

Surveying Cloud Storage Correctness using TPA with BLS

Priyanka Dehariya¹, Prof. Shweta Shrivastava², Dr. Vineet Richaraya³

¹M.Tech Scholar (CSE), LNCT, Bhopal

²Asst.Professors, (CSE Department), LNCT, Bhopal

³HOD, (CSE Department), LNCT, Bhopal

pdeheriya@gmail.com

Abstract— In cloud computing, data is moved to a remotely located cloud server. Cloud storage in which data is stored on remote servers and is accessed by the user. Even though there is a benefit in accessing the data from the remote servers there raises security problem. After moving the data to the cloud data owner thinks that the data is secured and safe. But this hope may fail sometime because the data on the cloud may be altered, deleted or may be corrupted. For this scheme the user must download the data each time to validate it. But this scheme is very inefficient for the large data files. Thus by enabling public auditability user can resort to a third-party-auditor (TPA) to check the integrity of the outsourced data.

In this Paper we proposed a system which is based on Bi-linear pairing, which covers many recent proof of storage systems because existing system are based in HLA and HLA-based systems are not privacy preserving. Our main scheme is based on a specific BLS scheme SHA2 based solution. There we are going to present a new technique for the key generation scheme bases on bilinear pairing unlike BLS & it is assume to be proven best experimentally & to increase user experience. Also here we are using SHA2 in place of cryptography map to point hash function.

Keywords—Bi-linear pairing, BLS, Secure Hashing Algorithm-2, Public Auditing, TPA, Secure cloud authentication, Secure Data verification, IAAS & PAAS.

INTRODUCTION

You can put the page in this format as it is and do not change any of this properties. You can copy and past here and format accordingly to the default front. It will be easy and time consuming for you. This Cloud computing means store and access data and programs over the Internet instead of our computers hard drive. Cloud computing, often referred to as simply “the cloud,” is the delivery of on-demand computing resources—everything from applications to data centers—over the Internet on a pay-for-use basis. Cloud resources are usually not only shared by multiple users but are also dynamically reallocated per user’s demand. NIST [1] defines cloud computing by describing five essential characteristics, three cloud service models, and four cloud deployment models.

They are summarized in visual form in figure 1. Organizations use the Cloud in a variety of different service models (SaaS, PaaS, and IaaS) and deployment models (Private, Public, and Hybrid). As the user store data remotely to the cloud, it brings many benefits like user is free from the burden of storage management, hardware and software maintenance, location independency and has the ease of universal data access. But it also brings many challenges in the context of security of outsourced data.

As the data is not physically present in user’s storage we cannot adopt traditional cryptography methods to encrypt the data, downloading complete file only for verification is very difficult. If the outsourced data is very large in size the task of auditing is formidable and very expensive. All these problems can be solved by enabling public auditing. In public auditing user can rely on the third party for the verification of his outsourced data. Enabling public auditing leads to many rewards like:

1. Auditor checks the integrity of data periodically.
2. Save the cloud user’s computational resources and reduces online burden.
3. It is easier and affordable for the user to check the data’s integrity and correctness.
4. It also reduces the time consumption as we don’t need to download data to check its correctness. It can be verified on the cloud.
5. Using public auditing the user not suffer from the complexity in verifying data and hence it increases efficiency.

