International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 3 Issue: 6 3750 - 3754

Group Based Secure Sharing of Cloud Data with Provable Data Freshness

Hussein Ali Ghadhban Salman
Department Of Ms.C
Osmania University
Hyderabad, Telangana, India
hussein.alsalman85@gmail.com

T. Ramdas Naik
Assistant Professor (B.E, MCA, M.Tech, Ph.D)
Department Of IT Nizam College
Hyderabad, Telangana, India
ramdas_teja@gmail.com

Abstract-With cloud computing technology it is realized that data can be outsource and such data can also be shared among users of cloud. However, the data outsourced to cloud might be subjected to integrity problems due to the problems in the underlying hardware or software errors. Human errors also may contribute to the integrity problems. Many techniques came into existence in order to ensure data integrity. Most of the techniques have some sort of auditing. Public auditing schemes meant for data integrity of shared data might disclose confidential information. To overcome this problem, recently, Wang et al. proposed a novel approach that supports public auditing and also do not disclose confidential information. They exploited ring signatures that are used to compute verification metadata on the fly in order to audit the correctness of shared data. The public verifiers do not know the identity of the signer. It does mean that the verifier can verify data without knowing the identity of the signer. However, this scheme does not consider the freshness of data which is very important in cloud services. Obtaining latest copy of data is very important to avoid stale data access in cloud. Towards this end, in this paper, we proposed an algorithm for ensuring freshness of the data while retrieving the outsourced data in multi-user environment. Our empirical results revealed that the proposed algorithm is efficient.

Index Terms - *Cloud computing, data integrity, public auditing, and provable data freshness*

*****

I. INTRODUCTION

Distributed computing technologies enable multiple servers to work together. One such technology is known as cloud computing. This technology allows a pool of computing resources to be shared to users in pay per use fashion. With this technology, outsourcing data became common practices. In this context, the cloud users are worried about the integrity of data as the data is in an untrusted environment. Some of the examples for storage offerings in cloud computing include Google Drive, Drop Box and iCloud. There are many reasons for integrity problems in cloud computing including hardware problems, software problems and problems with human error [4]. The conventional approaches followed procedures that need entire data to be taken in order to verify integrity. The mechanisms used for security include MD5 [8] and RSA [7]. Since the cloud data is very huge, this approach is not feasible. Moreover, the verifier taking the actual data for checking integrity itself causes security issues again. Later on many techniques came into existence that allow a public verifier to check the integrity of data without the need for downloading entire data [17], [15], [13], [11], [10], [9].

The problem with the latest techniques is that they do not work for shared data in multi-user environment. Moreover, the techniques are prone to disclose confidential information to public verifier. Therefore a new technique was required in order to safeguard the confidentiality of cloud users. Towards this end recently Wang et al. [34] proposed a method that makes use of ring signatures that prevent the verifier to know the identity of the signer. Without knowing identity of the signer, the verifier will check the integrity of the outsourced data. In this paper we take this research forward. Since Wang et al. [34] did not focus on the data freshness; we proposed an algorithm that can take care of data freshness. Our work is based on the work done in [34].

Our contributions in this paper include the new algorithm proposed for data freshness and the prototype application that could demonstrate the proof of concept. The remainder of this paper is structured as follows. Section 2 provides review of literature. Section 3 presents the proposed system. Section 4 presents experimental results while section 5 provides conclusions and recommendations for future work.

II. RELATED WORK

This section provides review of literature that throws on prior works. In [9] public auditing was proposed which helped the public verifier to verify the integrity of data without gaining access to full data. Another such technique named Proofs of Retreivability (POR) [32] which does the same with slight difference as it makes use of the concept of sentinels. In [10], [27] and [33] some improvements are made to the existing techniques. BLS signatures are used in [27] while the [12] uses Merkle Hash Tree and BLS signature to support dynamic data in public auditing.

