International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org, editorijettcs@gmail.com
Volume 3, Issue 4, July-August 2014 **ISSN 2278-6856**

Advanced Approach of Data Integrity Verification for Multicloud Storage

Vaibhav Bharati[1], Manisha Patil[2]
[1]PG Student, SKNCOE, Pune, University of Pune, Pune.

[2]Asst. Professor, SKNCOE, Pune, University of Pune, Pune.

Abstract

In the field of storage area, cloud computing is an emerging field which is raising its importance to large extent. Every user wants to store large capacity of data. Users want to acquire storage devices for storing this data. The capacity of the storage devices is the limitation parameter. And so the data is stored online or it is outsourced which makes the security of data vulnerable. In this paper the prime focus is on maintaining the data integrity and hence security. Security can be maintained by using three main parameters that are confidentiality, integrity and availability. Provable Data Possession (PDP) can handle these issues more effectively which is nothing but the proof given by service provider to the data owner according to demand. The project proposes an effective PDP model for verification of data integrity on distributed multicloud storage by using web-servers and Trusted Third Party. The use of techniques such as, Advanced Cooperative Provable Data Possession (Advanced CPDP), Homomorphic Verifiable Response and Hash Index Hierarchy for Multiprover Zero Knowledge Proof System (Multiprover-ZKPS) which is an Interactive Proof System are proposed.

Keywords: Advanced Cooperative Provable Data Possession, Cloud Service Provider, Multicloud storage, Web Server based Multicloud, Zero Knowledge Property.

1. INTRODUCTION

In this paper, the primary focus is on the maintenance of security of owner's data that is outsourced at the remote storage. Cloud is the most selected technology for outsourcing the data. Its service is based on the scheme of pay per use. When data owner stores data on the Cloud Service Provider (CSP), there can be a possibility of misusing stored data by CSP itself. There are also the techniques available in cryptography which prevents misuse of data but it provides minimum security. Security is based on three main pillars which are confidentiality, integrity and availability. If any of this parameter fails to achieve results, the data becomes vulnerable to the outside attack. In this model all the parameters mentioned above are assessed along with the detailed focus on the integrity. Confidentiality of the data can be retained by the authentication. The availability of it can be achieved by storing data redundantly on multiple cloud based web servers. In the storage, file data are large and remotely located. And so the time of accessing an entire file becomes expensive with I/O costs for data storing and transmitting data activities across a network. Other operations like reading an entire file periodically also limit the scalability of network storage. As discussed above, the main aim is to achieve integrity. In addition, I/O related to perform data possession involves with on-demand bandwidth for data storage and retrieval. PDP is the process that guarantees the consistency of outsourced data. It concludes with the necessity that, clients should verify consistency of the file data retained by the server without accessing or retrieving the whole data from the server. The basic principle of PDP is, data owner can challenge CSP to provide the proof and CSP provides it for the sake of guarantee of outsourced data consistency i.e. integrity of data. The challenge is nothing but the query containing some metadata i.e. credentials in the form of tags of the uploaded data calculated before outsourcing it. The CSP replies with the response containing proof of data possession to the data owner. Later on the data owner verifies the proof with the original credentials. If these credentials matches with each other then that indicate that the owner data is retained with consistency which leads to the satisfaction of the owner; failing results into the compromised data. PDP is used in many previous techniques with some variations in it. There are some enhancing results of the operation along with various parameters. This model introduces a parameter named as trusted third party (TTP) which reduces overhead and burden on the Data Owner and acts as the communicator between data owner and CSP. The rest of the sections discuss the related work, proposed model of Advanced CPDP, algorithm of Advanced CPDP, illustration of the system, Analysis and Conclusion along with Future scope.

2. LITERATURE SURVEY

2.1 Survey

Anteniese, *et al*. proposed provable data possession (PDP) model [4] which is defined for data possession which provides probabilistic proof for the file stored by third party. The model follows the expected requirement by allowing the server to access small portions of the file instead of accessing entire file. These small portions are used for generating proof for verification purpose. This model gives first secure scheme which is provable for remote data checking. The metadata stored at client side verifies proof given by server that is of constant complexity *O(1).* The bandwidth required for scheme is also of the order *O(1).* The challenge by the client and the response by the server are having size slightly more than *1 Kb*. There is also one more efficient version of this scheme that proves data possession using single modular exponentiation at server, but it provides a weaker guarantee. A. Juels, *et al*. proposed POR model [5] it states, before transmitting file on the remote storage

