International Journal of Electrical and Computer Engineering (IJECE)
Vol. 7, No. 6, December 2017, pp. 3593 – 3601
ISSN: 2088-8708, DOI: 10.11591/ijece.v7i6.pp3593-3601

Ternary Tree Based Approach For Accessing the Resources by Overlapping Members in Cloud Computing

Amar Buchade[1] and Rajesh Ingle[2]
[1]Department of Computer Engineering, College of Engineering Pune, Savitribai Phule Pune University
[2]Department of Computer Engineering, College of Engineering Pune and Pune Institute of Computer Technology, Savitribai Phule Pune University

Article Info

Article history:
Received: Feb 14, 2017
Revised: Jul 6, 2017
Accepted: Jul 22, 2017

Keyword:
Key Management
Ternary Tree
Cloud Computing
Overlapping members
Join Leave

ABSTRACT

In cloud computing, immediate access of resources is important due to cost incurred to customer by pay per use model of cloud computing. Usually resource is protected by using cryptography technique. The resource may be shared by multiple members in group. There can be overlapping members to access the multiple resources. Group key management is important to form the group key to access the resource. Group key formation time is crucial for immediate access of protected resource in cloud computing. Thus ternary tree based approach is proposed to form the key for overlapping members accessing resources. Membership event such as join and leave also considered. Through the analysis, it is found that computational overhead is reduced by 23% if ternary key trees are combined than independent ternary key trees. It is also observed that combined ternary key tree outperforms the combined binary key tree approach for group key formation by considering overlapping members. Security requirement analysis of group membership for key formation is also provided in the paper.

Corresponding Author:
Amar Buchade
Research Scholar
College of Engineering, Pune
Savitribai Phule Pune University
amar.buchade@gmail.com

1. INTRODUCTION

In recent days, the use of cloud computing is increasing. Usage of cloud based social media applications whatsapp, twitter, facebook and collaborative applications, Pay TV [1] systems are increasing. Many governments have initiated digital move including cashless transactions, m-wallet etc. Thus security is major concern over the usage of many cloud based applications. Obviously there is important role of cryptographic algorithm to secure the resource. Resource can be considered as data, applications, storage, CPU, virtual machine etc. Thus key management plays significance role to access the resources from cloud computing.

In cloud computing, it is important to have instant (on demand) access of resources. It is important to form the group key within a time for immediate access of resources. For collaborative environment, group key among the members needs to be formed to protect the access of resources in cloud computing. There can be member accessing multiple resources. Such member in group/s is called overlapping member. Thus in this paper, the problem of overlapping members accessing multiple resources is proposed.

Group key can be formed by TGDH approach [2],[3]. Presently for group key formation separate key trees are formed even if member has access to multiple resources. This approach leads to computational overhead to form the group key.We preferred ternary key tree approach over binary tree based approach for group key formation because as the number of members increased in ternary key tree, the height is also reduced compared to binary. Hence computational cost for forming group key is also reduced. The novelty of the approach is that it is computational efficient that the other approaches.

[4],[5],[6] proposes binary tree based approach for multiple members overlapped to access the resources. [7] specified computational time analysis for group key formation by multiple members accessing resources.Binary tree based approach [8],[9],[10] proposes group key management protocol in distributed group communication. Group members are managed in the hierarchical manner logically. Diffie-Hellman key agreement is applied. [11] proposed suite of group key management protocols that allows a group of users to agree on a shared group key, which can be used to protect a shared file system stored remotely in the cloud. [12] proposes and applies key management methods to various cloud environments. [13] proposes group leader, group administrator approach.Data is shared and accessed by group member based on the group key. [14] describes group key management technique. [14] sharing of files among different users maintained at Cloud. These users form group. The tresor contains all the encrypted files. It is protected by group key. The concept of Key Lock Box is proposed. Every directory has key-lock-box and contains the keys of files within directory. Thus our contribution towards this paper is

1. Combining ternary key trees algorithm.
2. Consideration of membership event such as join and leave.
3. Computational and communication cost analysis of group key formation by considering separate key trees and combining ternary based key trees.
4. Security analysis of proposed scheme.