In [11] dynamic provable data possession (DPDP) was proposed that makes use of authenticated dictionaries. In [15] fragment structure is used in order to reduce signature storage as part of public auditing mechanism. To leverage existing mechanisms in [21] aggregate signatures concept was used. The correctness of eraser codes for data integrity was explored in [13]. In multi-server scenario auditing was carried out using network codes instead of eraser codes [14]. All these techniques improved the data integrity capabilities of cloud. Thus the cloud data owners are able to outsource their data confidently. In [16] LT odes-based security mechanism was explored. Recently Wang et al. [34] proposed a mechanism that helps public verifiers to check the data integrity without disclosing the identity of signers. To achieve this, they used ring signatures. In this paper our work is based on the work of

 ISSN: 2321-8169
3750 - 3754

[34] and we introduce an algorithm that takes care of data freshness.

III. PROPOSED SYSTEM

In this paper we proposed an algorithm for ensuring data freshness. This algorithm takes care of finding the type of data and applies the freshness logic to it. Only the latest data is considered and it avoids using stale data. However, our work is based on the work of [34] and the reader can get more concepts from that paper. Here our description is confined to the proposed algorithm. Figure 1 shows all the modules of the proposed system including the data freshness that has been introduced by us.

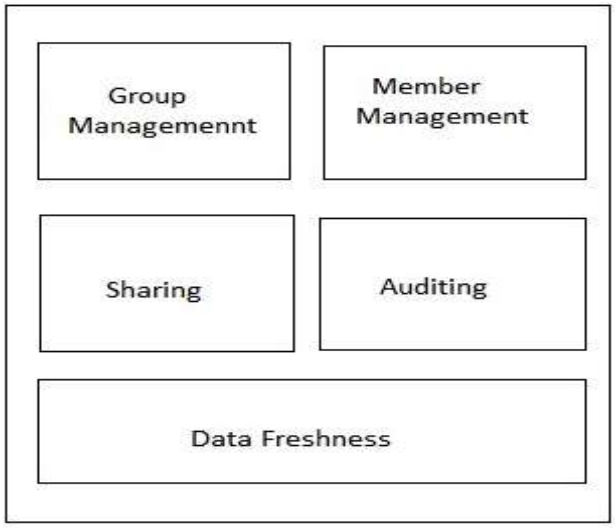

Figure 1 – Block diagram showing modules of the system

As can be seen in Figure 1, it is evident that there are five modules in the system. They include group management, member management, sharing, and auditing and data freshness. Group management module takes care of managing groups in cloud. It does mean that related members are grouped together. The member management module takes care of member dynamics. It includes adding new members, removing existing members and related operations. The sharing module takes care of sharing of data in multi-user environment. The auditing module takes care of the verification of data for integrity without disclosing the identity of the signer using ring signatures. More details on these four modules can be found in [34]. We focus more on the data freshness module as we proposed it and explored using a prototype application.

A. Freshness Verification Algorithm

In this paper we proposed an algorithm to verify freshness of data that can be applied to cloud environment. The algorithm takes cloud transactions as input and finds the freshness status. The pseudo code of the algorithm is as presented in listing 1.

Algorithm: Freshness Verification Algorithm

Input : Cloud Transactions ***T***
Output : Data Freshness Status ***s***
Process:
Finding Category of Data
1 Initialize ***dataCat*** to null
2 Initialize ***s*** to zero
3 Initialize ***h*** to null
4 For Each ***t*** in ***T***
    Find the discrimination in temporal domain
    Update ***h***
5 End For
6 If ***h*** reflects stable data Then
    dataCat = “sd”
7 Else if ***h*** reflects long term changing data Then
    dataCat = “lcd”
8 Else if ***h*** reflects long term changing data Then
    dataCat = “fcd”
9 End If
Finding Freshness
10 If ***dataCat*** = “sd”
    Do nothing
11 Else if ***dataCat*** = “lcd”
    Do nothing
12 Else if ***dataCat*** = “fcd”
13 For Each ***t*** in ***T***
    Apply concurrency metric
    Apply obsolescence metric
    Apply timeliness metric
    Find freshness rate
    Update ***s***
14 End For
15 End If
16 Return ***s***

Listing 1 – Pseudo code of the proposed algorithm

As can be seen in Listing 1, it is evident that the algorithm has two phases in it. First of all it finds the category of data. The category of data is of three types. They are stable data which will not be subjected to changes, long term changing data which is changed rarely and frequently changed data which is subjected to frequent changes. In the first phase the kind of data is computed and then the second phase stars. In the second phase, the actual freshness of the data is computed. There are many metrics applied to know the freshness. They are concurrency metric, obsolescence metric and timeliness metric. Afterwards the freshness rate is computed and finally the freshness status is returned by the algorithm. This algorithm can be used to ensure that the users in the group take the latest data.

