International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2015): 6.391

Public Integrity Checking for Dynamic Data Sharing among Multi Groups

Athulya Prabhakaran[1], Nithya VP[2]

[1]Calicut University, M Dasan Institute of Technology, Ulliyeri, Kozhikode, Kerala, India

[2]Sri Krishna college of Technology, Anna University, Chennai, India

Abstract: ***Cloud is un-trusted file storage, and there is still a challenging issue, i.e., frequent change of the data in an un-trusted cloud. For that a number of techniques have been proposed for data integrity auditing to ensure user's confidence of their shared data on cloud. Most of the paper focus on the original data owner who will create the data can modify the data, and the other users have only the reading permission. To enhance this work to multi-user modification therefore any user (in group) can modify the data in the cloud. Main issues of multi-user modification is used revocation, here a user can revocate, just delete the user information no need to update the signature, because the data in depend to the user. Multi file auditing is also efficiently supported in our scheme by mechanism of batch auditing. So it reduces the computational/communication cost. This work enhanced to multiple groups to access the data. This paper concluded as, a efficient integrity auditing scheme for cloud data characterized by multi-user modification, public auditing, high error detection probability, efficient user revocation, resist user impersonation attack, Multi file auditing.***

Keywords: Public auditing, user revocation, cloud computing, Batch auditing.

1. Introduction

In recent years, the concept of third-party data warehousing and more generally, data outsourcing has become quite popular. Outsourcing of data essentially means that the data owner (client) moves its data to a third-party provider (server) which is supposed to – presumably for a fee – faithfully store the data and make it available to the owner (and perhaps others) on demand. Appealing features of outsourcing include reduced costs from savings in storage, maintenance and personnel as well as increased availability and transparent up-keep of data. Early work concentrated on data authentication and integrity, i.e., how to efficiently and securely ensure that the server returns correct and complete results in response to its clients' queries. Real-world examples (Dropbox [1], Sugarsync [2], Version Control Systems) are cloud-based storage synchronization which are multiple team members to work in sync, so execution of this kind of collaborative applications, made a problem i.e., assure data integrity. This scheme support the dynamic data operation so that each data modification operation is performed by an authorized group member and the data remains intact and update to date thereafter. This is an important problem but may be failure due to hardware/software failures, human errors and external malicious attacks [3], [4]. By using auditing process to ensure that the server stores all their latest data without any corruption.

Existing schemes [5]-[19] provide the data owner who holds secret keys can only modify the data. And all other users (who share data with the data owner only) have read permission. Here we extend these schemes to support multiple writers with data integrity assurance. But it leads to another problem i.e., the data owner has to stay online, collecting all modified data from other users and regenerating authentication tags for them. This may increase the tremendous workload to the data owner; especially in case of a large number of writers (users) and/or a high frequency of data modification operations are done.

When some paper can explained that all users share data with each other in cloud have both read and write privileges. But it creates several problems; i.e., scalability is limited by the group size and data size and not considered user revocation. If anther paper consider the user revocation [18] but it leads to non-trivial computational cost which is linear to the number of modifiers. And it also suffer limited in scalability due to the number of checking tasks. The concept of user revocation is based on the assumption that the cloud node responsible for updating signatures will not be compromised nor encounters internal errors. These internal errors may lead to the disclosure of user's secrets. It generate security issues.

In this work, focus on an efficient public integrity auditing scheme supporting multi-user modification, public auditing, high error detection probability, efficient user revocation, resist user impersonation attack, Multi file auditing. This novel design based on polynomial-based authentication tags, which can aggregate authentication tags from multiple writers into one when sending the integrity proof information to the verifier. By using this design, no matter how large the audited file is and how many writers are associated with the data blocks because a constant size of integrity proof information and a constant number of computational operations are needed for the verifier. Here we consider that to generate the keys for each data blocks so that no matter in the case of user revocation. In addition, our proposed scheme also allows aggregation of integrity auditing operations for multiple tasks (files) through our batch integrity auditing technique.

