A METADATA VERIFICATION SCHEME FOR DATA AUDITING IN CLOUD ENVIRONMENT

Muralikrishnan Ramane[1] and Bharath Elangovan[2]

[1]Department of Information Technology, University College of Engineering Villupuram, Villupuram, India
murali.itpro@gmail.com
[2]Department of Information Technology, University College of Engineering Villupuram, Villupuram, India
bharath.elan@gmail.com

ABSTRACT

Cloud, being the most vulnerable next generation architecture consists of two major design elements i.e. the cloud service provider and the client. The architecture shifts data, applications and development environments to large data centers thereby providing storage, software and platform services online. The notion on verifying data for intactness is termed as data auditing and the process is carried out by the client or by a third party auditor delegated by the client. The delegation model lessen client's overhead but may not be fully trustworthy since the third party has direct access to client's private data. The third party auditing brings in many new security challenges, and one among those is untrusted TPA. In the process of avoiding such an access this work designs a well secured, novel verification scheme that allows the vulnerable third party to perform verification as well as ensures data privacy in cloud.

KEYWORDS

Metadata Verification, Data Privacy, Cloud TPA, Public Auditability, Security Technique.

1. INTRODUCTION

Cloud [1,2,3] basically comprises of three schemes which provide services such as storage, platform and infrastructure denoted as SaaS, PaaS and IaaS respectively. These three play a vital role in the present day computing world. The era of cloud computing brought in tremendous changes on how data and its attributes are managed in distributed environments. Cloud was initiated to revolutionize online data storage concepts as an easier and cheaper alternative. However, this alternative has a created a backdoor in its architecture for enormous amounts of private data residing in cloud. This new data storage paradigm poses challenges regarding the integrity of the client's data.

Data Security [4,5,6] is an important research topic in cloud computing. Security in cloud can only be remotely implemented by the client since they do not have access to data centers and protocols in the system. The service provider must achieve two major security objectives; (1) confidentiality, for secure data access and transfer, and (2) auditability, for attesting whether security setting of applications has been tampered or not. Confidentiality is achieved using cryptographic protocols, whereas auditability is achieved using remote attestation techniques.

Cloud Service Provider holds huge amounts of private and critical data from diverse organizations and individuals who trusts [7,8,9] the storage provider on the privacy and integrity of the data. This is the cause that makes it the most vulnerable design element in the cloud architecture. On the other hand, the storage provider may experience hardware failures leading to data inconsistency and may decide to hide the errors from the owner of the data. Further, it may neglect consistency errors based on significance of the data and the client.

Data verification [10,11] is the fundamental function performed at the service provider to ensure the integrity of the client data. Several schemes [12] and security models are proposed to solve the problem of data integrity checking and some of them are Bilinear Aggregate Signature, Block Storage Integrity, Proofs of Retrievability, Pairing-Based Cryptography, and Public Verifiability. The Verifier is responsible for the efficiency of the verification process which basically is done using two scenarios i.e. Public and Private Verification, where the former allows anyone with authorization to challenge the service provider on the integrity of the data but, the latter achieves higher efficiency.

Public verifiability is the most preferred scheme as it does not necessitate the data owner to exploit private resources to verify the data. As in figure 1 public verification applies delegation approach where the verification function is performed by a Third Party Auditor. The TPA is an independent authority and it plays the role of verifier who informs the client on the integrity of his data. Normally the TPA has to act in favour of client by reporting any data inconsistency noticed in cloud storage. But, TPA being autonomous can be compromised by service providers on hiding errors or by intended misbehaviours (competing individuals or organizations) to disclose critical data.

