International Journal of Scientific Engineering and Research (IJSER)
www.ijser.in
ISSN (Online): 2347-3878
Volume 2 Issue 6, June 2014

Privacy-Preserving Public Auditing using TPA for Secure Cloud Storage

Jyoti R Bolannavar[1]

[1]P G Student, Department of Computer Science, Gogte Institute of Technology, Belgaum-590008, Karnataka, India

Abstract: ***Using Cloud Storage, users can remotely store their data and enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources, without the burden of local data storage and maintenance. However, the fact that users no longer have physical possession of the outsourced data makes the data integrity protection in Cloud computing a formidable task, especially for users with constrained computing resources. Moreover, users should be able to just use the cloud storage as if it is local, without worrying about the need to verify its integrity. Thus, enabling public audit ability for cloud storage is of critical importance so that users can resort to a third party auditor (TPA) to check the integrity of outsourced data and be worry-free. To securely introduce an effective TPA, the auditing process should bring in no new vulnerabilities towards user data privacy, and introduce no additional online burden to user. In this we propose a secure cloud storage system supporting privacy-preserving public auditing. We further extend our result to enable the TPA to perform audits for multiple users simultaneously and efficiently. Extensive security and performance analysis show the proposed schemes are provably secure and highly efficient.***

Keywords: Homomorphic Linear Authenticator, Third Party Auditor, Public Auditing, Zero knowledge, Data storage, Cloud computing

1. Introduction

A privacy-preserving public auditing system for data storage security in cloud computing in this the homomorphic linear authenticator and random masking to guarantee that the TPA[1] would not learn any knowledge about the data content stored on the cloud server during the efficient auditing process. It not only eliminates the burden of cloud user from the tedious and possibly expensive auditing task, but also alleviates the users' fear of their outsourced data leakage.

Using cloud storage, usre can remotly store their data and enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources, without the burden of local data storage and maintenance. However, the fact the user on longer have physical possession of the outsourced data makes the data integrity protection in cloud computing a formidable task, especially for the users with constrained computing resource. Enabling public audit ability for cloud storage is of critical importance so that user can resort to a third party auditor (TPA) to check the integrity of outsourced data and be worry-free. To securely introduce an effective TPA, the auditing process should bring in no new vulnerabilities towards user data privacy, and introduce no additional online burden to user. Here the a secure cloud storage system supporting privacy-preserving public auditing is proposed.

2. Literature Survey

The public auditability, i.e. "provable data possession" (PDP) is a model for ensuring possession of data files on untrusted storages. The scheme utilizes the RSA based homomorphic non-linear authenticators for auditing outsourced data and suggests randomly sampling a few blocks of the file. However, the public auditability in their scheme demands the linear combination of sampled blocks exposed to external auditor. When used directly, the protocol is not provably privacy preserving, and thus may leak user data information to the auditor. Juels et al. [11] describes a "proof of retrievability" (PoR) model, where spot-checking and error-correcting codes are used to ensure both possession and retrievability of data files on remote archive service systems. However, the number of audit challenges a user can perform is a fixed priori, and public auditability is not supported in their main scheme. Although they describe a straightforward Merkle-tree construction for public PoRs, this approach only works with encrypted data. Dodis et al. [5] give a study on different variants of PoR with private auditability. Shacham et al.[13] design an improved PoR scheme built with full proofs of security in the security model defined in [11]. Similar to the construction in [8], they use publicly verifiable homomorphic non-linear authenticators that are built from provably secure BLS signatures. Based on the elegant BLS construction, a compact and public verifiable scheme is obtained. Again, their approach does not support privacy preserving auditing for the same reason as [8]. The propose allowing a TPA to keep online storage honest by first encrypting the data then sending a number of pre computed symmetric-keyed hashes over the encrypted data to the auditor. The auditor verifies both the integrity of the data file and the server's possession of a previously committed decryption key. This scheme only works for encrypted files and it suffers from the auditor statefullness and bounded usage, which may potentially bring in online burden to users when the keyed hashes are used up. The dynamic version of the prior PDP scheme, using only symmetric key cryptography but with a bounded number of audits. Consider a similar support for partial dynamic data storage in a distributed scenario with additional feature of data error localization. In a subsequent work, Wang et al. [10] propose to combine BLS-based HLA with MHT to support both public auditability and full data dynamics. Almost simultaneously developed a skip lists based scheme to enable provable data possession with full dynamics support. However, the verification in these two protocols requires the linear combination of sampled blocks just as [8], [13], and thus does not support privacy preserving auditing. While all the above schemes provide methods for efficient auditing and

