International Journal of Interdisciplinary Innovative Research &Development (IJIIRD)
ISSN: 2456-236X
Vol. 05 Special Issue 01 | 2020

A Cloud Storage System With Proxy Re-Encryption Scheme

[1]Mr. Mahesh Vijay Shastri, [2]Mr. Shrikrushna Govind Jadhao, [3]Mr.Harshal Shriram Deshpande

[1.] *Lecturer, Computer Science & Engineering Department, Pdm.Dr.V.B.Kolte College Of Engineering, Malkapur, Maharashtra, India*
[2.] *Programmer, Computer Engineering Department, Rambhau Lingade Polytechnic, Buldana, Maharashtra, India*
[3.] *Student M.E (Second year), Computer Science & Engineering Department, Pdm.Dr.V.B.Kolte College of Engineering, Malkapur, Maharashtra, India*

ABSTRACT

Cloud computing is a model for enabling convenient, on demand network access to a shared pool of computing resources. The cloud storage system is a collection of storage servers and key servers. Storing data in a third party cloud system causes serious concern on data confidentiality. In order to provide strong confidentiality for messages in storage servers, a user can divides the data into blocks, encrypt messages by a cryptographic method and stores them in various storage servers. General encryption schemes protect data confidentiality but also limit the functionality of the storage system. In the proposed system a secure distributed storage system is formulated by integrating a threshold proxy re-encryption scheme. Their main technical operations are encrypting, encoding and forwarding.

Keyword: -Distributed Storage system; Encoding; Proxy Re-Encryption; Encrypted data;Cryptographic keys;

1. INTRODUCTION

As high-speed networks and ubiquitous Internet access become available in recent years, many services are provided on the Internet such that users can use them from anywhere at any time. Cloud computing is an evolutionary new model for distributed computing consisting of centralized data center that provide resources for massively scalable units of computing. These computational facilities are delivered as a service to users over an insecure medium such as the Internet, and may be bridged to wireless packet data networks.[2]

Cloud storage is a service model in which data is maintained, managed and backed up remotely and made available to user via internet. A cloud storage server is servers located at different places and provides continuous access. Cloud storage system is a collection of such storage servers. A cloud storage system can be considered to be a network of distributed data centers which typically uses cloud computing technologies like virtualization, and offers some kind of interface for storing data. To increase the availability of the data, it may be redundantly stored at different locations. It is desired that data stored in the system remain private even if all storage servers in the system are compromised. The major challenge of designing these distributed networked storage systems is to provide a better privacy guarantee while maintaining the distributed structure. To achieve this goal, we introduce secure decentralized erasure code, which combines a threshold public key encryption scheme and a variant of the decentralized erasure code. Our secure distributed networked storage system constructed by the secure decentralized erasure code is decentralized and robust. Because of storage server can join or leave without control of a central authority, a decentralized architecture for storage systems offers good scalability.In this paper, we focus on designing a cloudstorage system for robustness, confidentiality, and functionality.[5]

A cloud storage system is considered as a large scale distributed storage system that consists of many independent storage servers. Because of the huge amount of data stored by a cloud, efficient processing and analysis of data has become a challenging one. Data robustness is a major requirement for storage systems. There have been many proposals of storing data over storage servers. One way to provide data robustness is to replicate a message such that each storage server stores a copy of the message. It is very robust because the message can be retrieved as long as one storage server survives. 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 decentralized erasure code is an erasure code that independently computes each codeword symbol for a message. Thus, the encoding process for a message can be split into n parallel tasks

of generating codeword symbols. A decentralized erasure code is suitable for use in a distributed storage system. [1]

Decentralized storage systems aggregate the available disk space of participating computers to provide a large storage facility. These systems rely on data redundancy to ensure durable storage despite of node failures. After the message symbols are sent to storage servers, each storage server independently computes a code word symbol for the received message symbols and stores it. This finishes the encoding and storing process. The recovery process is the same.[9]

In straightforward integration method Storing data in a third party's cloud system causes serious concern on data confidentiality. Constructing centralized storage system for the cloud system makes hackers stole data easily. 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.[7]

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.[10]There are three problems in the above straightforward integration of encryption and encoding:

- The user can perform more computation and communication traffic between the user and storage servers is high.
- The user has to manage his cryptographic keys. If the user's tool of storing the keys is vanished or compromise, the security is broken.
- The data storing and retrieving, it is hard for storage servers to directly support other functions.

