Scientific Journal Impact Factor (SJIF) : 1.711
International Journal of Modern Trends in Engineering and Research
ISSN (ONLINE) : 2349 - 9745
ISSN (PRINT) : 2393 - 8161

SECURED DATA SHARING USING MULTI PROTOCOL IN CLOUD ARCHITECTURE

DHIVYABHARATHI.S[1] , Mr.T.K.P.RAJAGOPAL[2]
[1]*Computer Science And Engineering ,Kathir College of Engineering*
[2]*Computer Science And Engineering ,Kathir College of Engineering*

Abstract- 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.

Keywords-Confidentiality, decentralized erasure code , proxy re-encyption , key server, robustness , Distributed storage system.

I. INTRODUCTION

Cloud storage architecture will have a collection of storage servers with higher end configuration which will provides long-term storage services over the Internet and also for the cloud storage system. Here storing and retrieving the data in a third party's cloud system causes serious problems and conflict over data confidentiality during the data transactions. Whenever third party storage will involved with the cloud server this conflict will occur naturally. Even thou there are various methods are available to overcome this problem like cryptography, key encryption and etc. But general encryption schemes protect data confidentiality during the transaction, but along with this process the main drawback will, it limits the functionality of the storage system. This is because; a few operations only supported over encrypted data. These methods will cause failure. In order to constructing a secure storage system that supports multiple functions is challenging when the storage system is distributed and has no central authority.

This paper proposes a secured threshold proxy re-encryption server and integrates it with a decentralized erasure code such that a secure distributed storage system is formulated. In this method multiple users can interact with the storage system. Users can upload their data in to the distributed storage system. 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. This makes the ownership data unused and secured during the time of retrieval. The main technical contribution is that the proxy re-encryption scheme supports encoding operations along with a key over encrypted messages, as well as forwarding operations over encoded and encrypted messages. The content in the database will be in the decrypted format. So that even intruder cant able to access the data even they access the database.

The encrypted data will become unused even the data obtained by the intruder. This makes the system so stronger. This project deals with fully integrates encrypting, encoding, and forwarding. The application can be shown in both cloud servers as well as in local host as per the environment. The storage and robustness are more flexible with the users. So that user will authorize the sender request to generate the key. Using the authorized one time key sender can access the encrypted file in decrypted format at once. The key will become invalid after one use. This is method is implemented for secured data forwarding. During data forwarding a proxy server will be created virtually to access the encrypted data from the sender side. The original data from the cloud server will be transmitted to the proxy virtually. This makes less traffic and the original data content will not get affected during the time of data transaction. After the transaction the proxy server will be deleted. Thus we implemented "A Secure Erasure Code-Based Cloud Storage System with Secure Data Forwarding"

II. OBJECTIVE

- The main objective of this project is to develop a cloud architecture using privacy preservation protocol
- Shared authority based data forwarding can be done through proxy server. Using Proxy re encryption method.
- To create a cloud storage server for long term storage over the internet
- The Storage server will act as a data base server.
- Uploaded data stored in the cloud server through proxy re encryption method.
- To generate proxy re encryption key for one time data access
- A proxy server will be created virtually for one time data access. To create fully integrates encrypting, encoding, and forwarding

III. EXISTING SYSYEM

We identified a new privacy challenge in cloud storage, and address a subtle privacy issue during a user challenging the cloud server for data sharing, in which the challenged request itself cannot reveal the user's privacy no matter whether or not it can obtain the access authority. Propose an authentication protocol to enhance a user's access request related privacy, and the shared access authority is achieved by anonymous access request matching mechanism. Apply cipher text -policy attribute based access control to realize that a user can reliably access its own data fields, and to provide temp authorized data sharing among multiple users. Storing data in a third party's cloud system causes serious concern on data confidentiality.

DISADVANTAGES OF EXISTING SYSTEM:

- 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.
- Data robustness is a major requirement for storage systems.