The section 2 presents the proposed method, section 3 presents research method details, section 4 presents the results and analysis section, section 5 concludes the paper.

2. PROPOSED METHOD

2.1. Motivation

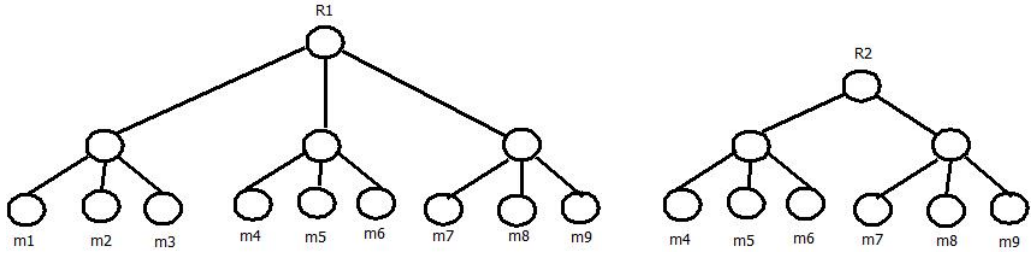


Figure 1. Resource R1 and R2 key trees formations with the members

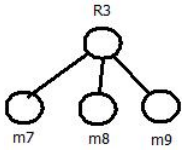


Figure 2. Resource R3 key tree

Figure 1 represents resource key tree R_1. Members $\{m_1, m_2, m_3, m_4, m_5, m_6, m_7, m_8, m_9\}$ are part of resource R_1. Members $\{m_4, m_5, m_6, m_7, m_8, m_9\}$ are part of resource R_2. Members $\{m_7, m_8, m_9\}$ are part of R_3. Respective group key can be formed independently by three way Diffie hellman key exchange protocol as explained in subsection 2.3.. From figures 1 and 2, it is observed that if we calculate each resource group key separately (independently), it causes the extra computations and extra partial keys incurred by members $\{m_4, m_5, m_6, m_7, m_8, m_9\}$. This is because these members are common to access resource R_2 and resource R_1. Members $\{m_7, m_8, m_9\}$ are common to access resource R_3. Thus key formed at resource R_3 can be reused to form the resource group key at R_2 while key formed at resource R_2 can be reused to compute resource group at R_1. This approach causes less computational cost to form the key than the independent group key formation. Thus we use overlapping members approach for accessing multiple resources.

2.2. Overlapping Members

Let R be the set of resources i.e. $\{R_1, R_2, R_3, ..., R_n\}$
Members $\{m_1, m_2, ..., m_n\} \in R_1$.
Members $\{n_1, n_2, ..., n_n\} \in R_2$
Overlapping member can be defined as $\{x|(x \in R_1) \wedge (x \in R_2)\}$. From figure 3, members $\{m_2$,m_3,$n_2\}$ are overlapping members accessing resources R_1 and R_2.

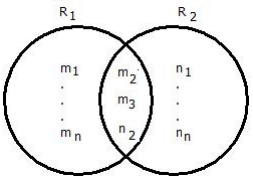


Figure 3. Overlapping members access to resources

In earlier approach, we proposed binary key tree approach [4], [7] for overlapping resource access members. In this approach, we can prove how ternary tree data structure improves the performance over earlier proposed approach.

2.3. Group key formation by using ternary key tree

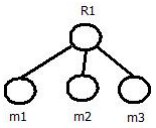


Figure 4. Ternary based resource key tree

Group key can be formed by TGDH approach [2],[3]. It uses bottom-up approach. Basic TGDH uses two way Diffie hellman key exchange algorithm. In this paper, we modified it for three way Diffie hellman key exchange algorithm .

