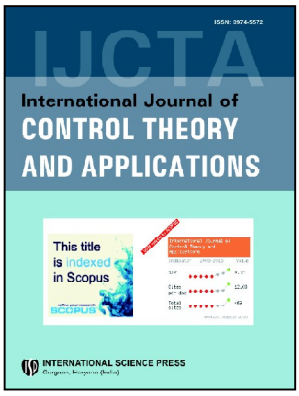

International Journal of Control Theory and Applications

ISSN : 0974-5572

Volume 9 • Number 50 • 2016

Privacy and Integrity Preserving Trusted Third Party Auditing for Outsourced data in the Cloud by using Security Tags

S. Ahamed Ali[1*] and M. Ramakrishnan[2]

[1]Assistant Professor, Department of Information Technology, Velammal Engineering College, Anna University, Chennai, Tamil Nadu, India
[2]Professor and Chairperson, School of Information Technology, Madurai Kamaraj University, Madurai, Tamil Nadu, India
**Correspondence: haiahamed@gmail.com*

Abstract: Cloud computing is a cutting edge technology which is widely used in all applications. It is a technology which provides computation, software, data access, and storage services that do not require end-user knowledge of the physical location and configuration of the system that delivers the services. In the current scenario, organizations produce a large amount of sensitive data including personal data, health records and financial informations.These organizations depend on cloud to store all their sensitive records. The data or the information that is stored in a cloud server is referred as data outsourcing and generally they are managed by a third party. By outsourcing data storage, organizations delegate the storage and management of their data to a Cloud Service Provider with pre-defined SLAs that include payment of fees metered in GB/month. The fact that data owners no longer physically possess their sensitive data raises new challenges. Once customer's data has been outsourced to the cloud computing servers, eûcient veriûcation of the completeness and correctness of the outsourced data becomes a great challenge. The traditional methods used to achieve confidentiality and integrity cannot be directly adopted in the cloud since the data is stored and accessed from the remote site. Considering these problems, we propose a very practical approach of providing security and integrity to the data outsourced in the cloud by the data owner. To save computational resources of the data owner and to ensure the confidentiality and the integrity of the outsourced data this work propose a trusted third party auditing scheme. The proposed approach generates a unique security tag for each outsourced data blocks. Security tag is used for secure and effective trusted third party auditing. In our approach the entire data block is not downloaded for auditing hence the storage and communication cost associated with the auditing process is minimized. We hope our model provides enhanced security to the outsourced data and perform better than the existing systems.

Keywords: Cloud Computing, Data outsourcing, Cloud Service Provider,Security and Integrity, Security Tag, Trusted third party auditing

1. INTRODUCTION

Cloud computing from the last few years has rapidly grown from a business initiative into one of the fastest growing Emerging Technologies of Information and Communication Technology. Cloud computing describes the use of networks of remote servers - usually accessed over the Internet - to store, manage, and process data.

NIST (2009) defines cloud computing as a model for providing an enabled, convenient, on-demand, pay-as-you-use shared collection of a computing resources that include applications and programs, storage, servers, services and networks that can be customized easily with speed and limited management effort. Hence organizations and enterprises are showing huge interest in migrating their sensitive and important data to the cloud. More than half of all businesses utilize cloud services, with 95 percent of the general public using cloud in at least one form, according to a recent study from Soliant Consulting. Cloud usage is expected to become even more widespread in 2017, with Soliant estimating that 36 percent of all data will be stored in the cloud by the end of 2017.

When data is migrated to cloud, security of the data must be verified since the sensitive and the critical data lies outside the data owner premises. Thus security violations that include data loss, unauthorized data access and modifications can occur. Hence guarantee for data availability, data confidentiality and data integrity must be given with high priority. Data availability means data must be always accessed even in the case of natural and other unpleasant disasters such as power outages. Data confidentiality is the most essential aspect of outsourced data security. Data confidentiality means preventing the access of the outsourced data by unauthorized third party that includes the third party auditor and also the cloud service provider. Data confidentiality can be provided by using appropriate encryption algorithms with secure key management schemes. Data integrity deals with accuracy and consistency of the data stored in the cloud. Mechanisms that include MAC's, hash functions and digital signatures can provide integrity to an extent.