IV. PROPOSED SYSYEM

In the proposed system, a protocol based threshold proxy re-encryption scheme is introduced and integrate it with a decentralized erasure code such that a secure distributed storage system is formulated. By using the threshold proxy re-encryption scheme, presenting a secure cloud storage system that provides secure data storage and secure data forwarding functionality in a decentralized structure. And also we propose three protocols for more security. They are one time key generation, time based key generation and date based key generation. All these protocols are implemented using proxy key protocol(PKP). Data in the cloud server will get splitted into encrypted file and decrypted file and then it will be stored. In order to provide strong confidentiality for messages in storage servers, a user can encrypt messages by a cryptographic method before applying

an erasure code method to encode and store messages. When he wants to use a message, he needs to retrieve the codeword symbols from storage servers, decode them, and then decrypt them by using cryptographic keys.

ADVANTAGES OF PROPOSED SYSTEM

- In this paper, we focus on designing a cloud storage system for robustness, confidentiality, and functionality.
- A cloud storage system is considered as a large scale distributed storage system that consists of many independent storage servers.

V. SYSYEM MODEL

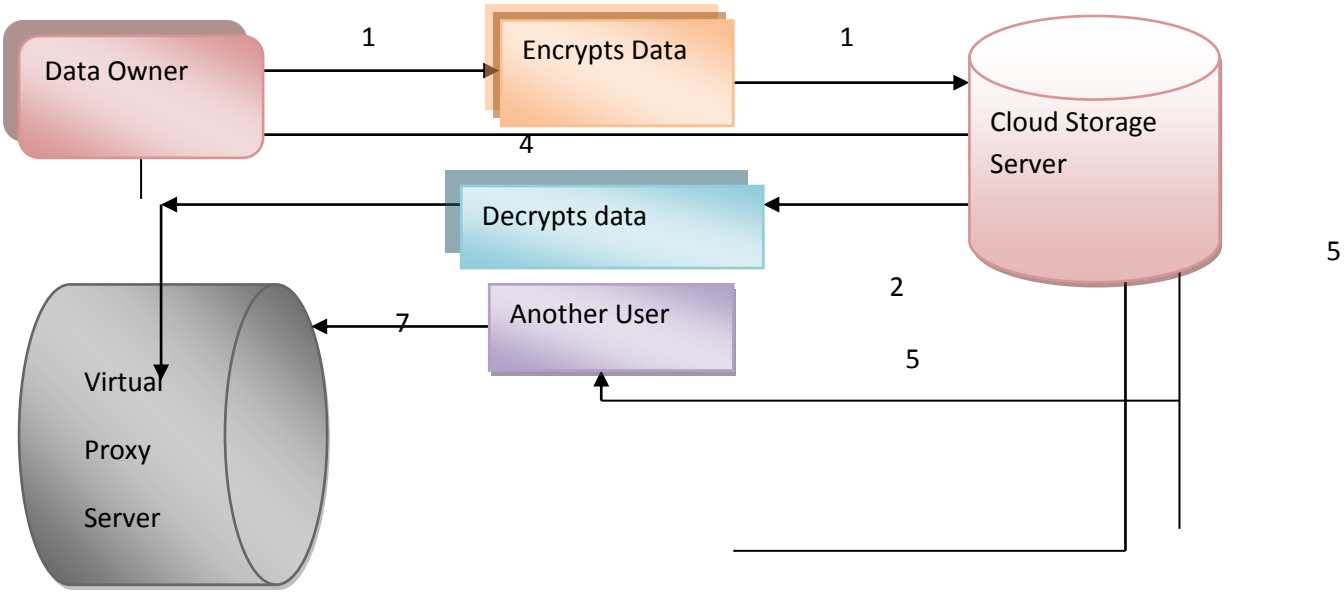

Fig 3.1 : System Model

1. Data owner Store the original data in to the cloud storage server. The data will encrypt and stored in the cloud storage server.
2. Any can view the uploaded data. But the data will be in the encrypted format. Another user can only view the file name of the data. And another user will send request to the cloud server to view the data.
3. Cloud server will forward the request from the user to the data owner.
4. Data owner wants to accept the request from the cloud server.
5. Cloud server will forward a de encrypted key to the user.
6. Simultaneously cloud server will create a virtual server and decrypts the data from the cloud storage server.
7. User can enter the de encrypted key to the proxy server to view the original data. The key will be valid for one time only.
8. After the data view from the proxy server, the virtual data will be deleted automatically.