Figure 4 shows the ternary tree. Root node is called as R1 and leaf nodes represents the members m_1, m_2 and m_3. Here we assume that g is generator and p is prime number. Members m_1, m_2 and m_3 have β_1, β_2, β_3 private keys respectively. Each member forms the key called as blinded key. Thus group key among these members is formed by the following steps.

1. Member m_1 forms key called as $g^{\beta_1} \ mod \ p$
2. Member m_2 forms key called as $g^{\beta_2} \ mod \ p$
3. Member m_3 forms key called as $g^{\beta_3} \ mod \ p$
4. Member m_1 sends $g^{\beta_1} \ mod \ p$ to member m_2
5. Member m_2 forms $g^{\beta_1\beta_2} \ mod \ p$
6. Member m_2 sends $g^{\beta_1} \ mod \ p$, $g^{\beta_2} \ mod \ p$ and $g^{\beta_1\beta_2} \ mod \ p$ to member m_3
7. Member m_3 forms $g^{\beta_3\beta_2} \ mod \ p$, $g^{\beta_3\beta_1} \ mod \ p$ and $g^{\beta_1\beta_2\beta_3} \ mod \ p$
8. Member m_3 sends $g^{\beta_3\beta_2} \ mod \ p$, $g^{\beta_3\beta_1} \ mod \ p$ to member m_1 and m_2
9. Member m_1 forms $g^{\beta_1\beta_2\beta_3} \ mod \ p$
10. Member m_2 forms $g^{\beta_1\beta_2\beta_3} \ mod \ p$

By steps 1, 2, 3, 5, 7,9 and 10, total modular exponential operations (MEO) required are nine. Group key formed as $g^{\beta_1\beta_2\beta_3}\ mod\ p$. Total messages involved are mainly two unicast messages namely at step 4 and step 6 and one broadcast message namely at step 8. Thus total three messages are used to establish the group key. In general formula for calculating total number of exponential operations is as below.

$$N = 9(3^{log_3^n} - 1)/2$$

Where N = Number of modular exponential operations, n = number of members e.g if n=3, by above formula, N = 9

3. RESEARCH METHOD

3.1. Member Resource Access Matrix (MRAM)

Any member for resource access, broadcast message containing resource membership details and current resource request for which resource. Thus each member makes entry in MRAM. Rows represents members m_1,m_2,m_3,...,m_n . Columns represents resources R_1, R_2, R_3,...,R_n

$$\begin{bmatrix} 1 & 0 & \cdots & \cdots \\ 1 & 0 & \cdots & \cdots \\ 1 & 1 & \cdots & \cdots \\ 1 & \cdots & \cdots & \cdots \end{bmatrix}$$

$m_1 \in R_1$ and $m_3 \in R_1, R_2$ i.e. m_3 overlapped to access the resources R_1 and R_2. These are indicated by '1' in the MRAM.

3.2. Combining Ternary Key Trees Algorithm

In existing key management algorithm [8],[9],[10],[11],[13],[15],[16] separate key tree is built for each resource, even if members are accessing multiple resources. Thus we can combine multiple resource key trees. Algorithm 1 illustrates combining ternary key trees algorithm. Computation cost analysis is given in subsection 3.3.

Algorithm 1 Combining Ternary based Resource Key Trees Algorithm

1: Begin
2: The member which wants the access of particular resource, broadcast request to access the resource. Each member makes the entry in member resource access matrix.
3: Let R_1, R_2, R_3, R_4, ...R_n be the set of resources.
4: Let m_1,m_2,m_3,....,m_n be set of members.
5: Each member keep the track of resource membership in member resource access matrix.
Rows represents members m_1,m_2,m_3,....,m_n
and Columns represents resources R_1, R_2, R_3,....R_n

$$\begin{bmatrix} 1 & 0 & \cdots & \cdots \\ 1 & 0 & \cdots & \cdots \\ 1 & 1 & \cdots & \cdots \\ 1 & \cdots & \cdots & \cdots \end{bmatrix}$$