Whenever the data owner outsources the data files in to the cloud, he loses the control over his outsourced data and further he has to trust the security mechanism provided by the CSP. Thus the confidentiality of the outsourced data cannot be completely guaranteed. Hence we need specific and efficient encryption mechanisms that are suitable for the cloud environment. Downloading data files for the purpose of checking their integrity is not an efficient solution due to the high network bandwidth requirement and hence it is practically an infeasible solution. The traditional methods used to achieve integrity such as MAC schemes, Hash methods and digital signatures cannot be directly adopted in the cloud based distributed environments since the user data is stored and accessed from the remote cloud site. Hence effective remote integrity verification methods are needed.

This work mainly focuses on ensuring confidentiality and integrity of the data outsourced to the CSP by the data owner. This work ensures that the trustworthiness of the data can be verified without completely downloading it. The proposed solution also guarantees that the data cannot be accessed by unauthorized third party users including the cloud service provider and the third party auditor. To save computational resources of the data owner and to check the integrity of the outsourced data this work propose a trusted third party auditing scheme, to audit the outsourced data behalf of the data owner.

2. RELATED WORKS

Message Authentication Code(MAC) based remote integrity check was implemented by [7] . MAC uses an algorithm that produces a fixed length code called MAC irrespective of the length of the message. MAC algorithm needs a secret key to compute MAC. The data owner before uploading his data file into the CSP will generate a finite number of MACs depending on the number of auditing attempts required using a different secret key for each MAC. The data owner will store the computed MAC and the secret key in his private storage space. To verify the integrity of data, the data owner can each time challenge the CSP by providing one of the available keys. The CSP then has to respond by computing the MAC using the received key. If the stored MAC equals the computed MAC by the CSP the data stored by the CSP is not altered. When all keys are used for auditing, the user has to retrieve the data and compute new MACs. This is the serious limitation of this method. Another major setback of this method is it supports only limited number of auditing, and it is impossible to perform public auditing.

Digital signature based schemes are also available in literature to preserve the integrity of the outsourced data[7].Digital signature based schemes use asymmetric key algorithm .In this method message digest is generated

using a hashing algorithm such as SHA,MD5 etc., for the given block and it is encrypted with the senders private key. In this scheme, users pre-compute the signature for each data block and then send both the blocks and their corresponding signatures to the CSP. To verify the integrity of data, the data owner can each time challenge the CSP by requesting the CSP to generate digital signatures for some randomly selected blocks. The CSP then has to respond by returning the required blocks along with their signatures. The data owner then computes the signature of the received block using his public key. If the stored and computed digital signature matches the data stored by the CSP is not modified. Thus, this scheme provides a probabilistic assurance for remote data because the verifier checks the integrity of a certain number of data blocks instead of all data blocks. The major issue with this scheme is the communication overhead involved in downloading the data blocks for integrity verification creates a major bottleneck in performance enhancement.

Proofs of Retrievability (PoR) based schemes for data auditing in cloud based storage systems are also found in literature. This approach uses a cryptographic method for remotely auditing the integrity of files stored in the cloud, without keeping a copy of the original files in local storage. One of the most popular approach for PoR scheme is proposed by Juels and Kaliski[8].Initially the data blocks are encoded using some error correction codes before outsourcing to the CSP. This approach is based on sentinels. In their scheme, the file F is encrypted and randomly embedded with a set of check blocks called sentinels. Sentinels are generated to perform remote integrity auditing. Sentinels are generated in such a way that they cannot be differentiated from other data blocks. The position of the sentinels is stored along with their value at the data owner's storage space. To verify the data integrity, the verifier challenges the CSP by identifying the positions of a set of sentinels and requesting the CSP to return the sentinel values. The main advantage of this scheme is the CSP has to access only a part of the stored data blocks (the portion of the block that contains sentinels).The problem with this approach is the problem of preprocessing overhead imposed on the data owner.

