Journal of Advances in Science and Technology
Vol. 19, Issue No. 1, March-2022, ISSN 2230-9659

A Study of Hierarchical Group Key Management

Ghanshyam Shivcharan Nikhade[1]*, Dr. Tryambak Hiwarkar[2]

[1] Research Scholar, Department of Computer Science, Sardar Patel University, Balaghat ,M.P.

[2] Professor, Department of Computer Science and Engineering, Sardar Patel University, Balaghat, M.P.

Abstract - Data security in the cloud is where the majority of the most recent work has focused its attention. For safe data transfer in a cloud context, a new key management method that allows for multicast communication must be presented. The study proposes a novel hierarchical key management technique suitable for multicast communication in the cloud. For big, ever-changing multicast systems, the suggested approach is the best option due to its efficiency and scalability. Managing access control to approve and authenticate users utilizing the data is another significant difficulty in a cloud setting. The paradigm of access control used in a conventional setting is very different from that used in the cloud. Therefore, encryption methods, in addition to the access control strategy, must be implemented to safeguard data in group communication.

Keywords - Hierarchical, Group, Key Management, Access Control, Cloud Communication.

----------X----------

INTRODUCTION

The proliferation of internet-based services has led to an increase in the usage of group communication mechanisms in a wide variety of contexts, including Pay TV, online video games, and others. We are now more concerned than ever with the safety of our group chats. In the multicasting method, one participant in a group transmits information to all other participants. Group key management, in which the secret key is shared among all group members, is used to provide secure group communication. Encrypting the message using the private key facilitates its transmission. (1) Key creation and key distribution are the two major components of group key management. There is less complexity in key management and distribution when the number of users or members in a group is low. Nonetheless, a scalability issue in group key management emerges as the number of members grows. Consequently, the hierarchical group key management is proposed to improve scalability and efficiently handle a larger number of members. (2)

Group Key Management in Cloud Computing

Due to the ever-changing nature of network topologies, cloud computing security has been identified as one of its primary challenges. (3) More and more people these days are trusting cloud services with their personal data. To ensure privacy and security of transmitted data, effective key management mechanisms must be used. In this study, we take a key agreement method, in which each node generates a unique key for its own protection, and a key distribution strategy, in which a single node is in charge of both key production and distribution to the rest of the network. (4) The primary concerns for safety in this study are:

- The leaving member should be denied from accessing the future keys in group – Forward secrecy.
- A new member should be denied from accessing the previous keys in the group – Backward secrecy.
- The keys generated should be absolutely different and independent from the key which was generated previously to avoid prediction – Key independence.

Group key management schemes are categorized into three main categories namely distributed, centralized and decentralized key management schemes. Figure 1 shows thetaxonomy of group key management. (5)

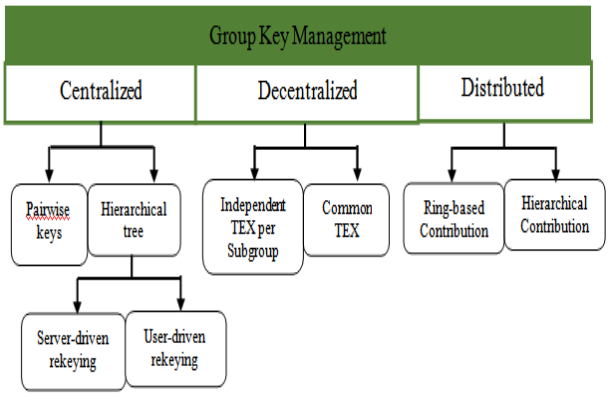

Figure 1: Taxonomy of Group Key Management

- **Centralized:** In this scheme, the group

communication is performed by the single entity. This entity is responsible for both the key generation and key distribution.(6)

- **Decentralized:** In this scheme, the members of the group are split into several subgroups which is managed by a Sub-Group Controller (SGC) thus reduces the load and also the single point of failure problem is solved.(7)

- **Distributed:** In this scheme, there is no group controller involved. The members of the group of multicast session will co-operate each other for the key generation.

RESEARCH METHODOLOGY

The primary goal is to successfully disseminate the group's secret to all members of the group. Therefore, the confidentiality is always protected by changing the key anytime there is a shift in the group. Iolus theory underpins the group's division into several subgroups. The LKH method is used by each subgroup to organize itself into a tree structure. Group Controllers (GCs) and Intermediate Controllers (ICs) are the two controllers that make up the proposed protocol (IC). The information may be securely transmitted to the IC's subgroup members thanks to the GC's distribution of the group key to the ICs. For their respective subgroup, ICs function as servers. We assume in our protocol that ICs are more reliable than other entities and that they are unable to depart the network once they have joined. Organizational structure is kept straight using a binary tree. (8) New members are assigned by GC when they join the group. When there is a shift in the organizational structure, like when members join or depart, the key route and the group key are updated to ensure confidentiality.