2. Literature Survey

To audit the data by access the entire file in past years, which is not feasible when it dealing with large amounts of data. So G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner,

Z. Peterson, and D. Song, proposed "Provable Data Possession at Untrusted Stores [6]". Here introduce a model for provable data possession (PDP) which able to verify that a server has retained file data without retrieving the data from the server. And also not need to be accesses the entire file from cloud. This model use the concept probabilistic proof , this proof generated by sampling random sets of blocks from the server. Therefore, to verify the data is corrupted or not just verify that proof not be verify the entire data from cloud. All other techniques must access the entire file to auditing. So by using this concept the client maintains a constant amount of metadata to verify the proof. Therefore this model drastically reduces I/O costs and minimizes network communication by challenge/response protocol transmits a small, constant amount of data. This model is very useful in case of large data set. This paper also constructs two provably-secure PDP schemes that are more efficient than previous solutions, even when compared with schemes that achieve weaker guarantees. In both schemes use homomorphic verifiable tags, ie, with homomorphic property, tags computed for multiple file blocks can be combined into a single value. So it will reduce the cost. Main process of PDP model client C pre-processes the file, generating a piece of metadata that stored locally, and transmits the file to the server S. Server stores the file and responds to challenges that are generated by the client(which file want to audit). Then server generates the proof of possession based on client challenge message. Then verify the file based on the proof. The client is thus convinced of data possession, without actually having to retrieve file blocks.

In previous paper i.e., Provable Data Possession (PDP) [6] explain how to frequently, efficiently and securely verify that a storage server is faithfully storing its client's (potentially very large) outsourced data. Not support additional features but by using "Scalable and Efficient Provable Data Possession [7]" introduced by Giuseppe Ateniese, Roberto Di Pietro, Luigi V. Mancini, and Gene Tsudik that overcome disadvantages of previous one. In this work, construct a highly efficient and provably secure PDP technique based entirely on symmetric key cryptography, while not requiring any bulk encryption. Also this paper allows outsourcing of dynamic data, i.e, it efficiently supports block modification, deletion and append. Before outsourcing the file, OWN pre-computes a certain number of short possession verification tokens, which covering some set of data blocks. Then actual data is outsourced to SRV. When OWN wants to obtain a proof of data possession, it challenges SRV with a set of random-looking block indices. SRV must compute a short integrity check over the specified blocks and return it to OWN. Then check the proof with corresponding value pre-computed by OWN.

By using the concept of Merkle Hash Tree (MHT) to achieve efficient data dynamics, to improve the Proof of Retrievability model as"Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing [10]" by Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou. The Merkle Hash Tree (MHT) construction for tag authentication for each block. For this scheme consider the task of allowing a third party auditor (TPA), on behalf of the cloud client, to verify the integrity of the dynamic data stored in the cloud. Here this work focused on studies the problem of ensuring the integrity of data storage in Cloud Computing. TPA can used for the behalf of the cloud and client, to verify the integrity of the dynamic data stored in the cloud. These paper achieve both ensuring remote data integrity often lacks the support of either public verifiability or dynamic data operations. Main advantages of this paper are the dynamics data operation, such as block modification, insertion and deletion. If any of the file block is modify, the computation overhead is unacceptable because the signatures of the file blocks should be re-computed with the new indexes. Solution of this limitation, we remove the index information by MHT so, individual data operation on any file block will not affect the others. The Merkle Hash Tree is a binary tree the leaves contain the hashes of authentic data values. And it is authentication structure, which is intended to efficiently and securely proves that a set of elements are undamaged and unaltered. In this paper we further employ MHT to authenticate both the values and the positions of data blocks. In our design, the leaf nodes as the left-to-right sequence, so any leaf node can be uniquely determined by following this sequence and the way of computing the root in MHT.

