ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016

Privacy-Preserving Simultaneous Public Auditing on Multi-Cloud Data Storage

Vilas C. Rathod, Prof. Angad Singh

M.Tech Student Dept. of I.T, NIIST Bhopal, Madhya Pradesh, India

Assistant Professor, Dept. of I.T, NIIST Bhopal, Madhya Pradesh, India

ABSTRACT: Cloud computing is Internet based processing which empowers sharing of services. Utilizing cloud storage, clients can remotely store their information and enjoy the on-interest excellent applications and services. However in cloud computing, since the data is put away anyplace over the globe, the client organization has less control over the stored data. To build the trust for the development of cloud computing the cloud suppliers must protect the client information from unauthorized access and disclosure. One technique could be encrypting the information on client side before putting away it in cloud storage; however this strategy has an excessive amount of trouble from client viewpoint as far as key management, maintenance perspective and so forth. Other way could be this kind of security service like figuring hash of data and verifying integrity of data, encryption/decryption service if gave by same cloud storage supplier, the data trade off can't be ruled out since same supplier has entry to both storage and security. A trusted third party cloud provider is used to give security services, while the other cloud supplier would be information storage provider. The trusted outsider security service provider would not store any data at its end, and it's only confined to providing security service. Thus, enabling public auditability for cloud storage is of discriminating significance so that client can resort to third-party auditor (TPA) to check the integrity of information which is outsourced and be effortless. The application will provide data integrity verification by using hashing algorithm, and encryption/decryption using algorithm like AES. This shows the proposed scheme is highly efficient and data modification attack /replace attack. In this system, TPA can perform multiple tasks auditing on multiple cloud servers in batch manner efficiently.

KEYWORDS: Multi cloud, TPA, Public auditing, Data integrity, Encryption, Hashing.

I. Introduction

Cloud computing allows sharing of computing resources to handle the applications. It also allows sharing of software and information. Cloud systems working on the basis of pay per use approach. Means customer have to pay for only what they uses. Cloud has several advantages including, high computational power, low service cost, good performance, accessibility, scalability and availability. It also reduces the expenditure of customer required for hardware, software, storage and many other services.

The switching from single cloud to multi cloud is very essential, because cloud computing with single cloud server has following limitations:

1.1.1 The data stored in the cloud storage, may modify by vulnerable entities. If the data is stored in single cloud, then all amount of data of users are loosed and it is very hectic to recover it. Therefore we can store the data on multiple servers in distributed manner, so that we reduce the whole loss of data possibility.

1.1.2 In case of single cloud, if password is hacked by unauthorized entities, all data may be loosed. Sometimes hacker may erase all data of users. Therefore it is better to store their data in multi cloud servers.

1.1.3 Another major concern in single cloud service is service availability.

There are also other challenging issues found in cloud computing, such as:

- Privacy
- Integrity
- Trust

DOI: 10.15680/IJIRCCE.2016. 0404255

- Access Control
- DoS

To overcome all these challenging issues, in this paper we working of multi cloud systems.

Cloud is used to store data by multiple users. This information is access by many other users. This data may be corrupted or modified by vulnerable entities. Cloud does not allowed users to access the data at physical level. User also unaware about the location of data storage. Therefore users has the big question about data integrity. Means whether the cloud service provider maintain data integrity or not. It is very important to secure the data store on cloud. To achieve this, the concept of Third Party Authority (TPA) is introduced in our system. TPA ensures that the data stores by users are not modified or altered by any other unauthorized entities.

TPA is nothing but the type of inspector. TPA having the particular resources to check the integrity and reducing the cost and load of users. The auditing results generated by TPA are valuable for users to ensure that their data is safe and for cloud to detect the unauthorized entities and improving performance.

TPA should take following points into consideration during the audit process:

1. The data for integrity checking should not be leak or access by other entities.
2. Perform audit process without downloading the data files.
3. Check data integrity with minimum communication overhead
4. TPA should be able to perform simultaneous auditing of multiple users data without bottleneck problem.
5. It can perform the auditing process on multi-cloud servers simultaneously.