Shacham and Waters [13] proposed a method that guarantee the shortest query and response when compared with the scheme proposed by Juels and Kaliski[8]. In their scheme, the encoded file (F) is divided into a number of data blocks m1,m2,…,mn. The data owner generates a private key, which is a combination of a random number and a key for the Pseudo Random Function. After generating the private key, the data owner generates verification metadata for each data block using the private key. The data blocks {mi} and the verification metadata {Ti} are then sent to the CSP. To check the integrity of the data, the verifier generates a verification challenge that will be sent to the CSP. The challenge specifies a set of data blocks, which is chosen at random, to be audited. Upon receiving the challenge, the CSP will compute a proof from the specified data blocks and then send it to the verifier. Having received the proof, the verifier can check the correctness of the received proof by comparing it with the stored metadata. Even though this approach is better than the previous approach proposed by Juels and Kaliski[8] , modification of the data blocks that include insertion, deletion, alteration are not considered. Further if any modification is needed on the outsourced data the data owner must regenerate the verification metadata for each modified data block using the private key. Thus, communication overhead involved in retrieving the data blocks re-computation of metadata for verification proved to be a costly one.

Provable Data Possession popularly called as PDP schemes were introduced by Ateniese et al. [2]. PDP is a technique that allows users, and the data owner, to check the integrity of their data without retrieving it. Ateniese et al. [2] have proposed a provable data possession (PDP) scheme that supports data appending operation which allows users to add new blocks at the end of the file. In their scheme the data owner before uploading the data to the cloud will split the File F into equal size blocks {b1,b2,b3…bn},where n is the total number of blocks. The data owner will generate a unique tag for every outsourced block and store these tags in his trusted storage space. The data blocks are then outsourced in the cloud. During auditing phase the data owner send challenge message that contains the set of data blocks to be verified. The data owner will request the cloud service provider to generate tags for the specified data blocks. The data owner can then verify the response by using tags stored in his trusted storage space. If the returned tags match the stored ones, the data file F is not tampered. The

absence of dynamic operations is the major pitfall in this method. To overcome this Erway et al. [7] proposed a scheme that supports dynamic operations on the outsourced blocks using rank-based authenticated skip lists which supports provable updates. This scheme is not efficient due to the use of authenticated skip list.

3. TRUSTED THIRD PARTY AUDITING BASED SCHEMES

Trusted third party auditor based schemes were proposed by Wang, Qian [15]. A trusted third party auditor (TTPA) is needed to ensure the privacy and security of the data outsourced in the cloud. These schemes relieve data owners from the burden of complex computation that are needed to perform effective auditing and in general a trusted TPA helps to perform effective data auditing behalf of the data owner. The following two important properties must be considered in order to make the third party audit most effective.

- The TPA should not demand the local copy of the data
- No new vulnerabilities shall be introduced towards the privacy and security of the user data

The TPA auditing schemes would verify the correctness of data without the complete data hence public auditing can be done on outsourced data with privacy preserving. In general the trusted TPA based auditing scheme is a four tuple algorithm , KeyGen, SignGen, GenProof, and VerifyProof .

1. KeyGen: this algorithm is run by the data owner to produce public and private keys.
2. SignGen: this algorithm is also run by the data owner to generate a signature that will be used for data auditing.
3. GenProof: this algorithm is run by the CSP to generate a proof of the stored data.
4. VerifyProof: this algorithm is run by the TPA to verify the proof generated by the CSP.

