International Journal of Engineering Research in Computer Science and Engineering (IJERCSE) Vol 3, Issue 3, March 2016

Hybrid Attribute Based Encryption for Secured Data Transaction in Cloud Storage System

[1] M.Preethika [2] Priyanka.S [3] Suganya.S [4] Yamuna.A [5] Mrs.R.Manjula
[1][2][3][4] Student [5] Associate Professor
[1][2][3][4] Department of Information Technology
Panimalar Engineering College

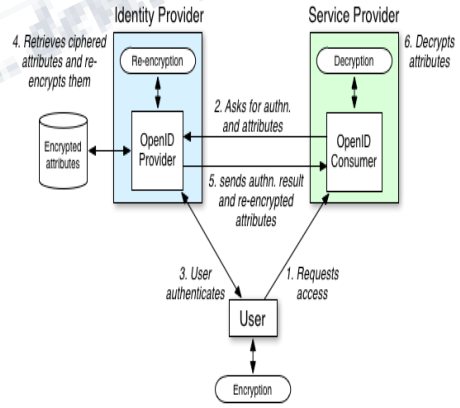
Abstract: **People have the power to access the internet anywhere and anytime. Cloud computing is a concept that treats the resources on the Internet as an unified entity, namely cloud. The data center operators virtualized the resources according to the needs of the customers and expose them as storage pools, which the customers can themselves use to store files or data objects. Physically, the resources may get stored across multiple servers. So, Data robustness is a major requirement for such storage systems. In this paper we have suggested one way to provide data robustness that is by replicating the message such that each storage server stores a copy of the message. We have enhanced the secure cloud storage system by using a threshold proxy re-encryption technique. This encryption scheme supports decentralized erasure codes applied over encrypted messages and data forwarding operations over encrypted and encoded messages. Our system is highly distributed where each storage server independently encodes and forward messages and key servers independently perform partial decryption.**

Key terms: erasure code, proxy re-encryption.

I. INTRODUCTION TO WEB ANALYTICS

A. Proxy Re-Encryption:

Proxy re-encryption schemes are a crypto system which allows third parties to alter a cipher text which has been encrypted for one user, so that it may be decrypted by the users at the time of viewing. By using proxy re-encryption technique the encrypted data (cipher text) in the cloud is altered again that is re-encrypted. It provides secured information stored in the cloud. The messages are first encrypted by the owner and then stored in a storage server. When a user wants to share or download his messages, he sends a re-encryption key to the storage server. The storage server re-encrypts the encrypted messages for the authorized user. Thus, the system has data confidentiality and supports the data forwarding function. An encryption scheme is called multiplicative homomorphic if it supports a group operation on encrypted plaintexts without decryption. This multiplicative homomorphic encryption scheme supports the encoding operation over encrypted messages. A secret key is shared to the key servers with a threshold value t. To decrypt for a set of k message symbols, each key server independently queries to the storage servers and partially decrypts two encrypted codeword symbols. As long as 't' key servers are available, 'k' codeword symbols are obtained from the partially decrypted cipher texts. Every user will have a public key and private key.



B. Erasure Code:

Erasure coding is the method of data protection in which data is broken into smaller fragments, expanded and encoded with redundant data pieces and stored across a set of different locations or storage media. The goal is to enable data that becomes corrupted at some point in the disk storage process to be reconstructed again by using information about the data that is stored elsewhere in the array. This technique creates a mathematical function to describe a set of numbers so they can be checked for accuracy and recovered if one is lost The protection offered

by erasure coding can be represented by the following equation: n=k+m where k is the original amount of data or symbols, m stands for redundant symbols that are added to provide protection from failures and n is the total number of symbols created. For example, in a 10 of 16 configuration, or EC 10/16, six extra symbols 'm' would be added to the 10 base symbols 'k'. The 16 dat fragments 'n' would be spread across 16 drives, or geographic locations. The original file can be reconstructed from 10 verified fragments.

C. *Existing System:*

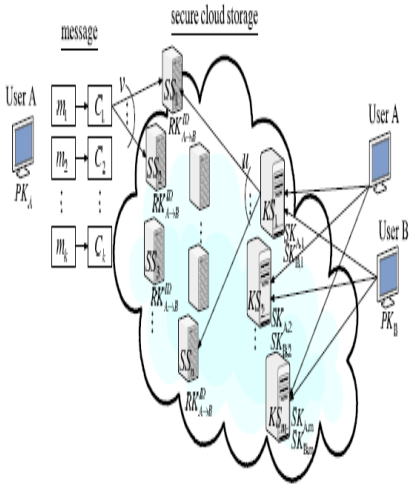
In Existing System we use the straight forward integration technique. In this technique Storing data in a third party's cloud system causes serious concern on data confidentiality. In order to maintain confidentiality for messages in storage servers, a user can encrypt messages by any one of the cryptographic method before applying an erasure code method to encode and store messages. When he needs to use the message, he needs to retrieve the Codeword symbols from storage servers, decode them, and then decrypt them by using his cryptographic keys. General encryption methods protect data confidentiality, but also restrict the functionality of the storage system because a very few operations are supported over encrypted data.

