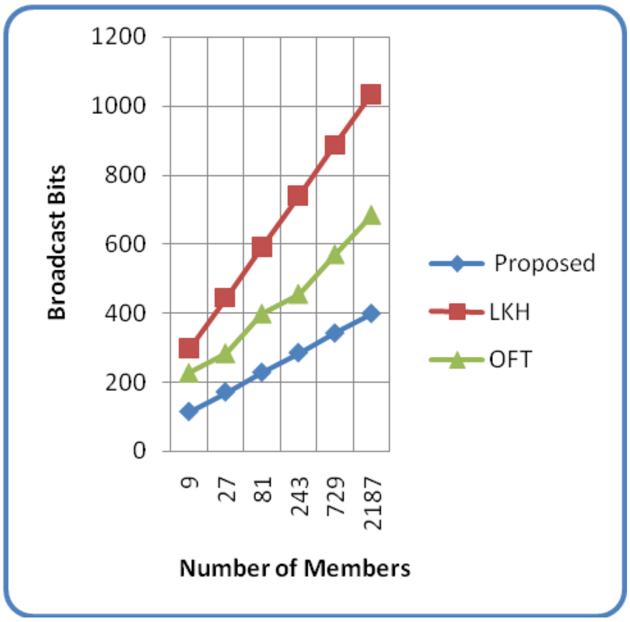


Figure 6: Chart of Broadcast bits required for membership change in LKH, OFT & Proposed Scheme.

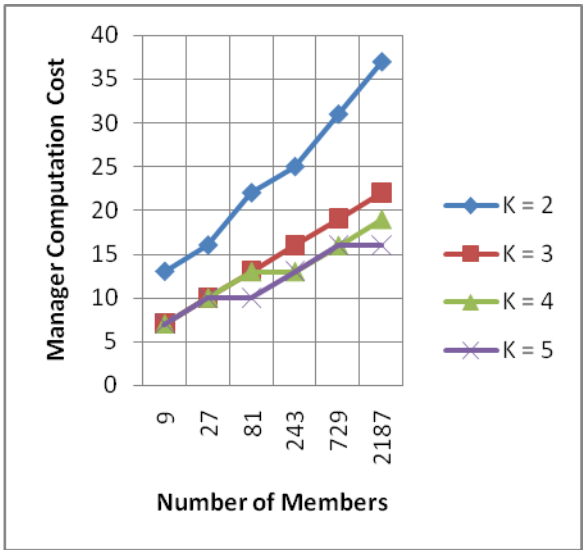


Figure 7: Chart showing Group Manager Computation for Join / leave operation with different Degrees of OFT Trees

5. CONCLUSIONS

In comparison with the first proposed hierarchical method—the Logical Key Hierarchy (LKH) and OFT our ternary OFT algorithm reduces more than half number of broadcast bits by the manager, per add or evict operation. The Proposed scheme is based on one-way function tree by changing the degree of the tree. The proposed scheme reduces manager computation as well as member key storage required during group operations. Dynamic tree is used for implementation of Key-tree, so that height of the tree is maintained at minimum level. From performance analysis of all the above schemes for computation and communication cost (i.e. overall run-time) of join / leave operation in dynamic multicast group, it has been observed that overall run-time is minimum for the proposed scheme. Communication cost when one-way function tree is used, is less as compared to logical key hierarchy. As one-way function tree is used in proposed scheme, communication cost is minimized.

6. REFERENCES:

[1] A.T. Sherman and D.A. McGrew, “Key Establishment in Large Dynamic Groups Using One-Way Function Trees,” IEEE Transaction on Software Engineering Vol. 29 No. 5, 2003

[2] Y. Amir, C. Danilov, and J. Stanton, “A Low Latency, Loss Tolerant, Architecture and Protocol for Wide Area Group Communications,” Proc. Int’l Conf. Dependable Systems and Networks (ICDSN), pp. 327-336, June 2000.

[3] D.M. Balenson, D.K. Branstad, D.A. McGrew, and A.T. Sherman, “Dynamic Cryptographic Context Management (DCCM): Report #1: Architecture and System Design,” TIS Report, 0709, TIS Labs at Network Associates, Inc., Glenwood, Md., June 1998.

[4] D.M. Balenson, D.A. McGrew, and A.T. Sherman, “Key Management for Large Dynamic Groups: One-Way Function Trees and Amortized Initialization,” Advanced

Security Research J.—NAI Labs, vol. 1, no. 1, pp. 27-46, 1998.

[5] D.M. Balenson, D.A. McGrew, and A.T. Sherman, "Key Management for Large Dynamic Groups: One-Way Function Trees and AmortizedInitialization," InternetDraft(workin progress), Internet Engineering Task Force, draft-irtf-smug-groupkeymgmt-oft-00.txt., July 2000.