IV. EXPERIMENTAL RESULTS

We built a prototype application using Java platform in order to demonstrate the proof of concept. The experiments are made in such way that they show the performance of all the modules. The results also reveal the experiments made using the data freshness module as well. The results reveal that the proposed mechanism is efficient.

ISSN: 2321-8169
3750 - 3754

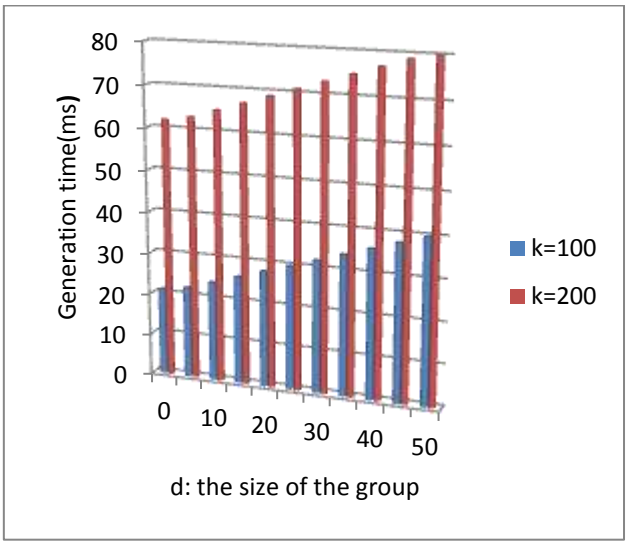

Figure 2 – The signature generation time vs. the size of the group

As shown in Figure 2, it is evident that the generation time for signature is more when value of k is more. It also reflects the increasing value of time taken for generating ring signatures when the size of the group is increased. Therefore it reveals the fact that there is impact of size of the group on the generation time.

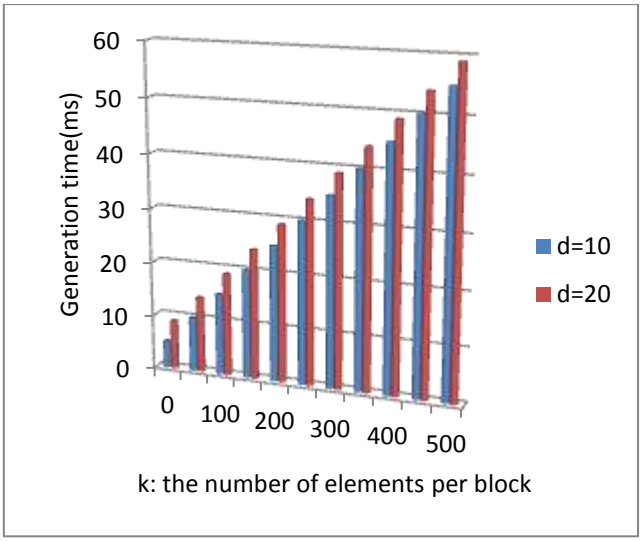

Figure 3 – The signature generation time vs. the number of elements per block

As shown in Figure 3, it is evident that the generation time for signature is more when value of d is more. It also reflects the increasing value of time taken for generating ring signatures when the size of the elements is increased. Therefore it reveals the fact that there is impact of size of the elements per block on the generation time.