6: Identify the members which are overlapped to access multiple resources.
7: Build the key graph of overlapped members. Maintain the entries such as resources and overlapping members in table 1.
8: Identify the members which are not overlapped. Build the key tree of members which are not overlapped.
9: Combine the trees which are formed during Step 7 and Step 8.
10: End

Table 1. Resources containing overlapped members

Index	Resources	Overlapping members
1	R_1,R_2	m_3

3.3. Computational Cost for Group Key Formation

There can be multiple members overlapping to any resources.
Modular exponential operations (MEO) after combining ternary key trees
= MEO for separate ternary key trees - MEO due to overlapping members

$$\text{MEO for separate ternary key trees} = \sum_{i=1}^{K}(9(3^{log_3^{N_i}} - 1)/2)$$

MEO due to overlapping member =
$\sum_{index=1}^{T}(Rcount[index]) - 1)(9(3^{log_3^{C[index]}} - 1)/2)$ ... from Table 1

where
N = Number of members per ternary key tree
K = total number of ternary key trees
T = Number of entries formed as per table 1
Rcount = Total Resource count per entry as per table 1
C = Number of Members overlapped per entry as per table 1

It is observed that computation cost in terms of number of modular exponential for separate key tree is O(N) while for ternary key trees combined is O(N – RC) where N is number of members of resource key trees, R is number of resources considered and C is overlapping members.

Thus we can observe that number of modular exponential operations required in separate key trees is more i.e RC compared to the combined key trees. Table 2 describes complexity in terms of modular exponential operations.

Table 2. Complexity in terms of MEO

Best Case	Worst Case
$\Omega(N)$	O(2N)

Best case complexity when all members of resource groups are overlapped to access the resources. Worst case complexity is when members of resource groups are not overlapped to access the resources.

3.4. Security Requirement

The members which are not part of resource group must not be able to access the resource.Following is the list of security requirement for group membership which should not be violated.

1. Backward secrecy: The members which joined to access the resource recently should not get access of past key. This property is called as backward secrecy.
2. Forward secrecy: The members which left from resource group should not get access of future key. This property is called as forward secrecy.

Section 4.1. illustrates the detailed security analysis of the proposed approach.

3.5. Group Key Formation Steps for Overlapping Members

Figure 5 describes overlapping members access to resources R1, R2 and R3.
Members $\{m1, m2, m3, m4, m5, m6, m7, m8, m9\}$ have access to resource R1.
Members $\{m4, m5, m6, m10, m11, m12, m13, m14, m15\}$ have access to resource R2.
Members $\{m7, m8, m9, m10, m11, m12, m13, m14, m15\}$ have access to resource R3.
Resource access representation is shown in the following.
OG1 $\rightarrow \{m4, m5, m6\}$
OG2 $\rightarrow \{m7, m8, m9\}$

