World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:9, No:9, 2015

Bit Model Based Key Management Scheme for Secure Group Communication

R. Varalakshmi

***Abstract*—**For the last decade, researchers have started to focus their interest on Multicast Group Key Management Framework. The central research challenge is secure and efficient group key distribution. The present paper is based on the Bit model based Secure Multicast Group key distribution scheme using the most popular absolute encoder output type code named Gray Code. The focus is of two folds. The first fold deals with the reduction of computation complexity which is achieved in our scheme by performing fewer multiplication operations during the key updating process. To optimize the number of multiplication operations, an O(1) time algorithm to multiply two N-bit binary numbers which could be used in an N x N bit-model of reconfigurable mesh is used in this proposed work. The second fold aims at reducing the amount of information stored in the Group Center and group members while performing the update operation in the key content. Comparative analysis to illustrate the performance of various key distribution schemes is shown in this paper and it has been observed that this proposed algorithm reduces the computation and storage complexity significantly. Our proposed algorithm is suitable for high performance computing environment.

***Keywords*—**Multicast Group key distribution, Bit model, Integer Multiplications, reconfigurable mesh, optimal algorithm, Gray Code, Computation Complexity, Storage Complexity.

I. Introduction

MULTICAST communication is an efficient way to deliver data from a sender to a group of authenticated users.

Multicasting enables an application to scale to a large number of users without overloading the network and server resources. Due to the dynamic nature, at any time users may 'leave' or 'join' the group. Multicast schemes are used to build an efficient multicast tree dynamically upon each user leaving or joining (by means of pruning or grafting). Some of the multicast applications include stock quote services, video-conferencing, pay-per-view TV, and Internet radio and so on. These multicast applications require security in data transmission, i.e., data can only be exchanged among an exclusive group of authenticated users. As multicast communication runs closer towards widespread deployment; security issues have become a central concern and are increasingly important.

Many applications like pay-per–view, distribution of digital media etc., require secure multicast services along with parallel and distributed systems in order to restrict group membership and enforce accountability of group members. A major issue associated with the deployment of secure multicast delivery services is the scalability of the key distribution scheme. This is particularly true with regard to the handling of group membership changes, such as membership departures and/or expulsions, which necessitate the distribution of a new session key to all the remaining group members.

As the frequency of group membership change increases, it becomes necessary to reduce the cost of key distribution operations. One solution is to let all authorized members use a shared key to encrypt the multicast data. To provide backward and forward confidentiality [1], this shared key has to be updated on every membership change and redistributed to all authorized members securely which is referred to as rekeying. The efficiency of rekeying is an important issue in secure multicast as this is the most frequently performed activity with dynamic change in the membership.

Group key must be updated with the group membership changes to prevent a new member from deciphering messages exchanged before it join the group; this is de-fined as backward secrecy [2]. Group key revocation in case of one member joins or multiple members join could be achieved by sending the new group key to the old group members encrypted with the old group key. Also, group key must be must be updated with the group membership changes to prevent an old member (departed or expelled) from deciphering current and future communication which is defined as forward secrecy [2]. Group key revocation, when one member departs or multiple members depart, is more complicated in case of join because of the disclosure of the old group key. The old group key is known to the leaving member(s) so there is a need to re-key the group using valid key(s) in a scalable way. The trivial scheme for rekeying a group of n members is through using individual secret key shared between the Key distribution Centre KDC and each member. This is not a simple or scalable method and consumed large bandwidth especially for large group with high member-ship changes: furthermore, it takes more time and needs more resources per hosts than using multicasting to re-key the group.

In this paper, we study the problem of key Distribution for multimedia multicast services which can be performed in O(1) on an N x N bit model of reconfigurable mesh. We begin in Section II by recollecting the concepts of key distribution using Huddle Hierarchy based Key Management scheme using gray code. In Section III, we introduce a Bit model based key distribution scheme for multicast key Distribution by employing the radar transform to minimize the computation overhead, storage complexity and the number of rekeying

R. Varalakshmi working as an Assistant Professor in Madras Christian College, Chennai (e-mail: rvaralakshmi697@gmail.com).

operations. Section IV analyses the performance of the proposed scheme using the multiplication algorithm, followed by the conclusion in Section V.