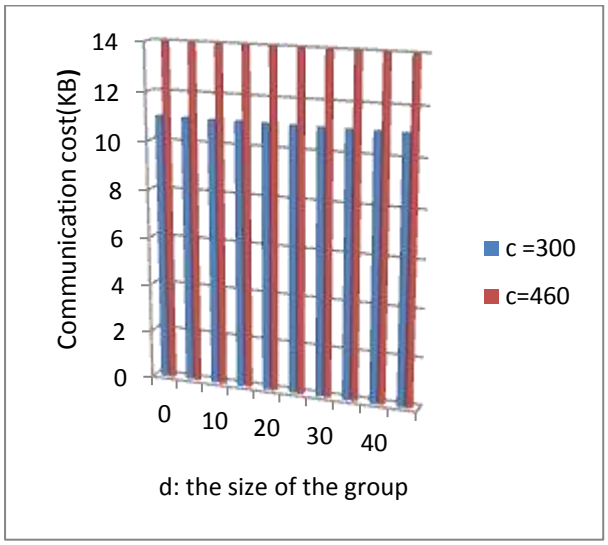

Figure 4 – The signature generation time vs. the size of the group

As shown in Figure 4, it is evident that the generation time for signature is more when value of c is more. It also reflects the increasing value of time taken for generating ring signatures when the size of the group is increased. Therefore it reveals the fact that there is impact of size of the group on the generation time.

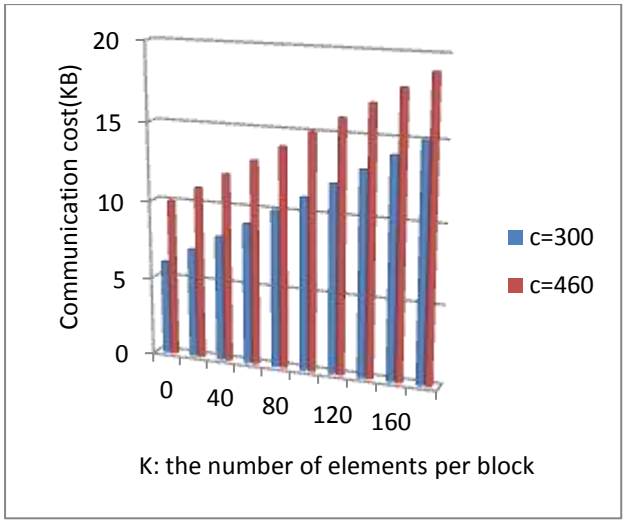

Figure 5 – The signature generation time vs. the number of elements per block

As shown in Figure 5, it is evident that the generation time for signature is more when value of c is more. It also reflects the increasing value of time taken for generating ring signatures when the size of number of elements per block is increased. Therefore it reveals the fact that there is impact of number of elements per block on the generation time.

 ISSN: 2321-8169
3750 - 3754

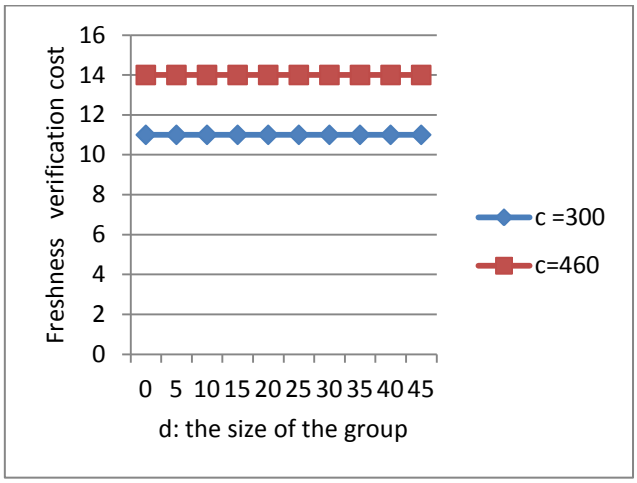

Figure 6 – Freshness verification cost vs. the size of the group

As shown in Figure 6, it is evident that the time taken for freshness verification is more when value of c is more. It also reflects the increasing value of time taken for freshness verification when the size of the group is increased. Therefore it reveals the fact that there is impact of size of the group on the signature verification cost.

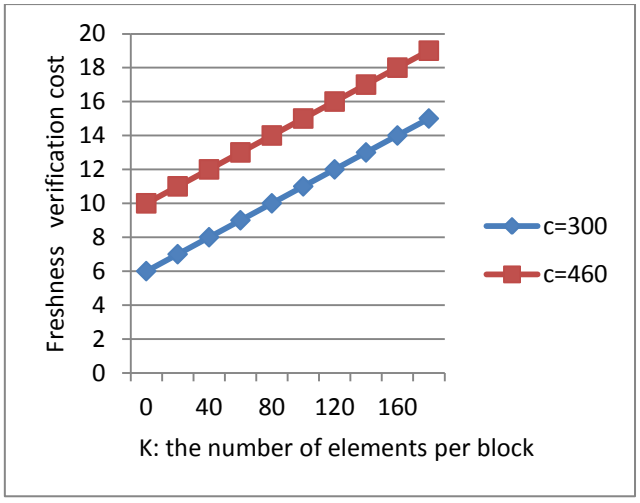

Figure 7 – The signature generation time vs. the size of the group

As shown in Figure 7, it is evident that the generation time for signature is more when value of c is more. It also reflects the increasing value of time taken for freshness verification when the size of the elements per block is increased. Therefore it reveals the fact that there is impact of size of the elements per block on the freshness verification cost.

