An Effective Key Computation Protocol for Secure Group Communication in Heterogeneous Networks

[1]S.JABEENBEGUM [2]Dr.T.PURUSOTHAMAN [3]KARTHI.M [4]BALACHANDAR.N [5]ARUNKUMAR.N

[1] Professor & HOD/ CSE, Velalar College of Engineering and Technology, Erode-12, Tamil Nadu, India.
[2]Assistant Professor/CSE, Government College of Technology, Coimbatore – 13, Tamil Nadu, India.
[3, 4, 5] UG Students (CSE), Velalar College of Engineering and Technology, Erode-12, Tamil Nadu, India.

ABSTRACT

In a Heterogeneous Environment to communicate among the group members securely and to minimize the complexity in forming the group key, we are proposing a new protocol which consumes minimum number of key computations. Secure group communication in dynamic and large group is more complex than securing one-to-one communication due to the inherent scalability issue of group key management. In particular, cost for key establishment and key renewing is usually relevant to the group size and subsequently becomes a performance bottleneck in achieving scalability. To address this problem, we are proposing a new approach that features decoupling of group size and computation cost for group key management. By using a Cluster based Hierarchical Key Distribution Protocol, the load of key management can be shared by a cluster of dummy nodes without revealing group messages to them. The proposed scheme provides better scalability because the Key Computation cost is $\log_8 \lceil n \mid m \rceil$. Specifically, our scheme incurs constant computation overhead for key renewing. Our proposed protocol provides the best performance in key computations than existing approaches and it is applicable for wired / wireless environments also.

KEYWORDS

Group Key Management, Secure Group Communication, Elliptic Curve Cryptography, Clustering, Complexity Analysis.

1. Introduction

Group key management is an important functional building block for any secure multicast architecture. The phenomenal growth of the Internet in the last few years and the increase of bandwidth in today's networks have provided both inspiration and motivation for the development of new services, combining voice, video and text over Internet Protocol. Although unicast communications have been predominant
so far, the demand for multicast communications is increasing both from the Internet Service Providers and from content or media providers and distributors. Indeed, multicasting is increasingly used as an efficient communication mechanism for group-oriented applications in the Internet such as video conferencing, interactive group games, video on demand, TV over Internet, e-learning, software updates, database replication and broadcasting stock quotes. Key management is the base for providing common security services (data secrecy, authentication and integrity) for group communication. The main goal of this paper is to demonstrate how provably our proposed secure group key management protocol can be combined with reliable group communication services to obtain provably efficient communication and computation costs [7].

Manuscript received February 5, 2010
Manuscript revised February 20, 2010

2. Group Key Management Protocol

In order to establish a group communication a common group key is to be distributed to all the member of the group. The group is to be changed when a member leaves or joins in the group (to ensure forward and backward secrecy) and also the key is to be refreshed periodically so that it would not be possible for the hackers to find the group key [2]. Figure.1 shows Taxonomy of Group Key Management Protocols. Below we are going to discuss about the three major approaches to group key management.

2.1 Centralized Group Key Management Architecture

In this type of key management a single entity is employed for controlling the whole group. Hence a centralized key management seeks to minimize storage requirements, computational power on both client and server sides, and bandwidth utilization. But the major problem of single point of failure exists. The Protocols used in Centralized Group Key Management are LKH, OFT, Canetti et al, ELK, CFKM, Keystone, GKMP, Wong et al. etc...

2.2 Distributed Key Management Architecture

In distributed key management there is no explicit KDC, and the members themselves do the generation. The members need not depend on third party. All members can perform access control and the generation of key is

contributory, meaning that all members contribute some information to generate the group key. So the security level has been raised but this method is suitable for a small group only. For large groups collecting the contribution from every user s tedious and time consuming, due to this reason scalability criterion is not fulfilled. The Protocols used in Distributed Key Management are CKA, Octopus, STR, DH−LKH, D−LKH, D−OFT, D−CFKM etc….

2.3 Decentralized Architecture

In a decentralized architecture the management of a large group is divided among subgroup managers, trying to minimize the problem of concentrating the work in a single place. The Protocols used in Decentralized Key Management are SMKD, IGKMP and Hydra etc...[1], [3].

