Securing Outsourced Data On Cloud Using ElGamal Cryptosystem

Arockia Panimalar.S[1], Subhashri.K[2]

[1,2] Assistant Professor, Department of BCA & M.Sc SS, Sri Krishna Arts and Science College, Tamilnadu

---***---

Abstract - *Cloud computing is a popular technology which grants storing and accessing data over Internet instead of storing it on local machine's hard drive. Cloud empowers users to store their data on cloud without dreading about its accuracy and reliability. However storing data on cloud forces certain security challenges. Outsourcing data in cloud result in data owners losing their physical control over the data. Certain Cloud Service Providers (CSPs) may work unscrupulously with the cloud user's data, they may sneak the data from cloud and sell it to third parties in order to earn profit. In this manner despite the fact that outsourcing data on cloud is economical and decreases long duration storage and maintenance complexity, there is minimum assurance of data integrity, protection, security and availability on cloud servers. Various arrangements have been prescribed to explain the security issues in cloud. This paper concentrates on the integrity verification methodology for outsourced data. The proposed scheme combines the encrypting mechanism along with integrity verification strategy. The encrypting scheme utilized here is public key cryptographic algorithm like ElGamal and SHA-256 hash function is utilized for ensuring data storage correctness on untrusted server.*

Key Words: **Cloud Computing, ElGamal Algorithm, Public Auditing, SHA-256, TPA**

1. INTRODUCTION

Cloud Computing provides services which permits users to upload their data remotely on cloud servers and access that data everywhere around the network at any time. Cloud computing offers different service models such as SaaS (software as a service), IaaS (infrastructure as a service), PaaS (platform as a service), STaaS (storage as a service), SECaaS (Security as a service) & many more. Storage is one of the most commonly used cloud services. It provides many advantages, client can store their files on cloud to avoid the inconvenience of storing and upholding the data files locally. Also it provides data access from any geographical location and reduces the hardware and software maintenance. However, since the stored data is on cloud server i.e. at a remote location, how to get the verification about the stored data. Since the cloud users do not have physical check over outsourced data, this makes data integrity checking in cloud environment a significant job.

One of the major challenges in cloud storage service is cloud data integrity verification. One easy way is to load the entire data files on local system and carry out the integrity checking. However, this results in severe I/O overhead on the server and increases the network traffic to transmit the entire data file over the network. Also it may be difficult to identify the corrupted data files while accessing the stored data and thus their recovery might get too late. Therefore to assure data integrity and privacy, it is necessary to introduce an effective method for clients to validate the authenticity of the data stored on the cloud. To completely guarantee on cloud user’s data integrity, it is more significant to allow public auditing service for client’s outsourced data.

Public auditing service makes use of an auditor, usually a Third Party Auditor (TPA) whose job is to frequently audit the data files uploaded by data owner on cloud. These TPAs possess knowledge and expertise that clients do not and are allowed to check the integrity of client’s outsourced files on cloud when needed. Thus the Third Party Auditing mechanism provides an efficient solution for cloud users to verify their data storage correctness on cloud. The service providers can also gain valuable insights from the audit results provided by these TPAs which can further help to improve their cloud service.

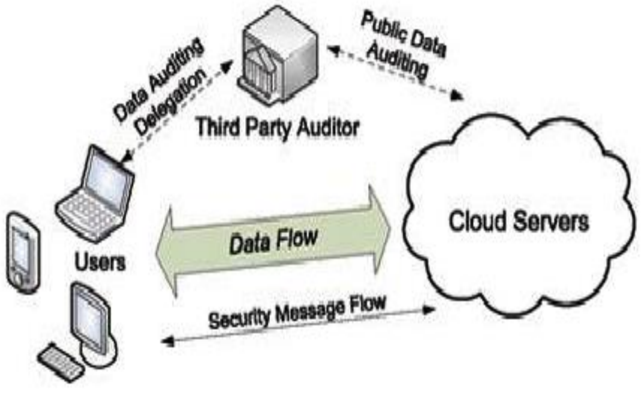

Fig 1: Cloud Storage Architecture

In public auditing, TPA will inquire the CSPs to prove that data files uploaded by a particular cloud user are safe and unmodified. However in this process, the original data gets revealed to Third Party auditors. TPA must not be permitted to access original data contents for security and privacy reasons. Thus encrypting the original data files before integrity verification is also necessary. The proposed system makes use of public key based ElGamal encryption scheme to improve data storage security in cloud.