Multi-cloud uses more than one cloud servers concurrently. It reducing the load of incoming data on single server and also protect the data from losing. In this paper we propose a system, which makes use of TPA for data integrity checking with hashing technique. It increases the security of data stored on multiple cloud servers. And handle the auditing of multiple user data files simultaneously. We also improve the security of data, by encrypting it before storing on cloud.

In this paper we will see: Section II presents the Literature survey of the existing systems then, we provide the detailed description of our proposed system in Section III. Section IV and V describes architectural view and implementation details respectively. Section VI presents Result and discussions, followed by Section VII that presents the Conclusion, at the end we have mentioned various references used in this paper.

II. LITERATURE SURVEY

Kai, He, et al. propose a public batch auditing protocol for multi-cloud storage in [1] for data integrity. In this, a TPA can verify the multiple auditing requests simultaneously from different users on multiple servers. To achieve this system makes use of homomorphic cipher text verification and recoverable coding approach. This system provides quick detection of corrupted data with minimum communication cost.

Public auditing of data distributed on multiple cloud servers is done with Aggregate Signature Scheme in [3]. In this, the identity of user and there data is not disclosed to third party auditor. System makes use of a Multi-clouds Database Model (MCDB) to implement traceability.

MuR-DPA, a public auditing system is proposed in [4]. System combines Authenticated Data Structure (ADS) with the Merkle Hash Tree (MHT). This system assigns the value of nodes in top-down order. It allows efficient verification of data files replicas. The MuR-DPA scheme has less communication overhead for data updation, verification and integrity checking. It provides more security against dishonest cloud service providers.

A novel public verification technique [5] to audit the data integrity of multiple users in suspicious cloud by using the concept of multi signatures. Data verification time and overhead of multi signature is depend on the data of multiple users.

eCloudIDS is a next-generation security framework proposed in [6]. This is a two tier hybrid architecture with uX-Engine and sX-Engine. This is a unique security solution applicable for any kind of cloud environments, such as public and hybrid cloud. This system enhance the identity management of Cloud at VM Level.

A.L. Ferrara et. al introduce a batch auditors including regular, identity-based, group, ring and aggregate signature schemes in [11]. This paper have implemented all this algorithms and compare batching algorithms. This paper identify that whether the batch auditing is practical or not.

To achieve efficient data dynamics, Q. Wang et. al enhance the proof of storage models in [12]. For block tag authentication system have made some changes in classic Merkle Hash Tree construction structure. System also includes the bilinear aggregate signature for simultaneous auditing of multiple user requests with TPA. This system is secure and more efficient.

Some systems provides data confidentiality using private key encryption techniques during TPA audit process. But some system does not consider data freshness property during development of new system. This disadvantage is overcome in [13] by implementing HMAC mechanism. This scheme is beneficial for metadata secrecy, integrity checking. This system randomly verify the data instead of whole data checking. This is the disadvantage of this system.

Chakraborty Et al implements the homomorphic encryption scheme for provide the data confidentiality [14]. This scheme is based on the Elliptic curve cryptography technique. System also implements a data possession scheme. It has provided dynamic operations on data. The third party auditor verifies and modifies the data on behalf of the client. Merkle hash tree (MHT) is used to fast access on data stored in cloud. This system checks the correctness of data and also identifies the misbehavior activities in server.

Yan et al have implemented a novel remote integrality checking approach in cloud computing [15]. It have efficient data integrity checking, dynamic update and data confidentiality. This system have checked the mass of files remotely with constant storage and resources of communication.

III.PROPOSED SYSTEM

We have proposed Privacy-Preserving Public Auditing of multiple cloud users on multi Cloud Storage in batch manner. To achieve this, we introduce the Third Party Authenticator (TPA) for public auditing of files store on multi-clouds. TPA performs data integrity checking efficiently and reducing the load of key management and file verification at individual cloud users.

1.2 Public Auditing by TPA

KeyGen: Executed by user for key generation. This key is use for file encryption.