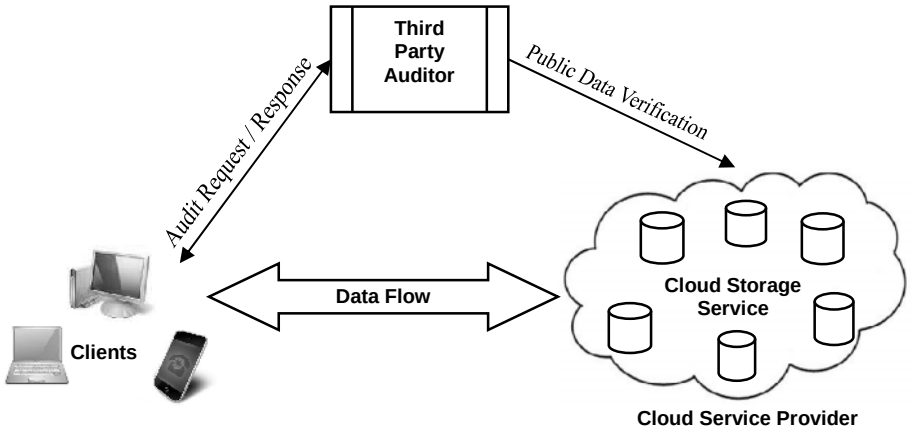

Figure 1. Third party data auditing setup in cloud

This paper contributes to public auditing schemes and enforces a novel security technique on data storage in Cloud. This paper proposes a metadata verification scheme where the TPA's have restricted access to the data thereby solving trust issues in the cloud architecture. The rest of this paper is structured as follows; Section 2 summarizes the related work on previous auditing methods. Section 3 introduces basic verification schemes. Section 4 proposes the idea for metadata verification scheme. Section 5 concludes the paper and suggests future work.

2. RELATED WORK

Data Privacy and Verification in cloud have been handled extensively in many existing works. On surveying the field of public auditability it is evident that considering the third party auditor as the vulnerable component is not addressed anywhere. The previous works do not address all the security threats and are all focusing on single server scenario. Most of them do not consider dynamic data operations and the problem of supporting both public auditability and dynamism have been recently addressed where the data is vulnerable in the hands of third party auditor. The following are some related papers in the field of public auditability in cloud.

2.1 Remote Data Possession at Untrusted Stores

This paper [13] states that cloud storage can achieve the goal that getting all storage resources in a plug-and-play way, it becomes a focus of attention. When users store their data in cloud storage, they mostly concern about whether the data is intact. This is the goal of remote data possession checking schemes. This paper proposes an efficient RDPC scheme which has several advantages as follows. First, it is efficient in terms of computation and communication. Second, it allows verification without the need for the challenger to compare against the original data. Third, it uses only small challenges and responses, and users need to store only two secret keys and several random numbers. Finally, a challenge updating method is proposed based on Euler's theorem.

2.2 Public Verifiability for Storage Security

This work [14] states that by data outsourcing, users can be relieved from the burden of local data storage and maintenance. It also eliminates their physical control of storage dependability and security, which traditionally has been expected by both enterprises and individuals. This unique paradigm brings about many new security challenges, which need to be clearly understood and resolved. This work studies the problem of ensuring the integrity of data storage in Cloud Computing. To ensure the correctness of data, we consider the task of allowing a third party auditor, on behalf of the cloud consumer, to verify the integrity of the data stored in the cloud. This scheme ensures that the storage at the client side is minimal which will be beneficial for thin clients.

2.3 Public Auditability for Storage Security

This paper [15] studies the problem of ensuring the integrity of data storage in Cloud Computing. It considers the task of allowing a third party auditor, to verify the integrity of the dynamic data stored in the cloud. This paper achieves both public auditability and dynamic data operations. It first identifies the difficulties and potential security problems of direct extensions with fully dynamic data updates from prior works and then shows how to construct an elegant verification scheme for the seamless integration of these two salient features in our protocol design. Extensive security and performance analysis show that the proposed schemes are highly efficient and provably secure.