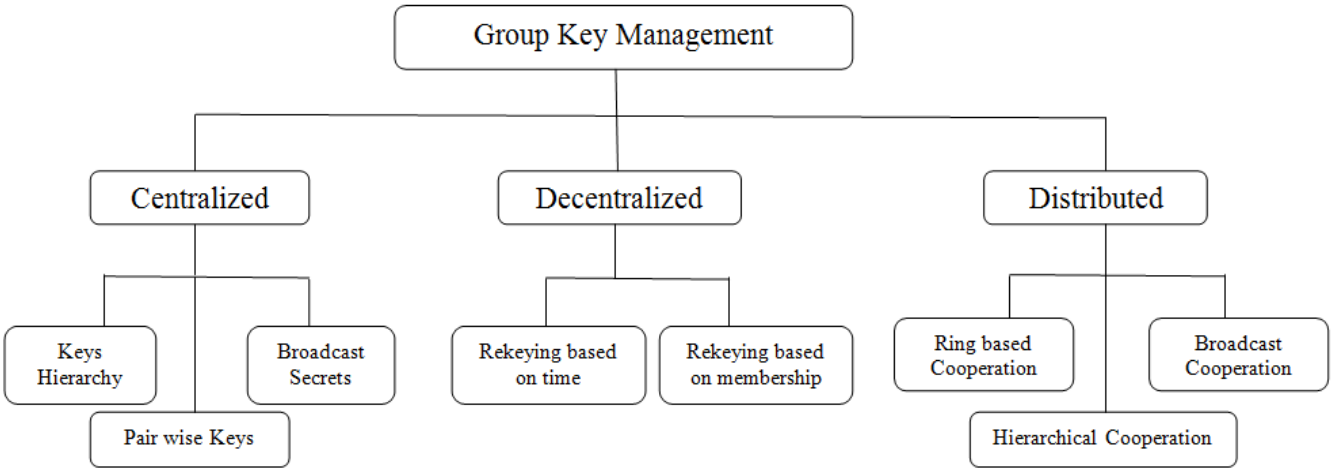

Figure.1 Taxonomy of Group Key Management Protocols

3. Clustering

Clustering is nothing but grouping the items or nodes which are having similar properties but here we have to cluster the members based on the properties like key, position, time etc…Clusters are usually deployed to improve performance and/or availability over that of a single user, while typically being much more cost-effective than multiple users of comparable speed or availability.

3.1 Why Clustering?

In order to overcome the difficulties faced by an individual member, we are using the efficient clustering technique. In this technique we are maintaining a Cluster Controller; It controls its own member by means of sending / receiving the messages and join / leave operations.

3.2 Cluster Model:

The following Figure.2 illustrates the typical cluster model:
Here the leaf node indicates the group member and non-leaf node indicates the dummy nodes. If the members are having the same parent then such type of members are grouped together which results cluster formation. The capacity of cluster is controlled by the cluster limit. It indicates how many nodes that the cluster has. i.e., each parent node can have maximum of 8 Children (We can take cluster limit based on the application).

3.3 CLUSTER TYPES:

Clustering is classified into three types based on their parameters as follows:

1) Key based clustering
2) Position based clustering
3) Time based clustering

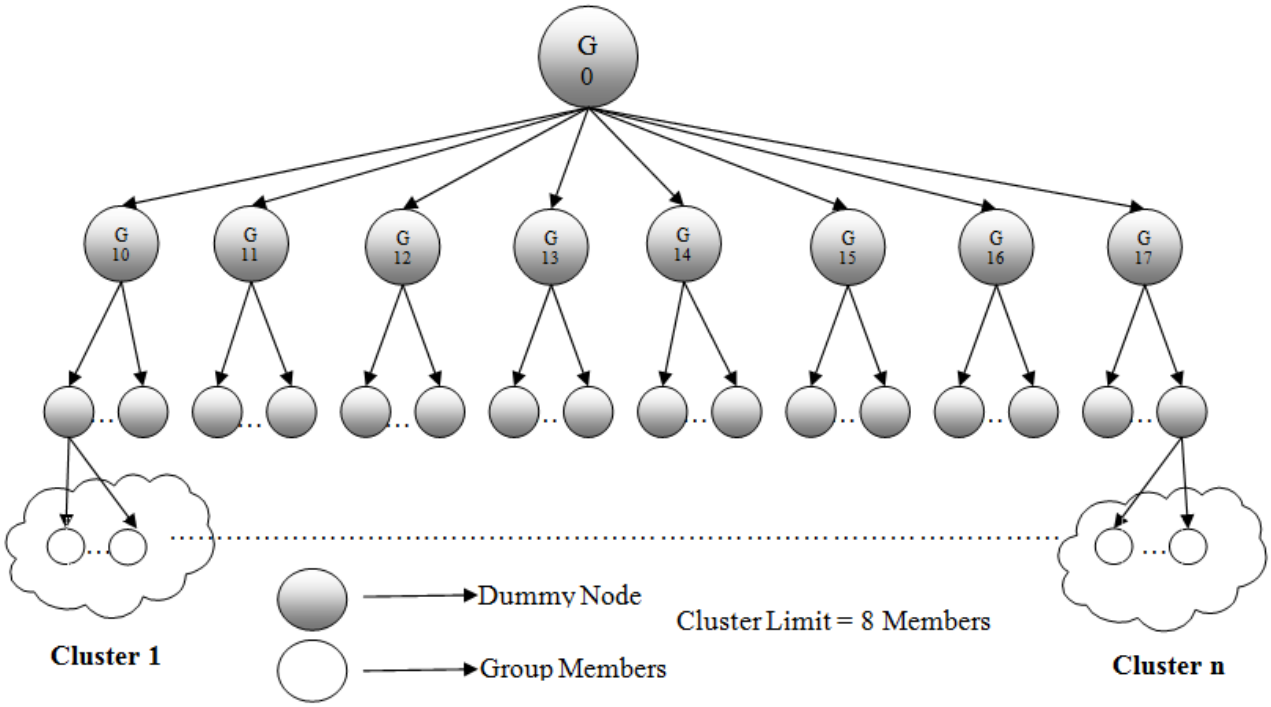

Figure.2 Typical Cluster model

Key Based Clustering:
Key based clustering is done by grouping the members those are having same key.

Position Based Clustering:
We have to determine the area after that we have to calculate the members who are belongs to that particular area. Based on that area we have done clustering and it is said to be as position based clustering. The PBC can be achieved by differentiating position or area like Computer Lab, building that contains number of system, city etc...

Time Based Clustering:
In order to achieve time based clustering, we have to maintain a database that contains joining time and name of the user. Based on the two parameters we have to calculate the number of clusters needed to achieve the cluster generation. For that we have to calculate the time, based on the interval we have to group the members if the member belongs to particular interval.

Table.1 Clustering Type

Clustering Type	Maximum No of Clusters	Total No of Levels Needed	Maximum Users
Binary (2 Node)	1	2	2
	2	3	4
	4	4	8
	8	5	16
	16	6	32
4 Node	1	2	4
	4	3	16
	16	4	64
	64	5	256
	256	6	1024
8 Node	1	2	8
	8	3	64
	64	4	512
	512	5	4096
	4096	6	32768
16 Node	1	2	16
	16	3	256
	256	4	4096
	4096	5	65536
	65536	6	1048576