provable assurance on the correctness of remotely stored data, none of them meet all the requirements for privacy preserving public auditing in cloud computing. More importantly, none of these schemes consider batch auditing, which can greatly reduce the computation cost on the TPA when coping with a large number of audit delegations.

A.MAC Based Solution

It is used to authenticate the data. In this, user upload data blocks and MAC to CS provide its secret key SK to TPA. The TPA will randomly retrieve data blocks & Mac uses secret key to check correctness of stored data on the cloud. Problems with this system are listed below as it introduces additional online burden to users due to limited use (i.e. Bounded usage) and stateful verification.

- Communication & computation complexity
- TPA requires knowledge of data blocks for verification
- Limitation on data files to be audited as secret keys are fixed
- After usages of all possible secret keys, the user has to download all the data to recomputed MAC & republish it on CS.
- TPA should maintain & update states for TPA which is very difficult
- It supports only for static data not for dynamic data.

B. HLA Based Solution

It supports efficient public auditing without retrieving data block. It is aggregated and required constant bandwidth. It is possible to compute an aggregate HLA which authenticates a linear combination of the individual data blocks.

C. Using Virtual Machine

Abhishek Mohta proposed Virtual machines which uses RSA algorithm, for client data/file encryption and decryptions [5]. It also uses SHA 512 algorithm which makes message digest and check the data integrity. The Digital signature is used as an identity measure for client or data owner. It solves the problem of integrity, unauthorized access, privacy and consistency.

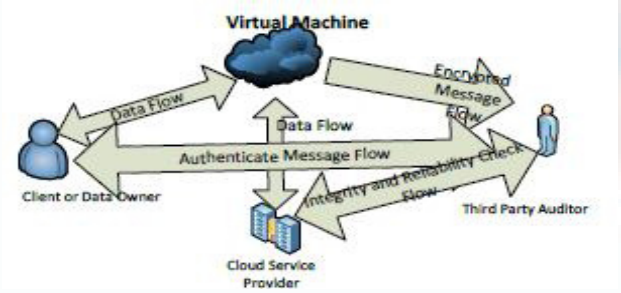

Figure 2.1: Architecture of Cloud server with CU and TPA

D. Non Linear Authentication

D. Shrinivas suggested Homomorphic non linear authenticator with random masking techniques to achieve cloud security [7]. K. Gonvinda proposed digital signature method to protect the privacy and integrity of data [8]. It uses RSA algorithm for encryption and decryption which follows the process of digital signatures for message authentication.

E. Using EAP

S. Marium proposed use of Extensible authentication protocol (EAP) through three ways hand shake with RSA. They proposed identity based signature for hierarchical architecture. They provide an authentication protocol for cloud computing (APCC) [9]. APCC is more lightweight and efficient as compared to SSL authentication protocol. In this, Challenge handshake authentication protocol (CHAP) is used for authentication. When make request for any data or any service on the cloud. The Service provider authenticator (SPA) sends the first request for client identity. The steps are as follows

1. When Client request for any service to cloud service provider, SPA send a CHAP request / challenge to the client.
2. The Client sends CHAP response/ challenges which is calculated by using a hash function to SPA
3. SPA checks the challenge value with its own calculated value. If they are matched then SPA sends CHAP success message to the client. Implementation of this EAP-CHAP in cloud computing provides authentication of the client. It provides security against spoofing identity theft, data tempering threat and DoS attack. The data is being transferred between client and cloud providers. To provide security, asymmetric key encryption (RSA) algorithm is used. B. Dhiyanesh proposed Mac based and signature based schemes for realizing data audit ability and during auditing phase data owner provides a secret key to cloud server and ask for a MAC key for verification [11].