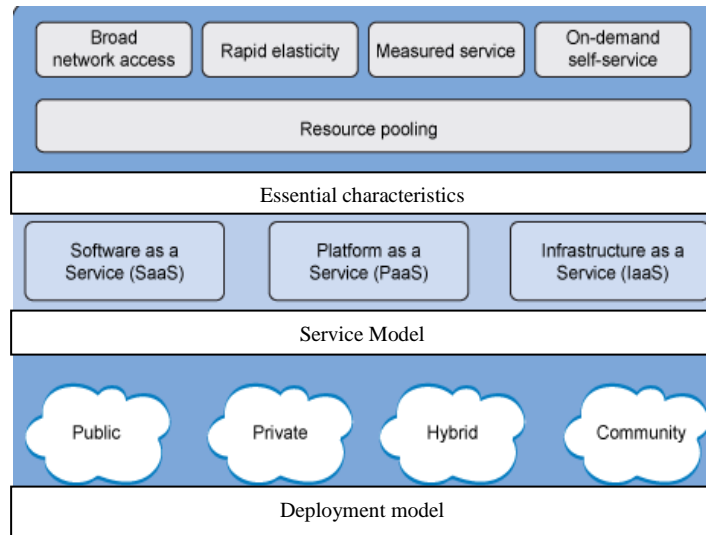


Fig.1. NIST Model of Cloud Computing

Most of the schemes described in paper [8],[9],[11],[12] had not consider the privacy of outsourced data in opposition to external auditors. Third party auditing provides cost effective method for uses. We assume that external auditor in the business of auditing is reliable but it may harm the user if it could learn the data in the audit. This severe drawback greatly affects the integrity and privacy of data. The data can be leaked and can be misused by the external auditors.

To address these problems we work on the techniques for achieving public auditing without the information leakage. Without learning the data contents the third party auditor has to perform auditing process.

PRELIMINARIES AND DEFINITION

This section contains introduction to the important terms used in our paper and their definitions. System model is also introduced in this section.

a) The System Model

We consider a cloud data storage service involving three different entities, as illustrated in Fig. 2:

1. *The cloud user*, who has large amount of data to be stored in the cloud;
2. *The cloud server*, which is managed by the cloud service provider to provide data storage services & resources;
3. *The third-party auditor*, who has the capabilities to check the integrity and correctness of the data stored in cloud.

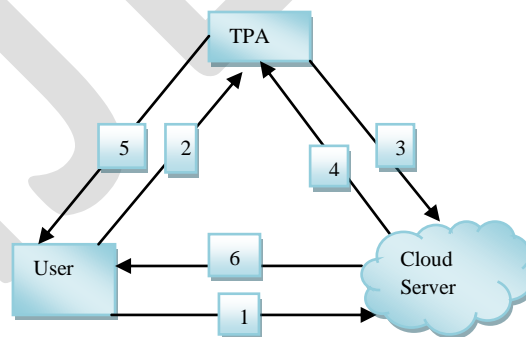


Fig 2.The architecture of cloud data storage service

Each steps mentioned above are consisting of different algorithm to perform the task. They are described as follows:

1. Data outsourcing: In this phase data to be outsourced is first encoded.

I. Key generation phase: First they generate some keys for coding and call encode function to create blocks of file and then perform the encoding the file

II. Signature generation phase: In this phase signature is generated of the coded data file.

Now user needs to send the data to the server with the verification metadata and deletes its local copy.

2. Verification: In this phase user sends the request for the verification of the outsourced data.

3. Public Verification (Challenge): In this phase TPA creates a challenge and sends the challenge to the cloud server and get back a corresponding response from the server.

4. Public Verification (Response): In this step server generate the proof of the challenge sent by TPA. And send back the proof as a response message to the TPA.

5. Verification Response: In this step TPA verify that the data block is damaged or not by calling the verification algorithm.

6. Data Extraction: If data is verified then user can extract the data from the server.

b) Bilinear Pairing

Pairing-based cryptography is the use of a pairing between elements of two cryptographic groups to a third group to construct cryptographic systems. Now we understand the bilinear mapping mathematically: Let G_1, G_2 , and G_t be multiplicative cyclic group of order p . Let g_1 and g_2 be the generator of G_1 and G_2 respectively. Then, a bilinear map e is a map where

$$e: G_1 \times G_2 \rightarrow G_t$$

Bilinearity: For all $u \in G_1, v \in G_2$ and $a, b \in \mathbb{Z}_p$,

$$e(u^a, v^b) = e(u, v)^{ab}.$$

Computability: There exists efficiently computable algorithm for computing map e .

c) Secure Hash Algorithm(SHA-2)