D. *Disadvatages:*

- Computation and Cmmunication traffic between the user and storage servers is high.
- The user has to manage his cryptographic keys otherwise the security has to be broken.
- The data storing and retrieving is hard for storage servers to directly support other functions.

E. *Proposed System:*

In our proposed system we address the problem of forwarding the data to another user by storage servers directly under the command of the data uploader. Here we consider the system model that consists of distributed storage servers and key servers. Since storage of cryptographic keys in a single device is quite risky, the user distributes his key to key servers that shall execute cryptographic functions on behalf of the user. The distributed systems require independent servers to perform all operations. We propose a new threshold proxy re-encryption technique and integrate it with a secure decentralized code to form a secure distributed storage system. This technique supports encoding operations over encrypted messages and forwarding operations over encrypted and encoded messages.



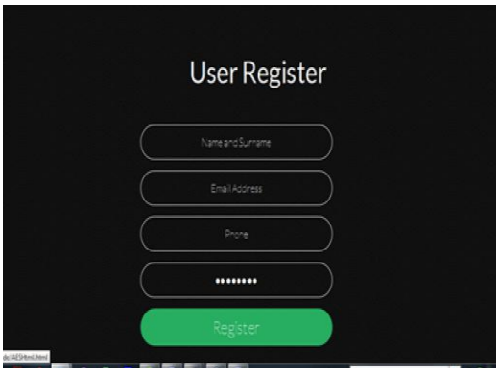
F. *Advantages:*

- Tight integration of encoding, encryption, and forwarding makes the storage system efficiently meet the requirements of data robustness, data confidentiality, and data forwarding.
- The storage servers independently perform encoding and re-encryption process and the key servers independently perform partial decryption process.
- The flexible adjustment between the number of storage servers and robustness.

II. MODULES

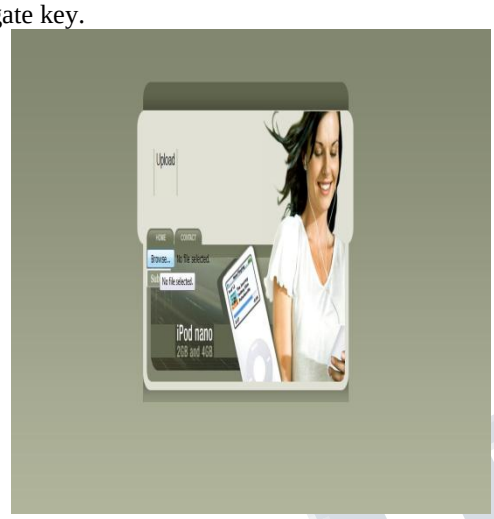
A. *User Registration:*

The group manager randomly selects a number for the registration of user with identity ID. Then the group manager adds user list into the group which will be used in the traceability phase. User obtains a private key after completing the registration process which will be used for group signature generation and file decryption.



B. *Sharing The Data:*

The canonical application is data sharing. When we expect the delegation to be efficient and flexible, the public auditing property is useful. The schemes enable a content provider to share their data in a confidential and selective way, with a fixed and small ciphertext expansion, by distributing to each authorized user a single and small aggregate key.



C. *Secure Cloud Storage:*

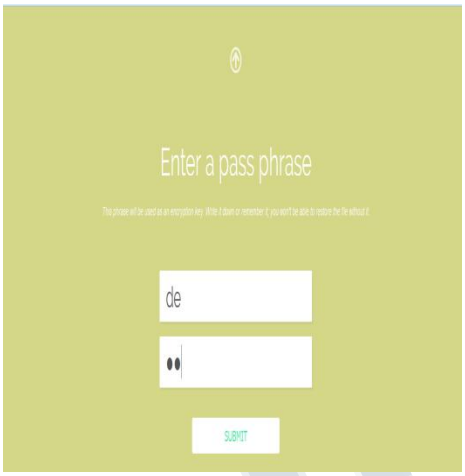
Data robustness is a major need for storage systems. One way to provide data robustness is to replicate a message such that each storage server stores a copy of the message. A decentralized erasure code is suitable for use in a distributed storage system.

D. *Proxy Re-Encryption:*

Proxy re-encryption scheme is a crypto system which allows third parties to alter a cipher text which has been encrypted for one user, so that it may be decrypted by another user. By using proxy re-encryption technique the encrypted data (cipher text) in the cloud is again altered by the user. It provides highly secured information stored in the cloud. Every user will have a public key and private key

E. *Data Retrieval:*

Two primary forms of the retrieved data from servers are reports and data. There are some overlaps between them, but queries generally select a relatively small portion of the server, while reports show larger amounts of data. Queries also present the data in a standard format; whereas reports allow formatting of the output however you like and is normally retrieved.