F. Using Automatic Protocol Blocker

Balkrishna proposed efficient reed Solomon technique for error correction which check data storage correctness [13].

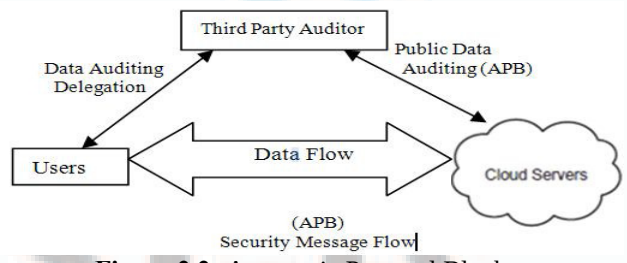

Figure 2.2: Automatic Protocol Blocker

Kiran Kumarproposed automatic protocol blocker to avoid unauthorized access [14]. When an unauthorized user access user data, a small application runs which monitors user inputs, It matches the user input, if it is matched then it allow user to access the data otherwise it will block protocol automatically. It contains five algorithms as keygen, SinGen, GenProof, VerifyProof, Protocol Verifier. Protocol Verifier is used by CS. It contains three phases as Setup, Audit and PBlock.

G. Random Masking Technique

Jachak K. B. proposed privacy preserving Third party auditing without data encryption. It uses a linear combination of sampled block in the server's response is masked with randomly generated by a pseudo random function (PRF) [16].

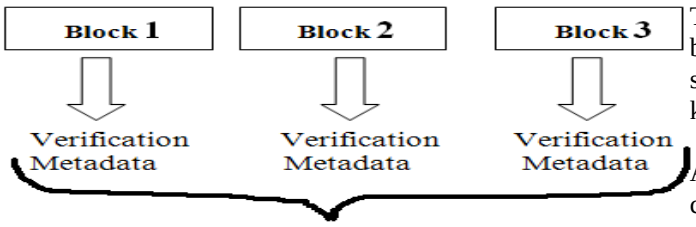

Figure 2.3: Homomorphic Authenticator

H. Homomorphic Authenticator

Researchers of [17] use the concept of virtual machines, The RSA algorithm is used to encode and decode the data and SHA 512 algorithm is used for message digest which check the integrity of information Dr. P.K. Deshmukh uses the new password at each instance which will be transferred to the mail server for each request to obtain data security and data integrity of cloud computing [17]. This protocol is secure against an untrusted server as well as third party auditor. Client as well as trusted third party verifier should be able to detect the changes done by the third party auditor. The client data should be kept private against third party verifier. It supports public verifiability without help of a third party auditor. This protocol does not leak any information to the third party verifier to obtain data security. This proposed protocol is secure against the untrusted server and private against third party verifier and support data dynamics. In this system, the password is generated and that will be transferred to email address of the client. Every time a key is used to perform various operations such as insert, update delete on cloud data. It uses time based UUID algorithm for key generation based on pseudo random numbers. If an intruder tries to access the users' data on a cloud, that IP address will be caught and transferred to the user so that user will be aware of.

3. Problem Definition

3.1 System and Threat Model

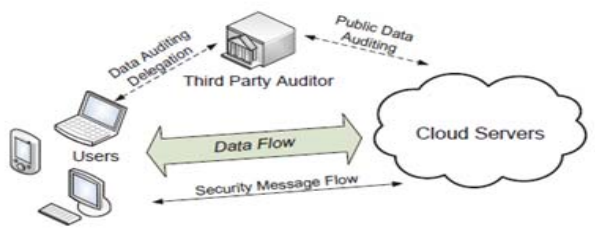

Figure 3.1: The architecture of Cloud Data Service

Consider a cloud data storage service involving three different entities, as illustrated in Fig. the cloud user (U), who has large amount of data files to be stored in the cloud; the cloud server (CS), which is managed by the cloud service provider (CSP) to provide data storage service and has significant storage space and computation resources, the third party auditor (TPA), who has expertise and capabilities that cloud users do not have and is trusted to assess the cloud storage service reliability on behalf of the user upon request. Users rely on the CS for cloud data storage and maintenance. They may also dynamically interact with the CS to access and update their stored data for various application purposes. To save the computation resource as well as the online burden, cloud users may resort to TPA for ensuring the storage integrity of their outsourced data, while hoping to keep their data private from TPA.