Based on Figure 2, we may infer that there are m=8 people in this subgroup, and that the tree has a height of h=log28 = 3. The node keys are $K_{1,2}$, $K_{1,4}$, $K_{3,4}$, $K_{5,6}$, $K_{5,8}$ and $K_{7,8}$ for member u_1, u_2, … , u_8. Finally, the sub-group key KSG is agreed by group members.

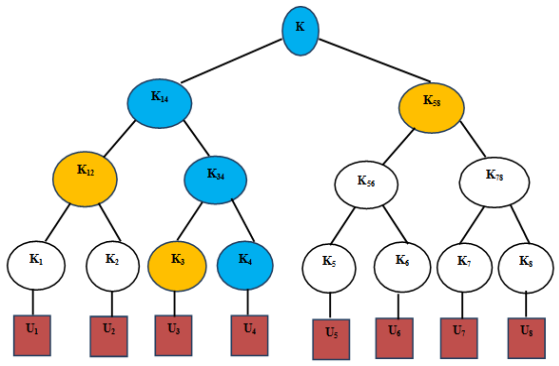

Figure 2: Structure of subgroup hierarchy

As part of this suggested protocol, ICs would be in charge of safeguarding and disseminating the subgroup's secret key. In addition to distributing the data to all members of the subgroup, the ICs are responsible for re-encrypting it using the subgroup key. In this case, the subgroup key, rather than the symmetric key, must be memorized by the group member. (9) The proposed subgroup hierarchy has its leaf nodes secured by keys held by members of the subgroup itself.

Key Structure

Each Ui in the subgroup is given a unique secret id Pi and is also obliged to share a secret value Si of any length (64/128/256 bits). The equation is used to determine each member's unique subgroup key.

$$M_i = ((P_1 \oplus S_1) * (P_2 \oplus S_2), \ldots, * (P_n \oplus S_n)) + K_i \; for \; i = 1, 2, \ldots n$$

Mi is the message; Ki is the random secret key for that group, P1, P2, etc. The members U1, U2,... Un each have a secret id, denoted by Pn, that will never be revealed. Similarly, S1, S2,... Sn represent a different random secret value for each member U1, U2,... Un.

$$K_{SGj} \equiv M_j \; mod \, (P_i \oplus S_i) \; for \; all \; i \; \& \; j$$

If I is the total number of members in the group and j is the group id, then I represents the total number of members and j represents the group id. The message M and subgroup key KSG will be uniquely created for each distinct group.

Every member of the group can independently produce the KSG subgroup key if they know both the secret value and the message that was generated.

This approach is illustrated using a simple example with seven group members. Members u_1, u_2,…,u_7 own keys K_1, K_2,…,K_7 respectively and also node keys $K_{1,2}$, $K_{3,4}$, $K_{5,6}$ and $K_{5,7}$ are at level 2. $K_{1,4}$ and $K_{5,7}$ are node keys at level 1. Here M_1 represents the message generated for group 1. The keys are calculated as follows;

Level 1:

$$K_1 \equiv M_1 \bmod (P_1 \oplus S_1)$$

$$K_2 \equiv M_1 \bmod (P_2 \oplus S_2)$$

$$K_3 \equiv M_1 \bmod (P_3 \oplus S_3)$$

$$K_4 \equiv M_1 \bmod (P_4 \oplus S_4)$$

$$K_5 \equiv M_1 \bmod (P_5 \oplus S_5)$$

$$K_6 \equiv M_1 \bmod (P_6 \oplus S_6)$$

$$K_7 \equiv M_1 \bmod (P_7 \oplus S_7)$$

Level 2:

$$K_{1,2} \equiv M_1 \bmod (P_1 \oplus S_1)(P_2 \oplus S_2)$$

$$K_{3,4} \equiv M_1 \bmod (P_3 \oplus S_3)(P_4 \oplus S_4)$$

$$K_{5,6} \equiv M_1 \bmod (P_5 \oplus S_5)(P_6 \oplus S_6)$$

Level 3:

$$K_{1,4} \equiv M_1 \bmod (P_1 \oplus S_1)(P_2 \oplus S_2)(P_3 \oplus S_3)(P_4 \oplus S_4)$$

$$K_{5,6} \equiv M_1 \bmod (P_5 \oplus S_5)(P_6 \oplus S_6)(P_7 \oplus S_7)$$

After the IC has received the partial keys, it will produce the subgroup key KSG using Equation, and then it will use multicast to send the KSG to everyone in the subgroup.