SHA-2 is a set of cryptographic hash function. We are using SHA-2 in our proposed scheme. The SHA-2 family mainly consists of six hash function: SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, SHA-512/256. SHA-256 has the longer hash value than SHA-1 and hence it is more secure than SHA-1. The SHA-256 Algorithm works on the following steps: Message is processed in 512-bit blocks sequentially, just like SHA-1. Message digest is 256 bits instead of SHA-1's 160-bits. 64 rounds instead of 80 rounds of compression

Algorithm structure same as SHA-1

- Step 1: Padding bits
- Step 2: Appending length as 64 bit unsigned
- Step 3: Buffer initiation
- Step 4: Processing of message
- Step 5: Output

We are using SHA-256 in our proposed scheme as it is more securing than other algorithm Here (in TABLE I) we have given a comparison among different hashing algorithm.

d) BLS Signature Scheme

There is a basic signature scheme known as the BLS scheme. It has the shortest length among signature schemes in classical cryptography. The scheme is based on Weil pairing and can be obtained from the private key extraction process of Boneh-Franklin's [11] ID-based encryption scheme. BLS short signature needs a special hash function. The signature scheme is consist of three steps:

- Key generation,
- Signing and
- Verification

COMARISION AMONG HASHING ALGORITHMS

S. No.	FACTORS	MD-5	SHA-0	SHA-1	(SHA-256)
1.	O/p size (bits)	128	160	160	256
2.	Internal state size (bits)	128	160	160	256
3.	Block size (bits)	512	512	512	512
4.	Max message size (bits)	$2^{64} - 1$	$2^{64} - 1$	$2^{64} - 1$	$2^{64} - 1$
5.	Word size (bits)	32	32	32	32
6.	Rounds	64	80	80	64
7.	Operations	+, And, Xor, Rot	+, and, or, xor, rot	+, and, or, xor, rot	+, and, or, xor, shr, rot
8.	Collision	Yes	Yes	Yes	Not yet

LITERATURE REVIEW

Recently, much of growing interest has been attended in the context of remotely stored data auditing [2], [6], [7], [8], [9], [10], [11], [12].

Ateniese et al. [9] are the first who considered the public auditability in their defined “provable data possession” model for ensuring possession of files on untrusted storages. In their scheme, they utilize homomorphic tags used for verification process and it is based on RSA with HLAs for auditing outsourced data, thus public auditability is achieved.

Disadvantage: In this research paper they had not considered the case of dynamic data, and the direct adoption of their scheme from static data to dynamic data suffered design as well as security problems.

In their subsequent work of “Scalable and Efficient Provable Data Possession” [10], Ateniese et al. propose a extended and dynamic version of the prior PDP scheme. They only extend their previous scheme in their next paper.

Disadvantage: However, their system is dynamic still their system demands a priori bound on the number of queries. It also does not support fully dynamic data operations, that is, it allows basic block operations with limited functionality.

In [11], authors considered the dynamic data storage with a distributed scenario. Here Wand et al. proposed a challenge-response protocol by which they can determine both the data correctness and locate possible errors.

Disadvantage: Similar to [10], they only considered partial support for dynamic data operation.

In [8] authors describe the “proof of retrievability” (PoR) model. According to the Juels et al. in “Pors: Proofs of Retrievability for Large Files” model spot-checking and error-correcting codes are used. They are used to ensure both possession and retrievability of data files on remote archive service systems. A POR is a protocol in which a server proves to a client that a target file F is intact or unchanged. In a native POR, a client might simply download F itself and check an accompanying digital signature.

Disadvantage: However, the number of audit challenges a user performs is fixed a priori. And hence public auditability is not achieved in their main scheme. And this approach only works with encrypted data.

Shacham and Waters [6] design an improved PoR scheme with full proofs of security in the security model defined in [8]. They use publicly verifiable homomorphic authenticators or HLAs built with BLS signatures [14], based on which the proofs can be aggregated into a small authenticator value, and public retrievability is achieved.