2. SYSTEM MODEL

The system model consists of three main components: Cloud Service Provider, TPA and Data Owner.

i. Cloud Service Provider (CSP)

This entity possesses the infrastructure and proficiency to host indefinite and extendable data storage and computational resources.

ii. Data Owner (Client)

This entity utilizes cloud server to uphold huge volume of files and leaves IT operations on data to third party professionals and focuses on his/her business requirements.

iii. Third Party Auditor (TPA)

This entity possess more capabilities and expertise than cloud user and performs integrity check on cloud data on behalf of client and then sends report to client regarding the status of data.

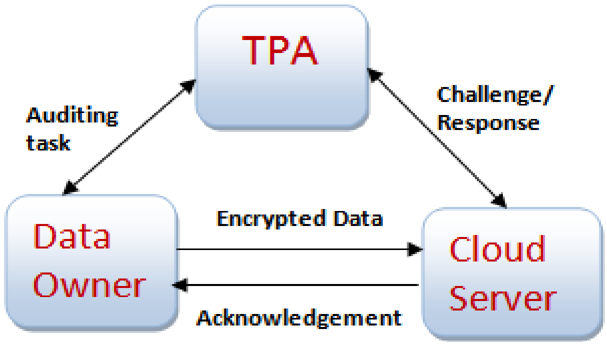

Fig 2: Cloud Data Storage Structure

A file F is uploaded by data owner on a cloud server. Owner encrypt file using ElGamal encryption scheme, which generates two keys; public key for encrypting data and secret key for decrypting encrypted contents. In addition, a secret hash key is also generated using SHA-256 for data integrity verification. Owner sends secret key to authorized users with whom the user wants to share the uploaded files and sends secret hash key to TPA for verifying cloud files integrity. Initially, the auditor challenges the CSP for initial verification of the entire data. The audit results are also sent to the data owner of the file.

Stage 1: Initialization

In this stage, before uploading data files on cloud server the owner generates the cryptographic keys and also the hash key.

The algorithm is given in Fig 3.

a. Data owner uploads data (file F) on cloud server.
b. Before uploading data on cloud server owner encrypt the data.
c. Generate keys i.e secret hash key(shk), secret key(sk), public key(pk)
d. Send the secret key to the authorized user.

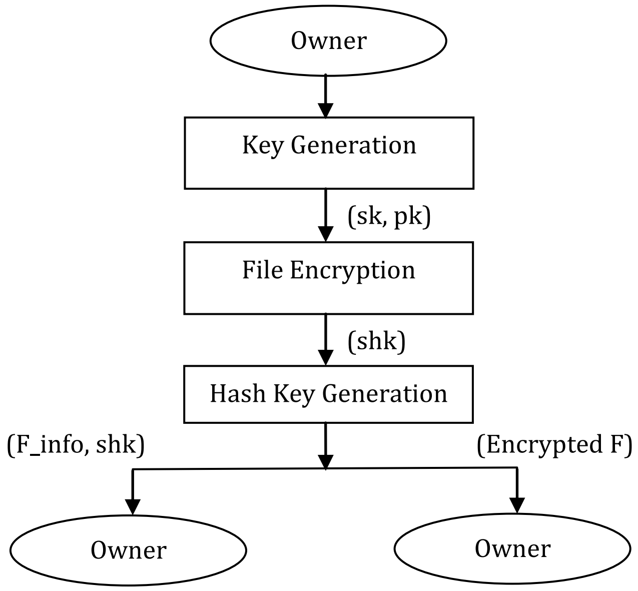

Fig 3: Initialization Stage

Stage 2: Verification Inspection

In this stage, the TPA checks whether the owner's data is correctly stored on the server (Fig 4). The algorithm is as follows:

a. The CSP stores the data files along with the metadata and the public key send by the owner.
b. Challenge (chall) is generated by the auditor on the basis of the secret hash key sent by the data owner.
c. The auditor sends challenge to the CSP.
d. The auditor will perform integrity checking on the basis of the response received from CSP.
e. If verify = successful then the data contents are original else modified.
f. The auditor then triggers the message to the data owner.

