Securing the Data in Cloud by Code Generation Method in Two- Tier Architecture

[1]K.Sathya, Mr. Syed Ismail .A[2]
[1]M.E. Scholar, [2]Asst. Professor,
CSE Dept., Mohamed Sathak A.J. College of Engineering,
Rajiv Gandhi salai, OMR, Siruseri, Chennai.

Abstract

The cloud security is one of the important roles in cloud. In this paper, the regenerating codes have gained popularity due to their lower repair bandwidth while providing fault tolerance. Here we can overcome the security issues in our project. In existing they are using a remote verification technique to audit by the third party or private auditing. In this project the data owners need to be online to manage that auditing. In our system we are using the own auditing based on the token generation. Using this token generation technique compare the token values from original tokens we can find out the changes about the file.

Generally users can login into their account then they upload our files. The files will be stored into the cloud storage. In this project we provide the two tier security for our uploaded files. The files does not stored directly it will be divided into the three files, it will be stored into three different cloud server locations. The original file content split into three parts and it will be store into each files. Not only stored also the content will be encrypted in the cloud server. If anyone try to hack at the cloud end is not possible to break the two tier block. They need first decrypt the files and also combine first decrypt the files and also combine the splited files from three different locations. Anyone can download the files from the server with file owner permission. At the time of download key generated (code based key generation) and it will send to the file owner.

Every data stored in the cloud will be generated with the Code generation (Token generation) technique, So that the security will be maintained. Once any unauthorized modification is made, the original data in the private cloud will be retrieved by the Owner and will be returned to the user.

Index Term: *Cloud storage, regenerating codes, public audit, privacy preserving, authenticator regeneration, proxy, privileged, provable secure.*

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 are 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 location independence, 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 toward users' outsourced data.

Since cloud service providers (CSP) are separate administrative entities, data outsourcing is actually relinquishing user's ultimate control over the fate of their data.

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.

Related Work. The problem of remote data checking for integrity, proposed a formal definition of the PDP (*provable data possession)* model for ensuring possession of files on untrusted storage, introduced the concept of RSA-based homomorphic tags and suggested randomly sampling a few blocks of the file.

II. LITERATURE SURVEY

A. *Toward Publicly Auditable Secure Cloud Data Storage Services:*

It does not offer any guarantee on data integrity and availability. This problem, if not properly addressed, may impede successful deployment of the cloud architecture.

Does not completely solve the problem of protecting data privacy but just reduces it to the one of managing the encryption keys. Unauthorized data leakage still remains a problem due to the potential exposure of encryption keys.

The auditing result should not only identify the data correctness but also be able to determine which entity (including owner, TPA, as well as cloud server) is responsible for the problem that may occur.

B. *Auditing to Keep Online Storage Services Hones*

We need both internal and external audits of OSPs. External audits can only confirm past behavior, so without internal audits, we could not predict upcoming problems or assess risk exposure. These audit trails are useful for detecting and pinpointing problems after the fact, and could support internal and external audits.

Currently, customers cannot make informed decisions about the risk of losing data stored with any particular service provider,

C. *Towards Secure and Dependable Storage Services in Cloud Computing*

Security risks towards the correctness of the data in cloud.

In order to address this new problem and further achieve a secure and dependable cloud storage service.

The problem of data error localization, thus only providing binary results for the storage Verification. We investigate the problem of data security in cloud data storage, which is essentially a distributed storage system.

D. *Decentralized Erasure Codes for Distributed Networked Storage*

In existing the method used here does not makes it optimally sparse, and lead to reduced communication, storage and computation cost over random linear coding. The key advantage of decentralized erasure codes is that there is no need for coordination among the data nodes. We show how data nodes acting randomly and independently, can create good erasure codes with sparse structure but in existing this technology are used.

Technique: Wiedemann algorithm Randomized network algorithm.

Drawback: The key issue is presented here so it is not possible to achieve this robust distributed storage with minimal computation and communication.

E. *Highthroughput File System for the Hydrastor Content- Addressable Storage System*

A content-addressable storage (CAS) system is a valuable tool for building storage solutions, providing efficiency by automatically detecting and eliminating duplicate blocks; it can also be capable of high throughput, at least for streaming access. However, the absence of a standardized API is a barrier to the use of CAS for existing applications. Additionally, applications would have to deal with the unique characteristics of CAS, such as immutability of blocks and high latency of operations.