SigGen: Executed by client to confirm the metadata with digital signature. It is used for authentication.

GenProof: Executed by multi cloud server. It will notify the users, once correct data is stored on server.

VerifyProof: Executed by TPA for batch manner public auditing. It will check the integrity of user data store on cloud by verifying the hash.

1.3 Design Characteristics

1.3.1 Verify the accuracy of data store on multi cloud servers without retrieving the whole copy of data that is preserving the data privacy.

1.3.2 TPA perform simultaneous public auditing of multiple users data.

1.3.3 Auditing is performed on multiple cloud servers instead of single cloud.

IV. ARCHITECTURAL VIEW

Overall view of proposed system is depicted in fig 1. This system is applicable for cloud service provider with multiple servers and multiple clients. System having three important components such as, Cloud Users, Cloud Service Provider (CSP) and Third Party Auditor (TPA). Users has large amount of multiple files to be stored on cloud server in distributed manner or on single server. CSP allows users to store their encrypted data files on their multi-cloud storage system. Finally, TPA performs the efficient auditing or verification of data files store in cloud server on receiving user requests. TPA is responsible for various activities including data verification and integrity checking, reducing the work load of users for verification etc.

Detail working of these three entities is explained bellow:

Cloud User: User has the choice to store their file either on single server or multiple servers. User can download these files whenever he wants to modify it. Using key generation algorithm, key is allocated for every user. The files are divided into number of blocks and encrypted using AES algorithm. Then at user side, the hash of each encrypted block is generated by using SHA1 algorithm. These encrypted blocks are uploaded on cloud servers and hash of each block is send to TPA.

Multi-cloud Server: Our Multi cloud server has four cloud servers. Each server has storage capacity to store large number of files. These servers are managed by cloud service provider.

Third Party Auditor (TPA): TPA is responsible for batch mannered-public auditing of data files stored on multi-servers, on receiving user requests. TPA performs auditing of large number of files simultaneously and on multiple cloud servers. It checks the data integrity of files. For this purpose, it challenging cloud server and in response cloud server provide hash of requested file or file block. Then this hash is compared with already received hash from users. If the hash is match, then it will notify user that the file stored on server is safe, it is not modified.