II. Multicast Key Distribution

During the design of a multicast application, there are several issues that should be kept in consideration when choosing a key distribution scheme. We now provide an overview of some of these issues.

- Dynamic nature of group membership: It is important to efficiently handle members joining and leaving as this necessitates changes in the session key and possibly any intermediate keying information.
- Ability to prevent member collusion: No subset of the members should be able to collude and acquire future session keys or other member's key encrypting keys.
- Scalability of the key distribution scheme: In many applications the size of the group may be very large and possibly on the order of several million users. The required communication, storage, and computational resources should not become a hindrance to providing the service as the group size increases.

The problem of designing efficient key updating schemes [1]–[11] has seen recent attention in the literature. One approach for achieving scalability is to apply hierarchical subgroups and map the KEKs to a logical hierarchy. The hierarchy-based approach to group rekeying was originally presented by [3]. Due to the hierarchy structure, the communication overhead is O(log n), while the storage for the center is O(n) and for the receiver is O(log n). We note that the O notation is presented to indicate that the constant factors are implementation dependent.

In most of the existing Key Distribution 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) [12] method instead of using encryption algorithms 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 [13] 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 lr + (n − 1)m where lr 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 lr = m = l, computation complexity is nl. The storage complexity is given by $[\log_2 L]$ + t bits for each member. L is number of levels of the Key tree. Hence Group Center has to store n ($[\log_2 L]$ + t) bits.

Secure communication using the extended Euclidean algorithm [14] 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 departs the multicast operation. This approach is only suitable for a star based key Distribution scheme.

The Data Embedding Scheme proposed in [15] 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 [16] based key Distribution scheme was proposed in [17] to reduce computation and storage complexity. A Key Distribution scheme using key graphs has been proposed by Wong Gouda [18] which consists of the creation of secure group and basic key Distribution 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 [19] 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, from the departs 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 of both time complexity as well as storage complexity.

Wade Trappe and Jie Song proposed a Parametric One Way Function (POWF) [20] based binary tree Key Distribution. 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 $(\tau^{L+1}-1)/(\tau-1)$ 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 $(\tau-1) \log_\tau n(\log_\tau n+1)/2$. This complexity can be reduced substantially if the number of multiplications is reduced. In the proposed Huddle Hierarchy scheme [21], users are grouped as huddle and they are assigned with unique IDs based on hierarchical representation. This scheme uses the existing extended Euclidean algorithm for finding the Greatest Common Divisor (GCD) of two positive integers using their prime powers. In order to provide computation efficient secure multicast communication, the following scenario is addressed: private communications are to be established within a restricted group. There is a Group Center in charge of key management. There is also a set of members which may communicate among them and/or with the Group Center. The focus deals with the reduction of computation complexity during the key generation process by reducing the number of multiplication operations by utilizing the Fast Fourier Transform (FFT) algorithm. To reduce the computation complexity further the bit model based key management scheme has been proposed. From this, it is clear that the general interest of research in this area is to reduce the complexity of key management. In this work, we made an attempt to improve the complexity over the existing algorithm. We identified some techniques to reduce the computation as well as storage complexity. The technique used in this paper is a Bit model based Key Distribution Scheme using Gray Code [21] that reduces computation time as well as storage by reducing the number of multiplications required in the existing approaches, and also refreshes the session key for secure communication. We also use the O(1) time algorithm [22], [23] to multiply two N-bit binary numbers which could be used in an N x N bit-model of reconfigurable mesh which yields better results than the previous scheme to optimize the multiplication operations used in the key distribution scheme 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 high performance computing environment.

