IACSIT International Journal of Engineering and Technology, Vol.2, No.4, August 2010
ISSN: 1793-8236

A Novel Hybrid Cryptosystem Based Approach for Secure Multicasting for Dynamic Groups

D.V. Naga Raju[1] and Dr.V.Vallikumari[2]

Abstract—**In the recent past the group-oriented applications like conferencing, chat groups and interactive gaming gaining popularity. In theses group based applications myriad messages are sent from one or more sources to multiple users. Multicasting is the optimum technique for such group oriented applications with effective network resource utilization. But maintaining security is a critical issue in group oriented protocols with frequent membership changes. Confidentiality can be achieved through changing the key material, known as rekeying every time a new member joins the group or existing member leaves from the group. Many techniques have been proposed earlier for this purpose. In centralized approach, a single key server is responsible to generate and distribute keys. In decentralized approach, a hierarchy of key managers distributes the keys. In distributed key-agreement protocol, the group members collectively generate and distribute a group key. This paper uses combination of both de-centralized and key-agreement approaches to prevent a single point of failures and to improve the reliability as well as the performance of the system. This paper proposes new a technique (SGKP-1), using hybrid cryptosystem, has certain advantages like secure channel establishment for the distribution of the key material, reducing the storage requirements and burden at each member, minimization of time requirement to become a new member of a group. The computational complexity further reduced using both the combination of public and private key crypto systems.**

Index Terms—**group key, key hierarchy, multicast, Secure Channel.**

I. INTRODUCTION

History is filled with examples of how technology helped usher a new eras of prosperity. Efforts are on streamline technology widening the knowledge canvas. This demands innovation and precision at every juncture. Toeing the identical line of perfection this paper focuses on key distribution using hybrid cryptosystem for reducing the computation overhead and improving the performance of the entire system. In the present competition of cutting edge technology, quality up gradation and improving accessibility are the demanding requisites. Any new idea ensuring viability reserves the glory of practicality in the world of computer science. With the increase of bandwidth along with several latest advancements in the internet technologies encouraging people for the development of new services like secure video conferencing, interactive gaming, stock quotes distribution which are based upon a group communications model where packets need to be delivered to a large community from one or more sources.

Many popular network applications like conferencing, chat groups, interactive video gaming etc., involve multi-sender and multi-receiver patterns. Multicasting is the best suited technology for such applications, where data can be sent to large user community effectively and efficiently via the Internet or in any wireless communication. As the nature of the membership is highly dynamic with frequent addition and eviction, providing security in a scalable manner is the challenging issue for any group communication protocols. The existing solutions of group key management are broadly classified into there major classes. 1) centralized key management protocols, which suffers from a single point of failure as a single entity takes the responsibility of generation and distribution of the keys 2) decentralized key management protocols, hierarchy of key managers share the responsibility of key generation and distribution 3) contributory key agreement protocols, share of each member is taken into consideration for the generation of the group key. This paper proposes a method where there is neither the need of establishing secure channels like LKH[3] for the distribution of the keys nor there is any chance of producing weak keys as in LKH[3], as many group members participate in the key generation. Whenever, a new member joins the group or an existing member leaves the group, the group key must be changed and should be securely re-distributed to the existing group members. The changing of the key material with respect to these events is known as re-keying. Since, re-keying is frequently performed activity, renewing the key material should be highly scalable. Two important aspects while maintaining the secrecy are (a) forward secrecy i.e., the users who left the group should not have any future access to the group and (b) backward secrecy i.e., a new user who joins the group should not have any past access to the group. This paper proposes a technique which eliminates the risk of establishing the secure channels like LKH [3] using public key trees like DLPKH [4]. The main objectionable issue with DLPKH [4] is that it reveals the private key to other members which is totally against the principle of public key cryptography. The proposed technique of this paper will not reveal any private keys. The storage requirements as well as work done by each member as part of re-keying and computational complexity are also been reduced further.