2.4. Remote Data Checking Using Provable Data Possession

This paper [16] introduces a model for provable data possession that can be used for remote data checking. The model generates probabilistic proofs of possession by sampling random sets of blocks from the server, which drastically reduces I/O costs. The challenge/response protocol transmits a small, constant amount of data, which minimizes network communication. The model

is also robust and incorporates mechanisms for mitigating arbitrary amounts of data corruption. It presents two provably-secure PDP schemes that are more efficient than previous solutions. In particular, the overhead at the server is low (or even constant), as opposed to linear in the size of the data. It proposes a generic transformation that adds robustness to any remote data checking scheme based on spot checking. The paper conducts an in-depth experimental evaluation to study the tradeoffs in performance, security, and space overheads when adding robustness to a remote data checking scheme.

2.5. Privacy Preserving Data Integrity Checking

This paper [17] proposes protocols that allow a third-party auditor to periodically verify the data stored by a service and assist in returning the data intact to the customer. The protocols are privacy-preserving i.e. it never reveals the data contents to the auditor. This solution removes the burden of verification from the customer, alleviates both the customer's and storage service's fear of data leakage, and provides a method for independent arbitration of data retention contracts. The solution provides storage service accountability through independent, third-party auditing and arbitration. The protocols have three important operations, initialization, audit, and extraction, and it primarily focuses on the latter two. For audits, the auditor interacts with the service to check that the stored data is intact. For extraction, the auditor interacts with the service and customer to check that the data is intact and return it to the customer.

3. VERIFICATION

In previous works, authors have proposed several verification schemes where client directly gets involved in the verification process which is a very time consuming process and usually not preferred for huge volumes of data. Cloud clients then introduced a third party who is responsible for the verification process thereby delegating the authority to TPA's (Third Party Auditors). Cloud clients are blindfolded and do not have an idea on the operations carried out in the server side since they solely believe in third party verification of their private data. The scenario this work attempts to work on is to protect the privacy of the client from vulnerable TPA's. Clients, at some point of time when come across inconsistent data, just then they realize the data loss. So in order to avoid such critical situations this work brings in a novel verification scheme where TPA's have restricted access to client's data i.e. the metadata of whatever the client owns in the cloud.

3.1 Basic Scheme – I

This basic scheme as in figure 2 gives an overview of how a file is divided into blocks and the process of creating the Authentication code (AC). The Authentication code is generated at the user for each file block using a specific key for encryption. As the encryption process is completed the file blocks along with the codes are transferred to the cloud storage. Then the user shares the key with the third party auditor for future verification.