To overcome this problems propose a threshold proxy re-encryption scheme and integrate it with a decentralized erasure code such that a secure distributed storage system is formulated. When the sender wants to share his messages, he sends a re-encryption key to the storage server. The storage server re-encrypts the original codeword symbol into a re-encrypted codeword symbol. The re-encryption scheme is integrated with a secure decentralized erasure code so that a secure distributed system is designed. The proxy re-encryption scheme supports the encoding operations over encrypted messages as well as forwarding operations. The key server retrieves re-encrypted codeword symbols and performs partial decryption to retrieve the data. This distributed storage scheme lets a user forward his data in the storage servers to another user without retrieving the data back.[6]

In our proposed system we address the problem of forwarding data to another user by storage servers directly under the command of the data owner. We consider the system model that consists of distributed storage servers and key servers. Since storing cryptographic keys in a single device is risky, a user distributes his cryptographic key to key servers that shall perform cryptographic functions on behalf of the user. These key servers are highly protected by security mechanisms. The distributed systems require independent servers to perform all operations. We propose a new threshold proxy re-encryption scheme and integrate it with a secure decentralized code to form a secure distributed storage system. The encryption scheme supports encoding operations over encrypted messages and forwarding operations over encrypted and encoded messages. Storing cryptographic keys in a single device is risky; a user distributes his cryptographic key to key servers that shall perform cryptographic functions on behalf of the user. These key servers are highly protected by security mechanisms.[12]

The tight integration of encoding, encryption, and forwarding makes the storage system efficiently meet the requirements of data robustness, data confidentiality, and data forwarding. Accomplishing the integration with consideration of a distributed structure is challenging. Our system meets the requirements that storage servers independently perform encoding and re-encryption and key servers independently perform partial decryption. Moreover, we consider the system in a more general setting than previous works. This setting allows more flexible adjustment between the number of storage servers and robustness.[11]

1.1 System Architecture-

In a proxy re-encryption scheme, when a user wants to share his messages, he sends a re-encryption key to the storage server such that storage servers perform the re-encryptionoperation for him. Thus, the communication cost of the user is independent of the length of shared message and the computation cost of re-encryption is taken care of by storage servers. The overhead of the data forwarding function in a secure storage system is significantly reduced by proxy re-encryption schemes. In a proxy re-encryption scheme, a proxy server can transfer a cipher text under a public key PKa to a new one under another public key PKb by using the re-encryption key RKa->b. The server does not know the plaintext during transformation. Our work further integrates re-encryption, and encoding such that storage robustness is strengthened.[3]

A. System Model

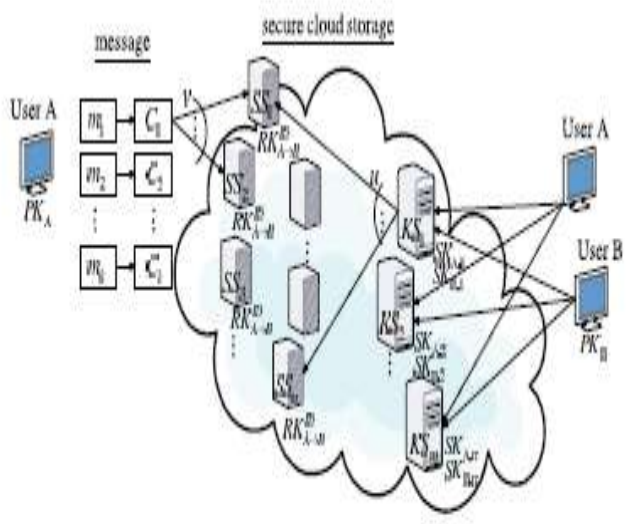

Fig- 1: Model of the System

Proposed system model consists of users, n storage servers SS1; SS2; . . . ; SSn, and m key servers KS1; KS2; . . . ; KSm as shown in Fig.[1] Storage servers provide storage services and key servers provide key management services. They work independently.[10]

In an integration processes, the splinted message is joined into an m number of blocks, and stored into lager storage server. User A encrypts his message M is decomposed into k number of blocks m1, m2, . . .,mk and which has an identifier ID. User A encrypts each block mi into a cipher text Ci and sends it to v randomly chosen storage servers. Upon receiving cipher texts from a user, each storage server linearly combines them with randomly chosen coefficients into a codeword symbol and stores it. Note that a storage server may receive fewer than k message blocks and we assume that all storage servers know the value k in advance. Integration is used to combine messages into m number of block, which is encrypted and stored into a large number storage server. Then forward to user B. Data which is encrypted by using single key. This is produced by using hash key algorithm.[8]

B. System Implementation

Once the system has been designed, the next step is to convert the designed one in to actual code, so as to satisfy the user requirements as excepted. System can be implemented if it is approved to be error free. The department was consulted for acceptance of the design, when the initial design was done for the system; so that further proceedings of the system development can be carried on. After the development of the system, a demonstration was given to them about working of the system.

