Public Auditability of Data for Checking Integrity in Cloud Storage

Vaishali Tupe
M.E. Student, Dept. of Computer Engg,
Shah and anchor Kutchhi Engg college,
Chembur, Mumbai-400088, India

Vidyullata Devmane
Assistant Professor, Dept . of Computer Engg,
Shah And Anchor Kutchhi Engg. College,
Chembur, Mumbai-400088, India

Abstract - Cloud computing is a latest technology, It consists of two words cloud and computing in which cloud refer to network and computing consist of services such as a software as a service, infrastructure as a service and platform as a service. One of the most important service provided by cloud computing is cloud data storage in which user store there data on a cloud server and it gets relived from the burden of storage overhead. User can store data remotely without maintaining local copy of data but achieving the integrity of the data is major problem. Recently many works focuses on providing public verifiability for checking the remote data integrity. In this work the integrity and confidentiality of data in cloud storage is achieved by using Homomorphic Encryption technique. Online burden of user or a owner can be removed by using public auditability for cloud data storage, that use external audit party to check integrity of outsourced data. Third party auditor (TPA) to audit outsourced data without demanding local copy of data. No additional online burden for the cloud owner, server and that can be achieved by using Privacy-Preserving Public key based homomorphic encryption scheme like Elgamal.

Keywords : Third Party Auditability, Data integrity in cloud storage Data security in cloud

I. INTRODUCTION

Cloud computing is an internet based computing in which computing resources such as hardware software and information is stored in large data centers made available to the public on demand. The well known cloud storage providers are Amazon, Google, Salesforce.com, and Microsoft. These companies provides on demand, virtualized, scalable resources to the cloud user. Cloud data storage is one of the important service in which data owner stores there data on cloud server. Prior to the development of the concept of cloud computing, critical industrial data was stored internally on storage media, protected by security measures including firewalls to prevent external access to the data and including organizational regulations to prohibit unauthorized internal access. Storing the data on personal devices, users have the highest privilege to operate on it and ensure its security. But once data move to the cloud, it is handled by cloud service provider. Sometimes the cloud service provider may dishonest, they may hide the data corruption to maintain their reputation. The cloud services also suffer from internal and external data attacks. So, data owners must take the security issues in cloud storage service into account. Traditionally the data integrity is measured by traditional hash algorithm, as traditional hash algorithms require original data so not suitable for privacy preserving. There are two ways to verify data integrity in cloud storage system [1] the owner auditing and the public auditing[2] [3]. With owner auditing, only owner checks the integrity of their remotely stored data, which could introduces heavy overhead and cost. Avoiding any side of, cloud service provider or the data owner conducting the auditing, public auditing, transferring the auditing procedure to third party auditor (TPA), is a natural choice.

In the existing system basic concept was encrypt the data before sending to the cloud server, but this one will have to decrypt each time when user want to work on the data, which break the confidentiality of data. Thus to hide the data from the cloud and to verify the correctness of data, the homomorphic encryption techniques such as RSA and ELGamal is the most suitable option in cloud environment. Homomorphic encryption allows operation on encrypted data without learning knowledge on data. In these work the data security in cloud environment is achieved using ELGAMAL cryptosystem with the help of public audit system model.

II. RELATED WORK

Ateniese et al. [4] has proposed public auditability in their "provable data possession" (PDP) model for ensuring possession of data files on untrusted storages. They utilize the RSA-based homomorphic linear authenticators for auditing outsourced data and suggest randomly sampling a few blocks of the file. This scheme provides data access to external auditor which may cause security problems

Juels et al. [5] use the concept of a "proof of retrievability" (PoR) model, where spot-checking and error-correcting codes areused to ensure both "possession" and "retrievability" of data files on remote archive service systems. The limitation of this solution is that the number of audits is fixed. Moreover this works with only encrypted data.

The authors complete their dynamic auditing system to be privacy preserving and it support the batch auditing for multiple owners [6]. However, due to the large number of data tags, their auditing protocols will incur a heavy storage overhead on the server.