Technique:

Wiedemann algorithm, Randomized network algorithm. The absence of a standardized API is a barrier to the use of CAS for existing applications additionally; applications would have to deal with the unique characteristics of CAS, such as immutability of blocks and high latency of operations.

F. *Scalable Secure File Sharing on Untrusted Storage*

We explain the mechanisms in Plautus to reduce the number of cryptographic keys exchanged between users by using file groups, distinguish file read and write access, handle user revocation efficiently, and allow an untrusted server to authorize file writes. We have built a prototype of Plautus on OpenAFS.

Technique:

Rabin's Information Dispersal Algorithm. Trace Evaluation Bench mark They showed that the performance impact, due mostly to the cost of cryptography, is comparable to the cost of two popular schemes that encrypt on the wire.

G. *The Least-Authority Filesystem*

Tahoe is a system for secure, distributed storage. It uses capabilities for access control, cryptography for confidentiality and integrity, and erasure coding for fault-tolerance. It had been deployed in a commercial backup service and is currently operational. The implementation is Open Source.

Technique:

cryptographic techniques, SHA256 algorithm. The Least-Authority File system is a practical secure, decentralized system, using several techniques which

have previously been analyzed in the theoretical literature but not widely deployed.

H. A Generic Construction for Proxy Cryptography

We present a novel proxy cryptosystem model: proxy cryptosystem based on time segmentation. Under this mode, a security analysis model will be proposed. Furthermore, a proxy cryptosystem scheme is presented as an example. We will show that the proposed scheme is proven security in the proposed security analysis model. Finally, we will give the ID-based version of this construction.

Technique:

The key generation algorithm, The proxy key derivation algorithm, The decryption algorithm A PCBTS scheme is secure against selective time segmentation, chosen ciphertext attacks (STSCCA) if no polynomial bound adversary.

III. PROPOSED WORK

An effective and flexible distributed scheme with data in the cloud. Here we are using the erasure code technique for distribute the data to cloud locations and access the data from cloud. User can register and login into their account. They have a option to store, share and access the data from cloud storage. Here we are using the two tier security scheme for storing data into the cloud. The first tier security is your data or file splited into multiple parts and it will store into different cloud server locations. Each and every file generates the token for auditing. Then second tier security is each and every splited file will encrypt before store into different locations. The shared users can edit the file in the cloud with file owner's permission. That file eligible of own public auditing. Then user can have to login and access the own files or some other files. User first can search and download the files, at the time of download user should use the security key. If authentication success it will be decrypt and combine to get the original data from cloud.

Compared to many of its predecessors, which only provide binary results about the storage state across the distributed servers, the challenge-response protocol in our work further provides the localization of data error.

Unlike most prior works for ensuring remote data integrity, the new scheme supports secure and efficient dynamic operations on data blocks, including: update, delete and append.

Extensive security and performance analysis shows that the proposed scheme is highly efficient and resilient against Byzantine failure, malicious data modification attack, and even server colluding attacks.

A. Architectural details

An architecture diagram is the one which describes the overall view of this work. It is the pictorial representation of the entire work which is to be carried out. This architecture consists of four modules. It is the first part of system design. The main aim of the input design is covering user oriented descriptions of the input to the computer oriented form. All the inputs are converted into a computer based format. The goal of designing input data is to make data entry easier and free errors as possible.