provider client encrypts it for archiving. This model makes challenge response protocols able to make bandwidth efficient; which gives probabilistic guarantee of the availability of file at remote storage provider. PORs use mainly the principle of spot-checking for adversarial detection through challenge-response protocol. Challenge by client contains the sampled file block's subset, whereas the response contains results of these block's computations which is returned to client by CSP. The returned response is checked using encrypted information embedded in the file. But, spot-checking is suitable only for large rate of adversarial corruption; the error correcting code (ECC) is used for making POR protocol resilient the small rate of adversarial corruption. Shacham and Waters [6] proposed Compact POR model which advances over the POR. This model works with two variations depending on supporting model. First, random oracle model which supports shortest query-response of any POR with the facility of public verifiability securely. Second, standard model which supports the shortest response of any POR but it needs longer query with private verifiability. It uses efficient homomorphism for authentication purpose. There are two variations secure POR scheme; first, gives by using pseudo random functions based on standard model and second, gives by using BLS signatures on random oracle model with public verifiability. This model supports the previous variations of POR like Juels - Kaliski model and provides the security against random adversaries. Techniques discussed above are useful for handling static storage only. Anteniese, *et al*. proposed Scalable PDP [7] supports dynamicity i.e. the operations performed on sampled file as block of data. This model entirely based on symmetric key cryptography and avoids any bulk encryption. This model gives assurance probabilistically as data stored on server in un-tampered form with same probability as of data being stored at the time of setup. This scheme provides security based on random oracle model. practicality of outsourcing data due to scalable PDP model, as it supports modification operations dynamically on stored data. Dynamicity helps for real time applications which need more interaction than just data warehousing. This scheme has some drawbacks like bandwidth-storage trade-off, limited number of verifications, computation overhead which limits the use. But, by compromising some parameters or allowing one of the trade-off performances of other parameters can be achieved. C. Erway, *et al.* enhances PDP model to give dynamic service on outsourced data updating online as a DPDP model [8]. It uses dictionaries which are authenticated and based on rank information. This scheme gives different variations of the model on the basis of service maintaining same data with better quality having varied performance complexity as from $O(1)$ to $O(\log n)$ for the file sampled into n blocks. The misbehavior of server for data is detected through this model and also slowdown is reduced with certain amount. It also supports version controlling of outsourced file systems. The rank based information is built by using skip list which yields into complexity of PDP up to log n without changing detection probability proposed as DPDP I. Also, by compromising complexity of computation it improves the probability of misbehavior detection which is proposed as DPDP II. Feifei Liu, *et al*. enhances DPDP model [8] discussed previously as Improved DPDP model [9]. It samples file data into blocks, generate tags for each block and computes hash values for each tag. Tags verify integrity of outsourced file, and hash values verify integrity of tags. There is markable improvement in complexity from log n to constant. It also provides operations like insert, edit and delete on unit blocks of an uploaded data.

2.2 Motivation

Cloud computing is the mechanism which provides service to the user which provides storage for outsourced data. There are CSP's to monitor service to the data owner. There are various techniques to store the data. One of the technique states that it can be stored on a single storage server. But there are situations like crashing of it which leads to loss of data. There may be possibility of corruption of data by indirectly or by mean. Also misuse can be done by the CSP itself; it can be prevented by outsourcing data on web server based multiple clouds. There is less or even not possible probability of compromization of different CSP's together to breach data. Also one additional actor mentioned in this model TTP who works as monitor of the cloud and performs the job as trustable communicator in system.

3 PROPOSED MATHEMATICS

3.1 Model

The system proposes scheme for outsourcing data by making use of multicloud storage. Data is stored on multiple clouds; this technique can be used for storing data redundantly so that lost or corrupted data can be prevented and retrieved from alternate server. Here web servers are used in the form of distributed file systems. CSP will provide space to the data owner after the request data is stored on allocated space by CSP, but the solo CSP cannot be a trustable one for the consistency of original data. As CSP is not fully trustable by data owner, the new actor trustable third party (TTP) is introduced in the model which facilitates mechanism to give guarantee of security for uploaded data. In this system, two algorithms are used; first is Multiprover ZKPS which gives completeness, knowledge soundness and zero-knowledge properties and second is CPDP which gives scalability for dynamic storage of data on multiple web-servers cooperatively based on HIH [3] and HVR [3] with features like high security, transparent verification, and high performance.