In [7], the authors proposed a dynamic auditing protocol that can support the dynamic operations of the data on the cloud servers, but this method may leak the data content to the auditor because it requires the server to send the linear combinations of data blocks to the auditor.

III. PROPOSED SYSTEM

This section provides information about proposed system model such as public audit system in which data security is achieved by using homomorphic encryption techniques such as ELGAML cryptosystem.

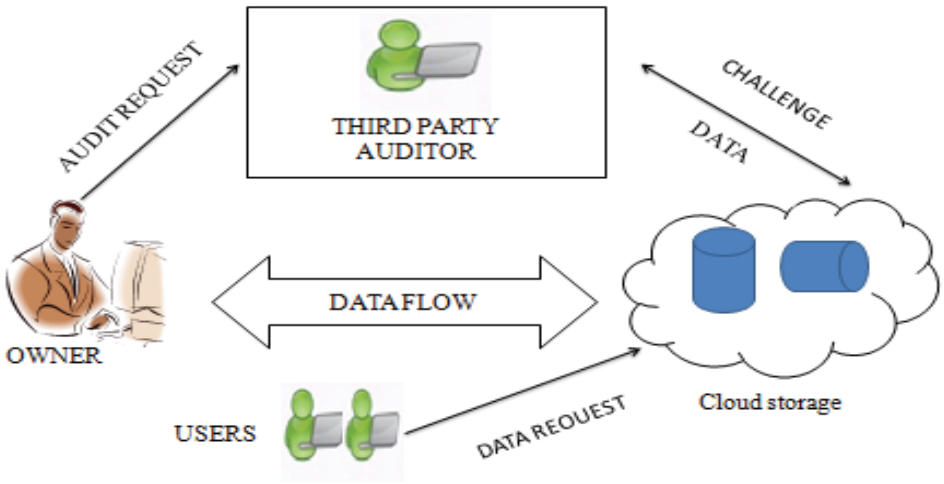

Figure. 1 block diagram of public audit system

As given in figure 1three entities are involved in data storage and sharing services in cloud environment.

1. User
2. Cloud server
3. Third party auditor

There are number of users in a group and one of them is data owner who will upload his data on a cloud server and share it with the other users. The authorized user who registered on cloud server can download the data. When owner or user can request third party auditor to check the integrity of data. Upon receiving the request from the owner, the third party auditor send an audit message to the cloud server and retrieve an audit proof about the data. With the help of that proof the third party auditor verify the correctness of data and finally send audit report to the owner or user. Third party auditor verifies the correctness of data without demanding local copy of data and introduces no additional burden on cloud server. TPA can audit user's outsourced data in cloud environment without

gaining knowledge on data contents. This can be implementing by using homomorphic encryption technique such as ELGamal.

Proposed work can provide:

- Light weight : TPA perform the auditing with minimum communication and computation overhead
- Storage correctness: TPA can audit the client's data without learning knowledge on data contents.
- This scheme support privacy preserving public auditing storage in cloud environment.

IV. IMPLEMENTATION

The ElGamal encryption system is an asymmetric key encryption algorithm for public-key cryptography which is based on the Diffie–Hellman key exchange. It was developed by Taher Elgamal in 1984. ElGamal encryption can be defined over any cyclic group G. Its security depends upon the difficulty of a certain problem in G related to computing discrete logarithms [8]. Data has got encrypted, tag has got calculated, data is stored on cloud and tag is with TPA for audit purpose.