3. IMPLEMENTATION

The proposed system is implemented on an Intel core i5 processor system running at 2.20 GHz, 3GB RAM using Java and Ulteo OVD virtual desktop for building cloud environment. The implemented system consists of 5 modules: User registration, encrypt and upload, file sharing, decrypt and download and verification auditing.

A. User Registration

The registration function allows users to create secure account. Here the user enters his/her information necessary for signing up like user's name, password, mobile no and email-address. The validations and required fields are effectively handled. Each user will be provided his/her own space on cloud.

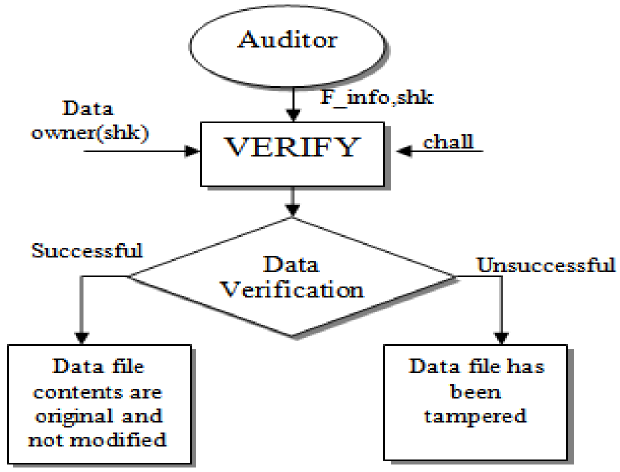

Fig 4: Verification Inspection

B. Encrypt and Upload

After registering, the user may login into the system. Every user is provided space on cloud where they may upload their files. The encrypt function will encrypt the data files of users before storing them on cloud storage using ElGamal cryptosystem. The owner will generate secrete hash key using SHA-256 and secrete key to download the uploaded file. The secret hash key is further mailed to TPA for data integrity verification. The time required to encrypt the files using ElGamal is also recorded.

C. File Sharing

The data owner may share the outsourced files with other users in cloud using the share module. The secret key generated during encryption is also mailed to the shared user in order to grant them access to the shared file. The shared user may download the file, make changes and again upload the file. In such a case, TPA informs the original data owner of that file about the latest modifications done by a shared user.

D. Decrypt and Download

The data owner or a shared user may need to download the file. Since the data files stored on cloud server are in encrypted form, decryption must be performed before downloading the file. Initially, the system validates whether the user requesting to download the file is a legitimate user by demanding the secret key from that user. The decryption module then performs data decryption using both RSA and ElGamal decryption scheme and downloads the data using secret key sent by the data owner. The time required to decrypt the file is also recorded.

E. Verification Auditing

In order to authenticate the integrity of the user's uploaded data, the TPA is granted access to the system. The TPA validates the integrity of the cloud data files on remote server on behalf of cloud user itself. TPA verifies the legitimacy of data using secret hash key sent by the cloud user. If the secret hash key matches with hash key in the cloud server, the verification proves to be successful, thus implying that the data files has not been modified. However, if the verification is unsuccessful, an email is dispatched to the data owner of the file informing about the last modifications done to his file.

4. ELGAMAL ENCRYPTION

The ElGamal cryptosystem[11] relies on discrete logarithm problem which implies that the discrete logarithms are difficult to compute in reasonable amount of time, whereas the inverse operations of the power are easy to compute. ElGamal is a public encryption algorithm which makes use of a random exponent k. This k is used in place of private exponent of receiver. Thus the entire operation is performed by one party i.e. the party which encrypts the data. Thus the encryption can be performed in one direction, without active participation of the second participant.