In previous work focused on TPA should be able to efficiently audit the cloud data storage without demanding the local copy of data, and introduce no additional on-line burden to the cloud user. This paper entirely different compare with previous in the case of privacy. "Privacy-preserving public auditing for data storage security in cloud computing [11] , " by C. Wang, Q. Wang, K. Ren, and W. Lou utilize and uniquely combine the public key based homomorphic authenticator with random masking to achieve the privacy-preserving public cloud data auditing system. In this design, propose to uniquely integrate the homomorphic authenticator with random masking technique. In this protocol, the linear combination of sampled blocks in the server's response is masked with randomness generated by a pseudo random function (PRF). By using random masking, the TPA no longer has all the necessary information to build up a correct group of linear equations and therefore cannot derive the user's data content, and no matter how many linear combinations of the same set of file blocks can be collected. Here, use public key based homomorphic authenticator specifically, the BLS based signature, to equip the auditing protocol with public auditability.

There are three processes comprised in audit service of this paper "Dynamic Audit Services for Integrity Verification of Outsourced Storages Clouds [12]" by D. Cash, A. Kp, and D. Wichs. Three processes are fragment structure, random sampling and index-hash table, can support provable updates to outsourced data, and timely abnormal detection. In this paper, audit service, constructed efficient approach based on probabilistic query and periodic verification for improving the performance of audit services. experiments showed that our solution has a small, constant amount of overhead, which minimizes computation and communication costs.

However, "Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud" by Boyang Wang, Baochun Li, public auditing for such shared data — while preserving identity privacy — remains to be an open challenge. In this

paper, propose the first privacy-preserving mechanism that allows public auditing on shared data stored in the cloud. In particular, we exploit ring signatures to compute the verification information needed to audit the integrity of shared data. With this mechanism, the identity of the signer on each block in shared data is kept private from a third party auditor (TPA), who is still able to publicly verify the integrity of shared data without retrieving the entire file.

In "Privacy-Preserving Public Auditing for Secure Cloud Storage Using Cloud Storage", proposed by Cong Wang, Sherman S.-M. Chow, Qian Wang, Kui Ren, and Wenjing Lou, users can remotely store their data in cloud. However, the fact that users no longer have physical possession of the outsourced data makes the data integrity protection, especially for users with constrained computing resources. Then, users should be able to just use the cloud storage as if it is local, without worrying about the need to verify its integrity. Thus, enabling public auditability for cloud storage is of critical importance so that users can resort to a third party auditor (TPA) to check the integrity of outsourced data and be worry-free. Here, introduce an effective TPA, the auditing process should bring in no new vulnerabilities towards user data privacy, and introduce no additional online burden to user. The proposed a secure cloud storage system supporting privacy-preserving public auditing. So extend this result to enable the TPA to perform audits for multiple users simultaneously and efficiently. Extensive security and performance analysis show the proposed schemes are provably secure and highly efficient. Extensive security and performance analysis show the proposed schemes are provably secure and highly efficient.

In this paper, we propose "Knox: Privacy-Preserving Auditing for Shared Data with Large Groups in the Cloud " by Boyang Wang, Baochun and Hui Li into a multi-user setting, where the TPA can perform multiple auditing tasks in a batch manner for better efficiency. Knox, a privacy-preserving auditing mechanism for data stored in the cloud and shared among a large number of users in a group. the identity of the signer on each block in shared data is kept private from the TPA. With Knox, the amount of information used for verification, as well as the time it takes to audit with it, are not affected by the number of users in the group. Moreover, we exploits homomorphic MACs to reduce the space used to store such verification information. Utilize group signature, a verifier is convinced that messages are correct and signed by one of the group members, but cannot reveal the identity of the signer. Meanwhile, only the group manager is able to trace these group signatures and reveal the identity of the signer. The efficiency of Knox is not affected by the number of users in the group. With the group manager's private key, the original user can efficiently add new users to the group and disclose the identities of signers on all blocks.

To design efficient user revocation proposed B. Wang, L. Baochun, and L. Hui "Public auditing for shared data with efficient user revocation in the cloud [18]" by. To ensure data integrity can be audited publicly, users need to compute signatures on all the blocks in shared data. Different blocks are signed by different users due to data modification performed by different users. Once a user is revoked from the group, the blocks, which were previously signed by this revoked user must be re-signed by an existing user. In previous use straightforward method, which allows an existing user to download the corresponding part of shared data and re-sign it during user revocation, is inefficient due to the large size of shared data in the cloud. We design a novel public auditing mechanism for the integrity of shared data with efficient user revocation. This model done by utilizing proxy re-signatures, we allow the cloud to re-sign blocks on behalf of existing users during user revocation, so that existing users do not need to download and re-sign blocks by themselves. In addition, a public verifier is always able to audit the integrity of shared data without retrieving the entire data from the cloud, even if some part of shared data has been re-signed by the cloud.