Intermediate Controller Join

In case of IC join, the group key is changed and the new message $\{K'(G)\}K(G), \{K'(G)\}K(GC, ICn+1)$ is broadcasted, where K'(G) is the new group key, and $K(GC, ICn+1)$ is the key shared between the GC and the new IC. In case of IC join, the change in key will not affect the subgroup members.

Member Join

Each time a member wishes to join a certain group, he sends a "join" request to the KGC server, which serves as the group's controller. It is the responsibility of the GC to assign the new member to the correct group. When a member of the parent group requests to join a subgroup, the IC of that subgroup will receive the request and make any necessary adjustments to the rules of that subgroup to ensure that its backward secrecy is not compromised. Let's say u8 is interested in joining the GC and sends a "join" request on his behalf. The GC will try to find the right subgroup, which in this case is SG1, and assign the new member there.

Figure 3 depicts the subgroup hierarchy after a new member has been added. When user u8 joins, the following procedures take place. Since u8 is a new member, we must generate a new key pair for the node, K7, 8.

Every member of the subgroup receives the inverse value of the departing members with identification P8-1 through broadcast. The new member receives unicast transmissions of the subgroup key and the inverse values of the other members.

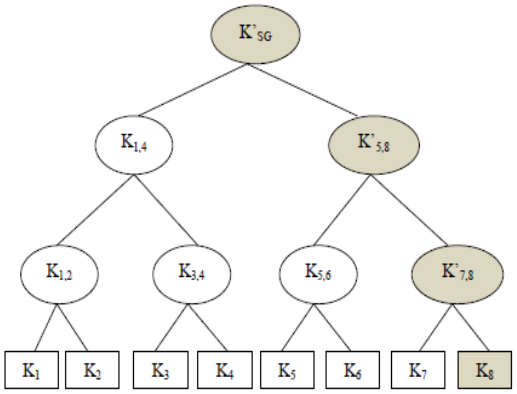

Figure 3: Subgroup 1 hierarchies when u_8 joins

The IC_1 sends the following message to the members in his subgroup:

$$\{K'_{SG1}, (P_8 \oplus S_8)^{-1}\}K_{SG1}, \{K_{5,8}\}K_{5,7}, \{K_{7,8}\}K_7, \{K'_{SG1}, K_{5,8}, K_{7,8}, (P_1 \oplus S_1)^{-1}, \ldots, (P_7 \oplus S_7)^{-1}\}K_8$$

The last message is decrypted as

$$K_{7,8} \equiv M_i \bmod (P_7 \oplus S_7)(P_8 \oplus S_8)$$

$$K_{5,8} \equiv M_i \bmod (P_5 \oplus S_5)(P_6 \oplus S_6)(P_7 \oplus S_7)(P_8 \oplus S_8)$$

$$K'_{SG1} \equiv M'_i \bmod (P_1 \oplus S_1) * (P_2 \oplus S_2) * \ldots * (P_8 \oplus S_8) + K'_i \text{ for } i = 1, 2, 3 \ldots n$$

Where i represent the group id

Member Leave

A "leave" request is sent to the subgroup's IC when a member decides he no longer wishes to be a part of the subgroup. The IC notifies everyone of the departing member's true identity. The IC also notifies the rest of the subgroup of the new keys they need to use. Subgroup members will update their keys with the departing member's inverse value once they get the IC message. (10) To keep forward secrecy once a member of a subgroup leaves, for instance when u8 decides to stop being a part of SG1, the keys KSG1, K5,8, K7,8 must be modified.

Our protocol dictates that the modified keys need not be sent to the other participants. However, the IC1 just has to prepare a single message to indicate u8 is leaving and the keys need to be changed. The

IC_1 sends the message $\{u_8, K'_{SG1}\}K'_{SG1}, \{K_{5,8}\}K_{5,8}$.