Erway et al. were the first to explore constructions for dynamic provable data possession. In their work [12] they extend the PDP model in [9] to support provable updates for stored data files using rank-based authenticated skip lists. This scheme is a fully dynamic version of the PDP solution.

In the paper "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing" Qian Wang et al.[7] particularly try to achieve efficient data dynamics, they improve the existing proof of storage models by manipulating the Merkle Hash Tree construction for block tag authentication.. To efficiently handle multiple auditing tasks, they explore the technique of bilinear aggregate signature to extend their main result. Here TPA can perform multiple auditing tasks simultaneously.

Boyang Wang et al. [2] proposed Oruta, a privacy-preserving public auditing technique for shared data. They use ring signatures to construct homomorphic authenticators, so that a public verifier is able to audit shared data without retrieving the entire data. They further extend their mechanism to support batch auditing.

Disadvantage: It suffers from the drawback of traceability, which means the ability for the original user to reveal the identity of the signer based on verification metadata in some special situations. As Oruta is based on ring signatures, where the identity of the signer is protected, this design does not support traceability.

PROBLEM FINDING

Before giving our main result, we study two classes of Schemes:

1. MAC based solution and
2. HLA based solution

Message Authentication Coded (MAC) based solution is also separated in two schemes:

Scheme I: In this scheme the user pre computes the MACs of each block of the data file and send both file and the MACs to the cloud server. Simultaneously it also sends the secret key to the TPA. At the time of audit TPA request for the data clocks and their corresponding MACs from the cloud server and check the correctness of the data. This simple solution for verification of data suffers from severe drawbacks. Firstly, as TPA demands for user's data it violates the privacy preserving guarantee & suffers from large communication overhead.

Scheme II: To improve the first scheme new solution is proposed. It avoid retrieving of data from the server .Before outsourcing the data the cloud user chooses s random MAC keys, pre compute s MACs for the complete data file, and this to verification data to the TPA. For verification, each time TPA reveals the secret key to the cloud server and ask for a fresh keyed MAC for comparison. Here auditing is limited by number of secret keys generated that to be fixed priori. And TPA has to maintain and update state of audits. And it is very difficult and error generating procedure for the TPA.

Homomorphic Linear Authenticator: Existing system based on HLA which is not preserve to privacy. HLA like MAC are verification metadata that are used for checking of the integrity of the data block. But unlike MAC, HLA can be aggregated. HLA based on homomorphic linear authentication which not privacy is preserving because in HLA linear combination of block may potentially reveal user data information to TPA.

PROPOSED WORK

In this Paper we proposed a system which is based on Bi-linear pairing. Both systems HLA & MAC are not privacy preserving. Our main scheme is based on a specific BLS scheme SHA-2 based solution.

Message authentication is a processing of message with a private key to produce a digital signature. Thereafter anyone can verify this signature by processing the signature with the signer's corresponding public key and compare that result with the message. Success confirms the message is unmodified since it was signed. And also the signer's private key has remained secret to the signer i.e. the signer, and no one else, performed the signature operation. In practice, typically only a hash or digest of the message, and not the message itself, is encrypted as the signature. A signature scheme consists of the following four algorithms:

1. *ParamGen* is a parameter generation algorithm.

2. *KeyGen* is a key generation algorithm.
3. *Sign* is a signature generation algorithm.
4. *Ver* is a signature verification algorithm.

Now, we are describing the new signature scheme as follows:

Step 1: ParamGen: The system parameters are $\{P, H, e, G1, G2, q\}$ are generated here. Here, P is a generator of $G1$ in order to q , H is a hash function and bi-linear pairing is given by:

$$e : G1 * G1 = G2.$$

Step 2: KeyGen: A user he/she randomly chooses x which belongs to Z_q^* and compute $P_{pub} = xP$.

Where Z_q^* is a collection of large random numbers.

Then the user's public key is P_{pub} . And his secret key is x .