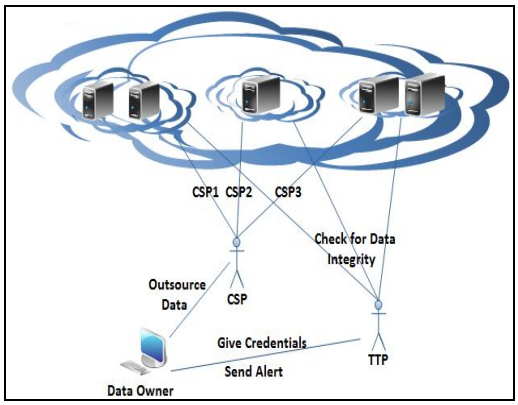


Figure 1 Architecture of Advanced CPDP Model [1]

Figure 1 shows architecture of Advanced CPDP model. Data Owner, CSP and TTP are three main actors of this model. Original data is partitioned into byte order form. Data Owner calculates credentials on partitioned data in the form of tags and stored them on TTP before outsourcing the data on central CSP. The HIH technique is used for maintaining integrity of tags. The whole data is then outsourced redundantly on multiple web server clouds by central CSP. Each cloud storage and security is maintained separately by separate CSP in the form of confidentiality, availability and integrity which is monitored by the TTP. TTP has whole access of clouds

Table 1: Symbol Table

Symbol	Representation
csp	Cloud Service Provider
c	Number of clouds to store a file
C	Set of csp's
n	Number of blocks in file
F	File with n blocks; i Є [1,n]
m_i	i^{th} block in file
T	Set of tags
H	Set of hash values
v	Set of number of times i^{th} block is modified
A	Set of alerts by TTP

data for keeping overlook on stored data; which prevents any modifications in the contents for security purpose. In this model the burden on data owner is reduced in large extent. Almost all activities required for verification purpose are done at TTP. When there is any modification in the outsourced data on clouds at any instant, TTP CHALLENGEs in the form of query. As TTP has full access of data, the RESPONSE is calculated which contains credentials as tags. According to this model data available on multiple clouds, each cloud generates different RESPONSE. So, all are in different forms. From these single homomorphic response is generated combining all responses by the technique of HVR. Initially stored credentials and calculated RESPONSE are compared. If both entity matches with each other then outsourced data is consistent, else there is a compromise in integrity of data. If integrity is lost alerts are generated to notify to data owner in the form of automated mail.

3.2 Set Theory

❖ **Data Sets**

- *Input:*

$C = \{csp_i\} \quad i \in [1, c]$
$F = \{ m_i \} \quad i \in [1,n]$

- **Intermediate Results**

$T = \{ T_i \} \quad i \in [1,n]$
$H = \{ h_i \} \quad i \in [1,n]$
$v = \{ v_i \} \quad i \in [1,n]$