III. RELATED WORK:

In [1] they considered the problem of constructing an erasure code for storage over a network when the data sources are distributed. Specifically, they assumed that there are n storage nodes with limited memory and sources generating the data. They needed a data collector, who can appear anywhere in the network, to query the storage nodes and be able to retrieve the data. Hence we introduce "Decentralized Erasure Codes".We show that decentralized erasure codes are optimally sparse, and lead to reduced communication, storage and computation cost.

In [2],they used "Plutus" which is a cryptographic storage system that enables secure file sharing.It makes novel use of cryptographic primitives to protect and share files. Plutus reduces the number of cryptographic keys exchanged between users by using filegroups, distinguishes file read and write access, handles user revocation efficiently, and allows an untrusted server to authorize file writes.

[3]explains the motivation, architecture and implementation of a new peer-to-peer storage system, called TotalRecall system.It automatically measures and estimates the availability of its constituent host components, predicts their future availability based on their past behavior, calculates the appropriate redundancy mechanisms and repair policies, and delivers user-specified availability.

[4]sketches the design of PAST

PAST is a peer-to-peer Internet application.PAST nodes serve as access points for clients, and participate in the routing of client requests, and contribute to the storage system. But the drawback is Nodes are not trustable, they may join the system at any time and may silently leave the system without warning.

In [5] they introduced HAIL (High-Availability and Integrity Layer), a distributed cryptographic system that allows a set of servers to prove to a client that a stored

file is intact and retrievable. HAIL strengthens, formally unifies, and streamlines distinct approaches from the cryptographic and distributed-systems communities.HAIL cryptographically verifies and reactively reallocates file shares.

In [6] they presented the design, implementation, and evaluation of HydraFS, a file system built on top of HYDRAstar, a scalable, distributed, content-addressable block storage system. HydraFS provides high-performance reads and writes for streaming access, achieving 82–100% of the HYDRAstor throughput, while maintaining high duplicate elimination.

In [7] they suggested an approach to build a cluster deduplication storage system with multiple deduplication storage system nodes. The goal is to achieve scalable throughput and capacity using extremely high throughput (e.g. 1.5 GB/s) nodes, with a minimal loss of compression ratio. The key technical issue is to route data intelligently at an appropriate granularity.

IV. CONCLUSION:

Erasure codes are promising for improving the reliability of the storage system due to its space efficiency compared to the replication methods. Traditional erasure codes split the data into equal sized data blocks and encode strips in different data blocks. This brings heavy repairing traffic when clients read parts of the data, since most strips read for repairing are not in the expected blocks. This paper proposes a novel method of discrete data dividing to completely avoid this problem. The key idea is to encode strips from the same data block. We could see that for repairing failed blocks, the strips to be read are either in the same data block with corrupted strips or from the encoded strips. Therefore, no data is wasted. We design and implement this data layout into a HDFS-like storage system. Experiments over a small-scale test bed shows that the proposed discrete data divided method avoids downloading data blocks that are not needed for clients during the repairing operations.

FUTURE ENHANCEMENT:

In Future, The storage system has to propose some content addressable file system and storage system are highly compatible. The storage servers act as storage nodes in a content addressable storage system for storing content addressable blocks. The key servers act as access nodes for providing a front-end layer such as a traditional file system interface.

REFERENCES:

[1]H.-Y. Lin and W.-G. Tzeng, “A Secure Decentralized Erasure Code for Distributed Network Storage,” IEEE Trans. Parallel and Distributed Systems, vol. 21, no. 11, pp. 1586-1594, Nov. 2010.

[2]A. Juels and B. S. Kaliski, “Pors, proofs of retrievability for large files,”in Proc. 14th ACM Conf. Computer and Commun. Security (CCS'07),2007, pp. 584–597

[3]G. Ateniese,R.Burns,R.Curtmola, J. Herring,O.Khan, L. Kissner, Z.Peterson, andD. Song, “Remote data checking using provable data possession,”ACM Trans. Inf. Syst. Secur., vol. 14, no. 1, pp. 12:1–12:34, Jun. 2011

[4] 4.G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson,and D. Song, “Provable data possession at untrusted stores,” in Proc.14th ACM Conf. Computer andCommun. Security (CCS'07), 2007, pp.

[5]S.-T. Shen and W.-G. Tzeng, “Delegable provable data possession for remote data in the clouds,” in Proc. 13th Int. Conf. Information and common. Security (ICICS'11), 2011, pp. 93–111.

[6]K. D. Bowers, A. Juels, and A. Oprea, “Hail, a high-availability and integrity Layer for cloud storage,” in Proc. 16th ACM Conf. Computer and Commun. Security (CCS'09), 2009, pp. 187–198.

[7]B. Chen, R. Curtmola, G. Ateniese, and R. Burns, “Remote data checking for network coding-based distributed storage systems,” in Proc. 2nd ACM Workshop Cloud Computing Security (CCSW'10), 2010, pp. 31–42.