[1]Department of IT, Shri Vishnu Engineering College for Women, Bhimavaram-534204, India
[2]Department of CSSE, College of Engineering (A), Andhra University, Visakhapatnam-530003, India

II. Related Work

Yacine Challal, Abdelmadjid Bouabdallah, Hamida Seba [1], conducted an excellent survey in group key management protocols, organized the existing solutions into three categories as centralized, decentralized and distributed key agreement. In centralized scheme, a single key server participates in symmetric key (Traffic Encryption Key, TEK) generation and distribution which suffers from a single point of failure and generation of weak keys. In decentralized approach, a hierarchy of key managers distributes TEK and there by avoid bottlenecks and single point of failures. In key-agreement protocols, the group members co-operate to establish a group key and as such improving the system's reliability. C. K. Wong, M. Gouda, and S. S. Lam [3] proposed Local Key Hierarchy (LKH) uses the centralized approach where a single entity known as central server maintains the keys. Each node is associated with a KEK (Key Encryption Key) and the members who occupy at leaf node holds secret keys shared with other members of the group. Each member knows all the KEKs corresponding to the nodes in the path from its leaf to the root .The key corresponding to the root of the tree is the TEK. For a balanced binary tree, each member stores at most $1+\log_2$ (n) keys, where n is the number of group members. This key hierarchy allows reducing the number of re-key messages to O (log n) instead of O(n) in GKMP[7,8]. McGrew and Sherman [5] proposed an improvement over LKH, called One-way Function Trees (OFT). OFT allows reducing the number of re-key messages from $2\log_2$ (n) to only log2 (n). Inoue and kuroda [4] have proposed the Fully Distributed Logical public key hierarchy (FDLKH), in which they used the concept of LKH without any central server. Moreover, in FDLKH [9] the members will not have any individual keys unlike LKH. For each sub-tree the task of generating and distributing of key material will be assigned to group members known as captains. The captains use DH key agreement[10]**.**.Rodeh et al [11] proposed a scheme which remove the central server but suffers from the drawback i.e beckward secrecy cannot be maintained for join operation. Rakesh Bobba,Himamshu Khurana[4] proposed DLPKH, Distributed Logical Public Key Hierarchy where they also used the concept of Logical Key Hierarchy. DLPKH uses public key trees for the secure distribution of the updated keys which have the advantage of secure distribution of the keys without establishing any secure channel. In DLPKH, each member knows all the private and public keys of their ancestor nodes and also the public keys of the nodes that are siblings to the set of nodes on the path from the leaf to the root. The responsibility of generation and distribution of keys is given to the sponsors and co-sponsors. The major weakness of this protocol is that the private keys of the nodes will be revealed which is totally against the concept of public key cryptography. This paper discuss a technique, a combination of decentralized and distributed key agreement approaches, shows improvement over DLPKH by reducing the number of keys, eliminating co-path updating and preventing the disclosure of private keys and minimizing the storage requirement as well as computational complexity.

III. Notations

The protocol proposed in this paper arranges entire group members in a logical key hierarchy as LKH[3] ,using the concept of binary trees .A node is represented as M (l,m), where l is the level of the tree from 0,1,2,… and m is the position of the node where $0 <= m <= 2^l-1$.Every node in the binary tree consists of three keys, viz, public key, private key and a common group key (session key). These three keys are identified with respect to (l,m) as PK,SK and GK respectively. The keys will be updated as follows.

New public key = old Public key $\times g^{newGK}$ mod p

New private key =old Private Key + new GK mod p

New Group key = new GK

(l, m)	m^{th} node at level l in a tree
(l, m)^I	updated m^{th} node at level l in a tree
M (l, m)	Member who occupies the node (l, m)
PK (l, m)	Public key associated with the node (l, m)
SK (l, m)	Private key associated with the node (l, m)
GK	Group Key
PK^I	New public Key of the node (l, m)
SK^I	New private Key of the node (l, m)
T (l,m)	Sub tree rooted at the node (l,m)
GL-i(l,m)	Member who represents the T(l,m)
E(PK, X)	Encryption of data X with public key PK
E (SK, X)	Encryption of data X with private key SK
n	Number of members in a group
p,g	EIGmal group parameters

IV. Basic Principles and Assumptions

Each member occupies the leaf node. Each member possesses all the public keys of its ancestors along with its own private key, public key and a group key.

In join, two group leaders will be selected dynamically and in leave, three group leaders (GLs) will be elected dynamically, two from the same sub-tree where the event (leave) is likely to happen and the other from the other sub-tree.

A. JOIN Protocol

When a new member wants to join the group, the join protocol will be executed. The member who wants to join the group broadcast a join request with its public key. The joining point and Leaders are determined by the existing group members who already know the complete tree

structure. The right most shallowest leaf node (from root) of the tree will be taken as the joining point.

In case of join, two group leaders (GLs) will be selected from the same sub tree where the join is going to happen. The joining point and GL are represented by GL-1, which is the right most shallowest leaf node of the tree. If the sibling of the GL-1 is a leaf node then it will be selected as GL-2. If this is not the case then the GL for sub tree rooted at the sibling node will be selected as GL-2.