- **Output**

- Alerts A ;If conflict in original and retrieved credentials
- Success Message

3.3 Planning of Advanced CPDP Model

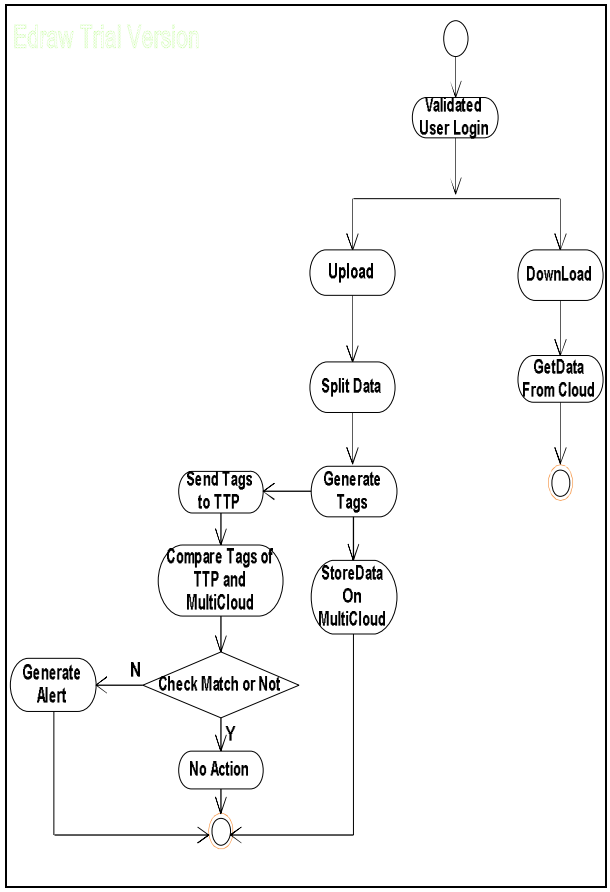


Figure 2 Execution of the Model

Figure 2 shows the overall idea of execution of Advanced CPDP model. Initially authorized user (or Data Owner) is allowed to use UPLOAD and DOWNLOAD facility. Authorized user has login details provided initially. When Data Owner wants to download the data directed towards download site containing contents only uploaded by own. The contents of other users are hidden. Downloading is single step process. In uploading data, the whole data is

partitioned into multiple partitions such a way that it can be stored redundantly. Redundancy is useful in the situations like crashing of one of the server; by which whole data can be retrieved by merging data from other servers. The user has to use service of central CSP to partition it into the blocks of byte orders. This byte ordered blocks are used for calculation of Tag values T. Tag values maintain the integrity of data whereas integrity of tag values is maintained by Hash values H. All such credentials i.e. sets of T and H are sent to trusted third party (TTP) before actually sending data to servers; which is intermediate trusted by both data owner and CSP. Whole data stored on different clouds of set C is under the vigilance of the TTP. It overlooks data throughout process for detecting any damage to it and maintaining the consistency of original data. When there are changes in data by some malicious activity are detected by TTP. Detection process consists of comparison between credentials stored on TTP and credentials calculated from response of CSP's. CSP's response is in heterogeneous forms as these are collected from multiple servers which are combined by HVR technique as shown in Step6 of Advanced CPDP Algorithm [1]. Whole verification process is as shown in the Step7 Advanced CPDP Algorithm [1]. If there is mismatch between two credentials that means integrity of data is compromised else data is consistent. Consistency lost results in notification to data owner with alerts.

3.4 Platform

The proposed model is being implementing on Windows 7 (Ultimate) OS. The platform for model implementation is .NET.

4 ILLUSTRATION

The Data Owner who wants to upload file on remote server has to submit it to service provider then credentials of file are calculated by system in the form of hash and tags as DIGEST. Uploaded file is partitioned into 3 blocks in this system at the central CSP. The partition number can be increased as with number of CSP's. The blocks are distributed on the three clouds. These are stored as block and it's DIGEST in database of cloud servers. In this system whole file is not stored at one location so that it cannot be targeted by attacker. In some previous systems it is seen that data can be compromised by CSP itself. But in this system as there are multiple CSP's chances of compromisation are negligible or can say impossible, because the storage is on block basis. There is no full access to whole file to an individual CSP. It has only one part of whole file and also it is in encrypted form. At the time of retrieval whole file is retrieved by combining all the three blocks together. Data from all clouds is fetched to provide whole file. When any block of file gets modified by any malicious activity, new actor TTP catches it. TTP visualize contents of file every time. When there are changes, comparison is done on original credentials and current credentials of file stored on it. If there is mismatch alerts are generated and sent to related user with detailed information containing filename and block number. The central backup server provides the facility of storing the copy of the original file in the encrypted form. And if a user wants to access this data then there is also a facility for recovering the changes done by malicious activity. All of the above steps are performed on the file having size 3.1 Mb. The file is uploaded in 260 mSec. The file is downloaded in 273 mSec. The signature for the credentials calculation is generated in 22 mSec. Also the size of the encrypted data is noted which seems to be incredibly negligible as compared to the actual file size. As the whole file is divided into three blocks each block has the size of 36 Kb and its digest of size is 12 Kb. Combining all the details the storage size becomes 144 Kb for the file of 3.1 Mb. By considering these calculations it can be concluded that the system has too much efficiency for the time and space complexity. The above tabular information represents the actual time required for (a) Uploading time, (b) Downloading time and (c) Signature Generation time along with the File Id, Username by whom file is uploaded, File name and File Size. Fig. 3 represents all the details for the previous CPDP Model [3] and Fig. 4 shows the current system Advanced CPDP Model [1]. Here the results of base system are generated by some observations and the formulae used in the base paper. From the above tables in Fig. 4 it is observed that, there is no relation between the size of the file and the required time for the specific action (a), (b) or (c), as it is dependent on the parameters such as network latency, availability of space and the speed of the local processor. Hence, the system performance is dependent on the unavoidable parameters.