Step 3: Signing: given a secret key x and a message m , then he/she computes $S = (H(m) + x)P$

Here, signature is S and $H(m)$ is the hash of the message.

The user uses cryptographic hash function SHA-2 for generating hash function. Then, user send message or file along with the verification data i.e. digital signature S to the server and delete file/message from local storage.

Step 4: Verifying: For the verification we are using TPA. If the TPA wants to check the integrity of message he creates a challenge (*chall*) for verification. In response server sends the proof of data storage correctness, which is the hash of message. Then TPA verifies the signature in the following manner: For a given public key P_{pub} , a message m and a signature S verify,

$$\text{if, } \{e(H(m)P + P_{pub}, S) = e(P, P)\}.$$

The verification work because of the following equation:

$$e(H(m)P + P_{pub}, S) = e((H(m)+x)P, (H(m)+x)P)$$

$$e(H(m)P + P_{pub}, S) = e(P, P)$$

EXPERIMENTAL SETUP

We have set up an experiment to justify our approach with the cloud. Our storage auditing protocol consists of two phases: Setup phase and Audit phase.

SETUP PHASE:

The first one is the setup phase. Here Cloud user register itself to the cloud, generate parameter and upload file to the cloud server. Here cloud user also generate signature and send it to the TPA.

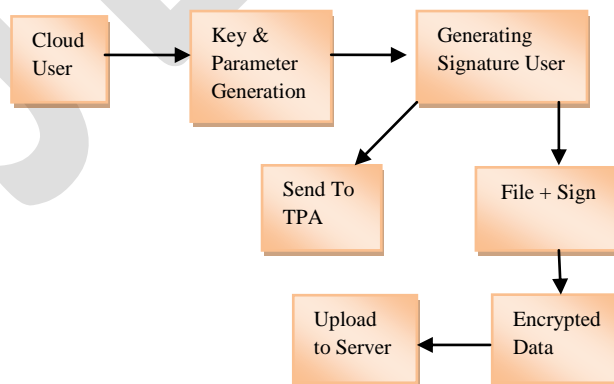


Fig. 3. Setup Phase of our Scheme

AUDIT PHASE:

Second one is the Audit phase. Here cloud user send request to TPA for the audit of outsourced data. TPA send auditing challenge to server. In response server generates a response message and send it to the TPA. Then, TPA performs the auditing of the data stored on the cloud server.

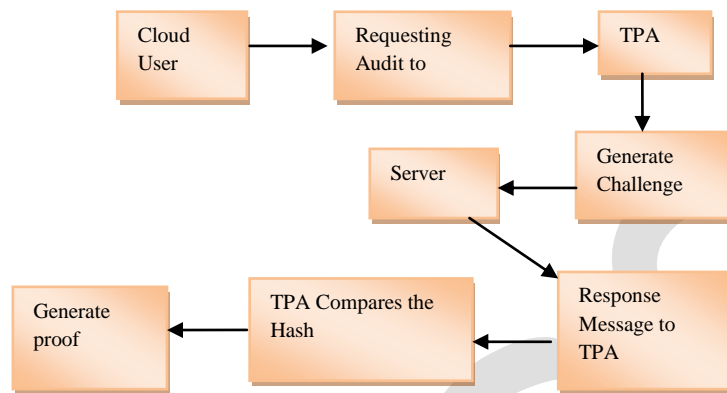


Fig.4 Audit Phase of our Scheme

IMPLEMENTATION FLOW

Before

Implementation flow describes the flow of our experimental setup i.e. how experiment is working. Here we have taken three modules for describing the flow of our implementation work. Three modules of our setup are as follow:

1. User module
2. Cloud server module and
3. TPA Module

The entire three modules provided with login credential to maintain its own account and can run on different machine from anywhere availability of access to the cloud.

Once the experimental setup is done with the machine, here we are started to perform with the data and to upload on to the machine and started performing to generate the signature and parameters using BLS algorithm to generate encrypted data in order to upload on to the cloud and perform a safely storage on to the cloud server.