In this paper, we propose a novel public verification mechanism to audit the integrity of multi-owner data in an untrusted cloud by taking the advantage of multisignatures. Introduce "Efficient Public Verification on the Integrity of Multi-Owner Data in the Cloud" by BoyangWang, HuiLi, XuefengLiu, FenghuaLi, and XiaoqingLi. With our mechanism, the verification time and storage overhead of signatures on multi-owner data in the cloud are independent with the number of owners. Using the proposed multisignature scheme as a building block, we design an efficient public verification mechanism on the integrity of multi-owner data in an untrusted cloud. With our mechanism, each block in cloud data is signed by multiple owners and is attached with one multisignature.

3. Problem Definition

We know that the cloud server to be curious- but-honest and cloud server will follow some protocol, but it may lie to users about the corruption of their data stored in the cloud for preserve the reputation of its services. Therefore, this kind of error may occur many times, being it internationally or not. It may effect on data that stored on cloud by the data loss events claimed by users. In this situation, we consider the some factors that may impact data integrity: hardware/ software failures and operational errors of system administrator; illegitimate user or external adversaries that corrupt data stored on the cloud, revoked users who no longer have data access privilege they will try to illegally attack the stored data. But valid users are always allowed to modify data, and we assume that the users are always honest in the group (group members). Moreover, assume that, there is a secure communication channel (e.g., SSL) in between each pair of entities. The existing system considers only the data owner can modify the data, extend the scheme with multi-user i.e., multiple groups.

One of the main problems is user revocation, i.e., a user can revocate from a group, the master user want to update the signature. But our system need not want to update operation so this paper can reduce the computational cost of the updation. And existing system consider the key generation process done by the master user. So that when a user can join in the group, the master user wants to key generation. And

update operation of a signature in the user revocation also managed by master user. So this scheme reduces the workload of master user.

4. System Model

The system model consists of three different entities: the cloud server, third-party auditor (TPA), and a large number of group members (i.e., the employees) as illustrated in Fig. 1. Cloud server is operated by cloud service providers and the fundamental service provides by them as storage as a service (SaaS). That means the cloud server offers data storage of group users.

The TPA is able to publicly audit the integrity of shared data in the cloud for users. If TPA detects a data corruption he/she will report the error to group users.

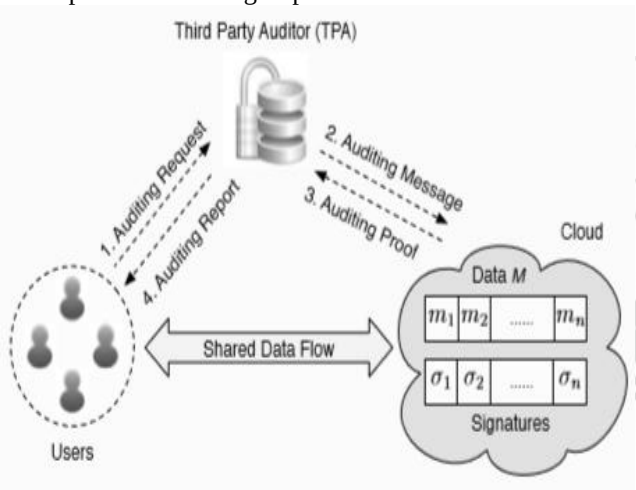


Figure 1: System model

In a group, there is one original user (master user) and a number of group users. The master user is the original owner of the data. This original user creates and shares data with other users in the group through the cloud and manages the membership of other group users. Both the original user and group users are able to access, download and modify shared data. As our proposed scheme, data can be uploaded/created by either the master user or other group users. Group members are the registered to join the group. In a example, the employee plays the role of group members. It allows the group members to be dynamically changed, due to the staff resignation and the participation of new employee in the organisation.

