International Research Journal of Computer Science (IRJCS)
Issue 07, Volume 3 (July 2016)
ISSN: 2393-9842
www.irjcs.com

Integrity Preserving Outsourcing Model in Cloud with Proxy Based Public Auditing

S.Ahamed Ali
Department of Information Technology
Velammal Engineering College, Chennai

Dr.M.Ramakrishnan
School of Information Technology,
Madurai Kamaraj University ,Madurai

Abstract *– **Cloud computing is the long dreamed vision of computing as a utility, where data owners can remotely store their data in the cloud to enjoy on-demand high-quality applications and services. Secured data storage and retrieval is challenging when the storage system is distributed and has no central authority further, security risk is developed towards accuracy of data in cloud i.e., the client's outsourced data is securely stored by the storage server or not. To address these issues we propose a scheme that uses Random key generation and Session tracking methodology that serves to identify unauthorised users who try to steal the information stored in cloud. We have also modularised the work done by cloud admin and data owner by introducing a proxy and a third party auditor which assists in auditing and repairing activity. Thus our scheme ensures the integrity of the outsourced data and also focuses on identifying and reporting unauthorised users.***

Keywords – Cloud computing, Random key, Session tracking, Proxy, Third party auditor.

I. INTRODUCTION

Cloud computing has been envisioned as the next-generation information technology (IT) architecture for enterprises, due to its long list of unprecedented advantages in the IT history: on-demand self-service, ubiquitous network access, location independent resource pooling, rapid resource elasticity,usage-based pricing and transference of risk. As a disruptive technology with profound implications, Cloud Computing is transforming the very nature of how businesses use information technology. One fundamental aspect of this paradigm shifting is that data is being centralized or outsourced to the Cloud. From users' perspective, including both individuals and IT enterprises, storing data remotely to the cloud in a flexible on-demand manner brings appealing benefits: relief of the burden for storage management, universal data access with independent geographical locations, and avoidance of capital expenditure on hardware, software, and personnel maintenances, etc. While Cloud Computing makes these advantages more appealing than ever, it also brings new and challenging security threats towards users' outsourced data. Since cloud service providers are separate administrative entities, data outsourcing is actually relinquishing user's ultimate control over the fate of their data. As a result, the correctness of the data in the cloud is being put at risk due to the following reasons. First of all, although the infrastructures under the cloud are much more powerful and reliable than personal computing devices, they are still facing the broad range of both internal and external threats for data integrity. Examples of outages and security breaches of noteworthy cloud services appear from time to time. Secondly, there do exist various motivations for Cloud service providers to behave unfaithfully towards the cloud users regarding the status of their outsourced data. For examples, Cloud service providers might reclaim storage for monetary reasons by discarding data that has not been or is rarely accessed, or even hide data loss incidents so as to maintain a reputation.

Although outsourcing data to the cloud is economically attractive for long-term large-scale data storage, it does not immediately offer any guarantee on data integrity and availability. This problem, if not properly addressed, may impede the successful deployment of the cloud architecture. As users no longer physically possess the storage of their data, traditional cryptographic primitives for the purpose of data security protection cannot be directly adopted. In particular, simply downloading all the data for its integrity verification is not a practical solution due to the expensiveness in I/O and transmission cost across the network. Besides, it is often insufficient to detect the data corruption only when accessing the data, as it does not give users correctness assurance for those un accessed data and might be too late to recover the data loss or damage. Considering the large size of the outsourced data and the user's constrained resource capability, the tasks of auditing the data correctness in a cloud environment can be formidable and expensive for the cloud users. Moreover, the overhead of using cloud storage should be minimized as much as possible, such that user does not need to perform too many operations to use the data (in additional to retrieving the data).For example, it is desirable that users do not need to worry about the need to verify the integrity of the data before or after the data retrieval. Besides, there may be more than one user accesses the same cloud storage, say in an enterprise setting. For easier management, it is desirable that the cloud server only entertains verification request from a single designated party. To fully ensure the data integrity and save the cloud users' computation resources as well as online burden, it is of critical importance to enable public auditing service for cloud data storage.