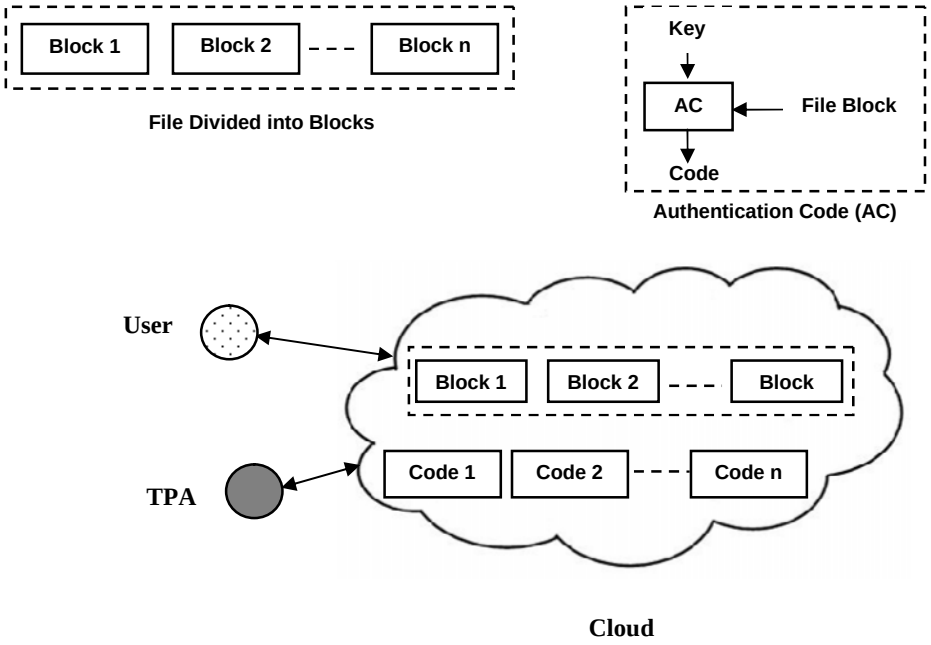

Figure 2. Basic Verification Scheme – I

When the user has the need to verify his data for intactness, the third party auditor is requested to proceed with the verification process. The TPA demands a random set of data blocks and their corresponding authentication codes from the service provider. The TPA uses the key that was shared at the time of encryption to decrypt and verify the data. So the security threat here needs to be addressed is key mishandling by TPA leading to unauthorized data access.

3.2 Basic Scheme – II

This scheme as in figure 3 is an enhancement to the previous scheme where data privacy is not preserved effectively. This scheme ensures that TPA is restricted from direct access to file blocks by sharing file blocks and corresponding codes. Here the user shares only the keys used for authentication and the codes generated. In this scheme user uses individual keys for each set of blocks and generates corresponding set of codes. Then user shares the keys and the authentication codes with TPA. In the auditing process the TPA sends a key to the CSP and requests the corresponding authentication codes. Once the TPA receives the codes it verifies them with the one it has for data intactness.

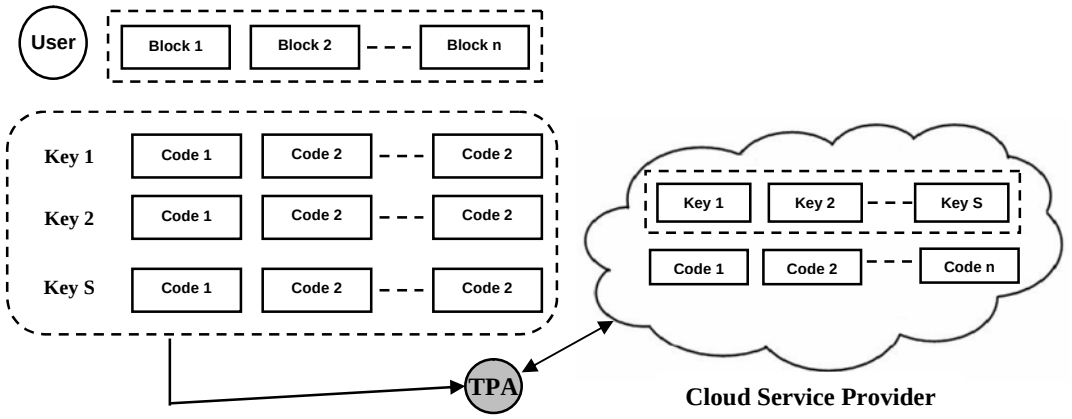

Figure 3. Basic Verification Scheme – II

The main drawback in this scheme is a key that is used once cannot be used for further encryption. Further the TPA has to keep a state remembering which key has been used previously. Both the schemes I and II are good for static data whereas it doesn't work in dynamic situations where data at the cloud is updated frequently.

4. METADATA VERIFICATION SCHEME

Public auditing schemes have achieved the support for dynamic data updates which is a critical need in environments like cloud where huge volumes of data are updated frequently. These verification schemes do not consider the effects on data privacy in the hands of a third party auditor. This work solves the issue of restricting the third party auditor from accessing the data openly. The following security model is designed for the above said purpose which gives access only to the metadata of the data being verified.

4.1 Security Model

The security model is basically designed as a data integrity scheme that supports integration of both public auditability and dynamic updates. The scheme verifies metadata rather than the actual data. The model is divided into two fundamental blocks,

i. Metadata Generation and
ii. Metadata Verification

4.1.1 Metadata Generation

The process as in figure 4 is initiated with the generation of a public key parameter P_k by the cloud client. Then the client generates a signature for individual file blocks. The signature is a form of metadata which is a combination of public key and file blocks and are called as codes. Finally the generated metadata is transmitted to the cloud storage.