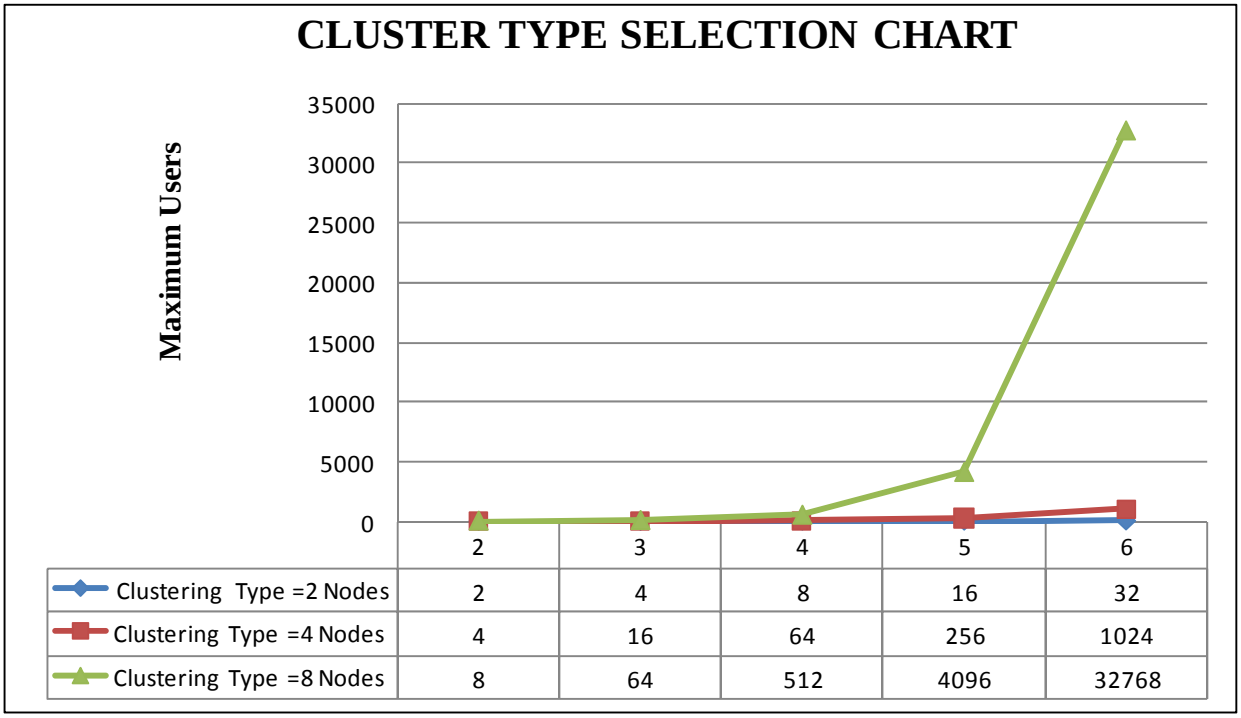

Figure.3 Cluster Type Selection Chart

3.4 Clustering Type Analysis

Table.1 shows Clustering Type Analysis. We can choose Clustering type based on the application. If more number of users are needed, we can choose maximum cluster limit. We had taken cluster limit as 8 here and analyzed. In Cluster Type Selection Chart, We had taken total no of levels needed in X-axis and maximum users in Y-axis.

4. Group Key Formation

In order to communicate within/between the group member we need a key identification so that we are maintaining public key and private key. Private Key is to be chosen by the member and then kept it as secret. By using private key each member calculate their own public key and it is announced by public manner. By using public and private key the members can communicate with each other and also encrypt/decrypt their messages. The necessity of group key is to provide authentication between/within the group members. The structure of Group Controller as follows:

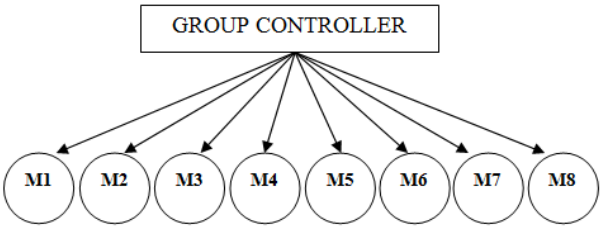

Figure.4 Typical Cluster Model

Here,
Cluster limit = 8 Nodes
M_i= Members where $1 \leq i \leq 8$
Group key formation consists of following steps

1) Identify the number of members present in the group.
2) Group controllers retrieve all the public keys.
3) After getting the public key, the group controller will calculate the group key by using the following expression

$GK=n_{GK} \times (P_1+P_2+P_3+\ldots\ldots P_n)$

It can be written as

$GK= n_G \times \sum_{i=1}^{m} P_i$ Where m is an Integer.

4) After completion of this process the Group Controller will distribute group key to all its group members.