Assuming that the data integrity threats towards user data can come from both internal and external attacks at CS. These may include: software bugs, hardware failures, bugs in the network path, economically motivated hackers, malicious or accidental management errors, etc. Besides, CS can be self-interested. For their own benefits, such as to maintain reputation, CS might even decide to hide these data corruption incidents to users. Using third-party auditing service provides a cost-effective method for users to gain trust in cloud. Considering the TPA, who is in the business of auditing, is reliable and independent. However, it may harm the user if the TPA could learn the outsourced data after the audit.

Note that in our project, beyond users' reluctance to leak data to TPA, it is also assumed that cloud servers have no incentives to reveal their hosted data to external parties. On the one hand, there are regulations, e.g., HIPAA [2], requesting CS to maintain users' data privacy. On the other hand, as users' data belong to their business asset [3], there also exist financial incentives for CS to protect it from any external parties. Therefore neither CS nor TPA has motivations to collide with each other during the auditing process.

To authorize the CS to respond to the audit delegated to TPA's, the user can issue a certificate on TPA's public key, and all audits from the TPA are authenticated against such a certificate.

4. Design Goals

To enable privacy-preserving public auditing for cloud data storage under the aforementioned model,
Our protocol design should achieve the following security and performance guarantees.

1. Public auditability: To allow TPA to verify the correctness of the cloud data on demand without retrieving a copy of the whole data or introducing additional online burden to the cloud users.
2. Storage correctness: To ensure that there exists no cheating cloud server that can pass the TPA's audit without indeed storing users' data intact.
3. Privacy-preserving: To ensure that the TPA cannot derive users' data content from the information collected during the auditing process.
4. Lightweight: To allow TPA to perform auditing with minimum communication and computation overhead.

International Journal of Scientific Engineering and Research (IJSER)
www.ijser.in
ISSN (Online): 2347-3878
Volume 2 Issue 6, June 2014

5. The Proposed Scheme

This section presents public auditing scheme which provides a complete outsourcing solution of data and its integrity checking.

4.1 Definitions and Framework

A public auditing scheme consists of four algorithms (KeyGen, SigGen, GenProof, VerifyProof). KeyGen is a key generation algorithm that is run by the user to setup the scheme. SigGen is used by the user to generate verification metadata, which may consist of MAC, signatures, or other related information that will be used for auditing. GenProof is run by the cloud server to generate a proof of data storage correctness, while VerifyProof is run by the TPA to audit the proof from the cloud server.

Running a public auditing system consists of two phases, Setup and Audit:

1. Setup: The user initializes the public and secret parameters of the system by executing KeyGen, and preprocesses the data file F by using SigGen to generate the verification metadata. The user then stores the data file F and the verification metadata at the cloud server, and delete its local copy. As part of pre-processing, the user may alter; the data file F by expanding it or including additional metadata to be stored at server.
2. Audit: The TPA issues an audit message or challenge to the cloud server to make sure that the cloud server has retained the data file F properly at the time of the audit. The cloud server will derive a response message from a function of the stored data file F and its verification metadata by executing GenProof. The TPA then verifies the response via VerifyProof. Our framework assumes the TPA is stateless, which is a desirable property achieved by our proposed solution.

The TPA is stateless, i.e., TPA does not need to maintain and update state between audits, which is a desirable property in the public auditing scheme [13].

4.2 Privacy-Preserving Public Auditing Scheme

Overview: To achieve privacy-preserving public auditing, we propose to uniquely integrate the homomorphic linear authenticator with random masking technique. In our protocol, the linear combination of sampled blocks in the server's response is masked with randomness generated the server. With random masking, the TPA no longer has all the necessary information to build up a correct group of linear equations and therefore cannot derive the user's data content, no matter how many linear combinations of the same set of file blocks can be collected. On the other hand, the correctness validation of the block authenticator pairs can still be carried out in a new way which will be shown shortly, even with the presence of the randomness. Our design makes use of a public key based HLA, to equip the auditing protocol with public auditability. Specifically, we use the HLA proposed in [13], which is based on theshort signature scheme proposed by Boneh, Lynn and Shacham (hereinafter referred as BLS signature) [17].