5 PERFORMANCE ANALYSIS

Id	UserName	FileName	SigGenTime mS	FileSize 10^{-6}Mb
13	aksh	2 States.mp3	45.2	7072178
14	aksh	POST_OFFICE.MP3	46.3	10768384
15	aksh	ME_CLED_april2013.pdf	49.02	10870535

a)

Id	UserName	FileName	UploadingTime mS	FileSize 10^{-6}Mb
13	aksh	2 States.mp3	453	7072178
14	aksh	POST_OFFICE.MP3	437	10768384
15	aksh	ME_CLED_april2013.pdf	793	10870535

b)

Id	UserName	FileName	DownloadingTime mS	FileSize 10^{-6}Mb
13	aksh	2 States.mp3	216	7072178
14	aksh	POST_OFFICE.MP3	851	10768384
15	aksh	ME_CLED_april2013.pdf	290	10870535

c)

Figure 3 Database Values for the Size and Time of existing system.

Figure 5 contains the (a) Uploading file, (b) Downloading file and (c) Signature Generation graphs which are plotted file size against required time. Graphs in the Fig. 5 are plotted by considering the values of the tables in the Fig. 3 and Fig. 4. Blue colored line shows the current model [1] and red one shows the previous model [3]. Graphs may

show diverse behavior for the same file size, because it does not dependent on the deterministic parameters. From the above conclusion of the behavior of graphs the completeness of the system can be stated. As the behavior of systems is dependent on the non-deterministic parameters the system may fall into the NP. Even though failing to predict the actual required time, there is always the guarantee of the solution of operation. Hence, it can be understood that the system falls into the NP completeness.

Id	UserName	FileName	SigGenTime mS	FileSize 10^{-6}Mb
13	aksh	2 States.mp3	45	7072178
14	aksh	POST_OFFICE.MP3	46	10768384
15	aksh	ME_CLED_april2013.pdf	49	10870535

a)

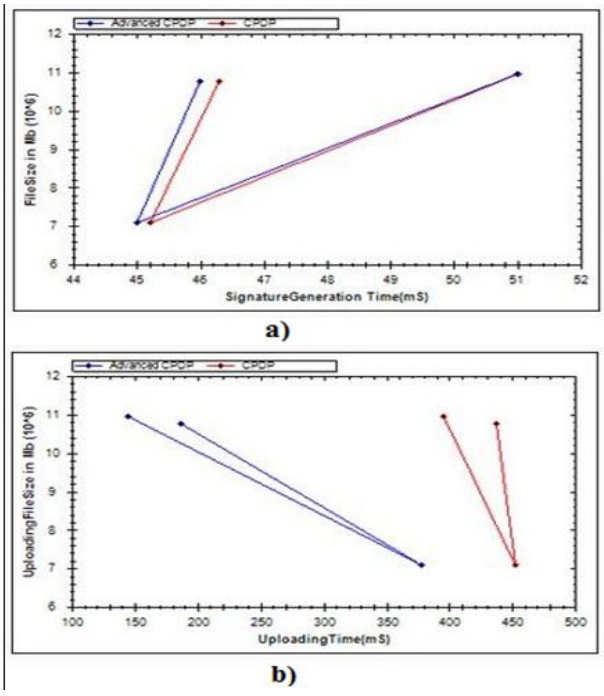
Id	UserName	FileName	UploadingTime mS	FileSize 10^{-6}Mb
13	aksh	2 States.mp3	378	7072178
14	aksh	POST_OFFICE.MP3	187	10768384
15	aksh	ME_CLED_april2013.pdf	293	10870535

b)

Id	UserName	FileName	DownloadingTime mS	FileSize 10^{-6}Mb
13	aksh	2 States.mp3	116	7072178
14	aksh	POST_OFFICE.MP3	626	10768384
15	aksh	ME_CLED_april2013.pdf	240	10870535

c)

Figure 4 Database Values for the Size and Time of Advanced CPDP Model.



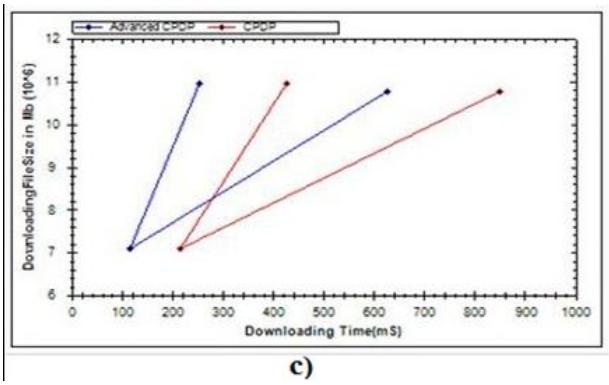


Figure 5 Graphs of File Size of operation against Time.