In the figure -1 the joining point and the GL-1 will be M8.The Gl-2 should be from the same subgroup. So this time it is M5. Two new nodes will be created by the existing group members and will be added to the joining point as their children. The GL-1 associated with the joining point is given to the right child (from root) of the joining point. The new joining member and its public key is given to the left child of the joining point. In the above figure two new nodes M8 and M7 will be created and the GL-1 will be assigned to M8 and the new member will be M7. The rekeying nodes in this case are (1,1) and (0,0).The new tree is shown in fig 2.

GL-1 and GL-2 perform Diffie-Hellman key exchange and get a group key. GL-1 encrypts the GK with the old group key and broadcast it to the group. All the existing members except the newly joined member get the GK and they update their key pair and group key and also the public keys of their ancestor nodes. The non-leaf nodes also update their key pair. Since the newly joined member does not know the old group key, it cannot get the GK. So the GL-1 encrypts the GK with the public key of the newly joined and also the updated public keys of the ancestor nodes of the newly joined member, the tree structure.

GL-1→G: E (old GK, new GK)

GL-1(M8) →M7: E ($PK_{(3,7)}$), $((2, 3)^1, (1, 1)^1, (0, 0)^1, GK)$))

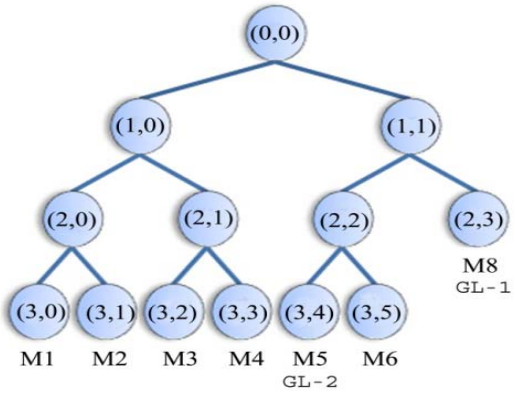

Figure: 1 Tree before join in SGKP-1

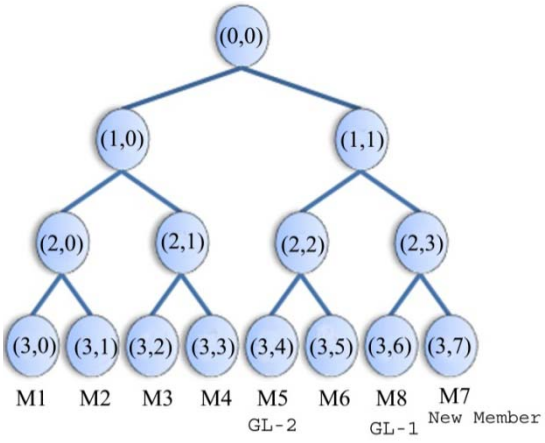

Figure: 2 Tree After join in SGKP-1

B. Leave protocol in SGKP-1

When an existing group member wants to come out of the group leave protocol will be executed. The member who wants to leave from the group sends a leave request to the group. The existing members will decide the GLs. Unlike join protocol, in leave protocol the number of GLs are three. Two are from the same sub tree where the event (leave) is likely to happen and the other from the other sub tree. The GL1 in this case is that for the sub tree rooted at the sibling of the leaving node. GL2 is the GL of the sub tree rooted at the sibling node of leaving node's parent. Members update the tree by removing the leaving node and promoting the sub tree rooted at the sibling node to be rooted at the leaving node's parent node. In order to maintain the forward secrecy the group key and the public keys of the root's sibling to which it belongs (e.g. (1, 0)) need to be changed and the public keys of the ancestor nodes of the leaving member are also changed. GL-1, GL-2 and GL-3 perform DH key exchange and get a secret key, GK.

Now GL-3 encrypts the GK with the old public key of (1, 1) which again broadcast the GK with the old private key and updates its own key pair and also the group key associated with it. All the group members {M6.M7, M8, M9} who already know the old public key of (1, 1) get the GK and update the public key of (1, 1).

Similarly, GL-2 also sends the GK encrypted with the old public key of (2, 1) and broadcasts to the group (2, 1) with its old public key. All the members of this sub tree {M4, M5} get the group key and update the public keys of its ancestor nodes. Finally GL-1sends the GK encrypted with the old public key of (3, 0) and send to (3,0),which will broadcast the GK encrypted with its old private key. So M2 also gets the GK and updates the group key and the public keys of its ancestor nodes.