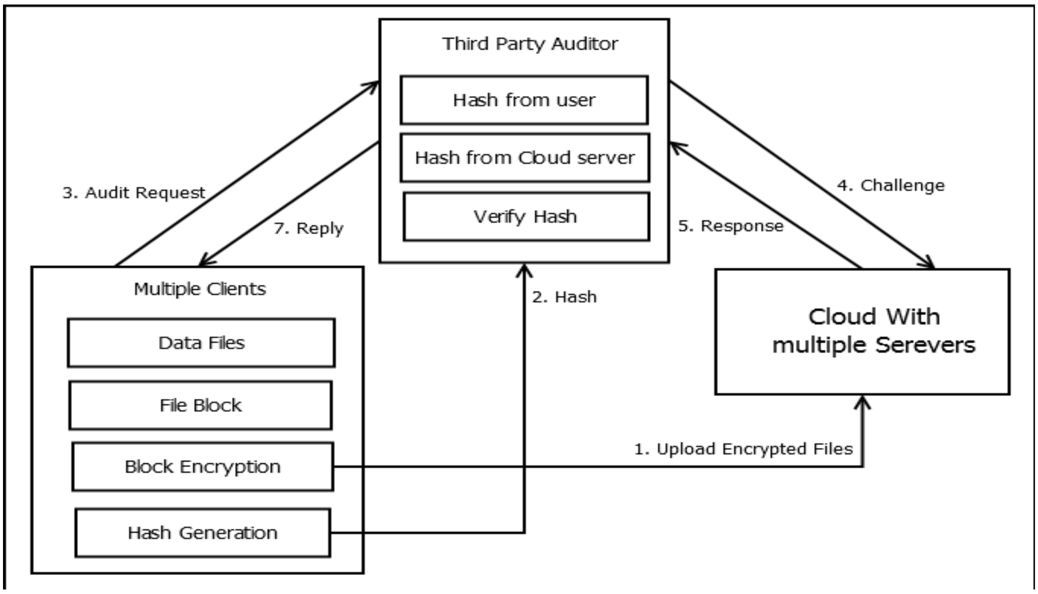

Fig.1System Architecture

V. IMPLEMENTATION DETAILS

A. Mathematical Model

Let, the system S is represented as: S = {Input, Process, and Output}

Input: Set of Text, Excel, Doc, PDF, & Image files

Output: File auditing

Process:

1. Set of users
   U = {u1, u2… un}
   Where, U is the set of cloud users, want to store their files on server.
2. Set of cloud server
   CS = {CS1, CS2, CS3, CS4}
   This system having multi-cloud structure with four cloud server.
3. Key generation
   K = {k1, k2… kn}
   Where, K is the set of keys generated using key generation algorithm, allocated for n number of users.
4. Signature Generation
   S = {s1, s2… sn}
   Where, S is the set of digital signatures, generated by users for authentication purpose.
5. File Divided into multiple blocks
   B= {B1, B2… Bn}

Where, B is the set of blocks of all files. These blocks will either store on single server or distributed on multiple servers.

6. Block Encryption
   BE = {BE1, BE2… BEn}
   Where, BE is the set of encrypted blocks store on cloud server
   Blocks are encrypted using AES algorithm.
7. Hash Generation at user side
   H = {h1, h2, …,hn}
   Where, H is the hash generated for each encrypted file block using SHA1 algorithm.
   These hash of all blocks are stored on TPA.
8. User requests for Auditing at TPA
   AR = {AR1, AR2, …,ARn}
   Where, AR is the set of audit requests from multiple users at TPA.
   TPA performs auditing of user requests simultaneously that is in batch manner.
9. TPA challenge to CS and get hash from server
   C = {c1, c2… cn}
   Where, C is the set of challenge made by TPA to CS.
   In return, CS provide hash of challenging file.
10. Comparing hash at TPA
    TPA compares the hash received from CS of particular file block with already received hash from user for that same file.If hash is match, notify user that, the file is not modified.

B. Algorithm

Algorithm 1: AES Algorithm

1. **Key Expansion:** Using Rijndael's key schedule Round keys are derived from the cipher key.
2. If DistanceToTree(u) > DistanceToTree(DCM) and First-Sending(u) then
3. **Initial Round:** AddRoundKey where each byte of the state is combined with the round key using bitwise XOR.
4. **Rounds**
   SubBytes: non-linear substitution step
   ShiftRows: transposition step
   MixColumns: mixing operation of each column. AddRoundKey
5. **Final Round:** It contain SubBytes, ShiftRows and AddRoundKey

Algorithm 2: Secure Hash Algorithm-1 (SHA-1)

Input: Encrypted file

Output: The contents in H0, H1, H2, H3, H4, and H5 are returned in sequence the message digest.

Process:

1. Appending Padding Bits.

The original message is padded (extended) so that its length (in bits) is congruent to 448, modulo 512. The padding rules are:

- The original message is always padded with one bit"1" first.
- Then zero or more bits "0" are padded to bring the length of the message up to 64 bits fewer than a multiple of 512.

2. Appending Length.

64 bits are appended to the end of the padded message to indicate the length of the original message in bytes. The rules of appending length are:

- The length of the original message in bytes is converted to its binary format of 64 bits. If overflow happens, only the low-order 64 bits are used.
- Break the 64-bit length into 2 words (32 bits each).
- The low-order word is appended first and followed by the high-order word.