Key Generation:

- Generate a large random prime p and choose generator

  g of the multiplicative group Zp* = {1, 2, ..., p-1}
- calculate public key

  public key is (p , g , $h = g^x \bmod p$)
- private key is x random integer in (1, p-2)
- ENCRYPTION

  Select random exponent k such that $0 < k < p-2$

  Compute chiper text

  $C1 = (g^k \bmod p)$

  $C2 = (h^k \bmod p) * M$
- DECRYPTION

  Compute $R = (c1)^{p-1-x} \bmod p$

  $m = (R * C2) \bmod p$

Example :

- P = 277, g = 183
- $x = 14$ $h = g^x \bmod p = 183^{14} \bmod 277 = 206$, k1 = 33
- Calculate the chpertext (C1 , C2)
- Compute $C1 = g^{k1} \bmod p$
  $= 183^{33} \bmod 277 = 61$
- Compute $S = (h^k) \bmod p$
  $= 206^{33} \bmod 277$
  $= 76$
- Cipher text = (m1 * s)
- Assume m1 = 7
- $C2 = (7 * 76) \bmod 277 = 255$

Decryption

- Compute $R= (61)^{277 -1 - 14} \bmod 277 = 113$
- $m = (113 * 255) \bmod 277 = 7$

Similarly if we take k2 = 22 and m = 2

- B1 = g^{k2} mod p = 183^{22} mod 277 = 220
- S = 206^{22} mod 277 = 185
- B2 = (2 * 185) mod 277 = 93

Decryption

- Compute R= $(220)^{277-1-14}$ mod 277 = 3
- m = (3 * 93) mod 277 = 2

Homomorphic Property

- (C1* B1 ,C2* B2) = (61*220 , 255*95) = (13420, 23715)

Decryption

- $((C1\ B1)^{P-1-X}$ mod p) * (C2 B) mod p = (62. 23715) = 14 = (7 * 2) = (m1 * m2)

The encryption message M1 and M2 using chipertext C1, C2 and B1, B2 and the random values are K1 and K2 respectively then the homomorphic properties are:

- (C1, C2) (B1, B2) = (C1* B1,C2*B2)

  = g^{k1} g^{k2} (m1 . s1) (m2 . s2))

  = (g^{k1+k2}, m1 m2 . s1+s2) is valid encryption of m1 m2.

ElGamal encryption is probabilistic, meaning that a single plaintext can be encrypted to many possible ciphertexts, with the consequence that a general ElGamal encryption produces a 2:1 expansion in size from plaintext to ciphertext. Encryption under ElGamal requires two exponentiations; however, these exponentiations are independent of the message and can be computed ahead of time if need be. Decryption only requires one exponentiation:

- There are efficient honest-verifier zero-knowledge proofs of knowledge to prove properties of ElGamal ciphertexts without revealing the plaintext.
- Encryption process of Elgamal is faster as compared to RSA.

This work has been deployed in ulteo cloud platform. Ulteo cloud is Open Virtual Desktop (OVD) was about delivering Linux and/or Windows applications to virtual desktops accessible from within or outside an organization [9].

System Requirements:

- OVD Application servers for Linux applications: x86 servers w/multi-core or quad CPU w/1GB or more RAM. Count 8GB or more RAM per 100 concurrent end-users. Supported Host OS (32 or 64 bit): Ubuntu 10.04.*, 12.04.*, RHEL 6.0, Centos 6.0, Open SuSE 11.3, SLES 11SP1.
- OVD Application servers for Windows applications: Windows 2003 (32 bit), 2008R2 (64bit) Server + Active Directory and Terminal Services on any hardware. 1GB or more RAM. Count a minimum of 12GB or more RAM per 100 concurrent end-users.

V. EXPERIMENTAL RESULT

Table 1. Encryption and Decryption time for different file size and Tag size for respective files

.