4. SYSTEM ARCHITECTURE

The new architecture proposed for the secured trusted third party auditing has the following communicating parties

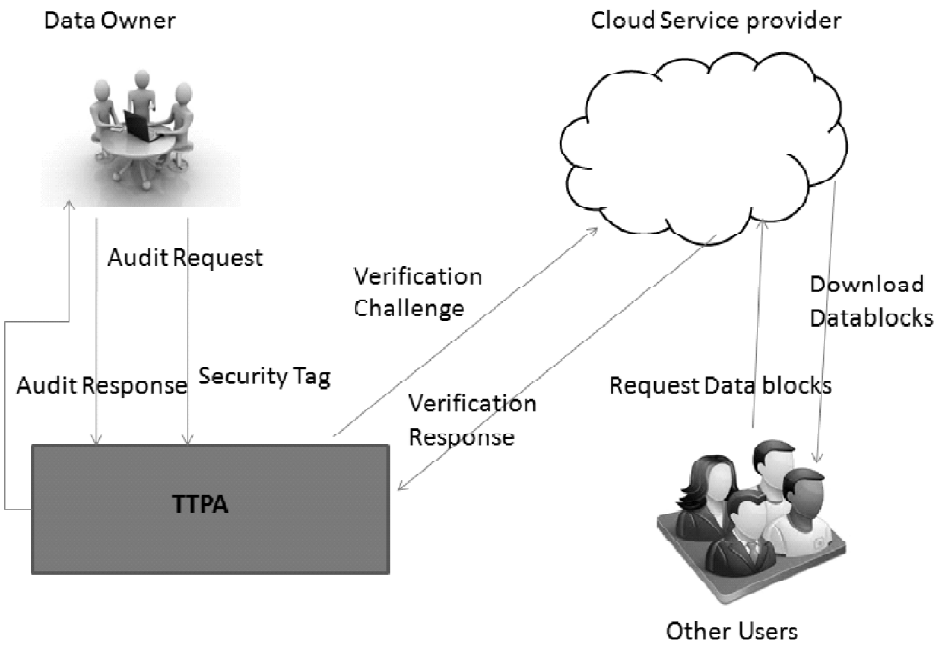

Figure 1 : TTPA Architecture Diagram

- The Cloud Service Provider(CSP): a third party who provides storage services to the outsourced data in the cloud.
- Data Owner: who outsources the data in the cloud and generate security tag for auditing. The data owner also request the TTPA to perform audit on his outsourced data
- Trusted Third Party Auditor (TTPA) : It is a trusted authority who audit the integrity of the outsourced data behalf of the data owner. However he has no access permission to view or modify the content of the outsourced data.
- The User (U): who has limited access rights to share or use the data block outsourced by the data owner

5. THE CONCRETE SCHEME

Before outsourcing the file(F) to the CSP the data owner will divide the (F) into fixed size blocks $F=\{b_0,b_1,b_2\ldots b_n\}$,where 'n' stands for the total number of file blocks. Each file block is encrypted before uploading to the cloud servers. For each block a unique security tag is generated and attached to the block. This tag is used for auditing purpose. The security tag along with the encrypted blocks is outsourced to the CSP and the security tags alone are uploaded to the TTPA. To verify the integrity of outsourced data blocks, the data owner can send a verification request to the TTPA to audit the outsourced data blocks that is available with CSP. The TTPA will then issue a verification challenge to the CSP. Upon receiving the challenge, the CSP has to respond with the necessary proof to the TTPA. The TTPA will then verify the response and there by the correctness of the outsourced data blocks and inform the status to the data owner.

The following notations are used in the proposed protocol design and it is listed in table 1

Table 1
Notations used in our protocol design

Notation	*Meaning*
F	Outsourced file
b_i	Represents the i^{th} block of the F
n	Total number of blocks
K_R	Private Key
K_U	Public Key
$Z"_n$	n^{th} residue of the integer
Ω	Security Tag
g	A generator for a large prime number 'p' of the multiplicative group $Z"_p$
r	A random Number
H	A hashing algorithm such as SHA

6. TTPA PROTOCOL DESIGN

The proposed TTPA is a protocol which runs six algorithms namely, ASYMKeyGen, EnDec, SecTagGen, ChallengeGen, ProofGen, and VerifyProof.

I. ASYMKeyGen: ASYMKeyGen stands for asymmetric key generation algorithm. The Data owner executes this algorithm to generate public and private key pair namely K_R,K_U. Here K_R is used by the data owner for encrypting the data blocks and K_U ,the public key is shared between the data owner and the other trusted users who have access rights over encrypted data blocks in the cloud. K_U is used by the users to decrypt the encrypted data blocks.