3. Preparing Processing Functions.

SHA-1 requires 80 processing functions defined as:

DOI: 10.15680/IJIRCCE.2016. 0404255

f (t; B,C,D) = (B AND C) OR ((NOT B) AND D) (0 <= t <= 19)
f (t; B,C,D) = B XOR C XOR D (20 <= t <=39)
f (t; B,C,D) = (B AND C) OR (B AND D) OR (C AND D) (40 <= t <= 59)
f (t; B,C,D) = B XOR C XOR D (60 <= t <=79)

4. Preparing Processing Constants. SHA1 requires 80 processing constant words defined as:

K (t) = 0x5A827999 (0 <= t <= 19)
K (t) = 0x6ED9EBA1 (20 <= t <= 39)
K (t) = 0x8F1BBCDC (40 <= t <=59)
K (t) = 0xCA62C1D6 (60 <= t <=79)

5. Initializing Buffers.

SHA1 algorithm requires 5 word buffers with the following initial values:

H0 = 0x67452301
H1 = 0xEFCDAB89
H2 = 0x98BADCFE
H3 = 0x10325476
H4 = 0xC3D2E1F0

6. Processing Message in 512-bit Blocks.

This is the main task of SHA-1 algorithm, which loops through the padded and appended message in blocks of 512 bits each. For each input block, a number of operations are performed.

Algorithm 3: Proposed System Algorithm

Input: Text, Doc, Excel, PDF, & Image files

Output: File downloaded if not corrupted or modified

Process:

1. Get Input file
2. Select number of clouds on which file will be store in distributed manner
3. Key generation for each user
4. Divide file into multiple blocks
5. Encrypt each block with AES algorithm
6. Generate hash of encrypted block
7. Store hash on TPA
8. Generate digital signature and store metadata on cloud server
9. Audit and download request to TPA from user to check data integrity
10. TPA challenging to CS and get hash in response
11. Hash verification at TPA
12. If (hash matched)
13. Download file successfully
14. Else
15. Download failed and alert message for file corruption

VI. RESULT AND DISCUSSION

6.1 Experimental Setup

The system is built using Java (Version JDK 8) to evaluate the efficiency, effectiveness. The development tool used is NetBeans (Version 8). The experiments performed on Core2Duo Intel processor, 2GB RAM under Windows 7 ultimate. The system does not require any specific hardware to run; any standard machine is capable of running the application.

6.2 Dataset

System is evaluated with random number of txt, doc or excel, PDF & image files ranging from 1kb to 10mb.

6.3 Results

Fig. 2 shows that system is efficient for encryption of file blocks by using AES algorithm. AES generate small size of key for encryption; therefore proposed systems require minimum time for file encryption. This encrypted file will store on cloud server. X-axis represents the number of files with various sizes and their encryption time is representing on Y-axis. From above figure it is clear that, as file size increases, time required for its encryption is decreases.Fig. 3 represent the time require for uploading the file on cloud server. Uploading time of file is depending on the speed of network. In general, with AES scheme, original file size remains same after encrypting it. Therefore this proposed system with AES takes minimum time as compared to system with other encryption schemes. The X-axis represents the number of files with various sizes and their uploading time is representing on Y-axis.

Fig. 4 Shows that, the proposed system is more secure than existing system. Proposed system is more secure because it uses AES and SHA1 algorithm for data privacy and intigrity checking. One more important reason is that, it uses TPA for public auditing and hide the original data by providing a hash of file instead of original file. Fig. 5 depicts the comparision of system with separate auditing and batch auditing. Proposed system requires less time for auditing of files in batch manner. Separate auditing performs auditing of single file at a time and batch auditing perform auditing of multiple number of files at a time. Therefore overall performance of proposed system is better in terms of minimum time required for auditing of all files in batch manner.