6 CONCLUSION

This paper proposes the model which is one of the effective PDP model for data integrity verification stored on distributed multicloud storage with using web-servers. This system works with the multiple private clouds by using principle of homomorphism with technique of homomorphic verifiable response and monitored by hash index hierarchy. This system proposes an Advanced CPDP model to give assurance to the data-owner by challenge-response protocol induced on the added new actor trusted third party; who provides the most effectiveness in the provability, confidentiality and the availability. This system can be a real time solution for the previous CPDP model [3]. It also reduces the burden on the data owner of the pre-processing. Hence, this system can be raised as some enhancement in the related systems till yet. This Advanced CPDP model can be enhanced to work on the hybrid clouds formed by combinations of private and public clouds.

References

[1.] Vaibhav Bharati, M. R. Patil, "Advanced Cooperative Provable Data Possession Based Data Integrity Verification For Multi-Cloud Storage", International Journal of Computer Applications, Vol 81-No.13, Nov. 2013.

[2.] Vaibhav Bharati, Manisha R. Patil, "Data Integrity Verification For Multi-Cloud Storage Based on Advanced Cooperative Provable Data Possession", Proceedings of Elsevier's 3rd International Conference on Recent Trends in Engineering & Technology (ICRTET'2014).

[3.] Yan Zhu, Hongxin Hu, Gail-Joon Ahn, Mengyang Yu, "Cooperative Provable Data Possession for Integrity Verification in Multicloud Storage", IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 23, NO. 12, DECEMBER 2012.

[4.] G. Ateniese, R.C. Burns, R. Curtmola, J. Herring, L. Kissner, Z.N.J. Peterson, and D.X. Song, "Provable Data Possession at Untrusted Stores," Proc. 14th ACM Conf. Computer and Comm. Security (CCS'07), pp. 598-609, 2007.

[5.] A. Juels and B.S.K. Jr., "Pors: Proofs of Retrievability for Large Files," Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), pp. 584-597, 2007.

[6.] 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.

[7.] 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. Netowrks (SecureComm '08), pp. 1-10, 2008.

[8.] C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession", In CCS '09, pp. 213-222, April 24,2012.

[9.] Feifei Liu, Davu Gu, Haining Lu,"An Improved Dynamic Provable Data Possession", Proceedings of IEEE CCIS2011, pp 290-295, 2011.

[10.]Zhifeng Xiao and Yang Xiao, "Security and Privacy in Cloud Computing", The University of Alabama, Tuscaloosa, 24 March 2012.

[11.]Venkatesa Kumar V, Poornima G, "Ensuring Data Integrity in Cloud Computing", Journal of Computer Applications ISSN: 0974 – 1925, Volume-5, Issue EICA2012-4, February 10, 2012.

[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," Proc. ACM Symp. Applied Computing, pp. 1550-1557, 2011.

[13.]Yashaswi Singh, Farah Kandah, Weiyi Zhang, "A secured cost effective multicloud storage in cloud computing", IEEE INFOCOM 2011 Workshop on Cloud Computing.

[14.]B. Sotomayor, R.S. Montero, I.M. Llorente, and I.T. Foster, "Virtual Infrastructure Management in Private and Hybrid Clouds," IEEE Internet Computing, vol. 13, no. 5, pp. 14-22, Sept. 2009.

[15.]Y. Zhu, H. Hu, G.-J. Ahn, Y. Han, and S. Chen, "Collaborative Integrity Verification in Hybrid Clouds," Proc. IEEE Conf. Seventh Int'l Conf. Collaborative Computing: Networking, Applications.

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

AUTHOR

Vaibhav Bharati received the B.E. degree from Pune Institute of Computer Technology (PICT), University of Pune in Computer Engineering in 2012 and pursuing M.E. in Computer Network from SKNCOE, Pune. He shave qualified the GATE-2012 with 92 percentile. He is currently working as an Asst. Professor in ZES's DCOER, Pune.

Manisha Patil is currently working as an Asst. Professor in SKNCOE, Pune.