[6] C. Blundo, A. de Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly-Secure Key Distribution for Dynamic Conferences," Advances in Cryptology: Proc. Crypto 92, E.F. Brickell, ed., pp. 471-486, 1992.

[7] M. Burmester and Y.G. Desmedt, "Efficient and Secure Conference Key Distribution," Secure Protocols, M. Lomas, ed., pp. 119- 130, 1997.

[8] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, and B. Pinkas, "Multicast Security: A Taxonomy and Efficient Constructions," Proc. IEEE Infocom 99, 1999.

[9] R. Canetti, T. Malkin, and K. Nissim, "Efficient Communication- Storage Tradeoffs for Multicast Encryption," Advances in Cryptology: Proc. Eurocrypt 99, Jacques Stern, ed., pp. 459-474, 1999.

[10] P. Dinsmore, D.M. Balenson, M. Meyman, P.S. Kruus, C.D. Scace, and A.T. Sherman, "Policy-Based Security Management for Large Dynamic Groups: An Overview of the DCCM Project," Proc. DARPA Information Survivability Conf. and Exposition (DISCEX '00), pp. 64-73, Jan. 2000.

[11] A. Fiat and M. Naor, "Broadcast Encryption" Advances in Cryptology: Proc. Crypto 93, D.R. Stinson, ed., pp. 481-491, 1993.

[12] FIPS Publication 180-1, "Secure Hash Standard," NIST, US Dept. of Commerce, Washington, D.C., Apr. 1995.

[13] A.T. Sherman, M. Harding, and D.A. McGrew, "A New Key-Management Algorithm for Large Dynamic Groups," transparencies from talk given by Alan Sherman at US NSA, Nov. 1997.

[14] D. Harkins and D. Carrel, "The Internet Key Exchange (IKE)," Internet Draft (work in progress), draft-ietf-ipsec-isakmp-oakley-08.txt, Internet Eng. Task Force, June 1998.

[15] H. Harney and E. Harder, "Multicast Security Management Protocol (MSMP): Requirements and Policy," Draft (work in progress), draft-harney-sparta-msmp-sec-00.txt, SPARTA, Inc., Mar. 1999.

[16] H. Harney and E. Harder, "Logical Key Hierarchy Protocol," Internet Draft (work in progress), draft-harney-sparta-lkhp-sec-00. txt, Internet Engineering Task Force, Mar. 1999.

[17] H. Harney and E. Harder, "Group Secure Association Key Management Protocol," Draft (work in progress), draft-harneysparta- gsakmp-sec-00, SPARTA, Inc., Apr. 1999.

[18] H. Harney, C. Muckenhirn, and T. Rivers, "Group Key Management Protocol (GKMP) Architecture," Request for Comments (RFC) 2094, Internet Eng. Task Force, July 1997.

[19] D.A. McGrew and A.T. Sherman, "Key Establishment in Large Dynamic Groups Using One-Way Function Trees," TIS Report No. 0755, TIS Labs at Network Associates, Inc., Glenwood, Md., May 1998.

[20] A.T. Sherman and D.A. McGrew, "Key Establishment in Large Dynamic Groups Using One-Way Function Trees," NAI Labs Technical Report No. 02-017, NAI Labs at Network Associates, Inc., Rockville, Md., July 2002.

[21] A.T. Sherman, "A Proof of Security for the LKH and OFC Centralized Group Keying Algorithms," NAI Labs Technical Report No. 02-043D, NAI Labs at Network Associates, Inc.,Rockville, Md., Nov. 2002.

[22] D.M. Wallner, E.J. Harder, and R.C. Agee, "Key Management for Multicast: Issues and Architectures," Internet Draft (work in progress), draft-wallner-key-arch-01.txt, Internet Eng. Task Force,Sept. 1998.

[23] C.K. Wong, M.G. Gouda, and S.S. Lam, "Secure Group Communications Using Key Graphs," Technical Report TR-97-23, Dept. of Computer Science, Univ. of Texas at Austin, July 1997.

[24] J. Alves-Foss, "An Efficient Secure Authenticated Group Key Exchange Algorithm for Large and Dynamic Groups," Proc. 23rd Nat'l Information Systems Security Conf. (NISSC), pp. 254-266, Oct.2000.

[25] William Stallings, "Cryptography and Network Security: Principles and Practices", Pearson Education, First Impression 2006.

[26] Yacine Challal, Abdelmadjid Bouabdallah, "Taxonomy of Group Key Management Protocols: Issues and Solutions" World Academy of Science, Engineering and Technology 2005

[27] M Vijaya saradhi , BH Ravikrishna "A Group Key Management Approach For Multicast Cryptosystems" Journal of Theoretical and Applied Information Technology.

[28] H. Krawczyk, M. Bellare , R. Canetti: HMAC: Keyed –Hashing for message Authentication, RFC 2104, Feb.1997

[29] B. Schneier, "Applied Cryptography", Second Edition, John Wiley & Sons Inc., 1996.