GL-1→T (3, 0): E (old PK (3, 0), GK)

GL-2→T (2, 1): E (old PK (2, 1), GK)

GL-3→T (1, 1): E (old PK (1, 1), GK)

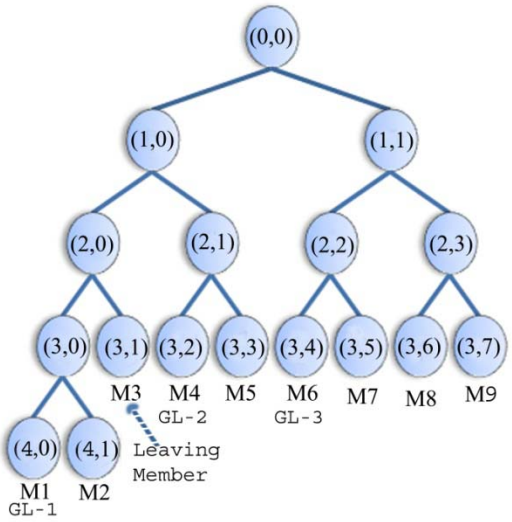

Figure: 3 Tree before Leave in SGKP-1

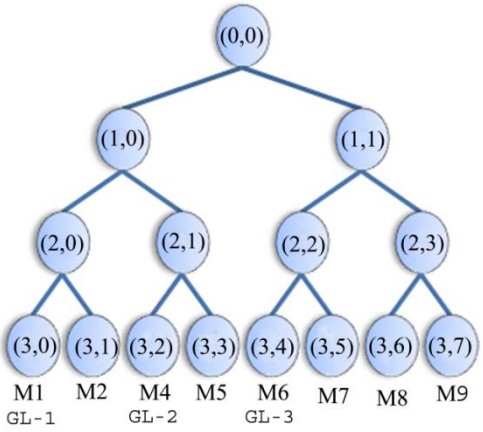

Figure: 4 Tree after leave in SGKP-1

V. ANALYSIS

We compare our technique, with DLPKH. We evaluate based on the number of keys at each member, number of key updates at each member and also key updating cost at Group leaders

A. Number of keys:

The number of keys maintained at each member is less than that of DLPKH member. In SGKP-1, every member is associated with public key, private key of its own, a group key and also the public keys of its ancestor nodes. i.e. a total of l+3, where 'l' is the level of that member. But in DLPKH, each member is associated with the public and private keys of its own, public and private keys of its ancestor nodes and also the public keys of the co-path nodes, a total of 3l+2, where 'l' is the level of that member.

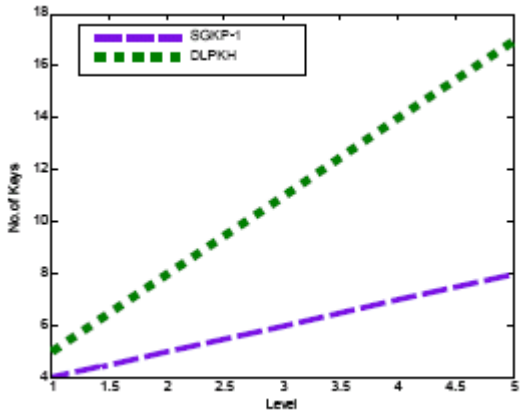

Figure: 5. No. of keys at each member.

B. Key updating cost for GL-1(SGKP-1)/Sponsor (DLPKH)

In SGKP-1, the GL-1 needs to update the public keys of the ancestor nodes of the joining member.

In DLPKH, the sponsor needs to update the public and private key of the ancestor nodes of the joining member and also the public keys of the co-path nodes of the newly joined member.

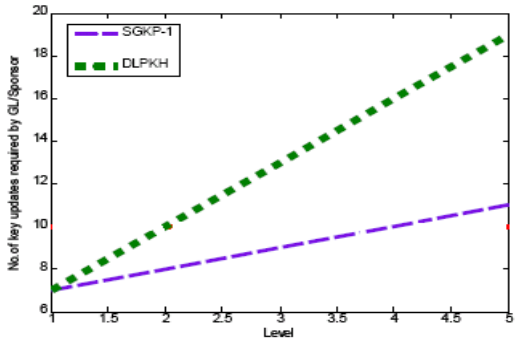

Figure: 6 Key updating costs for GL/Sponsor

C. Key updating cost at each member

In SGKP-1, the member needs to update the public keys of the ancestor nodes along its key path and also its own private and public key.