Scheme Details: Let G1, G2 and GT be multiplicative cyclic groups of prime order p, and $e : G_1 \times G_2 \rightarrow G_T$ be a bilinear map as introduced in preliminaries. Let g be a generator of G_2. H(·) is a secure map-to-point hash function: {0, 1}*→ G_1, which maps strings uniformly to G_1. Another hash function $h(\cdot) : G_T \rightarrow Zp$ maps group element of G_T uniformly to Zp.

The proposed scheme is as follows:

Setup Phase: The cloud user runs KeyGen to generate the public and secret parameters. Specifically, the user chooses a random signing key pair (spk, ssk), a random $x \leftarrow Zp$, a random element $u \leftarrow G_1$, and computes $v \leftarrow g^x$. The secret parameter is sk = (x, ssk) and the public parameters are pk =(spk, v, g, u, e(u, v)).

Given a data file F = (m1, . . . ,m_n), the user runs SigGen to compute authenticator for each block mi: $\alpha_i \leftarrow (H(W_i) \cdot u^{mi})^x \in G1$ for each i. Here W_i = name||i and name is chosen by the user uniformly at random from Zp as the identifier of file F. Denote the set of authenticators by $\Phi = \{ \alpha_i \}1 \leq i \leq n$.

The last part of SigGen is for ensuring the integrity of the unique file identifier name. One simple way to do this is to compute t = name||SSigssk(name) as the file tag for F, where SSigssk(name) is the signature on name under the private key ssk. For simplicity, we assume the TPA knows the number of blocks n. The user then sends F along with the verification metadata (Φ, t) to the server and deletes them from local storage.

Audit Phase: The TPA first retrieves the file tag t. With respect to the mechanism we describe in the Setup phase, the TPA verifies the signature SSigssk(name) via spk, and quits by emitting FALSE
if the verification fails. Otherwise, the TPA recovers name.

Now it comes to the "core" part of the auditing process. To generate the challenge message for the audit "chal", the TPA picks a random c-element subset I = {s1. . . sc} of set [1, n]. For each element i ∈ I, the TPA also chooses a random value v_i (of bit length that can be shorter than |p|, as explained in [13]). The message "chal" specifies the positions of the blocks that are required to be checked. The TPA sends chal = $\{(i, v_i)\}_{i \in I}$ to the server.

Upon receiving challenge chal = $\{(i, \_i)\}_{i \in I}$, the server runs GenProof to generate a response proof of data storage correctness. Specifically, the server chooses a random element $r \leftarrow Zp$, and calculates $R = e(u, v)^r \in G_T$. Let μ′ denote the linear combination of sampled blocks specified in chal: $\mu' = \sum_{i \in I} v_i m_i$ To blind μ′ with r, the server computes: $\mu = r + \gamma\mu' \bmod p$, where $\gamma = h(R) \in Zp$. Meanwhile, the server also calculates an aggregated authenticator $\alpha = \pi_{i \in I} \alpha^{vi} \in G1$.Itthen sends {μ, α, R} as the response proof of storage correctness to the TPA. With the response, the TPA runs VerifyProof to validate it by first computing γ= h(R) and then checking the verification equation.