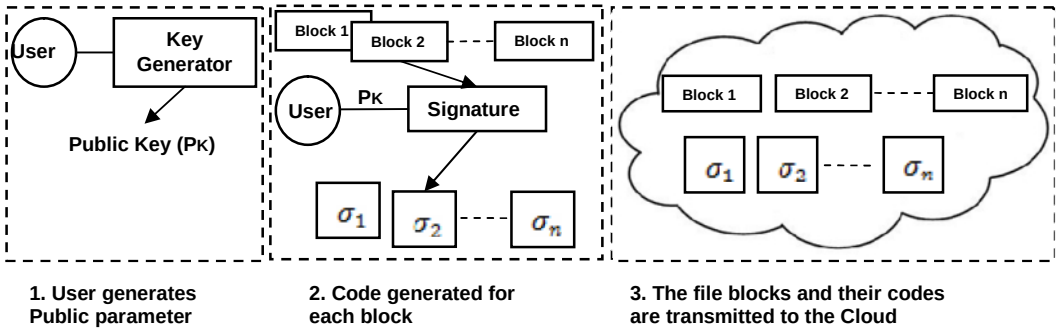

Figure 4. Metadata Generation

4.1.2 Metadata Verification

Once the metadata has been forwarded to the cloud, the TPA can perform data verification as in anytime represented in figure 5. When the TPA receives a request from the client for data verification, it sends an audit message to the service provider asking for a set of data blocks. The audit message contains the position of the blocks requested. The service provider makes a linear combination of blocks and applies a mask. The service provider sends the authenticator and masked blocks to the TPA. Finally the TPA compares the masked blocks from service provider and the metadata from the client.

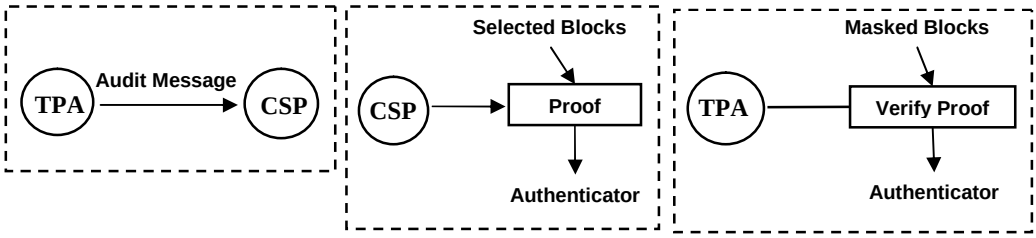

Figure 5. Metadata Verification

4.1.3 Assumptions and Equations

Let

P_k be the public key for encrypting the file blocks.

σ be the code generated for each block a.k.a metadata.

F be the actual file needs to be verified.

B be the single block of file.

Equation 1. File is divided into blocks

$$F \rightarrow \sum B1 + B2 + \cdots + Bn \quad \text{.....} \quad (1)$$

Equation 2. Code generated for each block

$$B_1, B_2, \ldots, B_n \xrightarrow{P_k} \sigma_1, \sigma_2, \ldots, \sigma_n \quad \text{.....} \quad (2)$$

Equation 3. Blocks and Codes are transferred to Cloud

$$B_1, B_2, \ldots, B_n \\ + \\ \sigma_1, \sigma_2, \ldots, \sigma_n \quad \Rightarrow \quad \text{CSP} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \textit{(3)}$$

Equation 4. TPA sends audit message to CSP

$$\text{TPA} \xrightarrow{\textit{Audit Message (File F, Pos)}} \text{CSP} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \textit{(4)}$$

Equation 5. CSP sends masked blocks TPA sends audit message to CSP

$$\text{CSP} \xrightarrow{\textit{Masked Blocks + Authenticator}} \text{TPA} \qquad \ldots\ldots\ldots\ldots\ldots\ldots \textit{(5)}$$

Equation 6. CSP sends masked blocks TPA sends audit message to CSP

$$\underset{\sigma_1, \sigma_2, \ldots, \sigma_n}{\textit{Client}} \quad \xleftrightarrow{\textit{metadata comparison}} \quad \underset{\sigma_1, \sigma_2, \ldots, \sigma_n}{\textit{CSP}} \qquad \ldots\ldots\ldots\ldots\ldots\ldots \textit{(6)}$$

4.1.4 Algorithm

Begin

metadata gen( )

Step 1: Initiate block splitting of the File F.