II. EnDec: EnDec stands for encryption and decryption. This is the algorithm responsible for encrypting and decrypting the data blocks. The encryption algorithm is executed in the trusted storage space available with the data owner. The encrypted data blocks are outsourced to the cloud.

III. SecTagGen: It stands for security tag generation. This algorithm is also run by the data owner to generate a unique field called Security Tag which appended with the data blocks. This tag is used during auditing phase. The security tag will also be sent to the TTPA for auditing purposes.

IV. ChallengeGen: This algorithm is run by the TTPA to send a verification challenge to the CSP.

V. ProofGen: The CSP will execute this algorithm in response to the challenge given by the TTPA. The output of this algorithm is given as the response to the challenge message.

VI. VerifyProof: This algorithm is run by the TTPA to verify the correctness of the response given by CSP for the given challenge

The flow of the six algorithms is illustrated in the figure <<xyz>>

I. ASYMKeyGen Algorithm

This algorithm is executed by the data owner to generate private and public key pair that is shared between the data owner and other trusted users who have limited rights over the outsourced data blocks. We use elgammal based public key cryptosystem proposed by Taher El Gamal [14] to generate private, public key pairs. This cryptosystem is based on discrete logarithmic problem. The security of this cryptographic technique depends on difficulty in computing discrete logs in a large prime modules. The Key Generation part of the elgammal cryptosystem is illustrated as follows

1. Generate a large prime p and a generator g of the multiplicative group Z^*_p
2. Select a random integer a between 1,2,….p-1.
3. Compute $(g^a) \bmod p$
4. $K_U = \{p,g,g^a\}$
5. $K_R = a$

The encryption and decryption part of this algorithm is explained below

II. EnDec Algorithm

EnDec is an algorithm of two components namely encryption and decryption. This algorithm is an asymmetric key encryption algorithm in which the encryption of plain text is done with the private key K_R and the decryption is done with the public key K_U.

Encryption: The encryption algorithm needs the public key K_U and the message block b_i as input and works as follows

1. Let bi be the message block to be encrypted. (Note that each message block contains an appended security tag).
2. Select a random integer "r" such that $r \in Z_n$.
3. Compute $\gamma = g^r \bmod p$
4. Let $\alpha = b_i * (g^a)^r$
5. The message block b_i can be encrypted using the equation $C_i = \{\alpha, \gamma\}$
6. The encrypted data blocks are uploaded to the CSP

Decryption

This algorithm needs the Private Key K_R to decrypt the security tag

$$b_i = (\gamma^{-a}) * \alpha \bmod p$$

III. SECTAGGEN

This algorithm generates security tag which is used by the TTPA to perform trusted third party auditing. The security tag is the enhanced version of the model proposed by ahamed *et al.* [1]. The security tag is attached with every block .This algorithm takes two inputs the first parameter is the meta data of the outsourced block and the second parameter is the write counter. Write counters are maintained for every outsourced block to keep track of number of modification done on any particular block. A unique function called process_block() generates a code 'ρ'. This code is used as a parameter for security tag generation.

```
Process_block()
{
        let u = H (b_i || b_i.wcnt)
        Encrypt u
        ρ = E(u).r^{bi.wcnt}, r is a random number
        return ρ
}
```

Security tags are generated using BLS scheme. We use this scheme because BLS signatures are short and homomorphic in nature. Now we explore the algorithm Sec_TagGen that takes two parameters (d1, ñ) from which the Security tag(Ù) is generated.

1. Compute $N = x * y$ and $\lambda = LCM\ (x\text{-}1, y\text{-}1)$, Where x represents the block index and y represents the number of writes done on the block bi
2. A token value called α is computed as $\alpha_i = (H(b_i).\ \rho^{bid1+r_i})^1 \in G_1$ Here G1 is a cyclic group of prime order.$H(b_i)$ represents hash value of the block b_i and r_i represents the random number generated using the pseudo random number generator. The random number is generated to avoid generation same tokens for similar data blocks
3. Security tag Ω computed as $\Omega = \alpha_i^{\ m} r^N \bmod N^2$

The security tag Ù generated for each block is sent to TTPA. The TTPA use this tag during auditing phase to challenge the CSP.