4.1 Group Key Formation Using ECC:

In order to provide an authorization between two friendly parties, we need a Group Key (GK). To generate GK efficiently, we need an impregnable cryptographic method. So we are using ECC technique [4], [5].

Table.2 Relative Strength of Public Key Cryptosystems

Security Bits	Symmetric Encryption Algorithm	DSA/ DH Bits	RSA Bits	ECC Bits
112	3DES	2048	2048	224
128	AES-128	3072	3072	256
192	AES-192	7680	7680	384
256	AES-256	15360	15360	512

Table.2 lists the minimum size (bits) of public-keys using DSA/DH, RSA and ECC which provide equivalent security. We see that the key length ECC scales linearly which is not the case with the key Lengths of RSA/DH. The relative computational performance advantage using ECC versus RSA/DH is indicated by the cube of key sizes: The 3072-bit RSA key requires 27 times the computation required for the 1024-bit RSA key. ECC increases the computational cost by just over 4 times. This will have a significant impact on cryptographic systems, especially in resource poor environments like ad hoc
and sensor networks.

4.2 Single Member Join / Removal

In Figure. 5 The cluster size is 8 and in the given example only 7 members are there and if a single member joins the group then we can add the member and the group key must be updated. The same procedure is followed for member removal. The group key formation is also below.

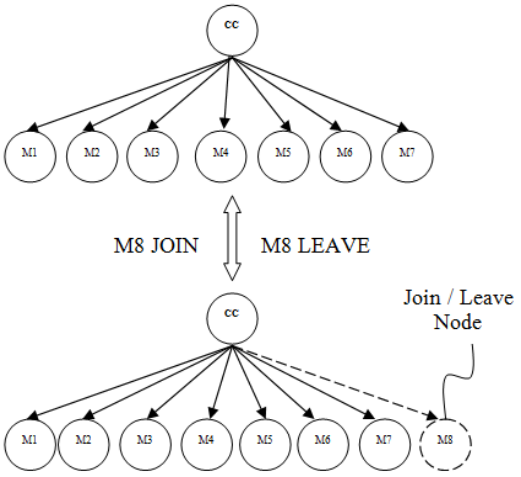

Figure.5 Single Member Join / Leave Operation

Single Member Join:

$$GK = n_G \times \sum_{i=1}^{m} P_i \text{ Where m=7}$$

$$= n_G \times \{ P_1+P_2+P_3+P_4+P_5+P_6+P_7 \}$$

$$GK\_new = n_G \times \sum_{i=1}^{m} P_i \text{ where m=8}$$

$$= n_G \times \{ P_1+P_2+P_3+P_4+P_5+P_6+P_7+ P_8 \}$$

i.e.,

$$\mathbf{GK\_new = GK + \{n_G \times P_8\}}$$

Single Member Removal:

$$GK = n_G \times \sum_{i=1}^{m} P_i \text{ Where m=8}$$

$$= n_G \times \{ P_1+P_2+P_3+P_4+P_5+P_6+P_7+ P_8 \}$$

$$GK\_new = n_G \times \{P_1+P_2+P_3+P_4+P_5+ P_6+P_7+ P_8 \} - \{n_G \times P_8\}$$

$$\mathbf{= GK - \{n_G \times P_8\}}$$

4.3Group Member Join / Removal

In Figure. 6 if the group of members is joining and if the cluster size is 8, then we will add a Cluster Controller and the new group members joins under the new Group Controller. The same procedure is followed for group member removal also. In our protocol we are fixing maximum 8 Cluster Controller under the Group Controller and if it extends according to the position based technique the new members joins under new Sub Cluster Controller the same procedure continues for the updation of the Secure Group Key.