III. Bit Model Based Key Management Scheme

In order to optimize the number of multiplication operations used in HHKM scheme, Bit Model based Key Management (BMKM) scheme has been proposed. This scheme uses an N x N reconfigurable mesh to perform the multiplication of two N-bit integers. The Cyclic convolution of two integer sequences of length 2N3/4 where each integer is in $[0, 2^{N^{1/4}} - 1]$ can be performed in O(1) time. This algorithm works with single processor system. The comparative analysis to illustrate the performance improvement of various key distribution schemes is shown in the upcoming section.

Let $L=\prod_{i=1}^{m} k_i$ be a multiplication function which is used for dynamic member operation, where ki = secret is the key of a user. Now, 'σi' is the size of the ki, where i = 1, 2, 3, . . . , n (n = size of the group).

Algorithm: Given two N bit binary numbers,

$$X = x_0 + x_1 2 + x_2 2^2 + \ldots + x_{N-1} 2^{N-1},$$
$$Y = y_0 + y_1 2 + y_2 2^2 + \ldots + y_{N-1} 2^{N-1}, \quad (1)$$

The multiplication problem is to compute the 2N-bit product of X and Y. This can be performed by computing

$$\sum_{i=0}^{N-1} X * y_i 2^i. \quad (2)$$

Lemma: Cyclic convolution of two integer sequences of length 2N3/4 where each integer is in $[0, 2^{N^{1/4}} - 1]$ can be performed in O (1) time.

Proof. Let Aj, Bj, $0 \le j \le 2N^{3/4} - 1$ be the two sequences to be convolved. Initially, Aj, Bj are in $(0, jN^{1/4} + w)$, $0 \le w \le N^{1/4} - 1$, $0 \le j \le N^{3/4} - 1$. By appending the $2N^{3/4}$ input points with 0s for performing four-dimensional convolution, this conceptual four-dimensional data array of size $2N^{3/16}$ X $2N^{3/16}$ X $2N^{3/16}$ X $2N^{3/16}$. These 4D data arrays generated from Aj and Bj are stored in lexicographic order in the top row of the 16N X 16N mesh. Now the resulting numbers in the top row are permuted in such a way that the numbers adjacent to each other along the second dimension of the 4D data array are adjacent in the top row of the 16N X 16N mesh. 16N bits are permuted and this can be performed in O (1) time on a 16N X 16N mesh. Using the above idea for each of the four dimensions, the convolution can be completed in O (1) time.

Theorem: Multiplication of two N bit numbers given in a row can be performed in O (1) time.

Proof: Multiplication of two numbers, $a_{N-1}, a_{N-2} \ldots a_0, b_{N-1}, b_{N-2} \ldots b_0$ can be represented by $C(2^{N^{1/4}})$ where

$$A(X) = \sum_{i=0}^{2N^{\frac{3}{4}}-1} A_i X^i$$
$$A_i = a_{(i+1)N^{\frac{1}{4}}-1} a_{(i+1)N^{\frac{1}{4}}-2} \ldots a_{iN^{\frac{1}{4}}}, 0 \le i \le N^{\frac{3}{4}} - 1$$
$$A_i = 0, N^{\frac{3}{4}} \le i \le 2N^{\frac{3}{4}} - 1 \quad (3)$$

$$B(X) = \sum_{i=0}^{2N^{\frac{3}{4}}-1} B_i X^i$$
$$B_i = b_{(i+1)N^{\frac{1}{4}}-1} b_{(i+1)N^{\frac{1}{4}}-2} \ldots b_{iN^{\frac{1}{4}}}, 0 \le i \le N^{\frac{3}{4}} - 1$$
$$B_i = 0, N^{\frac{3}{4}} \le i \le 2N^{\frac{3}{4}} - 1 \quad (4)$$

$$C(X) = \sum_{i=0}^{2N^{\frac{3}{4}}-1} C_i X^i$$
$$C_i = \sum_{k=0}^{i} A_k B_{i-k}, 0 \le i \le 2N^{\frac{3}{4}} - 1 \quad (5)$$

Clearly, $A_i, B_i \in [0, 2^{N^{1/4}} - 1]$and $C_i \in [0, 2^{3N^{1/4}-1} - 1]$ for all N such that N1/4 – 1 ≥ log N3/4. If compute Ci, $0 \le i \le 2N3/4 - 1$ in O(1) time, then the result is obtained by computing $C(2^{N^{1/4}}).C_i, 0 \le i \le 2N^{3/4} - 1$ by cyclic convolution of Ai and Bi. From Lemma, this can be performed in O (1) time. Now $C_i, 0 \le i \le 2N^{3/4} - 1$.