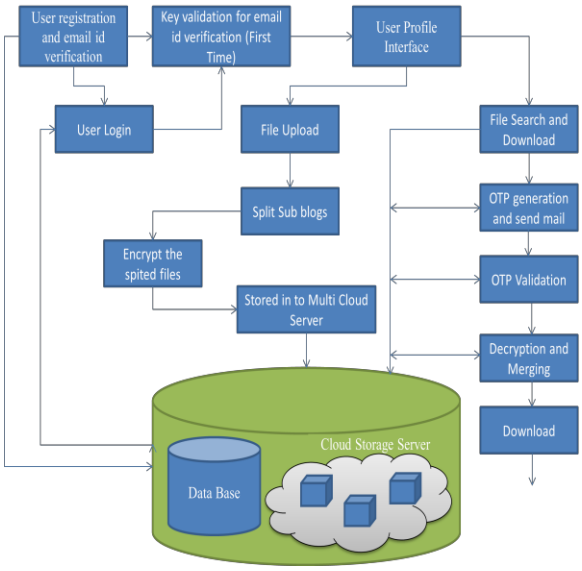

Fig1. System Model

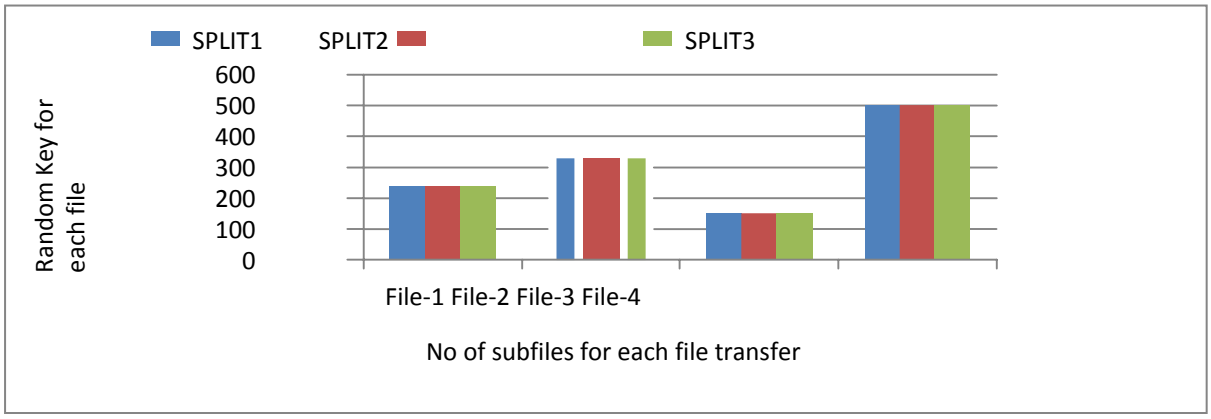

Fig2. Sample Graph for Key Generation

IV. SYSTEM ANALYSIS

System analysis is defined as the process of gathering and interpreting facts, diagnosing problem and using the facts to improve the system. The objectives of the system analysis phase are the establishment of the requirements for the system to be acquired, developed and installed. Fact finding or gathering is essential to any analysis of requirement.

A detailed study of the system is done by making use of various techniques. The data collected must be scrutinized to arrive at a conclusion. The conclusion is an understanding of how the system functions. This system is called existing system. Now, the existing system is subjected to close study and the problem areas are identified. The solutions are given as a proposal. The proposed system is presented to the user

A. Algorithm Specification

1) Secure Erasure Code Algorithm:

A cloud storage system, consisting of a collection of storage servers, provides long-term storage services over the Internet. Storing data in a third party's cloud system causes serious concern over data confidentiality. General encryption schemes protect data confidentiality, but also limit the functionality of the storage system because a few operations are supported over encrypted data. Constructing a secure storage system that supports multiple functions is challenging when the storage system is distributed and has no central authority. We propose a threshold proxy re- encryption scheme and integrate it with a decentralized erasure code such that a secure distributed storage system is formulated. The distributed storage system not only supports secure and robust data storage and retrieval, but also lets a user forward his data in the storage servers to another user without retrieving the data back. The main technical contribution is that the proxy re-encryption scheme supports encoding operations over encrypted messages as well as forwarding operations over encoded and encrypted messages. Our method fully integrates encrypting, encoding, and forwarding. We analyze and suggest suitable parameters for the number of copies of a message dispatched to storage servers and the number of storage servers queried by a key server. These parameters allow more flexible adjustment between the number of storage servers and robustness.

Steps for Secure Erasure Code Technique

Step 1: Given a signal of m blocks, recode to n

- Blocks where n > m
- Optimal: reconstruct signal given any m
- Unique blocks

Step 2: Suboptimal: Reconstruct signal using (1+e)

m unique blocks Rate r=m/n, and storage overhead is 1/r Optimal erasure codes have the property that any k out of the n code word symbols are sufficient to recover the original message (i.e., they have optimal reception efficiency). Optimal erasure codes are maximum distance separable codes (MDS codes).