All of the message's keys are used to encrypt the message itself. By receiving such a communication, the remaining members can determine that there has been a leave operation, and that member identification 8 has departed. They use the inverse value of u8 $(P_8 \oplus S_8)^{-1}$ to update the sent keys.

$$M'' = ((P_1 \oplus S_1) * (P_2 \oplus S_2) * ..... * (P_{j-1} \oplus S_{j-1}))((P_{j+1} \oplus S_{j+1}) * ... * (P_n \oplus S_n) + K'_i$$

$$K''_{SG1} \equiv M'' \bmod (P_i \oplus S_i) \text{ for } i = 1, 2, .., 7$$

$$K'_{5,8} \equiv (K_{5,8})(P_8 \oplus S_8)^{-1}$$

$$K'_{5,8} \equiv M'' \bmod (P_5 \oplus S_5)(P_6 \oplus S_6)(P_7 \oplus S_7)(P_8 \oplus S_8)(P_8 \oplus S_8)^{-1}$$

$$K'_{5,8} \equiv M'' \bmod (P_5 \oplus S_5)(P_6 \oplus S_6)(P_7 \oplus S_7)$$

RESULT ANALYSIS

Existing protocols are compared to the proposed one, and their performance is examined. Each subgroup in this protocol analysis has a size of m = 2h people. The proposed work is a Java programme that can handle groups of any size between 16 and 4096. The outcomes depend on how the implementation is carried out. We take into account both computational and storage cost in our comparisons.

Computational overhead includes the time spent on tasks like key creation and encrypting and decrypting data. Key generation overhead refers to the total amount of keys created by both the members and the server.

Table 1: Key generations for join and leave operation

Protocols	Key Server		Member Nodes	
	Join	Leave	Join	Leave
Simple	2	1	0	0
LKH	log2n	log2n-1	0	0
Proposed Protocol	log2m	0	0	1

In Table 1, we can see that as a member leaves the complexity of the proposed protocol decreases. During vacation, the IC won't produce any keys since they use other inverted values to generate them. Key generation for join and depart operations with varying group sizes is displayed in Figure.

If we assume that there are N subgroups and m individuals in each subgroup, then the total number of participants in our suggested procedure is n = N x m. Log2 (n/m) is the formula used to determine the total number of keys created by the key server during a join transaction. In contrast, we treated our procedure as log2m because it was compared to the LKH method. The key created during a join is log2 (n/m) if n is greater than or equal to 2m, and it is 2 otherwise.

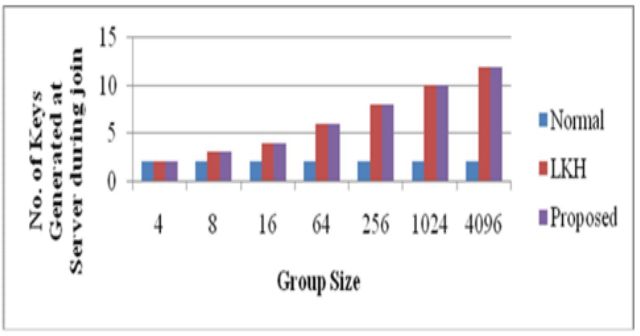

Figure 4: Number of key generation at server for join

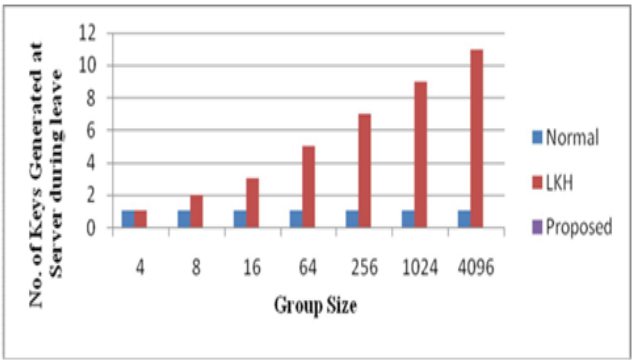

Figure 5: Number of key generation at server for leave

The number of encryption/decryptions at member side and server is said to be an encryption/decryption overhead.

Table 2: Number of Encryptions/Decryptions during join and leaveoperation

Protocol	Key Server	Member Node	Key Server	Member Node
	Join	Leave	Join	Leave
Simple	2	n-1	1	1
LKH	3.log2n	2.log2n	log2n	log2n
Proposed	log2n+1	log2n-1	1	0

Based on the data in Table 2, it is clear that a departure from membership decreases complexity on one side. When compared to other protocols, the number of encryptions is less. The proposed protocol improves upon the state-of-the-art when it comes to the encryption of messages at the key server during the join and departs operations, respectively. (11)