The desired output is $\sum_{i=0}^{2N^{\frac{3}{4}}-1} C_i 2^{iN^{\frac{1}{4}}}$.

Let $N' = \lfloor N^{3/4}/3 \rfloor$, then,

$$\sum_{i=0}^{2N^{\frac{3}{4}}-1} C_i 2^{iN^{\frac{1}{4}}} = \sum_{l=0}^{N'-1} C_{3l} 2^{3lN^{\frac{1}{4}}} + \sum_{l=0}^{N'-1} C_{3l+1} 2^{3lN^{\frac{1}{4}}+1} + \sum_{l=0}^{N'-1} C_{3l+2} 2^{3lN^{\frac{1}{4}}+2} \quad (6)$$

Note that $\sum_{l=0}^{N'-1} C_{3l} 2^{3lN^{1/4}}$ can be obtained by concatenation of C3l, $0 \leq l \leq N'-1$. The desired output can be obtained by adding the three 2N-bit numbers obtained by concatenation of C3l, C3l+1, C3l+2. This addition can be performed in O (1) time.

IV. Simulation Results

As the proposed Bit Model based Key Management (BMKM) approach further reduces the computation complexity of proposed huddle hierarchy, both these approaches are taken into consideration for comparison and Figs. 1 and 2 illustrate the simulated results. It is observed from Fig. 1 that, the proposed BMKM shows further reduction of GC key computation time in comparison with proposed HHKM. Since the key distribution algorithm of BMKM is similar to HHKM, user key computation time remains the same for both the schemes and the simulated result is shown in Fig. 2.

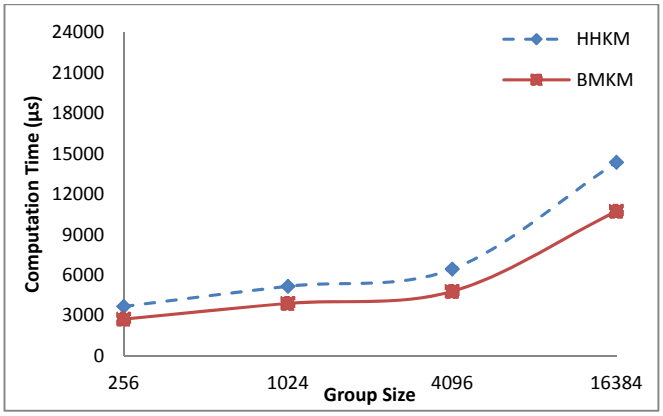

Fig. 1 GC Key Computation Time for BMKM vs HHKM schemes

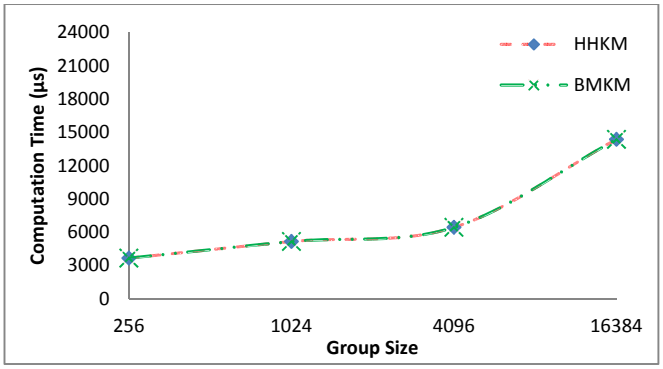

Fig. 2 User Key Computation Time for BMKM vs HHKM Schemes

Table I summarizes the complexity of proposed Huddle Hierarchy based Key Management and Bit Model based Key Management with the existing key distribution schemes such as Binary, and SKDC.