Step 3: A cloud storage system, consisting of a collection of storage servers, provides long-term storage services over the Internet. Storing data in a third party's cloud system causes serious concern over data confidentiality.

Step 4: General encryption schemes protect data confidentiality, but also limit the functionality of the storage system because a few operations are supported

over encrypted data.

Step 5: Parity check

Parity check is the special case where $n = k + 1$. From a set of k values $\{V_i\}$ $1 \le i \le k$, a checksum is computed and appended to the k source values:

$$V_{k+1} = -\sum_{i=1}^{k} v_i$$

The set of $k + 1$ values $\{V_i\}$ $1 \le i \le k+1$ is now consistent with regard to the checksum. If one of these values, Ve , is erased, it can be easily recovered by summing the remaining variables:

$$Ve = -\sum_{i=1, i \neq e}^{K+1} v_i.$$

ADVANTAGE:

- It supports dynamic data operations for both data owner and privileged user.
- It avoids direct server attack and similar attacks.
- It improves the storage space.
- Each time the data owner gets the notification messages if any change is made.

2) Secure Cloud Storage:

The Secure Erasure Code algorithm is used based on the encryption and decryption concept as RSA algorithm as such steps following

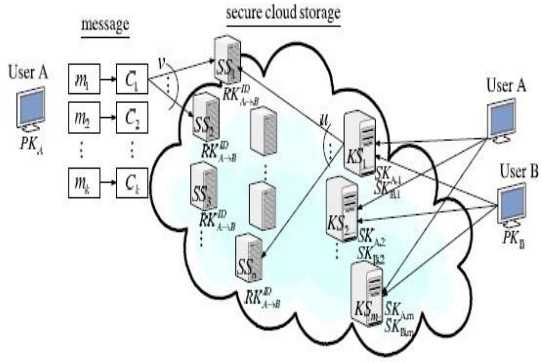

Fig. 1. A general system model of our work.

3) RSA ALGORITHM RSA key

Each user generates a public/private key pair by:
selecting two large primes at random p, q

- computing their system modulus N=p.q
- note ø(N)=(p-1)(q-1)
- selecting at random the encryption key e.

Where 1<e<ø(N), gcd(e,ø(N))=1

- solve following equation to find decryption key d
- e.d=1 mod ø(N) and 0≤d≤N
- publish their public encryption key: KU={e,N}
- keep secret private decryption key: KR={d,p,q} .

RSA use

To encrypt a message M the sender:
obtains
public key of recipient KU={e,N}
computes: $C=M^e \mod N$, where $0 \le M < N$
To decrypt the ciphertext C the owner:
uses their private key KR={d,p,q}
computes: $M=C^d \mod N$
Note that the message M must be smaller than the modulus N (block if needed)

RSA EXAMPLE

1. Select primes: p=17 & q=11 2. Compute $n = pq$ =17×11=187
3. Compute $ø(n)=(p–1)(q- 1)=16×10=160$
4. Select e *:* gcd(e,160)=1; choose e=7
5. Determine d*:* de=1 mod 160 and d < 160 Value is d=23 since 23×7=161= 10×160+1
6. Publish public key KU={7,187}
7. Keep secret private key KR={23,17,11}

B. Module Description

The modules which are to be used in the proposed schemes are:

1) 1. User interface:

Users can use them from anywhere at any time. For example, the email service is probably the most popular one. Cloud computing is a concept that treats the resources on the Internet as a unified entity, a cloud. Users just use services without being concerned about how computation is done and storage is managed. In this paper, we focus on designing a cloud storage system for robustness, confidentiality, and functionality. A cloud storage system is considered user interface entry level creation in this module.

2) 2. Secret key generation:

The data forwarding phase, user forwards his encrypted message with an identifier ID stored in storage servers to user such that can decrypt the forwarded message by using his secret key. The secret keys of target users, and the shared keys stored in key servers.