[1] User Module:

It is Cloud Client Registration module. This module is used to create network which consist of Users and Server to communicate with each other. So here we create a network which consists of Data Owner/ User and the Cloud server (Fig 5). Here the cloud user registers itself to the server so that cloud user can store data on the cloud server. User can also request TPA to audit the file stored on remote server (Fig 7).

[2] Cloud Server Module:

After registering with cloud and getting login details to access the operation the user can perform following operations on server like file uploading, file editing. In the mention diagram (fig 6) user get connected and can upload its required file to keep on to the server and before that user can encrypt it, and store on the cloud.

[3] TPA Module:

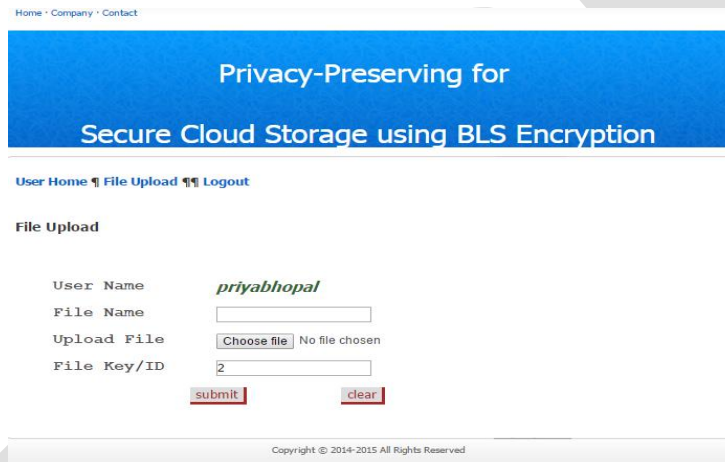
The Trusted Party Auditor is a module which is used to the audit the data that are uploaded by the Data Owner in the Server. So that TPA will audit data based on the Owner's request. Once it received the request from data owner, it checks the data integrity stored in cloud server.

In this phase (Fig 8) TPA is logged in and the file uploaded and requested by the user can be audited with the cloud block and perform the integrity check by generating our current scheme decryption and hashing technique. And once the data blocks are safe user can perform download operation with the data to get ensure about the data security.



The form is titled "Privacy-Preserving for BLS ALGORITHM". It contains a "User Register" section with the following fields: ID (value: 5), Name, User ID, Password, Mobile, Email ID, and Date (value: 30/01/2015). There are "submit" and "clear" buttons at the bottom.

Fig 5(User Registration)



The form is titled "Privacy-Preserving for Secure Cloud Storage using BLS Encryption". It includes a navigation menu: "User Home", "File Upload", and "Logout". The "File Upload" section contains: User Name (value: priyabhopal), File Name, Upload File (button: "Choose file", status: "No file chosen"), File Key/ID (value: 2), and "submit" and "clear" buttons. A copyright notice "Copyright © 2014-2015 All Rights Reserved" is at the bottom.

Fig 6(file upload and encryption phase)



The page is titled "Privacy-Preserving for Secure Cloud Storage Using BLS Encryption". It shows a user profile for "priyanka" and a table of files with "Send Request" buttons.

File ID	File Name	File Size	Date	Send Request
1	xyt	179 KB	12/12/2014	Request
2	qwerty	5983 KB	13/12/2014	Request

Fig 7(Send request for Auditing)