TABLE I
Comparison of Various Key Management Approaches

Parameters	Binary	SKDC	HHKM	BMKM
Computation complexity (GC)	$\tau \times \log_\tau n - 1$	h_τ (τ E+R)	$\tau + 2(\log_\tau n - 2) + \log_\tau n$	$\log_\tau n(1 + \log_\tau n)$
Computation complexity (user)	$(h_2 - 1)$ (M+ Ap)	h_τ (D)	$(\log_\tau n - 1)$(M+A)	$(\log_\tau n - 1)$(M+A)
Storage complexity (user)	$h_2 + 2$	h_τ	$\log_\tau n+1$	$\log_\tau n+1$
Storage complexity (GC)	$\frac{(\tau^{h_2+1} - 1)}{\tau - 1}$	$\frac{(\tau n - 1)}{(\tau - 1)}$	(cn × c) + tc + 1	(cn × c) + tc + 1
Communication complexity	$(\log_\tau n - 1)$ [(3 x p) + 1] + (2p + 1)	$(\tau h_\tau \times K)$	$(\log_\tau n - 1)$ [(3 x p) - 1] + 2p	$(\log_\tau n - 1)$ [(3 x p) - 1] + 2p

E, D, ED, R, M, Ap denotes the computation costs of encryption, decryption, erasure decoding, random key generation, modulo division appending and addition respectively. The notations used for comparison are defined as: n is the number of members, k is size of the key used in various algorithms, τ is the maximum number of users of each node of the tree, p is the size of the prime number used to define the multiplicative group. The length of a path of the key tree of Binary, SKDC and HHKM is h, in particular, $\log_\tau n$ for balanced key trees.

From the analysis of Table I, it is observed that the proposed HHKM scheme takes less computation complexity since it involves simple addition and modulo division operations in the user side. Using Fast Fourier Transform multiplication during key updation operation, the computation complexity reduction is found to be $\tau + 2(\log_\tau n - 2) + \log_\tau n$ in GC. The storage complexity is (cn*c)+tc+1 and the Communication complexity is found to be $(\log_\tau n - 1)$ [(3 x p) - 1] + 2p. Further reduction in computation complexity of $\log_\tau n(1 + \log_\tau n)$ is achieved in BMKM scheme of key generation process by using the Cyclic convolution of multiplication of two integer sequences of length $2^{N^{3/4}}$ where each integer is in $[0, 2^{N^{\frac{1}{4}}} - 1]$ .Since the key distribution algorithm of BMKM is similar to HHKM, the Storage and Communication Complexity remains the same as that of HHKM.

V. Conclusion

The proposed scheme has been compared with the existing HHKM scheme. To optimize the number of multiplication operations used in HHKM scheme, a bit model approach has been proposed. The proposed Bit Model based Key Management (BMKM) scheme shows that an N x N reconfigurable mesh can be used to perform the multiplication of two N-bit integers. Further reduction in computation complexity is achieved in BMKM scheme of key generation process by using the Cyclic convolution of multiplication of two integer sequences of length $2^{N^{3/4}}$ where each integer is in $[0, 2^{N^{\frac{1}{4}}} - 1]$ can be performed in O(1) time.

References

[1] Wallner, D. M., Harder, E. J., & Agee, R. C. (1997). "Key management for multicast: issues and architectures. Informational RFC, draft-Wallnerkey-arch-ootxt, July 1997.

[2] Chang, I., Engel, R., Kandlur, D., Pendarakis, D., & Daha, D. (1999). "Key management for secure internet multicast using Boolean function minimization technique". In ACM SIGCOMM' 99, March 1999.