International Journal of Scientific Engineering and Research (IJSER)
www.ijser.in
ISSN (Online): 2347-3878
Volume 2 Issue 6, June 2014

$$R \cdot e(\sigma^{\gamma}, g) \stackrel{?}{=} e((\prod_{i=s_1}^{s_c} H(W_i)^{\nu_i})^{\gamma} \cdot u^{\mu}, v) \quad (1)$$

The protocol is illustrated in Table 4.1. The correctness of the above verification equation is elaborated as follows:

$$\begin{aligned} R \cdot e(\sigma^{\gamma}, g) &= e(u, v)^{r} \cdot e((\prod_{i=s_1}^{s_c} (H(W_i) \cdot u^{m_i})^{x \cdot \nu_i})^{\gamma}, g) \\ &= e(u^{r}, v) \cdot e((\prod_{i=s_1}^{s_c} (H(W_i)^{\nu_i} \cdot u^{\nu_i m_i})^{\gamma}, g)^{x} \\ &= e(u^{r}, v) \cdot e((\prod_{i=s_1}^{s_c} H(W_i)^{\nu_i})^{\gamma} \cdot u^{\mu' \gamma}, v) \\ &= e((\prod_{i=s_1}^{s_c} H(W_i)^{\nu_i})^{\gamma} \cdot u^{\mu' \gamma + r}, v) \\ &= e((\prod_{i=s_1}^{s_c} H(W_i)^{\nu_i})^{\gamma} \cdot u^{\mu}, v) \end{aligned}$$

TPA		Cloud Server
1. Retrieve file tag t, verify its signature, and quit if fail;		
2. Generate a random challenge $chal = \{(i, \nu_i)\}_{i \in I}$;	$\xrightarrow{\{(i,\nu_i)\}_{i\in I}}$ challenge request $chal$	3. Compute $\mu' = \sum_{i \in I} \nu_i m_i$, and also $\sigma = \prod_{i \in I} \sigma_i^{\nu_i}$;
		4. Randomly pick $r \leftarrow \mathbb{Z}_p$, and compute $R = e(u, v)^r$ and $\gamma = h(R)$;
	$\xleftarrow{\{\mu,\sigma,R\}}$ storage correctness proof	5. Compute $\mu = r + \gamma\mu' \mod p$;
6. Compute $\gamma = h(R)$, and then verify $\{\mu, \sigma, R\}$ via Equation 1.		

Table 4.1: The privacy-preserving public auditing protocol

Specifically, the server chooses a random element r ← Zp, and calculates R = e(u, v)^{r} ЄG$_T$. Let μ′ denote the linear combination of sampled blocks specified in chal: μ′ = $\sum_{i \in I} v_i m_i$ To blind μ′ with r, the server computes: μ = r+γμ′ mod p, where γ= h(R) Є Zp. Meanwhile, the server also calculates an aggregated authenticator α=$\pi_{i \in I}$ α^{vi}ЄG1.Itthen sends {μ, α, R} as the response proof of storage correctness to the TPA. With the response, the TPA runs VerifyProof to validate it by first computing γ= h(R) and then checking the verification equation.

Properties of the Protocol: It is easy to see that our protocol achieves public auditability. There is no secret keying material or states for the TPA to keep or maintain between audits, and the auditing protocol does not pose any potential online burden on users. This approach ensures the privacy of user data content during the auditing process by employing a random masking r to hideμ, a linear combination of the data blocks. Note that the value R in our protocol, which enables the privacy preserving guarantee, will not affect the validity of the equation, due to the circular relationship between R and in γ = h(R) and the verification equation. Storage correctness thus follows from that of the underlying protocol [13]. Besides, the HLA helps achieve the constant communication overhead for server's response during the audit: the size of {μ, α, R} is independent of the number of sampled blocks.

6. Conclusions

Here proposed is a privacy-preserving public auditing system for data storage security in Cloud Computing. The homomorphic linear authenticator and random masking guarantees that the TPA would not learn any knowledge about the data content stored on the cloud server during the efficient auditing process, which not only eliminates the burden of cloud user from the tedious and possibly expensive auditing task, but also alleviates the users' fear of their outsourced data leakage.

7. Future Work

With the establishment of privacy-preserving public auditing in cloud computing, TPA may concurrently handle multiple auditing delegations upon different user's requests. The individual auditing of these tasks for TPA can be tedious and inefficient. Batch auditing not only allows TPA to perform the multiple auditing tasks simultaneously, but also reduces the computation cost on TPA side. And also we can extend our work to support for the data dynamics which includes the block level operations of modification, deletion, insertion.