Following are the steps involved in ElGamal encryption algorithm:

A. Key Generation

The key generation process works as follows:
a. Assume a large prime number **p**.
b. Choose a primitive element **g** modulo p.
c. Choose a private key **a** randomly from {1, ..., p-1}.
d. Compute public key **y** as follows: a. y = ga mod p

B. Encryption

The encryption algorithm is as follows:
The plaintext is expressed as a set of numbers modulo p. Data owner encrypts a message M, CP be the ciphertext; CP comprises of two values ciphertext1 (y1) and ciphertext2 (y2).
a. Generate a random number k less than p
b. Compute two values y1 and y2 , where
y1 = gk mod p
y2 = M xor yk

c. Transmit the ciphertext CP consisting two values y1 and y2.

C. Decryption

Upon receiving the ciphertext CT (y1 and y2), the receiver computes original message M as:
M = (y1 a mod p) **xor** y2.

5. RESULT AND ANALYSIS

Table 1 and 2 provides analysis of the implemented system by comparing the encryption time and decryption time of RSA and ElGamal after taking different file sizes ranging from 1 KB to 1000 KB. The average encryption time required by RSA is 2655 milliseconds and that of ElGamal is 60235 milliseconds. The average decryption time required by RSA is 72671 milliseconds and that of ElGamal is 37724 milliseconds. Thus it is observed that RSA requires less time for encryption as compared to ElGamal, but requires more time for decryption. Elgamal proves to be faster in decryption as compared to RSA even for larger file size.

Table 1: Comparative Analysis of Encryption Time

File Size (KB)	RSA(ms)	ElGamal(ms)
1	1076	261
2	277	16
10	707	89
100	1353	1319
500	4243	33212
1000	8277	326752
Average	2655	60235

Table 2: Comparative Analysis of Decryption Time

File Size (KB)	RSA(ms)	ElGamal(ms)
1	733	310
2	706	250
10	3019	1010
100	26586	13463
500	186825	81228
1000	218162	130312
Average	72671	37724

6. CONCLUSION

To authenticate the integrity of data uploaded on cloud server, it is significant to permit a third party auditor to assess the quality of data content outsourced on cloud server. Public auditing system permits the clients to allocate the data integrity authentication tasks to a third party as they themselves can be unstable or may not possess essential computational resources to perform periodic integrity verifications. However, delegating data integrity verification task to a third party (TPA) raises privacy issues since the TPA may derive the actual data content from server during verification process. Thus the proposed system uses public auditing scheme for data storage security on cloud while protecting the confidentiality of the user's data. ElGamal encryption along with SHA-256 hash algorithm are used to make sure that the TPA should not get access to the outsourced data on the cloud server while performing integrity check thereby increasing the effectiveness of the auditing process. This eliminates the overhead of performing auditing task from the client and also lessens the cloud users' concern that their uploaded data may be accessed by an untrusted organization or individual.

7. REFERENCES

[1]Giuseppe Ateniese, "Provable Data Possession at Untrusted Stores", Proc. of ACM Conference on Computer and Comm. Security (CCS), 2007.

[2]Ari Juels and Burton S. Kaliski Jr, "Pors: Proofs of Retrievability for Large Files", Proc. of ACM Conference on Computer and Comm. Security (CCS), pp. 584-597, 2007.

[3] Hovav Shacham and Brent Waters, "Compact Proofs of Retrievability," International Conf. on Theory and Application of Cryptology and Information Security: Advances in Cryptology, pp. 90-107, 2008.

[4] Giuseppe Ateniese, "Scalable and Efficient Provable Data Possession", International Conf. on Security and Privacy in Comm. Networks (SecureComm), 2008.

[5]Chris Erway, Alptekin Kupcu, Charalampos Papamanthou, Roberto Tamassia, "Dynamic Provable Data Possession", ACM International Conf. on Computer and Comm. Security (CCS),2009.

[6]Sarah Shaikh, Deepali Vora, "Review of Privacy Preserving Auditing Techniques", International Journal of Computer Applications (0975- 887), Volume 145 – No.13, July 2016.

[7]ElGamal Cryptosystem, http://lxmayr1 .informatik. tumuenchen. de/konferenzen/ Jass05/courses/ 1/papers/ meier_paper.pdf