[3] Varalakshmi, R., & Uthariaraj, V. R. (2011). "A new secure multicast group key management using gray code",IEEE-Xplore
[4] Li, M., Poovendran, R., & McGrew, D. A. (2004). "Minimizing center key storage in hybrid one-way function based group key management with communication constraints". Information Processing Letters, 93, 191–198.
[5] Poovendran, R., & Baras, J. S. (2001). "An information-theoretic approach for design and analysis of rooted-tree-based multicast key management schemes". IEEE Transactions on Information Theory, 47, 2824–2834.
[6] Kulkarni, S. S., & Bruhadeshwar, B. (2010). "Key-update distribution in secure group communication". Computer Communications, 33(6), 689–705.
[7] Bruhadeshwar, B., Kothapalli, K., & Deepya, M. S. (2009). "Reducing the cost of session key establishment." In ARES (2009),pp. 369–373.
[8] Bruhadeshwar, B., Kothapalli, K., Poornima, M., & Divya, M. (2009). "Routing protocol security using symmetric key based Techniques". In ARES (2009), pp. 193–200.
[9] Wang, S.-J., Tsai, Y.-R., Shen, C.-C., & Chen, P.-Y. (2010). "Hierarchical key derivation scheme for group-oriented communication systems". International Journal of Information Technology, Communications and Convergence, 1(1), 66–76.
[10] Imani, M., Taheri, M., & Naderi, M. (2010). "Security enhanced routing protocol for ad hoc networks". Journal of Convergence, 1(1), 43–48.
[11] Wong, C., Gouda, M., & Lam, S. (2002). "Secure group communications using key graphs". IEEE/ACM Transactions on Networking, 8, 16–30.
[12] Blaum, M., Bruck, J., & Vardy, A. (1996). "MDS array codes with independent parity symbols". IEEE Transactions on Information Theory, 42(2), 529–542.
[13] Trappe, W., & Lawrence, C. (2007). "Introduction to cryptography with coding theory" (2nd ed., pp. 66–70).Washington: Pearson Education.
[14] Lihao, Xu, & Huang, Cheng. (2008). "Computation-efficient multicast key distribution". IEEE Transactions on Parallel and Distributed Systems, 19(5), 1–10.
[15] Naranjo, J. A. M., Lopez-Ramos, J. A., & Casado, L. G. (2010). "Applications of the extended Euclidean algorithm to privacy and secure communications". In Proceedings of the 10th international conference on computational and mathematical methods in science and engineering, CMMSE, 703–713.
[16] Trappe, W., Song, J., Radha Poovendran, K. J., & Liu, R. (2001) "Key distribution for secure multimedia multicasts via data Embedding". IEEE International Conference on Acoustics, Speech, and Signal Processing, 3, 1449–1452.
[17] Lee, J. S., Son, J. H., Park, Y. H., & Seo, S. W. (2008). "Optimal level-homogeneous tree structure for logical key hierarchy". In: Proceedings of IEEE conference on communication system software and middleware workshop, COMSWARE.
[18] Je, D.-H., Lee, J.-S., Park, Y., & Seo, S.-W. (2010). "Computation and-storage-efficient key tree management protocol for secure multicast communications". Computer Communications, 33(6), 136–148.
[19] McGrew, A. D., & Sherman, A. T. (2003). "Key establishment in large dynamic groups using one-way function trees". IEEE Transactions on Software Engineering, 29(5), 444–458.
[20] Trappe, W., Song, J., Poovendran, R., & Liu, K. J. R. (2003). "Key management and distribution for secure multimedia multicast". IEEE Transactions on Multimedia, 5(4), 544–557.
[21] R. Varalakshmi, Dr.V.Rhymend Uthariaraj,"Huddle hierarchy based group key management protocol using gray code", Wireless Networks (2014) 20:695–704.
[22] Ng, W. H. D., Howarth, M., Sun, Z., & Cruickshank, H. (2007). "Dynamic balanced key tree management for secure multicast Communications". IEEE Transactions on Computers, 56(5), 590–605.
[23] Denis, T. S. (2003). "BigNum math implementing cryptographic multiple precision arithmetic". Rockland, MA: SYNGRESS Publishing. Pp. 91-128