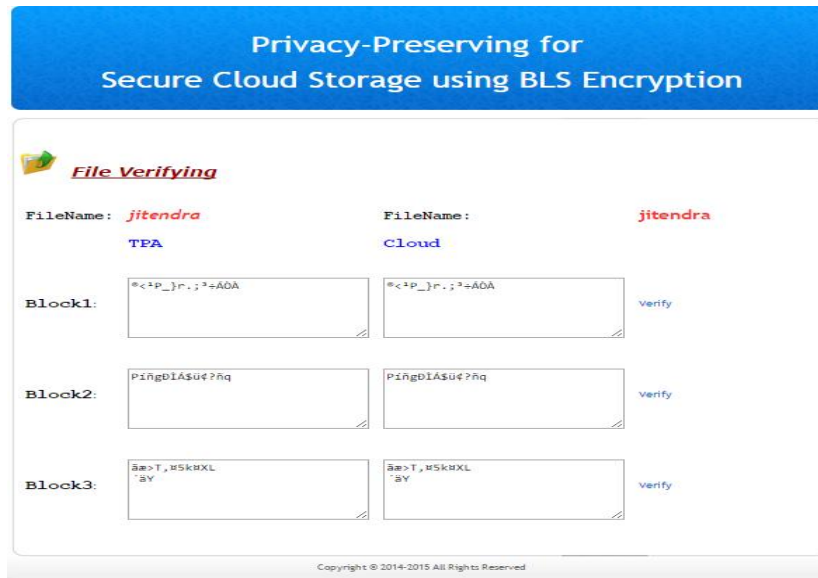


Fig 8(Auditing phase)

ANALYSIS OF OUR AUDITING PROTOCOL

Our main goal is to enable public auditing for cloud data to become a reality. Thus, the whole architectural design should not only be cryptographically strong, but, more importantly, it should be practical from a systematic point of view.

We briefly elaborate a set of suggested desirable properties below that satisfy our scheme. The in-depth analysis is discussed in this section.

- **Security Analysis:**

Thus the system is based on the BLS scheme where its giving the advantage over the current MAC and HLA based scheme, it is providing us a privacy preserving approach which tell the data is not being shared with the cloud or any other level.

As described above about the proposed system it is faster in process than the current system. The existing system may leak the data and give the poor performance. The direct adoption of HLA based technique is still not suitable because the linear combination of blocks, $\mu = \sum_i v_i m_i$. It may potentially reveal user data information to TPA, and violates the privacy guarantee. Specifically, by challenging the message block m_1, m_2, \dots, m_n using different sets of random coefficients $\{v_i\}$. TPA can accumulate different linear combinations $\mu_1, \mu_2, \dots, \mu_n$. With $\{\mu_i\}$ and $\{v_i\}$, TPA can derive the user's data M by simply solving a system of linear equations.

Where as in our system, as it won't have to go through those steps it makes our system faster and more security thus it increase our system performance.

- **Performance Evaluation:**

We now assess the performance of the proposed public auditing scheme. The experiment is conducted using Java on a Windows system with an Intel Core 2 Duo processor running at 2.00 GHz, 2 GB of RAM, and 32 bit of Operating System. All experimental results represent the mean of 30 trials.

We begin by estimating the performance of our proposed scheme in terms of the time required by the operations to execute successfully. We quantify the time taken by the privacy preserving auditing in terms of file uploading, server computation and auditor Computation. First before uploading we have taken a variable t which is being initialized by 0. Then on performing a particular task again we have calculated the time being taken by the system to execute the operation and monitored the new time on machine. Performance under different small size of data is illustrated in TABLE II

PERFORMANCE UNDER DIFFERENT SIZE OF DATA

	<i>Data Size (Byte)</i>	<i>File uploading Time (ms)</i>	<i>Server Computation time (ms)</i>	<i>TPA computation time (ms)</i>
Our Scheme	460	230	318	487
	300	223	217	355
Existing Scheme (15)	460	Not mention	335.17	530.60
	300	Not mention	219.27	357.53
(6)	460	Not mention	333.23	526.77
	300	Not mention	217.33	353.70

As we can see that our scheme works efficiently in comparison with the other scheme. Now to check the efficiency of our scheme we compare it with other scheme but with a large data file. We can see that our proposed scheme work accurately with large data files too. And hence, the proposed method provides an efficient and accurate solution for the auditing of the data through a third party using our proposed system. Illustrated table (TABLE III) gives the comparison of proposed scheme with the previous schemes under large sized of data files