ISSN(Online) : 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016

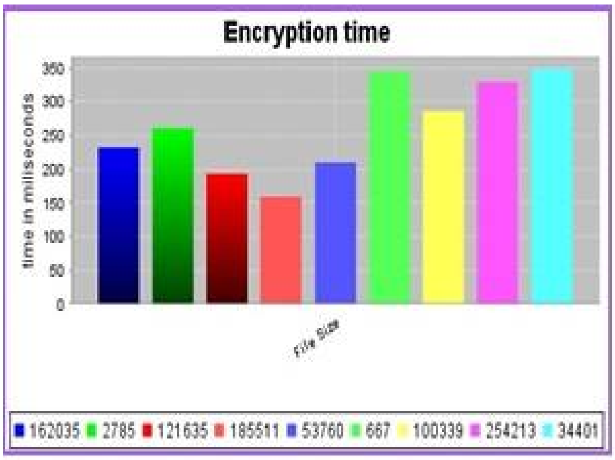

Fig.2 Block Encryption Time

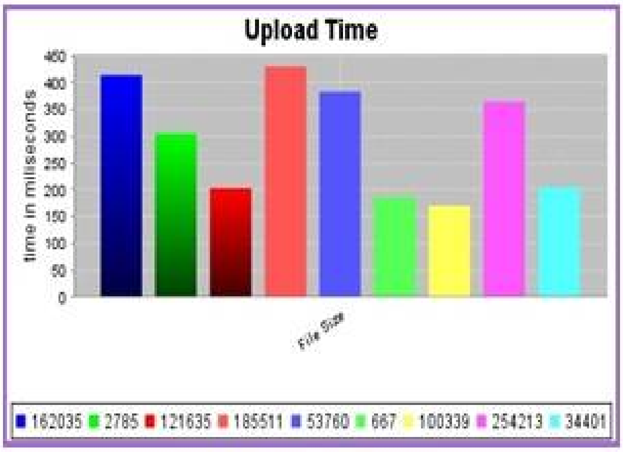

Fig.3 File Upload Time

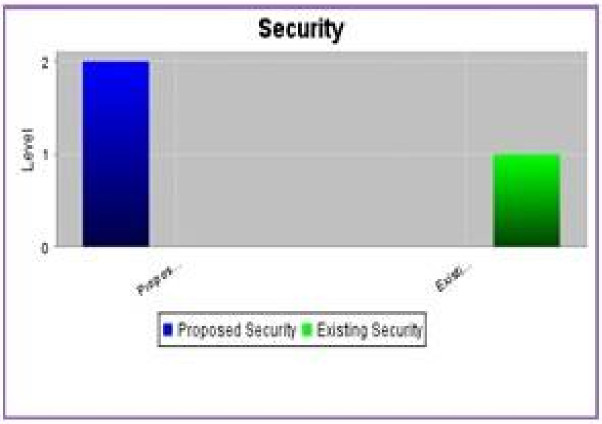

Fig. 3 Security Level

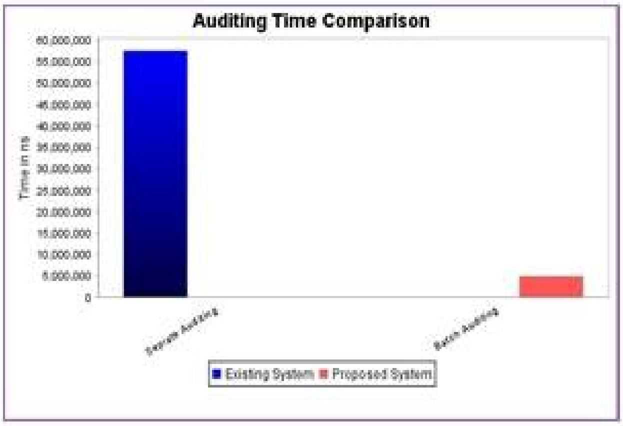

Fig. 4 Separate auditing Vs Batch auditing

VII. CONCLUSION AND FUTURE SCOPE

In this paper, we propose Privacy-Preserving Public Auditing for Secure multi cloud storage. This system simultaneously performs the auditing of multiple user requests. To provide the security of files, system encrypt the files using AES algorithm and then store on cloud server. For data integrity checking, TPA is introduced with the concept of SHA1 algorithm. TPA reduces the key management problem on client side. As system uses hash for integrity checking, there is no need to provide the original data file to the TPA for auditing purpose. Because of this the data confidentiality is increased. This will reducing the cost required for personal auditing process conducted by users themselves. Experimental results prove that the system is highly secure and efficient with TPA and reducing the cost of auditing at user side.