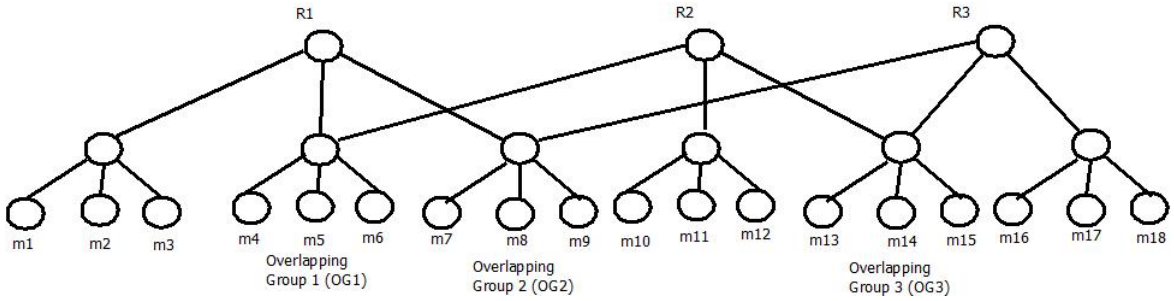


Figure 5. Overlapping members access to resources R1, R2 and R3

OG3 $\rightarrow \{m13, m14, m15\}$
R1 $\rightarrow \{m1, m2, m3, OG1, OG2\}$
R2 $\rightarrow \{OG1, m10, m11, m12, OG3\}$
R3 $\rightarrow \{OG2, OG3, m16, m17, m18\}$
When we form the group key using ternary key tree approach, the group key formation steps are as follows 1) Overlapping group (OG) member forms its partial group key as per TGDH approach.
2) Each subgroup forms it partial group key by approach used in subsection 2.3..
3) Group DH used to calculate the group key.
There are different events occurred during the key tree formation. a) Any member can join the group or overlapping members group. b) Any member can leave the group or overlapping members at any instant of time.

3.6. Scenarios

The following section elaborates about membership join at OG, non OG as well as moving from one OG to other OG. a) If the member m19 joins the group to have access to R1 as shown in figure 6. Member

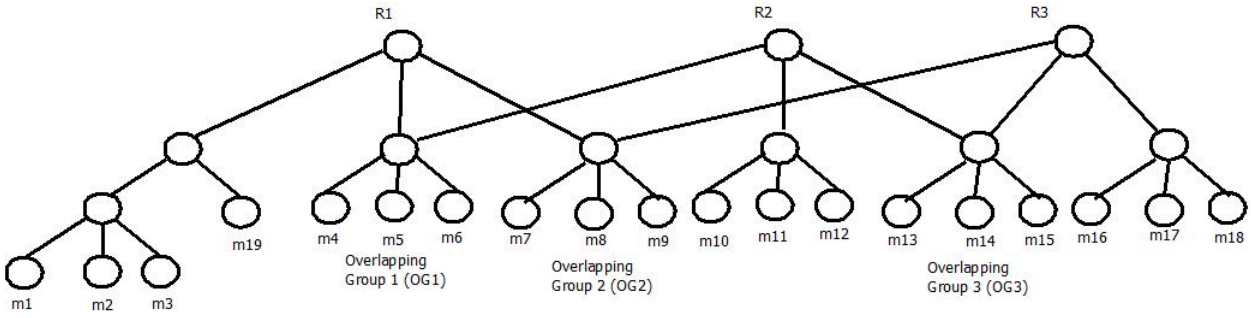


Figure 6. Member m19 join at non overlapping members group of resource R1

m19 is inserted at the rightmost node in the key tree of non-overlapping group. Thus group key can be formed by partial group key formed by sub group of non-overlapping members and overlapping group members. b) If member m19 joins OG1 with respect to figure 5. This is shown in figure 7. c) Member can move from one OG to other group. This is shown in figure 8. Member m6 can move to non-overlapping group members of R2 while member m12 can move to overlapping group 1 to access the resource R1 as well. This is shown in the figure 8.

4. RESULT AND ANALYSIS

Simulation is performed. The results are compared by considering separate key trees, combined key trees with ternary approach and binary key tree approach. From figure 9, when we consider number of Resources =2, total members 200, the number of overlapping members varied, the number of modular exponentiation operations are less for combing key trees (ternary) with overlapping than the other approaches. In case of combining key trees (binary) with overlapping, computational overhead (56%) is more than the combining key