ADVANTAGES OF USING PROXY SERVER:

A proxy server has a variety of potential purposes, including:

- To keep machines behind it anonymous, mainly for security.
- To speed up access to resources (using caching). Web proxies are commonly used to cache web pages from a web server.
- To prevent downloading the same content multiple times (and save bandwidth).

- To log / audit usage, e.g. to provide company employee Internet usage reporting.
- To scan transmitted content for malware before delivery.
- To scan outbound content, e.g., for data loss prevention.
- To bypass website restrictions at work

TYPES OF PROXY SERVERS:

- A proxy server may run right on the user's local computer or at various points between the user's computer and destination servers on the Internet.
- A proxy server that passes requests and responses unmodified is usually called a gateway or sometimes a tunnelling proxy.
- A forward proxy is an Internet-facing proxy used to retrieve from a wide range of sources (in most cases anywhere on the Internet).
- A reverse proxy is usually an Internet-facing proxy used as a front-end to control and protect access to a server on a private network, commonly also performing tasks such as load-balancing, authentication, decryption or caching.

VI. METHODOLOGIES

6.1 Distributed Erasure Code:

We consider the problem of constructing an erasure code for storage over a network when the data sources are distributed in the cloud server. Specifically, we assume that there are n storage nodes with limited memory and k < n sources generating the data. We want a data collector, who can appear anywhere in the network for accessing the data, to query any k storage nodes and be able to retrieve the data. We introduce Decentralized Erasure Codes, which are linear codes with a specific randomized structure inspired by network coding on random bipartite graphs with encrypted format. We show that decentralized erasure codes are optimally sparse, and lead to reduced communication, storage and computation cost over random linear coding over the cloud server.

6.2 Erasure code over a cloud Network

Decentralized erasure codes are random linear codes over a finite field Fq with a specific randomized structure on their generator matrix. Each data packet Di is seen as a vector of elements of a finite field fi . We denote the set of data nodes by V1 with |V1| = k and storage nodes by V2, |V2| = n. We will now give a description of a randomized construction of a bipartite graph that corresponds to the creation of a decentralized erasure code. Every data node i ∈ V1 is assigned a random set of storage nodes N (i). This set is created as follows: a storage node is selected uniformly and independently from V2 and added in N(i) and this procedure is repeated d(k) times. Therefore N (i) will be smaller than d(k) if the same storage node is selected twice. In fact, the size of the set N (i) is exactly the number of coupons a coupon collector would have after purchasing d(k) coupons from a set of n coupons. It is not hard to see that when d (k) ≪ n, N(i) will be approximately equal to d(k) with high probability. Denote by N (j) = {i ∈ V1 : j ∈ N(i)} the set of data nodes that connect to a storage node. Each storage node will create a random linear combination of the data nodes it is connected with:

Sj = X ∀i:∈N(j) fij Di where the coefficients fij are selected uniformly and independently from a finite field Fq. Each storage node also stores the fij coefficients, which requires an overhead storage of N(j)(log2 (q) + log2 (k)) bits. This construction can be summarized into s = mG where s is a 1 × n vector of stored data, m is 1 × k data vector and G is a k × n matrix with non-zero entries corresponding to the adjacency matrix of the random bipartite graph we described. The key property that allows the decentralized construction of the code is that each data node is choosing its neighbours independently and uniformly or equivalently, every row of the generator matrix is created independently and has N(i) = O(d(k)) nonzero elements. This row independence, which we call "decentralized property. We compare our results with random linear coding for distributed networked storage. A data collector querying k storage nodes will gain access to k encoded packets.

To reconstruct, the data collector must invert a k × k sub matrix G′ of G. Therefore, the key property required for successful decoding is that any selection of G′ forms a full rank matrix with high probability. Clearly d(k) is measuring the sparsely of G. Making d(k) as small as possible is very important since it is directly related with overhead storage, decoding complexity and communication cost. Our main contribution is identifying how small d(k) can be made for matrices that have the decentralized property to ensure that their sub matrices are full rank with high probability. The following theorems are the main results of this correspondence:

6.3 Theorem 1: Let G be a random matrix with independent rows constructed as described. Then, d(k) = c ln (k) is sufficient for a random k ×k sub matrix G′ of G to be non-singular with high probability. More specifically, P r det(G′) = 0] ≤ k q + o(1) for any c > 5 n k.