So that users may resort to an independent third party auditor (TPA) to audit the outsourced data when needed. The TPA, who has expertise and capabilities that users do not, can periodically check the integrity of all the data stored in the cloud on behalf of the users, which provides a much more easier and affordable way for the users to ensure their storage correctness in the cloud. Moreover, in addition to help users to evaluate the risk of their subscribed cloud data services, the audit result from TPA would also be beneficial for the cloud service providers to improve their cloud based service platform, and even serve for independent arbitration purposes.

Another major issue to be resolved is safe-guarding the data from networks attacks and hackers. While the data is being exchanged between the authorised entities, unauthorised user (hackers) may snoop around network and may steal the network packets, upon which they may apply various decryption algorithms to retrieve the encrypted data in the packet. To resolve this issue, we propose using random key generation that makes it difficult to find the respective decryption algorithm used. In addition to the above mentioned issue, an authorised user with limited privilege can acquire the credentials of an authorised user with higher level privilege to access the data. This becomes a serious problem as it is an inside attack. We have also proposed a method to overcome this issue by using a pair of keys and session tracking mechanism. An authorised user with higher privilege will request the data owner for the secret key for downloading the data. For doing so, the user must provide valid credentials that have been assigned to him by the data owner. Session tracking which uses the session ID of each user's matches the entered credentials with the list of available credentials and when a mismatch is found, that particular credentials are blocked automatically and a log information about that user is automatically sent to cloud admin, who can impose any action after that. Thus our work strengthens the security in cloud computing without increasing the overhead of entities.

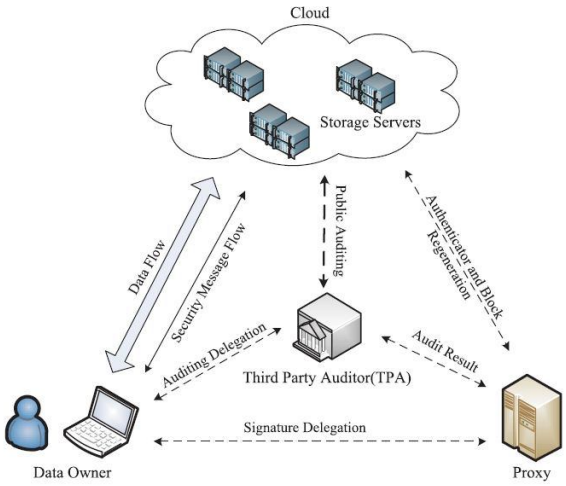

Fig. 1 Architecture Diagram of Existing system.

II. SYSTEM MODEL

We consider a cloud data storage service involving four different entities, as illustrated in Fig. 2: the *data owner* (DO), who has large amount of data files to be stored in the cloud; the *cloud server* (CS) 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 data owner. Data owners 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, data owners may resort to TPA for ensuring the storage integrity of their outsourced data; A *Proxy* (P) that maintains a backup copy of all the data files that are uploaded by data owners. Proxy regenerates and restores the missing file contents upon request.

We consider the existence of a semi-trusted CS. Namely; in most of time it behaves properly and does not deviate from the prescribed protocol execution. However, for their own benefits the CS might neglect to keep or deliberately delete rarely accessed data files which belong to ordinary cloud users. Moreover, the CS may decide to hide the data corruptions caused by server hacks or Byzantine failures to maintain reputation. We assume the TPA, who is in the business of auditing, is reliable and independent, and thus has no incentive to collude with either the CS or the users during the auditing process.

However, it harms the user if the TPA could learn the outsourced data after the audit. To authorize the CS to respond to the audit delegated to TPA's, the user can sign a certificate granting audit rights to the TPA's public key, and all audits from the TPA are authenticated against such a certificate.

III. PROPOSED MODEL

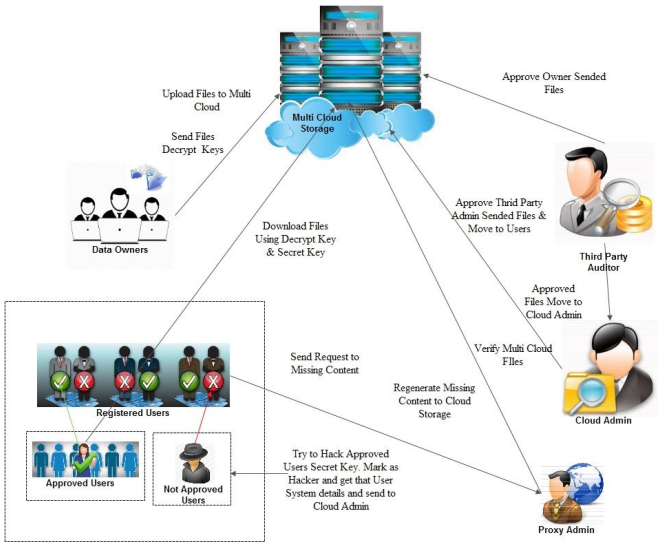

Fig. 2 Architecture Diagram of Proposed system.