References

[1] Krebs, "Payment Processor Breach May Be Largest Ever," Online at, http://voices.washingtonpost.com/securityfix/2009/01/payment processor breach may b.html, Jan. 2009.

[2] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in Proc. of ESORICS'09, volume 5789 of LNCS Springer-Verlag, Sep. 2009, pp. 355–370.

[3] H. Shacham and B. Waters, "Compact proofs of retrievability," in Proc. of Asiacrypt 2008, vol. 5350, Dec 2008, pp. 90–107.

[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. In Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007, pages 598–609, 2007.

[5] Cloud Security Alliance, "Security guidance for critical areas of focus in cloud computing," 2009, http://www. cloudsecurityalliance.org.

[6] Dr. P. K. Deshmukh, Mrs. V. R. Desale, Prof. R. A. Deshmukh, "Investigation of TPA (Third Party Auditor Role) foe Cloud Data Security", International Journal of Scientific and Engineering Research, vo. 4,no. 2,ISSn 2229-5518, Feb 2013.

[7] Jachak K. B., Korde S. K., Ghorpade P. P. and Gagare G. J. ,"Homomorphic Authentication with Random Masking Technique Ensuring Privacy & Security in Cloud Computing", Bioinfo Security Informatics, vol. 2, no. 2, pp. 49-52, ISSN. 2249-9423, 12 April 2012

[8] K. Kiran Kumar, K. Padmaja, P. Radha Krishna, "Automatic Protocol Blocker for Privacy-Preserving Public Auditing in Cloud Computing", International Journal of Computer science and Technology, vol. 3 pp, ISSN. 0976-8491(Online), pp. 936-940, ISSN: 2229-4333 (Print), March 2012
[9] Balkrishnan. S, Saranya. G, Shobana. S and Karthikeyan. S, "Introducing Effective Third Party Auditing (TPA) for Data Storage Security in Cloud", International Journal of computer science and Technology, vol. 2, no. 2, ISSN 2229-4333 (Print) | ISSN: 0976-8491(Online), June 2012
[10] Dhiyanesh "A Novel Third Party Auditability and Dynamic Based Security in Cloud Computing", International Journal of Advanced Research in Technology, vol. 1,no. 1, pp. 29-33, ISSN: 6602 3127, 2011
[11] S. Marium, Q. Nazir, A. Ahmed, S. Ahthasham and Aamir M. Mirza, "Implementation of EAP with RSA for Enhancing The Security of Cloud Computig", International Journal of Basic and Applied Science, vol 1, no. 3, pp. 177- 183, 2012
[12] K Govinda, V. Gurunathprasad and H. sathishkumar, " Third Party Auditing for Secure Data Storage in Cloud Through Digital Signature Using RSA", International Journal of Advanced science and Technical Research, vol 4,no. 2, ISSN: 2249-9954,4 August 2012
[13] Shrinivas, "Privacy-Preserving Public Auditing in Cloud Storage security", International Journal of computer science nad Information Technologies, vol 2, no. 6, pp. 2691-2693, ISSN: 0975-9646, 2011
[14] XU Chun-xiang, HE Xiao-hu, Daniel Abraha, "Cryptanalysis of Auditing protocol proposed by Wang et al. for data storage security in cloud computing", http://eprint.iacr.org/2012/115.pdf, and cryptologyeprintarchieve: Listing for 2012
[15] Abhishek Mohta, Lalit Kumar Awasti, "Cloud Data Security while using Third Party Auditor", International Journal of Scientific & Engineering Research, Volume 3, Issue 6, ISSN 2229-8 June 2012.

Author Profile

Ms. Jyoti Bolannavar, P G Student, Department of Computer Science, Gogte Institute Of Technology, Belgaum, Karnataka, India.

Under the guidance of:

Prof. P. M. Pujar, M. Tech. in CSE, Assistant Professor, Department of Computer Science, Gogte Institute Of Technology, Belgaum, Karnataka, India

Prof. Sharada Kulkarni, M. Tech in CSE, Assistant Professor, Department of Computer Science, Gogte Institute Of Technology, Belgaum, Karnataka, India