In our design data are stored in form of files, that file are further divided into a number of blocks. Each data block is attached with an authentication tag that is originally generated by the master user for integrity auditing,. If a user modifies a block, the user updates the corresponding authentication tag with his/her own secret key without contacting the master user.

5. Proposed System

Here, we consider detailed construction of our design. In this design, there are K users $u_k, 0 \le k \le K-1$ in a group sharing data stored on cloud and u_0 is the master user. And u_0 is also the owner of data and that having an ability to manage the membership of the group. So, u_0 can revoke any other group users when necessary. All users in the group can access and modify data stored on cloud. For simplicity of expression, we assume that the TPA performs the data integrity auditing procedure, who in practice can be any user knowing the public key.

Setup: Two procedures on setup algorithm i.e., Key Generation and File Processing. To setup the system, the master user u_o first runs the Key Generation part of the Setup algorithm and then generates public keys (PK), secret keys (SK) of each datablocks.

Algorithm: Setup $(1^{\lambda}, F) \rightarrow (PK, SK_k, \sigma_I)$

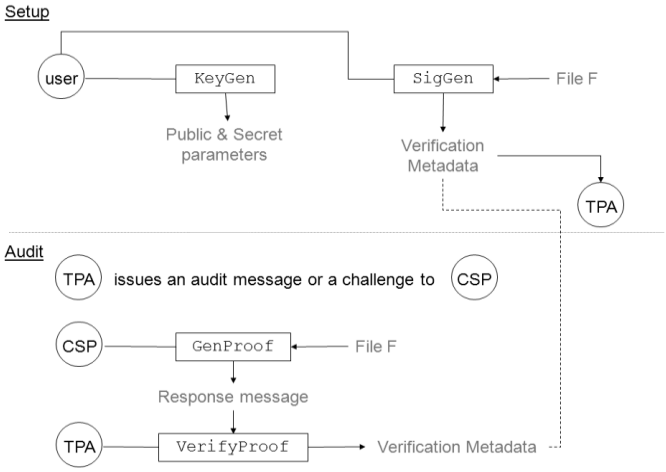
Key Generation: Select K random number $\varepsilon_k \leftarrow Z^*{}_q$ and generate $v = g^{\alpha \in_0}, k_0 = g^{\in_0}, \{k_k = g^{\in_k}, g^{\frac{\in_0}{\in_k}}\}$. Then randomly choose $\alpha \xleftarrow{R} Z^*{}_q$ and also generate $\{g^{\alpha^j}\}, 0 \le j \le s+1$. The public keys (PK) and secret keys (SK) of data blocks are:

$$PK = \{g, u, q, v, \{g^{\alpha^j}\} 0 \le j \le s+1, k_0 \{k_k, g^{\frac{\varepsilon_0}{\varepsilon_k}}\}\}$$
$$SK_k = \{\varepsilon_k\} 1 \le k \le K-1$$

File Processing: Divide the file F into n data blocks, and each block in to s elements: $\{m_{ij}\}$. Then generate authentication tag

$$\sigma_i = (u^{B_i} \cdot \prod_{j=0}^{s-1} g^{m_{ij}\alpha^{j+2}})^{\varepsilon_0} = (u^{B_i} \cdot g^{f_{\tilde{\beta}}(\alpha)})^{\varepsilon_0}$$

u_0 then uploads data blocks and tags to the cloud. And also u_0 publishes and maintains a Log for the file, which contains $\{i // t_i // k\}$ g information for each block.



Update: To allow group users to modify the shared data. A group user $u_{k,K\neq 0}$ modifies a data block m_i to m_i'. Then computes the tag for m_i' with data block secret key ε_k as:

$$\sigma_i' = (u^{B_i'} . \prod_{j=0}^{s-1} g^{m_{ij}' \alpha^{j+2}})^{\varepsilon_k} = (u^{B_i'} . g^{f_{\tilde{\beta}}(\alpha)})^{\varepsilon_k}$$

Challenge $(1^{\lambda}, PK) \rightarrow (CM = \{D, X, g^R, \mu\})$

1) Randomly choose d data blocks as a set D.