6.4 Theorem 2: (Converse) If each row of G is generated independently (Decentralized property), at least d(k) = Ω(ln(k)) is necessary to have G′ invertible with high probability. From the two theorems it follows that d(k) = c ln(k) is (order) optimal, therefore, decentralized erasure codes have minimal data node degree and logarithmically many nonzero elements in every row. Decentralized erasure codes can be decoded using Maximum Likelihood (ML) decoding, which corresponds to solving a linear system of k equations in GF(q). This has a decoding complexity of O(k 3).
Note however that one can use the sparsely of the linear equations and have faster decoding. Using the Wiedemann algorithm one can decode in O(k 2 log(k)) time on average with negligible extra memory requirements.

VII. CONCLUSION AND FUTURE WORK

7.1 CONCLUSION

Thus we are concluding that all the result obtained according to the committed abstract. In this paper, we consider a cloud storage system consists of storage servers and key servers. We integrate a newly proposed threshold proxy re-encryption scheme and erasure codes over exponents. The threshold proxy re-encryption scheme supports encoding, forwarding, and partial decryption operations in a distributed way. To decrypt a message of k blocks that are encrypted and encoded ton code word symbols, each key server only has to partially decrypt two codeword symbols in our system. By using the threshold proxy re-encryption scheme, we present a secure cloud storage system that provides secure data storage and secure data forwarding functionality in a decentralized structure. Moreover, each storage server independently performs encoding and re-encryption and each key server independently perform partial decryption. Our storage system and some newly proposed content addressable file systems and storage systems are highly compatible. Our storage servers act as storage nodes in a content addressable storage system for storing content addressable blocks. Our key servers act as access nodes for providing a front-end layer such as a traditional file system interface. Further study on detailed cooperation is required.

7.2 FUTURE WORK

- Layer 7 virtual proxy servers can be used.
- Output can be shown using some medical domain for real time implementation.
- EC2 Cloud server can be implemented.
- Key character length can be increased for more security
- HTTPS can be implemented.
- Network Address Translation (NAT) can be implemented
- Big data concepts can be included in case of huge number of data transaction.

REFERENCES

[1] R. Bhagwan, K. Tati, Y.-C. Cheng, S. Savage, and G.M. Voelker, "Total Recall: System Support for Automated Availability Management,"Proc. First Symp. Networked Systems Design and Implementation (NSDI),pp. 337-350, 2004.
[2] A.G. Dimakis, V. Prabhakaran, and K. Ramchandran, "Ubiqui-tous Access to Distributed Data in Large-Scale Sensor Net-works through Decentralized Erasure Codes,"Proc. Fourth Int'l Symp. Information Processing in Sensor Networks (IPSN),pp. 111-117, 2005.
[3] A.G. Dimakis, V. Prabhakaran, and K. Ramchandran, "Decen-tralized Erasure Codes for Distributed Networked Storage,"IEEE Trans. Information Theory,vol. 52, no. 6 pp. 2809-2816, June 2006.
[4] M. Mambo and E. Okamoto, "Proxy Cryptosystems: Delegation of the Power to Decrypt Ciphertexts," IEICE Trans. Fundamentals of Electronics, Comm. and Computer Sciences,vol. E80-A, no. 1, pp. 54-63, 1997.
[5] M. Blaze, G. Bleumer, and M. Strauss, "Divertible Protocols and Atomic Proxy Cryptography,"Proc. Int'l Conf. Theory and Applica-tion of Cryptographic Techniques (EUROCRYPT),pp. 127-144, 1998.
[6] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage,"ACM Trans. Information and System Security, vol. 9, no. 1, pp. 1-30, 2006.
[7] Q. Tang, "Type-Based Proxy Re-Encryption and Its Construction," Proc. Ninth Int'l Conf. Cryptology in India: Progress in Cryptology (INDOCRYPT),pp. 130-144, 2008.