V. CONCLUSION AND FUTURE WORK

In this paper we studied the need for data freshness in addition to the integrity of shared data that has been outsourced to cloud. Since the cloud is said to be untrusted, it is essential to ensure that the shared data has integrity. The integrity verification is done by third party verifier. However, the verifier does not know the true identity of the signer. Thus it ensures that the data is effectively verified for integrity and at the same time the confidential information is not disclosed. This research was carried out by Wang et al. [34] recently. However, their word does not consider the data freshness. In this paper we proposed an algorithm that takes care of data freshness in the multi-user environment. The shared data is verified by the verifier and the data freshness is also incorporated into the mechanism. We built a prototype application to demonstrate the proof of concept. The empirical results reveal that the scheme is very efficient. As future work, we focus on the data integrity of shared data in multi-cloud environment.

REFERENCES

[1] B. Wang, B. Li, and H. Li, "Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud," Proc. IEEE Fifth Int'l Conf.Cloud Computing, pp. 295-302, 2012.

[2] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A.Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A View of Cloud Computing," Comm. ACM,vol. 53, no. 4, pp. 50-58, Apr. 2010.

[3] K. Ren, C. Wang, and Q. Wang, "Security Challenges for the Public Cloud," IEEE Internet Computing, vol. 16, no. 1, pp. 69-73, 2012.

[4] D. Song, E. Shi, I. Fischer, and U. Shankar, "Cloud Data Protection for the Masses," Computer, vol. 45, no. 1, pp. 39-45, 2012.

[5] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing,"Proc. IEEE INFOCOM, pp. 525-533, 2010.

[6] B. Wang, M. Li, S.S. Chow, and H. Li, "Computing Encrypted Cloud Data Efficiently under Multiple Keys," Proc. IEEE Conf.Comm. and Network Security (CNS '13), pp. 90-99, 2013.

[7] R. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems," Comm. ACM,vol. 21, no. 2, pp. 120-126, 1978.

[8] The MD5 Message-Digest Algorithm (RFC1321). https://tools.ietf.org/html/rfc1321, 2014.

[9] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner,Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores," Proc. 14th ACM Conf. Computer and Comm.Security (CCS '07), pp. 598-610, 2007.

[10] H. Shacham and B. Waters, "Compact Proofs of Retrievability,"Proc. 14th Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT '08), pp. 90-107, 2008.

[11] C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia,"Dynamic Provable Data Possession," Proc. 16th ACM Conf. Computer and Comm. Security (CCS'09), pp. 213-222, 2009.

[12] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling Public Verifiability and Data Dynamic for Storage Security in Cloud Computing," Proc. 14th European Conf. Research in Computer Security (ESORICS'09), pp. 355-370, 2009.

[13] C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring Data Storage Security in Cloud Computing," Proc. 17th Int'l Workshop Quality of Service (IWQoS'09), pp. 1-9, 2009.

[14] B. Chen, R. Curtmola, G. Ateniese, and R. Burns, "Remote Data Checking for Network Coding-Based Distributed Storage Systems,"Proc. ACM Workshop Cloud Computing Security Workshop (CCSW'10), pp. 31-42, 2010.

[15] Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S.S Yau,"Dynamic Audit Services for Integrity Verification of Outsourced Storages in Clouds," Proc. ACM Symp. Applied Computing (SAC'11), pp. 1550-1557, 2011.