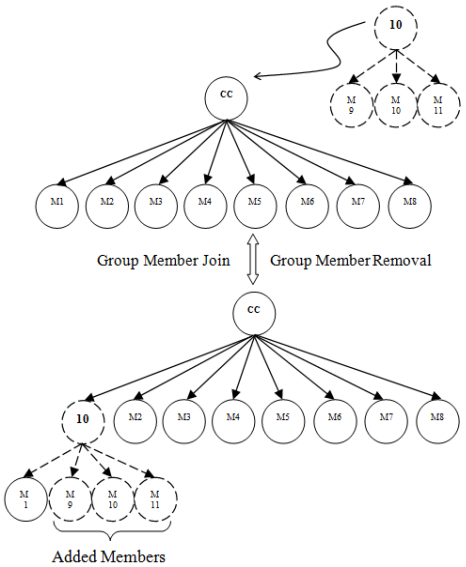

Figure.6 Group Member Join / Leave Operation

Table.3 Computation Cost

Techniques	Join	Leave
Simple Application	1	n
LKH	$2\log_2 n-1$	$2\log_2 n$
OFT	$\log_2 n+1$	$\log_2 n+1$
Key Graph	$\log_4 n+1$	$4\log_4 n-1$
Logical Key Tree	$\log_2 \lceil n \mid m \rceil$	$\log_2 \lceil n \mid m \rceil$
Our Protocol	$\log_8 \lceil n \mid m \rceil+1$	$\log_8 \lceil n \mid m \rceil$

Group Member Join

$GK = n_G \times \sum_{i=1}^{m} P_i$ Where m=8

$=n_G \times \{ P_1+P_2+P_3+P_4+P_5+P_6+P_7+ P_8 \}$

$GK\_new=n_G \times \{P_2+P_3+P_4+P_5+P_6+ P_7+ P_8 \}$

$= n_G \times \sum_{i=2}^{8} P_i$

i.e..,

$GK\_new = GK - \{n_G \times P_1\}$

$GK_{10} = n_{10} \times \{ P_1+P_9+P_{10}+P_{11} \}$

Group Member Removal

$GK\_new= n_G \times \{P_2+P_3+P_4+P_5+P_6+ P_7+ P_8 \}+\{n_G \times P_1\}$

$= \mathbf{n_G} \times \sum_{i=2}^{8} P_i + \{\mathbf{n_G} \times \mathbf{P_1}\}$

GK_{10}= NULL

5. Performance Evaluation

Here, we are analyzing the Computation Cost for effective response of Message Communication.

5.1 Computation Cost

When a member joins the group or leaves the group the new group key must be calculated to maintain the Forward and the Backward secrecy and the key Independence. Here under each Cluster Controller 8 members may join.

According to the new member join/ leave the updation of the group key takes place. In our protocol we are fixing maximum 8 Cluster Controller under the Group Controller and if it extends the new members joins under new Sub Cluster Controller, here m refers the maximum cluster size and n refers the number of members in the group. Our protocol consumes because $\log_8 \lceil n \mid m \rceil+1$ the total number of group members are cluster under a cluster controller with the size of 8 so we get maximum of log (n/m) and a public key is received when a new user joins the group. If the user leaves the group the rekeying cost is only $\log_8 \lceil n \mid m \rceil$. Table 3 shows the comparison of different techniques with our proposed technique and it shows only one message is needed for both join/leave event [6] ,[7].

6. CONCLUSION AND FUTURE WORK

Our protocol shows better results than the existing protocols. The computation cost is much reduced than the existing protocols by introducing the clustering technique. In heterogeneous environment either wired or wireless networks the secure group key formation must take minimum time, storage cost and the message send and received. Here we have concentrated on clustering, computation and we achieve better results using ECC and in future we are going to analyze the time efficiency in forming the key and the number of keys stored in cluster controllers. Here, we have shown the cluster size with 8 members, but we have analyzed for 16, 32, 64 members also.

References

[1] Sandro Rafaeli, David Hutchison, "A survey of Key management for Secure Group Communication", ACM Computing Surveys, Vol.35, No.3, September 2003, pp.309-329.

[2] Shanyu Zheng, David Manz, Jim Alves-Foss, "A Communication Computation Efficient Group Key Algorithm for Large and Dynamic Groups", Elsevier, Computer Networks, March 2006.

[3] Joe Prathap PM, Vasudevan.V, "Analysis of the various key management algorithms and new proposal in the secure multicast communication", IJCSIS, Vol.2, No.1, 2009.