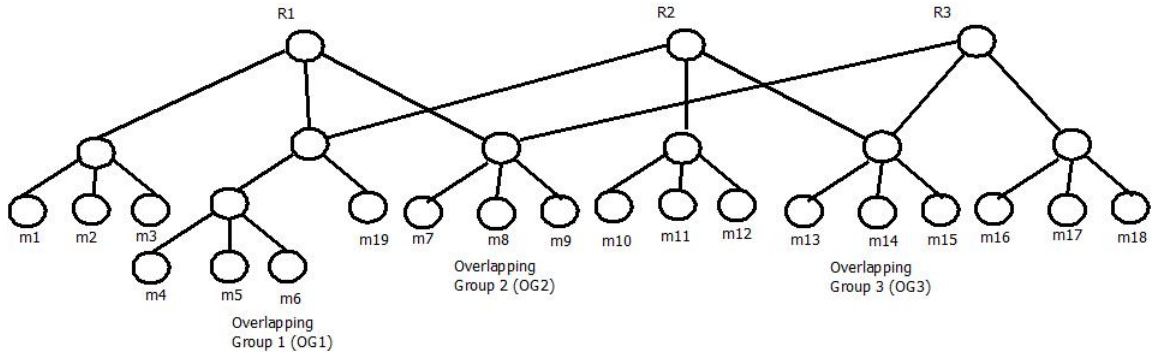


Figure 7. Member m19 join at OG1 of resource R1 and resource R2

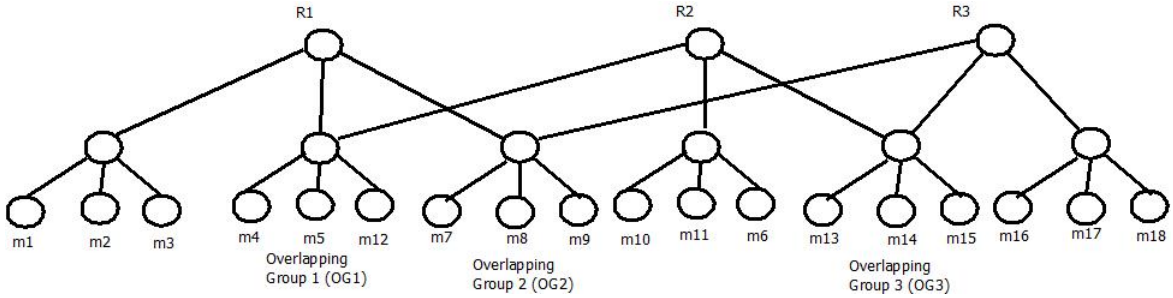
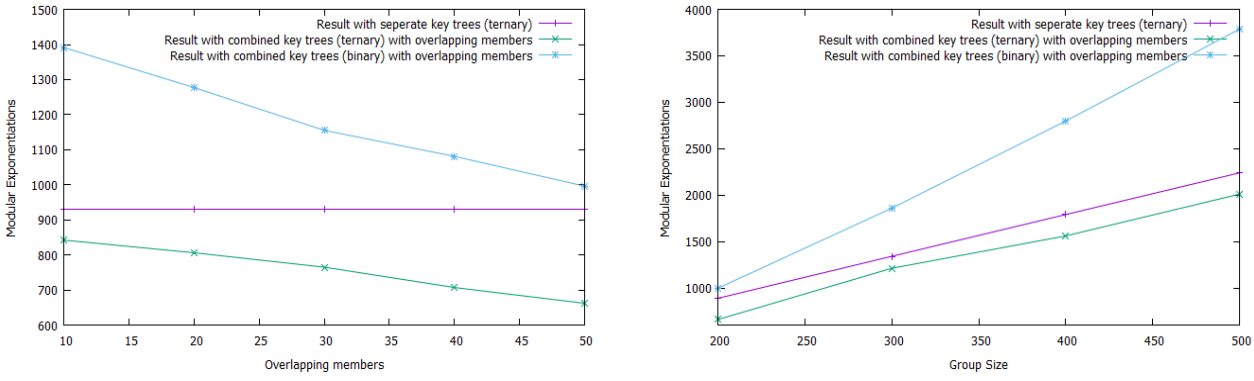


Figure 8. Member m6 move from OG1 to access resource R2 only and member m12 to OG1

trees (ternary) with overlapping approach. If we consider the independent (separate) key trees, the computational overhead (23%) is more than the combining key trees (ternary) with overlapping approach.Computational overhead is more in binary tree is more due to increase in height of the binary when members are increasing.

Figure 10 shows that as we vary the total number of members, modular exponentiation operations in combining key trees (ternary) with overlapping members are increased slowly as compared with other two approaches. The computational overhead is (73%) more in combining key tree (binary) with overlapping members and 15% more in group key formation with separate key trees compared with combined key tree (ternary) based approach.