[16] N. Cao, S. Yu, Z. Yang, W. Lou, and Y.T. Hou, "LT Codes-Based Secure and Reliable Cloud Storage Service," Proc. IEEE INFOCOM,2012.
[17] B. Wang, B. Li, and H. Li, "Certificateless Public Auditing for Data Integrity in the Cloud," Proc. IEEE Conf. Comm. and Network Security (CNS'13), pp. 276-284, 2013.
[18] C. Wang, S.S. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Secure Cloud Storage," IEEE Trans.Computers, vol. 62, no. 2, pp. 362-375, Feb. 2013.
[19] B. Wang, B. Li, and H. Li, "Public Auditing for Shared Data with Efficient User Revocation in the Cloud," Proc. IEEE INFOCOM,pp. 2904-2912, 2013.
[20] B. Wang, B. Li, and H. Li, "Panda: Public Auditing for Shared Data with Efficient User Revocation in the Cloud," IEEE Trans.Services Computing, 20 Dec. 2013, DOI: 10.1109/TSC.2013.2295611.
[21] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and Verifiably Encrypted Signatures from Bilinear Maps," Proc. 22nd Int'l Conf. Theory and Applications of Cryptographic Techniques:Advances in Cryptology (EUROCRYPT'03), pp. 416-432, 2003.
[22] B. Wang, H. Li, and M. Li, "Privacy-Preserving Public Auditing for Shared Cloud Data Supporting Group Dynamics," Proc. IEEE Int'l Conf. Comm. (ICC'13), pp. 539-543, 2013.
[23] B. Wang, S.S. Chow, M. Li, and H. Li, "Storing Shared Data on the Cloud via Security-Mediator," Proc. IEEE 33rd Int'l Conf. Distributed Computing Systems (ICDCS'13), pp. 124-133, 2013.
[24] D. Boneh, X. Boyen, and H. Shacham, "Short Group Signatures,"Proc. 24th Ann. Int'l Cryptology Conf. (CRYPTO'04), pp. 41-55, 2004.
[25] B. Wang, B. Li, and H. Li, "Knox: Privacy-Preserving Auditing for Shared Data with Large Groups in the Cloud," Proc. 10th Int'l Conf. Applied Cryptography and Network Security (ACNS'12),pp. 507-525, June 2012.
[26] E. Brickell, J. Camenisch, and L. Chen, "Direct Anonymous Attestation,"Proc. 11th ACM Conf. Computer and Comm. Security (CCS'04), pp. 132-145, 2004.
[27] D. Boneh, B. Lynn, and H. Shacham, "Short Signatures from the Weil Pairing," Proc. Seventh Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT'01), pp. 514-532, 2001.
[28] R.L. Rivest, A. Shamir, and Y. Tauman, "How to Leak a Secret,"Proc. Seventh Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT'01),pp. 552-565, 2001.
[29] D. Cash, A. Kupcu, and D. Wichs, "Dynamic Proofs of Retrievability via Oblivious RAM," Proc. 32nd Ann. Int'l Conf. Theory and Applications of Cryptographic Techniques: Advances in Cryptology (EUROCRYPT), pp. 279-295, 2013.
[30] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving Secure, Scalable,and Fine-Grained Data Access Control in Cloud Computing,"Proc. IEEE INFOCOM, pp. 534-542, 2010.
[31] X. Liu, Y. Zhang, B. Wang, and J. Yan, "Mona: Secure Multi-Owner Data Sharing for Dynamic Groups in the Cloud," IEEE Trans. Parallel and Distributed Systems, vol. 24, no. 6, pp. 1182-1191,June 2013.
[32] A. Juels and B.S. Kaliski, "PORs: Proofs of Retrievability for Large Files," Proc. 14th ACM Conf. Computer and Comm. Security (CCS'07), pp. 584-597, 2007.
[33] G. Ateniese, R.D. Pietro, L.V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Data Possession," Proc. Fourth Int'l Conf.Security and Privacy in Comm. Networks (SecureComm'08), 2008.
[34] . Boyang Wang , Baochun Li, , and Hui Li,. (JANUARY-MARCH 2014). Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud.*IEEE*. 2 (1), p43-56.

***Authors*:**

Hussein Ali Ghadhban Salman
Department Of MS.C IS OSMANIA University,INDIA
Hussein.alsalman85@gmail.com

T. RAMDAS NAIK
Assistant professor (b.e, mca, m.tech, ph.d)
Department of it nizam college
ramdas_teja@gmail.com