The aim of the system illustration was to identify any malfunctioning of the system. Implementation includes proper training to end-users. The implemented software should be maintained for prolonged running of the software. Initially the system was run parallel with manual system. To prove the system to be error-free and user-friendly, the system has been tested with data. Training was given to end -user about the software and its features. The encryption scheme supports encoding operations over encrypted messages and forwarding operations over encrypted and encoded messages. The tight integration of encoding, encryption, and forwarding makes the storage system efficiently meet the requirements of data robustness, data confidentiality, and data forwarding.

To provide data confidentiality, the data is been encrypted using cryptographic keys. Authenticating the users entering the network can also be done to secure the data. The cryptographic keys must be kept secret and it must not be lost by the user. Only after registration process or login process they must be allowed to enter the system. The following steps are followed in proposed system:

1. User creates an account.
2. His information will be stored in storage server and a key will be given.
3. The authenticated user can upload files.
4. He can also forward and retrieve files.

By sharing the id of the data and id of the user to the storage server the user can also forward data to other user. This reduces the computation done by the user.[11]

C. Process of the System

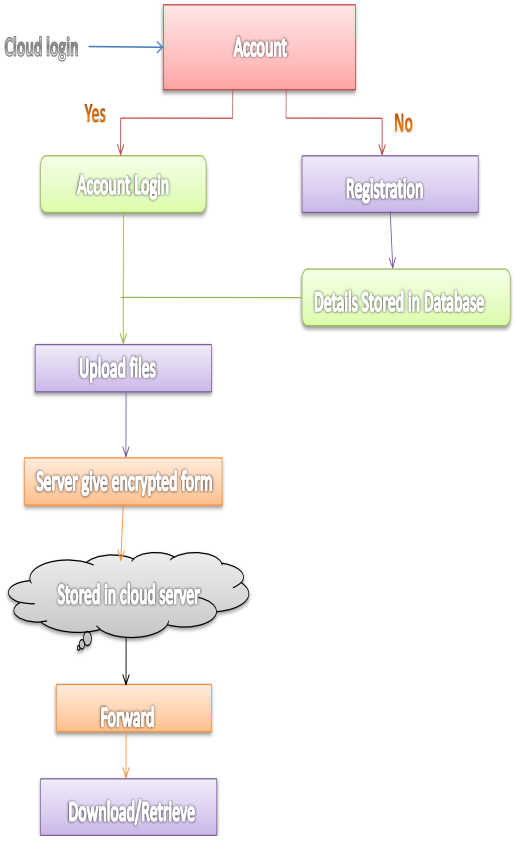

Figure 2: Process of the System

- *Login:*

First the user A will be allowed to enter into the cloud. Login page make user A to access an account in cloud server. If user A has account in cloud then he enters his login id and password to enter into the system otherwise he has to do the registration process. When user has an account in the cloud server for accessing data and provides other services. User A can sign up page directly. If the user A cannot have the account for cloud system, then he has to register his details for using and entering into the cloud system. After registration process completed, his all details stored in database.

- *User Detail Store in Database:*

The user details such as name, e-mail, qualification, location, department id, login id, password, etc, after registration process are stored in the database. The login id and password entered by user will be check from database and only if his credentials are correct then he will be allowed to enter into the cloud system. This process improves the security of the system.