Step 2: Generate a public key P_k.

Step 3: Generate authentication codes for each block using the key.

Step 4: Transmit authentication codes along with file blocks to the cloud.

End

Begin

metadata verify( )

Step 5: Generate an audit message containing position of file blocks and send it to CSP.

Step 6: Forward the response message containing metadata of requested blocks to TPA.

Step 7: Compare metadata from CSP and Client.

End

Figure 6. Algorithm for Metadata Verification scheme

4.1.5 Operation

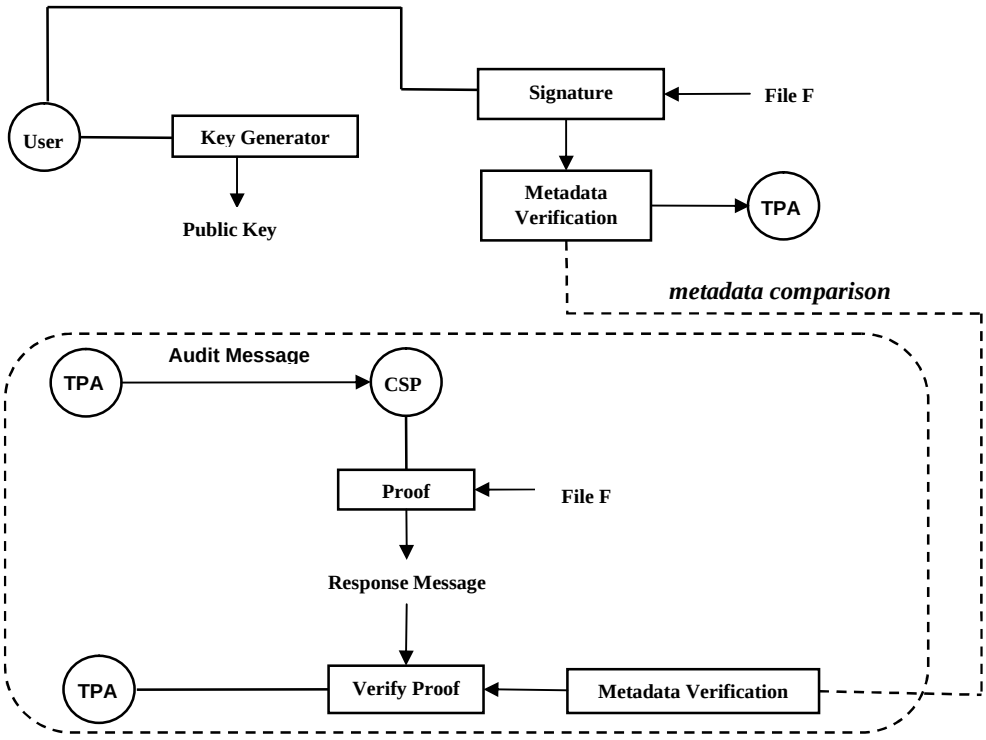

Figure 7. Metadata Verification – Operation

5. CONCLUSION & FUTURE ENHANCEMENTS

To conclude, the problem of trusting a third party auditor in verifying the data can effectively be handled by restricting the access to the owner's data. This metadata verification scheme is designed for such a purpose which restricts the third party to have access to the metadata of the data to be verified. The verification scheme can further be specialized using security protocols to check the auditor's reliability and confidentiality in handling the data and also can be checked for biasing. Further the data stored in cloud can be encrypted and the code generated for individual files can be sent over using secured transmission protocols.