2) Chosen d blocks are modified by a users, denoted by C. Generate two random numbers R and μ and produce set $X = \{(g^{\frac{\varepsilon_0}{\varepsilon_k}})^R\}_{k\in C}$.

To audit data integrity, the TPA generates the challenge message $CM = \{D, X, g^R, \mu\}$ by running the Challenge algorithm. The TPA is aware of the set C used in the Challenge algorithm by looking at records in the log file. The TPA then sends the challenging message CM to the cloud.

Prove $(CM, F, PK) \rightarrow (\Pr f = \{\pi, \varphi.y\})$

1) Generate $\{pi = \mu^i \bmod q\}, i \in D$ and compute $y = f_{\tilde{A}}(\mu) \bmod q$.

2) Divide the polynomial $f_{\tilde{A}}(x) - f_{\tilde{A}}(\mu)$ with $(x - \mu)$ using polynomial long division.

3) Data blocks in D that were last-modified by set user $u_{k,} k \in C$, and compute π_i as

$$\pi_i = e(\sigma_i, g^{\frac{\varepsilon_0 R}{\varepsilon_k}}) = e((u^{B_i} . g^{f_{\tilde{\beta}_i}(\alpha)}) . g)^{\varepsilon_0 R}.$$

When receiving the challenging message $CM = \{D, X, g^R, \mu\}$, the cloud will run the Prove algorithm to generate the proof information $\Pr f = \{\pi, \varphi.y\}$ which shows that it actually store the challenged data file correctly.

Verify $(\Pr f, PK) \rightarrow (VerifyRst)$

1) Compute $\eta = u^{\omega}$, where $\omega = \sum_{i\in D} B_i p_i$.

2)Verify the integrity of file as $e(\eta, k^R{}_0) . e(\varphi^R, v.k_0^{-\mu}) = \pi . e(k^{-y}{}_o, g^R)$.

User Revocation

Actually user revocation is performed by the group manager. Because when a user can revocate the group, all authentication tags generated by revoked users are updated so that the revoked users' secret keys are removed from the tags. Depending on how many the tags were modified by the revoked users, these update operations can be potentially communication/computation intensive. To reduce the overloads of admin and decrease the communication/ computation intensive the signature process done automatically and give the signature all the data blocks on the cloud. So, if any users can revocate the signature is not depending on the user. Therefore there is no need to update the signature just need to delete the information about the revocated user.

Batch Auditing

In cloud, data blocks are frequently modifications by users in the group. So TPA can need to audit data integrity often to ensure that the data is not corrupted. Therefore TPA can auditing the integrity of file by file is inefficient in terms of both bandwidth consumption and computational cost. So it leads to multi-file auditing i.e., batch auditing. Batch auditing performed by, consider a set of T different files, it is desirable if the TPA can aggregate integrity auditing operations of T files into one challenge and one verification to reduce cost, where n_t is the number of data blocks in file F_t and s_t is the number of elements in each block. In this design a batch verification algorithm based on a single file auditing solution, and enables the TPA to handle integrity auditing of T files at the cost comparable to the single file scenario.

6. Conclusion

In this work, propose a novel data integrity auditing scheme that supports multiple writers with multiple group for cloud-based data sharing services. These schemes are characterized by important features of public integrity auditing and constant computational cost on the user side. In case of scalability, we further empower the cloud with the ability to aggregate authentication tags from multiple writers into one when sending the integrity proof information to the verifier (who may be general cloud users). And also, consider a constant size of integrity proof information needs to be transmitted to the verifier no matter how many data blocks are being checked and how many writers are associated with the data blocks. Our novel design allows efficient user revocation operations to the cloud. Because of keys generated for the data not a user. In addition scheme allows aggregation of integrity auditing operations for multiple tasks (files) through our batch integrity auditing technique.

.