The Proposed model can be described by the following module description: *Data owner, Third party auditor, Proxy, Cloud Admin and User.*

A. *Data Owner:* Data Owner uploads files to group users via Cloud Storage like Drop box. If Data Owner uploads a file, in background, that the file will be encrypted using Blowfish Cryptography (V-A) and stored in to cloud storage.

B. *Third party auditor:* Third Party Auditor verify/audits all the files that are uploaded by owner and sends to the cloud admin for approval. If cloud admin approves the files, TPA will send the files to group users.TPA periodically audits for the integrity of the uploaded files.

C. *Proxy:* Proxy holds up the backup copy of every file that are uploaded by data owner on to the cloud server. When a request is generated, proxy restores the missing file contents into the cloud server.

D. *Cloud admin:* Cloud admin verifies and approves the files of data owners. It also checks the status of every file and keeps monitoring the users. When a threat is possessed by an user, it is alarmed and it has the ability to block that particular user, who possess threat to the stored data.

E. *Users:* Users register themselves to the data owner and acquires credentials with their accessibility rights. Each user's activity is monitored by session ID (V-B) and a log information about their credentials are maintained. An user who tries to violate the protocols are notified to the cloud admin along with their system details.

IV. DESIGN GOALS

A. Input Design:

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

Objectives:

- Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
- It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
- When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

B. Output Design:

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

- Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
- Select methods for presenting information.
- Create document, report, or other formats that contain information produced by the system.

Objectives:

- Convey information about past activities, current status or projections of the Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

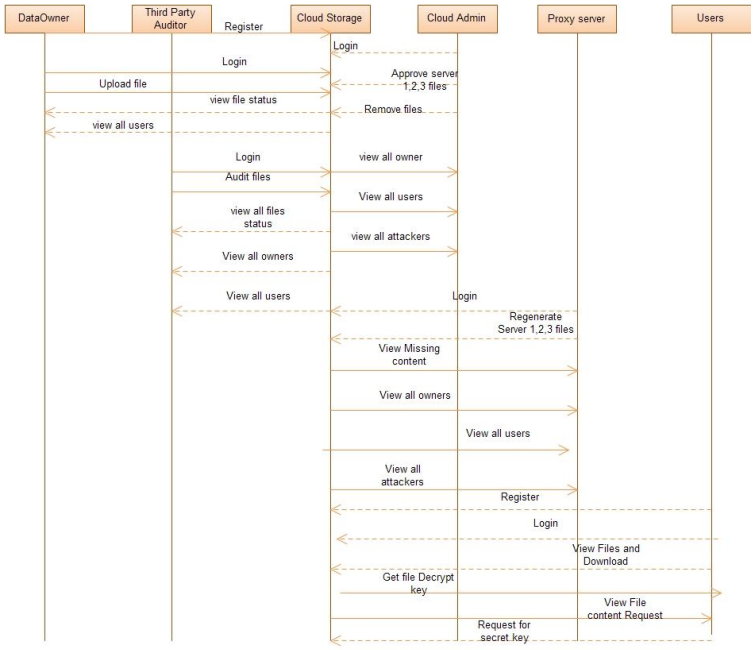

Fig. 2 Sequence Diagram of Proposed system.

V. METHODOLOGIES

A. Encryption and Decryption algorithm:

The encryption algorithm takes as input message blocks (a1,a2,a3,…..,an).,where each message block is assigned a private key (k1,k2,k3,….,kn) which is used while encrypting and decrypting the message blocks. The encryption scheme is Additive homomorphic encryption scheme which is proven to be more efficient for all the message blocks in the cloud.