[4] N.Gura, A.Patel, A.Wander, H.Eberle, S.C.Shantz, "Computing Elliptic curve Cryptography and rsa on 8-bit cupus", in: CHES, 2004, pp.119-132.

[5] S.A.Vanstone, "Next Generation Security for Wireless: Elliptic Curve Cryptography", Computing and Security 22 (2003) 412-415

[6] Alireza Nemaney Pour, Kazuya Kumekawa, Toshihiko Kato, Shuichi Itoh, "A Hierarchical group key management scheme for secure multicast increasing efficiency of key distribution in leave operation", Elsevier, Computer Networks, August 2007.

[7] Chung Kei Wong, Mohamed Gouda, and Simon S. Lam, "Secure Group Communications Using Key Graphs", IEEE/ACM Transactions on Networking, Vol. 8, NO. 1, February 2000.

Prof.JabeenBegum.S received her M.E in Computer Science from Government College of Technology and is working towards Ph.D. in Computer Science from the Anna University, Coimbatore. Currently she is working as a HOD & Professor in CSE Dept, Velalar College of Engineering and Technology, Erode, Tamil Nadu, India and she is having 18 years of Experience in Teaching. She had published 21 Papers in Various National Conferences and she had presented 5 Papers in Various International Conferences held at many Engineering Colleges. Her Research Paper regarding "Time Complexity in Key Management" has been published in AMSE, France and she had published her research paper in IEEE Explorer regarding Secure Group Communication. Her interests include Network Security, Distributed Systems and DBMS.

Dr.T.Purusothaman is currently working as Assistant Professor (RD) in the department of Computer Science and Engineering and Information technology, Government College of Technology, Coimbatore. He has twenty years of teaching experience. He has completed Ph.D in the area of Network Security and Grid Computing. In his thesis, a novel key management scheme was proposed to provide service only for the paid customers in Internet. He has successfully completed a project funded by DIT (Government of India) in the area of cryptanalysis in the year 2006. He has presented a number of papers in various National and International conferences. Many of his papers were published in IEEE Explore. He has to his credit several International Journal Publications in reputed journals including Journal of Grid Computing, Springer. His research interests include Network Security, Grid Computing and Data Mining.

Karthi.M, Pursuing his Final Year B.E (Computer Science and Engineering) in Velalar College of Engineering and Technology, Erode, Tamil Nadu, India. He had presented 20 Papers in Various National Level Technical Symposium held at Engineering Colleges in Tamil Nadu and got many Prizes. He had published 8 Papers in Various National Conferences held at many Engineering Colleges. His Research Paper regarding "Smart Card" has been published in International Journal of Computer Science and Network Security (IJCSNS), South Korea. He got certified from **Sun Microsystems** on Sun Certified Programmer for the Java Platform, Standard Edition 5.0 with 91%. He is Interested in Programming, Cryptographic Algorithms for Cryptography, Computer and Network Security.

Balachandar.N, Pursuing his Final Year Bachelor of Computer Science and Engineering in Velalar College of Engineering and Technology, Erode, Tamil Nadu, India. He had presented 20 Papers in Various National Level Technical Symposium held at Engineering Colleges in Tamil Nadu and got many Prizes. He had published 8 Papers in Various National Conferences held at many Engineering Colleges. His Research Paper regarding "Smart Card" has been published in International Journal of Computer Science and Network Security (IJCSNS), South Korea. He got certified from Microsoft as **Microsoft Certified Professional** (MCP) on Managing and Maintaining a Microsoft Windows Server 2003 Environment and Installing, Configuring, and Administering Microsoft Windows XP Professional with 94% and 100% respectively. His areas of Interests are System Administration, Cryptography and Network Security.

Arunkumar.N, Pursuing his Final Year B.E (Computer Science and Engineering) in Velalar College of Engineering and Technology, Erode, Tamil Nadu, India. He had participated Various National Level Technical Symposium held at Engineering Colleges in Tamil Nadu and got many Prizes. He had published 2 Papers in Various National Conferences held at many Engineering Colleges. He had attended a Workshop on Image Processing. He is Interested in Computer and Network Security.