(a) Analysis by varying overlapping members

(b) Analysis by varying members

Figure 9. Computational cost

Table 3 depict comparisons in terms of computational and communication cost. Authors[8],[9],[10],[11],[13] uses independent binary key trees based approach. Authors [6],[7] use binary key tree based approach with the consideration of overlapping members. Our approach uses ternary based key trees with overlapping resource access members. 'k' indicates total number of key trees (resources), n_i indicates total number of members per

resource key tree, 'c' indicates number of overlapping members among resource key trees. From the table 3 , it is observed that from our approach, as the overlapping members increases, computational and communication cost decreases.

Table 3. Comparisons: Computational and Communication cost

Approach	Modular Exponential Operations	Number of messages
Authors[8],[9],[10],[11],[13],[15],[16]	$(\sum_{i=1}^{k} n_i(1+\log_2 n_i))$	$\sum_{i=1}^{k}(2*n_i-2)$
Gu Xiaozhuo[6], Buchade [4]	$(\sum_{i=1}^{k} n_i(1+\log_2 n_i)) - c(1+\log_2 c)$	$\sum_{i=1}^{k}(2*n_i-2)-(2c-2)$
Ternary trees without overlapping	$\sum_{i=1}^{k} 9(3^{log_3^{n_i}}-1)/2$	$\sum_{i=1}^{k}(3(n_i-1)/2$
Our approach	$(\sum_{i=1}^{k} 9(3^{log_3^{n_i}}-1)/2) - (9(3^{log_3^{c}}-1)/2)$	$(\sum_{i=1}^{k}(3(n_i-1)/2) - (3(c-1)/2)$

4.1. Security Analysis

The group key formation for combined resource key tree is similar to TGDH. Thus attacker does not form any partial group key due to its discrete log problem of DH algorithm. By giving g and p, it is impossible to form $g^{\beta_1} \ mod \ p$, $g^{\beta_2} \ mod \ p$, $g^{\beta_1\beta_2} \ mod \ p$. It is impossible to get the group key by the attackers because private keys are kept with the use. Blinded keys are used to form the group key.
Backward secrecy is also achieved when new member joins the group. Thus group key is renewed after joining the members. So that member does not know previously formed group key.
Forward secrecy is also achieved when new member leaves the group. Thus group key is renewed after leaving the members. So that leaving member does not know future group key.
When member moves from one overlapping group to other, group secrecy is also achieved. The members in overlapping group updates the partial subgroup key when member joins or leave occurs thus backward and forward secrecy is achieved.

5. CONCLUSION

In cloud computing, immediate access of protected resource is important. Resource can be considered as data, applications, storage, CPU, virtual machine etc. Applications e.g social media that uses cloud computing also increasing. Key formation is important step for members of such applications. As soon as group key is formed, user should be able to access the resource in cloud computing and not to violate the on demand resource access property of cloud computing. We proposed novel ternary key based group key formation method for members with access to multiple resources. We proved that group key formation using ternary key tree approach by considering overlapping members is computationally efficient than the independent group key formation as well as combined key tree (binary) approach.

REFERENCES

[1] K.-Y. Chou, Y.-R. Chen, and W.-G. Tzeng, "An efficient and secure group key management scheme supporting frequent key updates on pay-tv systems," in *Network Operations and Management Symposium (APNOMS), 2011 13th Asia-Pacific*. IEEE, 2011, pp. 1–8.
[2] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theor.*, vol. 22, no. 6, pp. 644–654, 1976.
[3] Y. Kim, A. Perrig, and G. Tsudik, "Tree-based group key agreement," *ACM Transactions on Information and System Security (TISSEC)*, vol. 7, no. 1, pp. 60–96, 2004.
[4] A. Buchade and R. Ingle, "Key trees combining algorithm for overlapping resource access members," *International Journal of Network Security*, vol. 18, no. 5, pp. 855–860, 2016.
[5] R. Aparna and B. Amberker, "Key management schemes for multilayer and multiple simultaneous secure group communication," *ISRN Communications and Networking*, vol. 2012, 2012.
[6] X. Gu, Y. Zhao, and J. Yang, "Reducing rekeying time using an integrated group key agreement scheme," *Journal of Communications and Networks*, vol. 14, no. 4, pp. 418–428, 2012.