IV. Challenge Gen

Whenever the data owner request the TTPA to verify the integrity of the outsourced data block the TTPA will run this algorithm to send a verification challenge to the CSP. The algorithm has two phases namely (a) The Pre-process phase (b) The Challenge phase

(a) The Pre-Process Phase

1. The TTPA generate two large prime numbers namely p and q and compute modulus $m = p*q$
2. Computes $\phi(m) = (p\text{-}1) * (q\text{-}1)$ { like RSA algorithm }
3. Computes a value $S_i = \Omega_i \bmod \phi(m)$ where $1 \leq i \leq n$.

(b) The Challenge Phase

1. Generates a random number r.
2. Generate l (≤ n) random values V_i ($1 \le l \le n$)
3. Sends r, V_i as a challenge message to the CSP

V. Proof Gen

The CSP will execute this algorithm in response to the challenge given by the TTPA. The following are steps executed by the CSP

1. Computes a = $\Sigma_{i=1}^{l} V_{i*} \Omega_i$
2. Computes Response = $r^a \bmod n$

The computed response will be sent to the TTPA for verification.

VI. Verify Proof

This algorithm is run by the TTPA to verify the correctness of the response given by CSP for the given challenge. The following are the steps executed by the TTPA

1. Computes $a^| = \Sigma_{i=1}^{l} V_{i*} S_i \bmod \phi(m)$
2. Computes V_Response = $r^{a^|} \bmod n$
3. Checks whether Response = V_Response

The TTPA will sent the response of the audit to the data owner

7. PERFORMANCE ANALYSIS OF THE PROPOSED TTPA ARCHITECTURE

In this section we analyze the performance of our proposed scheme in various aspects. We have implemented our scheme using Java language. The experiments are conducted using local cloud servers. We measured computation time, storage cost, file upload and download times for varying file sizes. The data update operations are done on multiple blocks at a time by the data owner. The update operation includes block insert, modify, and delete operations in addition to creation of new security tags. We ran the experiments for block update on a 1 MB file with a file block size of 128 bytes. The experiments are run by inserting and modifying 2% to 50% number of file blocks. The results are depicted in figure 2

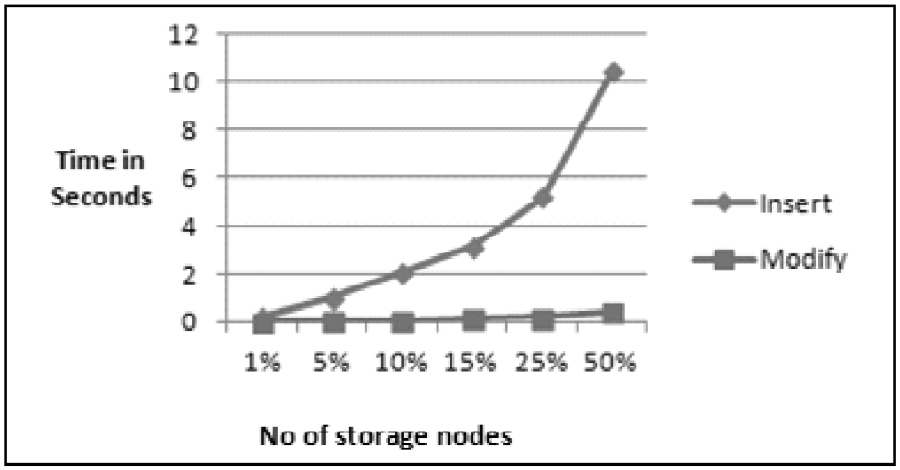

Figure 2: Block operations

We plot the upload and download time for text files in figure 3. The block size was typically set as 2 KB for the files.