Block size in bytes	File size in kb	Encryption time in bytes/sec	Decryption time in bytes/sec	Size of hash data file
32	**4**	**5**	**2.5**	**40 bytes**
	6	**8**	**4**	
	8	**14**	**7**	

64	**4**	**3**	**1.5**	**40 bytes**
	6	**4**	**2**	
	8	**7**	**3.5**	
128	**4**	**1**	**0.5**	**40 bytes**
	6	**2**	**1**	
	8	**4**	**2**	

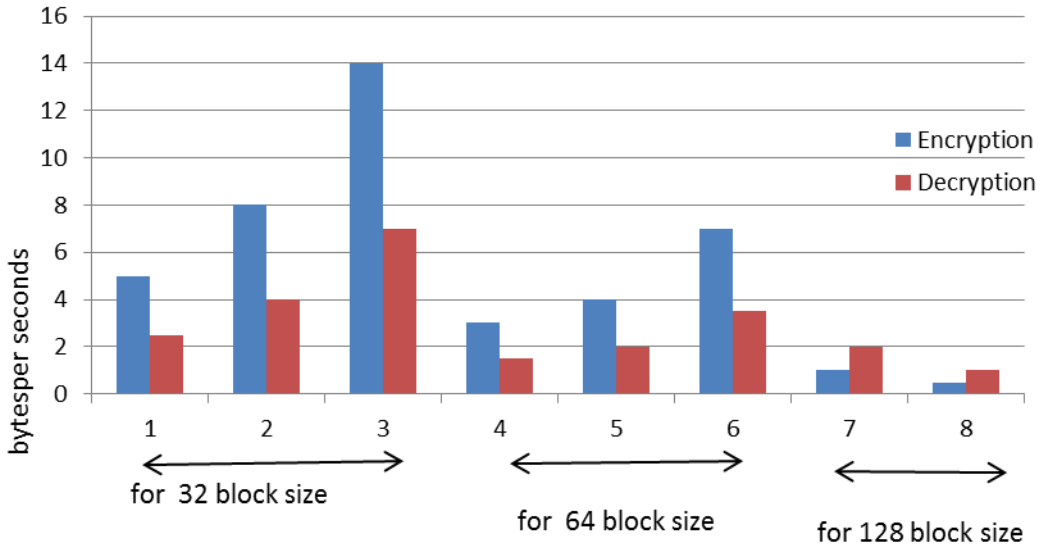

Figure 2. Encryption and Decryption time of different files

In the above table number 1 and graph given in fig. 2 time required to compute files encryption and decryption is mentioned along with their respective tag size. The files are taken with 4KB, 6KB and 8KB and considered in blocks size like 32 bytes, 64 bytes and 128 bytes to see the required time of encryption and decryption on it.

VI. CONCLUSION

In this work a privacy preserving public auditing system for data storage security has got implemented in cloud storage. Public auditability which plays very important role in cloud computing features which allows Third party auditor to verify the correctness of the cloud data without retrieving a copy of the whole original data or introducing additional online burden to the cloud user. Third party storage auditing service is efficient and secure. It protects the data privacy against the auditor by using the homomorphic cryptographic method such as ElGamal. The goal of this cryptographic scheme is to ensure confidentiality and integrity of data in communication and storage process. Third Party Storage Auditing Service incurs less communication cost and less computation cost of the auditor by moving the computing loads of auditing from the auditor to the server.

REFERENCES

[1] K. Yang and X. Jia,. “Data Storage Auditing Service in Cloud Computing: Challenge, Methods and Opportunities”, pp. 409-428, 2012.

[2] C. Wang, Q. Wang, K. Ren, and W. Lou, “Ensuring Data Storage Security in Cloud Computing,” IWQoS’09,Charleston, South Carolina, USA, 2009.

[3] Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S. S. Yau, "Cooperative Provable Data Possession," Cryptology Print Archive, Report 2010/234, 2010.

[4] 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.

[5] A. Juels and J. Burton, S. Kaliski, "PORs: Proofs of Retrievability for Large Files," Pro ACM Conf. Computer and Comm. Security (CCS'07), pp. 584-597, Oct. 2007.

[6] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing," Proc. IEEEINFOCOM, pp. 525-533, 2010.

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

[8] T. ElGamal, "A public-key cryptosystem and a signature scheme based on discrete logarithms", IEEE Trans. Inf. Theory 31(4), 469–472 (1985)

[9] https://www.ulteo.com/home/en/news (11/06/2013)