This system will be enhanced by allowing the uploading of image and multimedia data on cloud server.

VIII. ACKNOWLEDGMENTS

The authors would like to thank the researchers as well as publishers for making their resources available and teachers for their guidance. We are thankful to the authorities of RGPV University. We also thank the college authorities for providing the required infrastructure and support. Finally, we would like to extend a heartfelt gratitude to friends and family members.

REFERENCES

[1] Kai, He, et al. "An efficient public batch auditing protocol for data security in multi-cloud storage." *ChinaGrid Annual Conference (ChinaGrid), 2013 8th*. IEEE, 2013.

[2] Varghese, Lenny A., and Sayan Bose. "Integrity verification in multi cloud storage." *Advanced Computing (ICoAC), 2013 Fifth International Conference on*. IEEE, 2013.

[3] Suganthi, J., J. Ananthi, and S. Archana. "Privacy preservation and public auditing for cloud data using ASS in Multi-cloud." *Innovations in Information, Embedded and Communication Systems (ICIIECS), 2015 International Conference on*. IEEE, 2015.

[4] Liu, Chang, et al. "MuR-DPA: top-down levelled multi-replica Merkle Hash tree based secure public auditing for dynamic big data storage on cloud."*Computers, IEEE Transactions on* 64.9 (2015): 2609-2622.

[5] Wang, Boyang, et al. "Efficient public verification on the integrity of multi-owner data in the cloud." *Communications and Networks, Journal of* 16.6 (2014): 592-599.

[6] Srinivasan, Madhan Kumar, K. Sarukesi, and P. Revathy. "Architectural design for iCloudIDM Layer-II (iCloudIDM-LII) Subsystem of eCloudIDS generic security framework." *Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on*. IEEE, 2013.

[7] Cloud Security Alliance, "Top Threats to Cloud Computing," http://www.cloudsecurityalliance.org, 2010.

[8] P. Mell and T. Grance, "Draft NIST Working Definition of Cloud Computing,"http://csrc.nist.gov/groups /SNS/ cloudcomputing/ index.html, June 2009.

[9] Cloud Security Alliance, "Security Guidance for Critical Areas of Focus in Cloud Computing," http://www.cloudsecurityalliance. org, 2009.

[10] C. Wang, K. Ren, W. Lou, and J. Li, "Towards Publicly Auditable Secure Cloud Data Storage Services," IEEE Network Magazine, vol. 24, no. 4, pp. 19-24, July/Aug. 2010.

[10] A.L. Ferrara, M. Green, S. Hohenberger, and M. Pedersen, "Practical Short Signature Batch Verification," Proc. Cryptographers' Track at the RSA Conf. 2009 on Topics in Cryptology (CT-RSA), pp. 309-324, 2009.

[11] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing," IEEE Trans. Parallel and Distributed Systems, vol. 22, no. 5, pp. 847-859, May 2011.

[12] Priya, K., and I. Gunavathi. "Ensure cloud storage correctness based on public auditing mechanism." Communications and Signal Processing (ICCSP), 2015 International Conference on. IEEE, 2015.

[13] Chakraborty, TamalKanti, et al. "Enhanced public auditability & secure data storage in cloud computing." Advance Computing Conference (IACC), 2013 IEEE 3rd International. IEEE, 2013.

[15] Yan, Xiangtao, and Yifa Li. "A wew remote data integrity checking Scheme for cloud storage with privacy preserving." Communication Technology (ICCT), 2012 IEEE 14th International conference.

BIOGRAPHY

Mr. Vilas C. Rathod is M. Tech candidate in the Department of Information Technology at NRIIST, RGPV University, Bhopal, and Madhya Pradesh, India. He has achieved his Bachelor Degree in Information Technology from Pune University, Maharashtra, India in2009, He is currently a Lecturer with the Department of Information Technology, MIT Polytechnic, Pune, Maharashtra, India.