References

[1] "Dropbox - business,"https://www.dropbox.com/business.
[2] "Sugarsync," https://www.sugarsync.com/business/.
[3] "Amazon EC2 and Amazon RDS Service disruption," http://aiweb.techfak.uni-bielefeld.de/content/bworld-robotcontrol-software/.
[4] "Dropbox Forums on Data Loss Topic," https://www.dropboxforum.com/hc/enus/search?utf8=%E2%9C%93& query=data+loss&commit=Search.
[5] A. Juels and B. S. Kaliski, Jr., "Pors: Proofs of retrievability for large files," in *Proceedings of the 14th ACM Conference on Computer and Communications Security*, ser. CCS '07. Alexandria, Virginia,USA: ACM, 2007, pp. 584–597.
[6] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in *Proceedings of the 14th ACM Conference on Computer and Communications Security*, ser. CCS '07. Alexandria, Virginia, USA: ACM, 2007, pp. 598–609.
[7] G. Ateniese, R. Di Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in *Proceedings of the 4th International Conference on Security and Privacy in Communication Netowrks*, ser. SecureComm '08. New York, NY, USA: ACM, 2008, pp. 9:1–9:10.
[8] H. Shacham and B. Waters, "Compact proofs of retrievability," in Proceedings of the 14th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology, ser. ASIACRYPT '08. Melbourne, Australia: Springer-Verlag, 2008, pp. 90–107.
[9] C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring data storage security in cloud computing," in *In Proceedings of the 17th IEEE International Workshop on Quality of Service*, ser. IWQoS'09, Charleston, South Carolina, July 2009.
[10] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in *Proceedings of the 14th European conference on Research in computer security*, Saint-Malo, France, 2009, pp. 355–370.
[11] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in *Proceedings of the 29th IEEE International Conference on Computer Communications*, ser. INFOCOM'10, San Diego, California, USA, 2010, pp. 525–533.
[12] 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," in *Proceedings of the 2011 ACM Symposium on Applied Computing*, ser. SAC '11. New York, NY, USA: ACM, 2011, pp.
[13] X. Jia and C. Ee-Chien, "Towards efficient proofs of retrievability," in *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, ser. ASIACCS '12, Seoul, Korea, 2012.
[14] H. Wang, "Proxy provable data possession in public clouds," *Services Computing, IEEE Transactions on*, vol. 6, no. 4, pp. 551–559, Oct 2013.
[15] B. Wang, B. Li, and H. Li, "Oruta: Privacy-preserving publicnauditing for shared data in the cloud," in *Proceedings of the 2012 IEEE Fifth International Conference on Cloud Computing*, ser. CLOUD '12, Washington, DC, USA, 2012, pp. 295–302.
[16] C. Wang, S. S. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy preserving public auditing for secure cloud storage," *IEEE Transactions on Computers*, vol. 62, no. 2, pp. 362–375, 2013.
[17] J. Yuan and S. Yu, "Proofs of retrievability with public verifiability and constant communication cost in cloud," in *Proceedings of the International Workshop on Security in Cloud Computing*, ser. Cloud Computing '13. Hangzhou, China: ACM, 2013, pp. 19–26.
[18] B. Wang, L. Baochun, and L. Hui, "Public auditing for shared data with efficient user revocation in the cloud," in *Proceedings of the 32nd IEEE International Conference on Computer Communications*, ser. INFOCOM '13, Turin, Italy, 2013, pp. 2904–2912.
[19] D. Cash, A. Kp, and D. Wichs, "Dynamic proofs of retrievability via oblivious ram," in *EUROCRYPT 2013*, ser. Lecture Notes in Computer Science, T. Johansson and P. Nguyen, Eds. Springer Berlin Heidelberg, 2013, vol. 7881, pp. 279–295.
[20] M. Bellare, J. Garay, and T. Rabin. Fast batch verification for modular exponentiation and digital signatures. In Proc. of EUROCRYPT '98, LNCS, pages 236–250, 1998.

Author Profile

Athulya prabhakaran is pursuing her M.Tech degree in Computer Science and Engineering from M.Dasan Institute of Technology, Ulliyeri, Calicut University. She obtained her B.Tech Degree in Computer Science and Engineering from Erode Anna University, in 2014.