In DLPKH, each member needs to update the public and private key of the ancestor nodes along its key path, its own private and public keys and also the public keys of the co-path nodes.

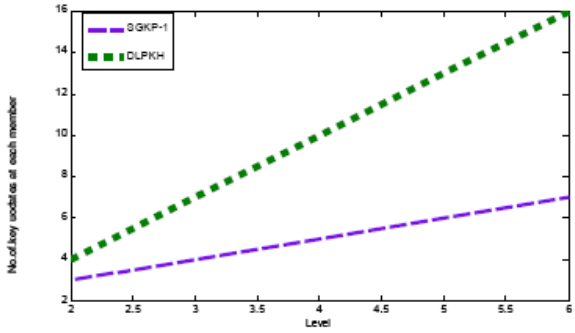

Figure: 7 Key updating costs for each member

D. Effort put by the members for key updating

Decryption in asymmetric cryptosystems takes much time when compared to that of symmetric cryptosystems.

SGKP-1

For group communication i.e if a member wants to send a message to the entire group, it will encrypt the message with the group key which is a session key using symmetric cryptography.

For newly joined message it will send the new group key encrypted with the public key of the new member where it uses asymmetric cryptography.

Conclusion: Burden on all the members except the new member will be less.

DLPKH

For group communication i.e if a member wants to send a message to the entire group, it will encrypt the message with the public key of the root asymmetric cryptography. So all the members need to decrypt using the private key of the root. So burden on all the members will be more.

For newly joined message it will send the new group key encrypted with the public key of the new member where it uses asymmetric cryptography.

Conclusion: Burden on all the members including the newly joined member will be more when compared to SGKP-1.

E. Reveal of private key

SGKP-1: In this scheme the members will know **only the public keys** of their ancestor nodes.

DLPKH: The members will know the private and public keys of their ancestor nodes i.e. the **private keys will be revealed** which is totally against the concept of public key cryptography.

VI. CONCLUSION

The dynamic nature of the group with frequent joins and leaves, the re-keying mechanism is the critical issue in any group key management protocols. Several techniques have been proposed earlier for efficient re-keying like LKH [3], FDLKH [9] requires the establishment of the secure channels for the distribution of the key material. DLPKH [4] eliminates the above said problem but it also suffers from the requirement of huge number of keys for each member as well as high computational resources because of the use of public key cryptosystem for all the operations. When compared to the above mentioned schemes, the technique proposed in this paper has certain advantages like the overhead of secure channel establishment for the secure group key distribution is completely eliminated with the use of public key trees and also the storage requirements for each member is also reduced and moreover less computational resources are required when compared to DLPKH.

REFERENCES

[1] A taxonomy of Group Key Management Protocols: Issues And Solutions, Yacine Challal, Abdelmadjid Bouabdallah, Hamida Seba, Proceedings of World Academy Of Science, Engineering And Technology Volume 6 June 2005.ISSN 1307-6884

[2] G. H. Chiou and W. T. Chen. Secure Broadcast using Secure Lock. IEEE Transactions on Software Engineering, 15(8):929–934, August 1989.

[3] C. K. Wong, M. Gouda, and S. S. Lam. Secure Group Communications Using Key Graphs. IEEE/ACM Transactions on Networking, 8(1):16–30, February 2000.

[4] DLPKH: Distributed Logical Public Key Hierarchy Rakesh Bobba, Himamshu Khurana, Volume 4812/2007 Springer Berlin / Heidelberg.

[5] D.A. McGrew and A.T. Sherman. Key Establishment in Large Dynamic Groups using One-way Function Trees, IEEE Transactions on Software Engineering- Volume 29, Issue 5.

[6] S. Mittra. Iolus: A Framework for Scalable Secure Multicasting.ACM SIGCOMM, 1997

[7] H. Harney and C. Muckenhim, Group Key Management Protocol (GKMP) Architecture," RCF 2094, July 1997.

[8] C. Muckenhim, Group Key Management Protocol (GKMP) Specification, RFC 2093, July 1997.

[9] FDLKH: fully decentralized key management scheme on a logical key hierarchy. Springer Berlin / Heidelberg, Volume 3089/2004.

[10] W.Diffie and M.E Hellman,New directions in cryptography.IEEE Transactions on Information Theeory. Vol. It-22, nov. 1976, pp. 644-654

[11] O.Rodeh,K.Birman and D.Dolev,optimized group re-key for group communication systems.Networks and distributed System security.