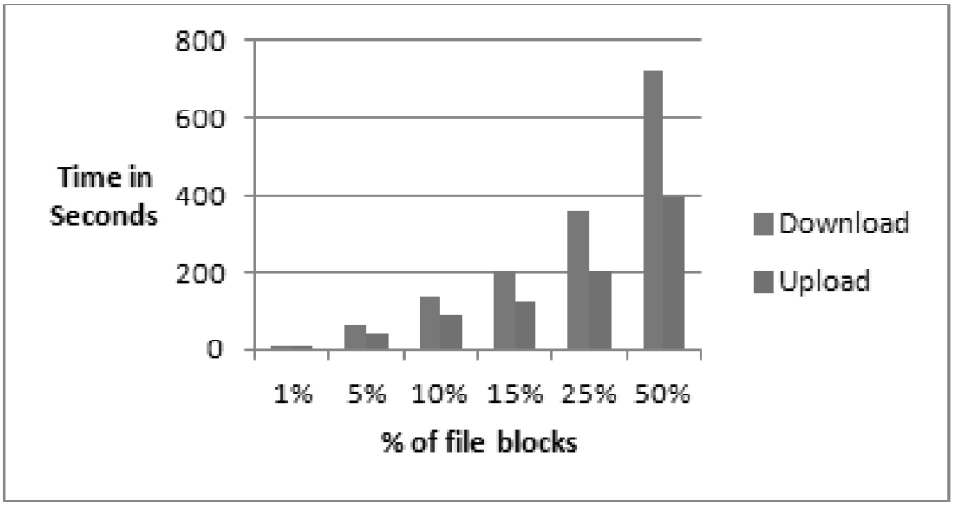

Figure 3: Upload and download time

The computation cost associated with the three entities the data owner,TTPA,CCSP is presented in table 2. The computation cost is measured for different file size varying from 20 MB to 100 MB. It can be noted that computation cost for the TTPA exceeds the data owner and the CCSP.

Table 2
Computation time Vs Computation time plot

File Size (in MB)	*Computation Time (In ms)*		
	Data Owner	*TTPA*	*CCSP*
20	230	430	350
40	300	522	500
60	350	671	552
80	440	720	632
100	500	745	648

We carried out experiments on text file sizes. In our work private key bit sizes are selected as suggested by NIST recommendation. Private key size is fixed as 160 bits for ElGamal and Paillier and our proposed scheme. The storage space required by the encrypted files based on the different schemes is presented in the table xx . The storage space required by the encrypted file in our scheme is lesser when compared to the other two schemes for the same key size.

Table 3
Encrypted Storage Space vs File Size plot

File Size (in KB)	*Storage Space needed for the encrypted files (In KB)*		
	Elgammal Scheme	*Paillier scheme*	*Proposed Scheme*
60	68	80	72
120	130	133	123
180	192	196	186
220	216	230	210

We also compare our proposed system design with other protocols that provide confidentiality and integrity to the outsourced data in table 4.

Table 4
Comparison with other protocols

Components Protocol	*Audit Attempts*	*Audit mechanism*	*Support for Dynamic Operation*	*Third Party audit feature*	*Privacy preserving*	*Integrity support*	*Confidentiality support*
MAC based Scheme	Limited	Downloading datablock mechanism	No	NO	No	No	No
Digital Signatures based Scheme	Not Limited	Without Downloading Audit is performed	No	NO	NO	NO	NO
Proofs of Retrievability (POR)			No	No	No	No	No
Sentinal approach based POR			No	No	No	No	No
Provable Data Possession			Partial Support	No	No	No	No
Dynamic PDPs			Supported	No	No	No	No
TTPA			supported	Yes	Yes	Yes	No
Our Approach			Supported	Yes	Yes	Yes	Yes

8. CONCLUSION

Cloud Computing is a technology that is continuously growing, and it is expected to successfully change the way we utilize information and communication technology resources. It also raises the demand for trust and security in cloud enabled technologies. Hence security will become more important and will be a decision criterion for enterprises moving services into cloud computing technology. In this paper we listed various security concerns that may arise because of migrating the sensitive data to the cloud by organizations. We have proposed a trusted third party auditing scheme which will preserve the privacy and integrity of the data outsourced in the cloud. The security tag generated in our approach strengthens our scheme and supports secure and effective trusted third party auditing. Our approach does not require download of the entire block for auditing. Further we use BLS signature scheme for generating the security tags which is proved to be easily generated. Extensive analysis shows that our scheme is provable secure, and the performance evaluation shows that our scheme is better than the existing protocols.