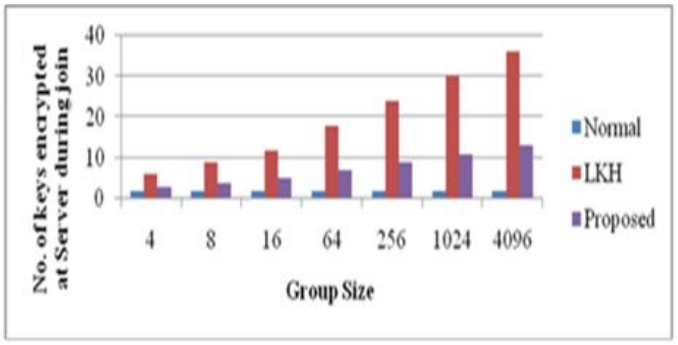

Figure 6: No. of messages Encrypted at the key server during joinoperation

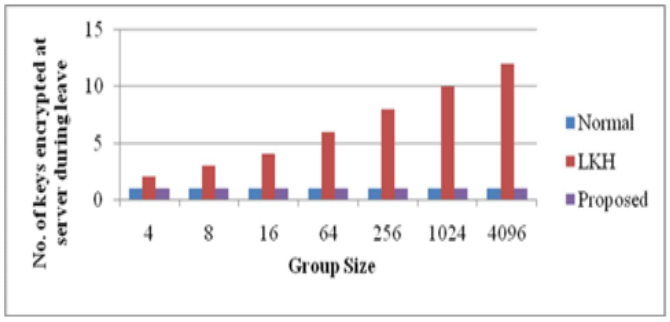

Figure 7: No. of messages Encrypted at the key server during leaveoperation

Figures 8 and 9 show how many keys are decrypted at each member node before a join or a depart operation. The suggested protocol performs a minimal amount of operations in both circumstances when compared to alternative methods. (12)

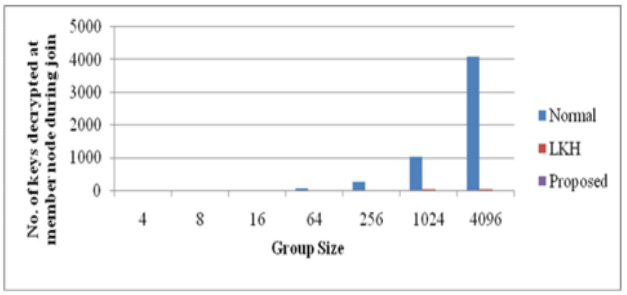

Figure 8: No. of messages decrypted at the member node during join

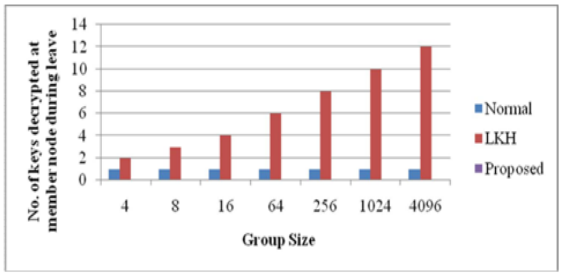

Figure 9: No. of messages decrypted at the member node during leave

The number of keys stored at member side as well as server side is said to be a storage complexity. Table 3 shows the storage complexity at key server and at member node.

Table 3: Storage Complexity

Protocol	Key Server	Member Node
Simple	n	2
LKH	2n	log2n+1
Proposed	2n	log2n+1

As part of the proposed protocol, the server keeps track of 2n node keys, which include the inverse values of the group member, while the members keep track of log2n + 1 key, which include both the path keys and the inverse values. The total number of keys saved in the server and in the member nodes is shown in Figure 10 and Figure 11.

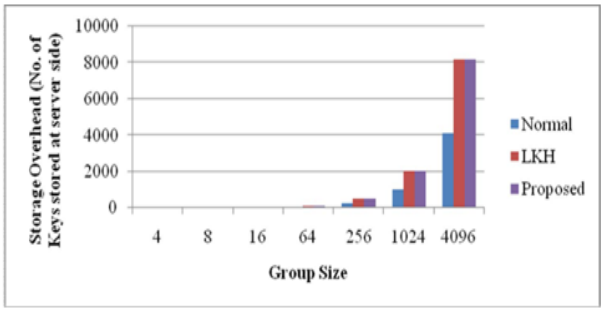

Figure 10: Storage overhead at key server

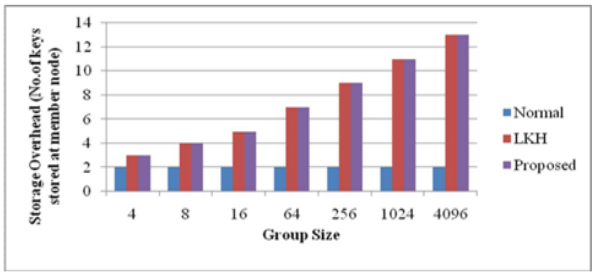

Figure 11: Storage overhead at member node

In general, the suggested solution reduces the computational and storage costs associated with leaving a group.