REFERENCES

[1] Michael A, Armando F, et al., "A View of Cloud Computing," Communications of the ACM, Vol. 53, April 2010, pp.50-58.
[2] George Pallis, "Cloud Computing: The New Frontier of Internet Computing," IEEE Internet Computing, September-October 2010, pp.70-73.
[3] Rimal, B., et al., "A Taxonomy, Survey, and Issues of Cloud Computing Ecosystems," Springer, London, 2010, pp.21-46.
[4] Qi Zhang, Lu Cheng et al., "Cloud Computing: state-of-the-art and research challenges," Journal of Internet Services and Applications, Volume 1, Springer, 2010, pp.7-18.
[5] Daniele Catteddu, "Cloud Computing: Benefits, Risks and Recommendations for Information Security," Communications in Computer and Information Science, Vol. 72, Springer 2010.

[6] Sean C and Kevin C, "Cloud Computing Security, International Journal of Ambient Computing and Intelligence," Vol. 3, pp.38-46, April-June 2011, IGI Publishing.
[7] Paresh D Sharma, "A classification of distinct vulnerabilities in cloud computing," World Journal of Science and Technology, Vol. 2, 2012.
[8] Fugini M., "Security and trust in Cloud scenarios," In 1st International Workshop on Securing Services on the Cloud (IWSSC), September 2011, pp.22-29.
[9] R.K.L. Ko, B.S. Lee and S. Pearson, "Towards Achieving Accountability, Auditability and Trust in Cloud Computing," Proc. International workshop on Cloud Computing: Architecture, Algorithms and Applications, Springer, 2011, pp.5-18.
[10] Hassan T, James B.D. Joshi, et al., "Security and Privacy Challenges in Cloud Computing Environments," IEEE Security and Privacy, vol. 8, pp.24-31, Nov-Dec 2010.
[11]Pardeep S, Sandeep K. Sood, et al., "Security Issues in Cloud Computing," Communications in Computer and Information Science, Volume 169, 2011, pp.36-45.
[12] Pardeep K, Vivek K.S., et al., "Effective Ways of Secure, Private and Trusted Cloud Computing", International Journal of Computer Science Issues, Vol. 8, May 2011, pp.412-421.
[13] Lanxiang Chen, Gongde Guo, "An Efficient Remote Data Possession Checking in Cloud Storage," JDCTA: International Journal of Digital Content Technology and its Applications, Vol. 5, 2011, pp.43-50.
[14] Mihir R. Gohel and Bhavesh N. Gohil, "A New Data Integrity Checking Protocol with Public Verifiability in Cloud Storage," Advances in Information and Communication Technology, Volume 374, 2012, pp.240-246.
[15] Q. Wang, C. Wang, et al., "Enabling public auditability and data dynamics for storage security in cloud computing," IEEE Transactions on Parallel and Distributed Systems, vol. 22, 2011 pp.847–859.
[16] Ateniese G., Burns R., et al., "Remote data checking using provable data possession," ACM Transactions on Information and System Security, volume 14, 2011.
[17] Z Hao, S Zhong, and N Yu, "A privacy-preserving remote data integrity checking protocol with data dynamics and public verifiability," IEEE Transactions on Knowledge and Data Engineering, vol. 99, 2011.

ACKNOWLEDGEMENTS

The authors would like to thank all those who have shared their valuable inputs, especially our students **Sivaramakrishnan** and **Rilvana Begum**, for their insights, suggestions and time throughout the course of this work.

Authors

Muralikrishnan Ramane has received post graduate degree in the field of Computer Science Engineering, specialized in Distributed Computing Systems. He is currently working as a Lecturer in University College of Engineering Villupuram, Villupuram, India. To his credit he has 1 International Journal publication. His research interests include Data Management in Distributed Environments (Grid, cloud, and P2P).

Bharath Elangovan has received post graduate degree in the field of Computer Science Engineering. He is currently working as a Lecturer in University College of Engineering Villupuram, Villupuram, India. To his credit he has 1 IEEE conference publication. His research interests include Service Oriented Architecture in Cloud.