REFERENCES

[1] S. Ahamed Ali and Dr.M. Ramakrishnan ," Security Tag and Hierarchical Tree Based Dynamic Key Management Technique for Ensuring Storage Security to the Data Outsourced in the Cloud" , International Journal of Printing, Packaging & Allied Sciences, Vol. 4, No. 2, December 2016, pages 1320-1329.

[2] Ateniese, Giuseppe, et al. "Provable data possession at untrusted stores."*Proceedings of the 14th ACM conference on Computer and communications security*. Acm, 2007.

[3] Atallah MJ , Blanton M, FazioN, FrikkenKB. Dynamic and efficient key management for access hierarchies. ACM Transactions on Information and System Security 2009; 12:18:1–43.

[4] Ayad F.Barsoum and M.Anwar Hasan,Provable Possession and Replication of Data over Cloud Servers - Centre For Applied Cryptographic Research (CACR), University of Waterloo, Report 2010/32, 2010, *http://www.cacr.math.uwaterloo.ca/techreports/2010/cacr2010-32.pdf*

[5] Blundo C, Cimato S, De Capitani di Vimercati S, De Santis A, Foresti S, Paraboschi S, et al. Efficient key management for enforcing access control in outsourced scenarios. In: Gritzalis D, Lopez J, editors. Emerging challenges for security, privacy and trust, IFIP advances in information and communication technol- ogy, 297. Boston: Springer; 2009. p. 364–75.

[6] Deyan Chen and Hong Zhao. Data security and privacy protection issues in cloud computing. In Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on, volume 1, pages 647–651, 2012.

[7] Erway, Chris, et al. "Dynamic provable data possession." *Proceedings of the 16th ACM conference on Computer and communications security*. Acm, 2009.

[8] Juels, Ari, and Burton S. Kaliski Jr. "PORs: Proofs of retrievability for large files." *Proceedings of the 14th ACM conference on Computer and communications security*. Acm, 2007.

[9] Li, Jin, et al. "An efficient proof of retrievability with public auditing in cloud computing." *Intelligent Networking and Collaborative Systems (INCoS), 2013 5th International Conference on*. IEEE, 2013.

[10] Li, Ling, et al. "Study on the third-party audit in cloud storage service." *Cloud and Service Computing (CSC), 2011 International Conference on*. IEEE, 2011.

[11] Miao Zhou , Yi Mu, Willy Susilo, Jun Yan Liju Dong , Privacy enhanced data outsourcing in the cloud. In Journal of Network and Computer Applications 35 (2012) 1367–1373 , Journal homepage: *www.elsevier.com/locate/jnca*

[12] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In Proceedings of the 17th international conference on Theory and application of cryptographic techniques, EUROCRYPT'99, pages 223–238, Berlin, Heidelberg, 1999. Springer-Verlag.

[13] Shacham, Hovav, and Brent Waters. "Compact proofs of retrievability."*Advances in Cryptology ASIACRYPT 2008*. Springer Berlin Heidelberg, 2008. 90-107.

[14] Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In Proceedings of CRYPTO 84 on Advances in cryptology, pages 10–18, New York, NY, USA, 1985. Springer-Verlag New York, Inc.

[15] Wang, Qian, *et al.* "Enabling public verifiability and data dynamics for storage security in cloud computing." *Computer Security–ESORICS 2009*. Springer Berlin Heidelberg, 2009. 355-370.

[16] Wang, Qian, *et al.* "Enabling public auditability and data dynamics for storage security in cloud computing." *Parallel and Distributed Systems, IEEE Transactions on* 22.5 (2011): 847-859.

[17] *http://www.soliantconsulting.com*

[18] *https://www.nist.gov*