PERFORMANCE UNDER DIFFERENT SIZED DATA

	<i>Data Size (Byte)</i>	<i>File uploading Time (ms)</i>	<i>Server Computation time (ms)</i>	<i>TPA computation time (ms)</i>
Our Scheme	4600	250	330	490-510
	3000	223	230	350-365
Existing Scheme (15)	4600	Not mention	361.56	547.39
	3000	Not mention	242.29	374.32
(6)	4600	Not mention	342.91	543.35
	3000	Not mention	223.64	370.29

ACKNOWLEDGMENT

I wish to express my sincere thanks to our Principal, HOD, and Professors and staff members of Computer Science & Engineering Department at Laxmi Narain College Of Technology, Bhopal. I would like to thank my mentor who have always been there to support and helped me to complete this research work.

CONCLUSION

Cloud Computing is emerging technology and gaining remarkable popularity in the recent years for its benefits in terms of flexibility, scalability, reliability, efficiency and cost effectiveness. Despite of all these benefits, Cloud Computing has one problem: Security, we analyze the problems of data security in cloud storage, which is a distributed storage system. An effective and flexible scheme is proposed in our paper to ensure the integrity and correctness of the data stored in cloud server. By detailed security and performance analysis, we show that our scheme is very efficient for checking integrity, probity and correctness of user's data in the cloud servers by third party authenticator (TPA). The future work for our research will be in the field of enhancing the security of the data by appending two signatures on the data to be stored on cloud so that no one can harm or change the data.

REFERENCES:

- Deyan Chen, Hong Zhao, "Data Security and Privacy Protection Issues in Cloud Computing", International Conference on Computer Science and Electronics Engineering 2012.
- Boyang Wang, Baochun Li, Hui Li," Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud," IEEE TRANSACTIONS ON Cloud Computing, VOL. 2, NO. 01, March 2014.
- "The Notorious Nine - Cloud Computing Top Threats in 2013," https://downloads.cloudsecurityalliance.org/initiatives/top_threats
- "Panda: Public Auditing for Shared Data with Efficient User Revocation in the Cloud," IEEE Transactions on Services Computing, VOL. X, NO. X, XXXX 2014, accepted.
- K. Ren, C. Wang, and Q. Wang, "Security Challenges for the Public Cloud," IEEE Internet Computing, vol. 16, no. 1, pp. 69–73,2012
- 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.
- Qian Wang, Cong Wang, Kui Ren, Wenjing Lou, Jin Li," Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing"Proc. IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 22, NO. 5, MAY 2011
- A. Juels and B.S. Kaliski Jr., "Pors: Proofs of Retrievability for Large Files," Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), pp. 584-597, 2007.
- G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores," Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), pp. 598-609, 2007.
- G. Ateniese, R.D. Pietro, L.V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Data Possession," Proc. Fourth Int'l Conf. Security and Privacy in Comm. Networks (SecureComm '08), pp. 1-10,2008.
- C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring Data Storage Security in Cloud Computing," Proc. 17th Int'l Workshop Quality of Service (IWQoS '09), 2009.
- C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, "Dynamic Provable Data Possession," Proc. 16th ACM Conf. Computer and Comm. Security (CCS '09), 2009.
- K.D. Bowers, A. Juels, and A. Oprea, "Hail: A High-Availability and Integrity Layer for Cloud Storage," Proc. 16th ACM Conf. Computer and Comm. Security (CCS '09), pp. 187-198, 2009..
- D. Boneh, B. Lynn, and H. Shacham, "Short Signatures from the Weil Pairing," Proc. Seventh Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT '01), pp. 514-532, 2001.
- Cong Wang, Member, IEEE, Sherman S.M. Chow, Qian Wang, Member, IEEE, Kui Ren, Senior Member, IEEE, and Wenjing Lou, Senior Member, "Privacy-Preserving Public Auditing for Secure Cloud Storage", IEEE IEEE TRANSACTIONS ON COMPUTERS, VOL. 62, NO. 2, FEBRUARY 2013