3)

3. File uploading process:

Storing data over storage servers one way to provide data robustness is to replicate a message such that each storage server stores a message. Another way is to encode a message of k symbols into a codeword of n symbols by erasure coding. To store a message, each of its codeword symbols is stored in a different storage server. A storage server corresponds to an erasure error of the codeword symbol. As long as the number of servers is under the tolerance threshold of the erasure code, the message can be recovered from the codeword symbols stored in the available storage servers by the decoding process.

4. Mail alert process:

The uploading and downloading process of the user is first get the secret key in the corresponding user email id and then apply the secret key to encrypted data to send the server storage and decrypts it by using his

secret key to download the corresponding data file in the server storage system's the secret key conversion using the Share Key Gen(SKA, t, m). This algorithm shares the secret key SKA of a user to a set of key servers.

5. File Downloading process:

File downloading process is to get the corresponding secret key to the corresponding file to the user mail id and then decrypt the file data. The file downloading process re-encryption key to storage servers such that storage servers perform the re-Encryption Operation. The length of forwarded message and the computation of re-encryption is taken care of by storage servers. Proxy re-encryption Schemes significantly reduce the overhead of the data Forwarding function in a secure storage system.

V. CONCLUSION

A privacy-preserving public auditing system for data storage security in cloud computing. We utilize the homomorphic linear authenticator and random masking to guarantee 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. Considering TPA may concurrently handle multiple audit sessions from different users for their outsourced data files, we further extend our privacy-preserving public auditing protocol into a multiuser setting, where the TPA can perform multiple auditing tasks in a batch manner for better efficiency.

VI. FUTURE ENHANCEMENT

In this project auditing can be done with the help of Third Party Auditing without locally downloading the file from cloud with the help of the homomorphic authenticator and Random mask technique. This will guarantee that TPA would not learn any knowledge about the data content stored on the cloud server during the efficient auditing process.

In future the project will have more enhancements as follows: Network traffic that happens in this while auditing by Owner from will be reduced. More privacy will issued to the files that are under process of Auditing by owner.

REFERENCES

[1] B. Wang, B. Li, and H. Li, "Oruta: Privacy- Preserving PublicAuditing for Shared Data in the Cloud," Proc. IEEE Fifth Int'l Conf.Cloud Computing, pp. 295-302, 2012.

[2] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph,

[3] R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A View of Cloud Computing," Comm. ACM, vol. 53, no. 4, pp. 50-58, Apr. 2010.

[4] K. Ren, C. Wang, and Q. Wang, "Security Challenges for the Public Cloud," IEEE Internet Computing, vol. 16, no. 1, pp. 69-73, 2012.

[5] D. Song, E. Shi, I. Fischer, and U. Shankar, "Cloud Data Protection for the Masses," Computer, vol. 45, no. 1, pp. 39-45, 2012.

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

[7] B. Wang, M. Li, S.S. Chow, and H. Li, "Computing Encrypted Cloud Data Efficiently under Multiple Keys," Proc. IEEE Conf. Comm. and Network Security (CNS '13), pp. 90-99, 2013.

[8] R. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems," Comm. ACM, vol. 21, no. 2, pp. 120-126, 1978.

[9] The MD5 Message-Digest Algorithm (RFC1321). https://tools. ietf.org/html/rfc1321, 2014.

[10] G. Ateniese, R. Burns, R. Curtmola, J. Herring,

[11] L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores," Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), pp. 598-610, 2007.

[12] H. Shacham and B. Waters, "Compact Proofs of Retrievability," Proc. 14th Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT '08), pp. 90- 107, 2008.

[13] C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, "Dynamic Provable Data Possession," Proc. 16th ACM Conf. Computer and Comm. Security (CCS'09), pp. 213-222, 2009.

[14] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling Public Verifiability and Data Dynamic for Storage Security in Cloud Computing," Proc. 14th European Conf. Research in Computer Security (ESORICS'09), pp. 355-370, 2009.

[15] C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring Data Storage Security in Cloud Computing," Proc. 17th Int'l Workshop Quality of Service (IWQoS'09), pp. 1-9, 2009.

[16] B. Chen, R. Curtmola, G. Ateniese, and R. Burns, "Remote Data Checking for Network Coding-Based Distributed Storage Systems," Proc. ACM Workshop Cloud Computing Security Workshop (CCSW'10), pp. 31-42, 2010.