Encryption:

$$c = Enck(a) = a + k \bmod m$$

Decryption:

$$a = Deck(c) = Enck(a) - k \bmod m$$

Thereby, $0 \leq a < m$, $0 \leq k < m$, and the modulus **m** is a large integer. Note that **k** represents the secret key, while a denotes the message to be encrypted. We perform the encryption and decryption in the following way.

$$Deck(c1 + c2 + \ldots + cn) = Deck(Enck1(a1) + Enck2(a2) + \ldots + Enckn(a2))$$
$$= (a1 + a2 + \ldots + an) \bmod m$$
$$\text{with } (k = k1 + k2 + \ldots + kn) \in [0, m-1] \text{ and } a1, a2, \ldots, an \in [0, m-1].$$

B. Session Tracking Mechanism:

An user who uses a legitimate user's credentials is identified by his session ID. Using session ID, the credentials used for login and request for download are compared. This comparison reveals the hacker who tries to steal the data by disguising as an authorized user. User's credentials are blocked and his details are sent to cloud admin.

C. Random Key generation:

Key Generation is done with Key generation algorithm which is implemented as follows

Key generation algorithm

Gen_Fixed_key()

```
For i=0 to No.Of.Blocks do
    fixedKey_i = generate Random Number()
    fixedKey_i = fixedKey_i mod No.Of.Blocks
  return fixedKey_i
end for
```

Data are sent into networks in form of packets which make them easily subjected to hacking. Even data in encrypted packets can be decrypted by brute force and other attacks. This is possible because the packets contains key of unique length which enables a professional hacker to identify the encryption algorithm used. To avoid such attacks, we implement random key size so that, even if someone interrupts and tries to decrypt the data, they would not do so as the size of key is random.

VI. CONCLUSION

Thus, we have implemented a a solution for security issues in real time environment with existing technologies. Furthermore, we provide this authenticator can be efficiently generated by the data owner simultaneously with the encoding procedure. Extensive analysis shows that our scheme is provable secure, and the performance evaluation shows that our scheme is highly efficient and can be feasibly integrated into a regenerating-code-based cloud storage system.

VII. REFERENCES

[1]. M. Armbrust *et al.*, "Above the clouds: A Berkeley view of cloud computing," Dept. Elect. Eng. Comput. Sci., Univ. California, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2009-28, 2009.
[2]. G. Ateniese *et al.*, "Provable data possession at untrusted stores," in *Proc. 14th ACM Conf. Comput. Commun. Secur. (CCS)*, New York, NY, USA, 2007, pp. 598–609.
[3]. R. Curtmola, O. Khan, R. Burns, and G. Ateniese, "MR-PDP: Multiple-replica provable data possession," in *Proc. 28th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2008, pp. 411–420.
[4]. K. D. Bowers, A. Juels, and A. Oprea, "HAIL: A high-availability and integrity layer for cloud storage," in *Proc. 16th ACM Conf. Comput. Commun. Secur.*, 2009, pp. 187–198.
[5]. B. Chen, R. Curtmola, G. Ateniese, and R. Burns, "Remote data checking for network coding-based distributed storage systems," in *Proc. ACM Workshop Cloud Comput. Secur. Workshop*, 2010, pp. 31–42.

[6]. Y. Hu, H. C. H. Chen, P. P. C. Lee, and Y. Tang, "NCCloud: Applying network coding for the storage repair in a cloud-of-clouds," in *Proc. USENIX FAST*, 2012, p. 21.
[7]. C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage," *IEEE Trans. Comput.*, vol. 62, no. 2, pp. 362–375, Feb. 2013.
[8]. C. Wang, Q. Wang, K. Ren, N. Cao, and W. Lou, "Toward secure and dependable storage services in cloud computing," *IEEE Trans. Service Comput.*, vol. 5, no. 2, pp. 220–232, Apr./Jun. 2012.
[9]. D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," *J. Cryptol.*, vol. 17, no. 4, pp. 297–319, 2004.
[10]. G. Ateniese, R. Di Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in *Proc. 4th Int. Conf. Secur. Privacy Commun. Netw.*, 2008, Art. ID 9.
[11]. K. D. Bowers, A. Juels, and A. Oprea, "Proofs of retrievability: Theory and implementation," in *Proc. ACM Workshop Cloud Comput. Secur.*, 2009, pp. 43–54.