[7] A. Buchade and R. Ingle, "Analysis of algorithms for overlapping resource access members in cloud computing," vol. 18, no. 5, 2016, pp. 978–986.
[8] S. A. Mortazavi, A. N. Pour, and T. Kato, "An efficient distributed group key management using hierarchical approach with diffie-hellman and symmetric algorithm: Dhsa," in *Computer Networks and Distributed Systems (CNDS), 2011 International Symposium on*. IEEE, 2011, pp. 49–54.
[9] S. Jarecki, J. Kim, and G. Tsudik, "Flexible robust group key agreement," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 5, pp. 879–886, 2011.
[10] Lin, Huang, F. Lai, and H. C. Lee, "Secure and efficient group key management with shared key derivation," vol. 31. Elsevier Science, 2009.
[11] S. Szebeni, L. Butty'n *et al.*, "Invitation-oriented tgdh: Key management for dynamic groups in an asynchronous communication model," in *2012 41st International Conference on Parallel Processing Workshops*. IEEE, 2012, pp. 269–276.
[12] A. R. Buchade and R. Ingle, "Key management for cloud data storage: Methods and comparisons," in *2014 Fourth International Conference on Advanced Computing & Communication Technologies*. IEEE, 2014, pp. 263–270.
[13] K. Xue and P. Hong, "A dynamic secure group sharing framework in public cloud computing," *IEEE Transactions on Cloud Computing*, vol. 2, no. 4, pp. 459–470, 2014.
[14] S. Szebeni and L. Butty'n, "Tresorium: cryptographic file system for dynamic groups over untrusted cloud storage," in *2012 41st International Conference on Parallel Processing Workshops*. IEEE, 2012, pp. 296–303.
[15] P. Qiao, "On the security of a dynamic and secure key management model for hierarchical heterogeneous sensor networks," *TELKOMNIKA Indonesian Journal of Electrical Engineering*, vol. 12, no. 10, pp. 7459–7462, 2014. [Online]. Available: http://iaesjournal.com/online/index.php/TELKOMNIKA/article/view/5579
[16] A. Diop, Y. Qi, and Q. Wang, "An improved key management scheme for hierarchical wireless sensors networks," *TELKOMNIKA Indonesian Journal of Electrical Engineering*, vol. 12, no. 5, pp. 3969–3978, 2014. [Online]. Available: http://iaesjournal.com/online/index.php/TELKOMNIKA/article/view/4241

BIOGRAPHY OF AUTHORS

Amar Buchade is Research Scholar at College of Engineering, Pune under Savitribai Phule Pune University. He has received B.E. and M.E. in Computer Engineering from Walchand College of Engineering, Sangli in 2002 and 2005 respectively. He is a member of IEEE and life member of ISTE. His research area is Distributed System, Cloud computing and Security.

Rajesh Ingle is adjunct professor at Department of Computer Engineering, College of Engineering Pune, India.He is professor at Pune Insitute of Computer Technology, Pune. It is affiliated to Savitribai Phule Pune University. He has received Ph.D. CSE from Department of Computer Science and Engineering, Indian Institute of Technology Bombay, Powai. He has received the B.E. and M.E. Computer Engineering from Savitribai Phule Pune University. He has also received M.S. Software Systems from BITS, Pilani, India. He is a senior member of the IEEE, IEEE Communications Society, and IEEE Computer Society. He is serving as Region 10 Asia Pacific Student Activities Chair 2015-18. His research area is Distributed system security, Grid middleware, Cloud security, Multimedia networks and spontaneously networked environments.