To address problems with symmetric key distribution among group members, including message size, storage complexity, and other overheads, this study proposes a scalable and efficient protocol for secure group communication. Due to the breakdown of this method, the complexity of a member join or leave operation is reduced from O(n) to O(log2m), where n is the total number of members in the group and m is the number of members in the subgroup. Both the LKH and Iolus protocols are used as benchmarks against which the performance of the proposed protocol may be evaluated. Key generation complexity, multicast message size, storage

overhead, and encryption/decryption overhead are all taken into account in this comparison. The outcomes demonstrate that the suggested protocol improves group performance in terms of computing overhead, particularly during the member exit process. The cloud might be used to execute and apply the results of future efforts in this area. This method will simplify storage while improving scalability and overall system performance.

CONCLUSION

The study proposes and implements novel strategies for protecting cloud infrastructure against cyber attacks. The study makes three significant contributions toward this goal. Our first major contribution is a suggestion for a key management strategy to prevent unwanted access to cloud-based software. The implements a novel, efficient technique for maintaining group key in a shared environment to prevent unwanted access in a cloud computing setting. To safely disperse the group key to the members of the group, the proposed key management protocol uses a distributed and decentralized key server to coordinate runs of the protocol. Second, a variety of access-control measures have been proposed to keep hackers out of the cloud. Our third original contribution is a plan for a safe online learning environment in the cloud, complete with multi-factor authentication and encrypted data storage.

REFERENCES

1. Sakshi Chhabra (2020) on “Security Enhancement in Cloud Environment using Secure Secret Key Sharing”, JOURNAL OF COMMUNICATIONS SOFTWARE AND SYSTEMS, VOL. 16, NO. 4.

2. Pradeep. K. V, Vijayakumar. V (2019) on “Secure Key Management System in Cloud Environment for Client data”, International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249-8958, Volume-8 Issue-5

3. Matthew Campagna, Shay Gueron (2019) on “Key Management Systems at the Cloud Scale”, Cryptography, 3, 23; doi:10.3390/cryptography3030023

4. Padinjappurathu Gopalan Shynu, Kumaresan John Singh (2017) on “An Enhanced ABE based Secure Access Control Scheme for E-health Clouds”, International Journal of Intelligent Engineering and Systems, Vol.10, No.5

5. Madhura Mulimani, Rashmi Rachh (2017) on “Analysis of Access Control Methods in Cloud Computing”, I.J. Education and Management Engineering, 3, 15-24

6. Sultan Aldossary, William Allen (2016) on “Data Security, Privacy, Availability and Integrity in Cloud Computing: Issues and Current Solutions”, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 7, No. 4

7. Kire Jakimoski (2016) on “Security Techniques for Data Protection in Cloud Computing”, International Journal of Grid and Distributed Computing, Vol. 9

8. S.Raghavendra, K.Meghana, P.A.Doddabasappa (2016) on “Index Generation and Secure Multi-user Access Control over an Encrypted Cloud Data”, Procedia Computer Science Volume 89, 2016, Pages 293-300

9. Surya Nepal, Carsten Friedrich, Catherine Wise (2016) on “key management service: enabling secure sharing and deleting of documents on public clouds”, Services Transactions of Cloud Computing (ISSN 2326-7550)

10. Mansura Habiba, Md. Rafiqul Islam (2015) on “A New Approach to Access Control in Cloud”, Arabian Journal for Science And Engineering 41(3) DOI:10.1007/s13369-015-1947-8

11. S. Muthurajkumar, S.Ganapathy, M. Vijayalakshmi, A. Kannan (2015) on “Secured Temporal Log Management Techniques for Cloud”, International Conference on Information and Communication Technologies, doi: 10.1016/j.procs.2015.02.098

12. Shabana Rehman, Rahul Gautam (2014) on “Research on Access Control Techniques in SaaS of Cloud Computing”, International Symposium on Security in Computing and Communication.

Corresponding Author

Ghanshyam Shivcharan Nikhade*

Research Scholar, Department of Computer Science, Sardar Patel University, Balaghat ,M.P.