TABLE I. User details stored in the database

Column Name	Data Type	Description
First Name	Char	First name of the user
Last Name	Char	Last name of the user
E-mail	String	E-mail id of the user
Qualification	Char	Qualification of the user
Location	Char	Address of the user
Department ID	Numeric	ID of the user department
Login ID	Char	Account login ID of the user
Password	Char	Password of the user

- *Upload Files:* User A after sign up into his account can upload or forward data to another using his account. For upload process the user A has to choose file from the system and enter the upload option.
- *Encrypted Form:* User A upload file along with a key which is used to encrypt the text. After uploading process, server from the cloud can give the encrypted form of the uploading file.
- *File Stores In Cloud Server:* Encrypted file stores in cloud servers SSi using re-encryption technique. Keys are distributed between the servers KSi.
- *Keys Stored In Cloud Storage Server:* Cryptographic keys storing in a single server is risky, a user distributes his cryptographic key to key servers, and then servers perform cryptographic functions on behalf of the user. These type security mechanisms highly protect the key servers.

TABLE II. User details stored in the database

Column Name	Data Type	Description
Login ID	Char	ID of the user
Prod Key	Char	Key of the user

- *File Forwarding:* In file forwarding process, user A forwards his encrypted file with id stored in storage server to user B. Then user B can decrypt the forwarded file by using his secrete key. To do so,user A uses his secrete key SKA and B's public key PKB to compute a re-encrypted key RKIDA->B and then sends to all storage servers. Every storage server uses the re-encryption key to re-encrypt its codeword symbol for later retrieval needs by B.

- *Forwarding Process:* User A forwards his encrypted file with id stored in storage server to user B. Then user B can decrypt the forwarded file by using his secrete key. This process provides the data forwarding with confidentiality.

TABLE III. Forwarding Process

Column Name	Data Type	Description
File name	Char	Selected file name which is forward
Encrypted id	Numeric	ID of file
Secrete key	Char	Secrete key of file
Security question	Char	Question ask by server for data forwarding

- *File Retrieval:* File retrieval is the final process of the system. User download or retrieve file using proxy re-encryption scheme text decode and partial decrypted. In file retrieval phase, user A retrieves a file which is either stored by user A or forwarded to user A from storage server. User A sends a recovery request to key servers. Upon receiving the recovery request and execute a proper verification process with user A, each key server KSi needs u randomly chosen storage servers to get code symbols and does partial decryption on the received code symbols by using the key share SKA,i. Finally, user A combine the partially decrypted codeword symbols to obtain the original message M.[8]

1.2 PHASES OF THE SYSTEM

Distributed storage system consists of four phases: system setup, data storage, data forwarding, and data retrieval. These four phases are described as follows:

*A. System Setup:*The Set Up process generates the system parameters. In this phase, the system manager chooses system parameters and publishes them. Each user is assigned a public-secret key pair. A user A uses KeyGen to generate his public and secret key pair to share his secret key to a set of m key servers such that each key server KSi holds a key and they shared key with a threshold t.[6]

*B. Data Storage:*When user A wants to store a message of k blocks he encrypts his message M and dispatches it to storage servers, in data storage phase. A message M is decomposed into k blocks m1; m2; . . .; mk and has an identifier ID. User A computes the identity token performs the encryption algorithm Encode and k blocks to get k original cipher texts. An original cipher text is indicated by a leading bit b = 0. After encrypting each block mi into cipher text Ci, user A sends it to v randomly chosen storage server. A storage server receives a set of original cipher texts with the same identity token from A. Upon receiving cipher texts from a user, each storage server linearly combines them with randomly chosen coefficients into a codeword symbol and stores it. The storage server inserts to the set, when a cipher text is not received. The storage server performs Encoding on the set of k cipher texts Ci and stores the encoded result in the storage server.[1]

C. *Data Forwarding:* User A wants to forward a message to another user B. In the data forwarding phase, user A forwards his encrypted message with an identifier ID stored in storage servers to user B such that B can decrypt the forwarded message by his secret key. To do so, A uses his secret key SKA and B's public key PKB to compute a re-encryption key RKida->b and then sends to all storage servers. Each storage server uses the re-encryption key to re-encrypt its codeword symbol for retrieval requests by B. The re-encrypted codeword

symbol is the combination of cipher texts under B's public key. User A does not take the risk of forwarding the data himself. He just gives the data id to data server and of the user to whom he wants to send the message to the key server. And the data will be sending to the user by the data and key server. This reduces the computation performed by the user.[13]

D. Data Retrieval: There are two cases for the data retrieval phase. In first case, a user A retrieves his own message. In the data retrieval phase, user A sends a retrieval request to key severs with the identity token. After receiving the retrieval request and executing a proper authentication process with user A, each key server KSi requests u randomly chosen storage servers to get codeword symbols and does partial decryption on the received codeword symbols by using the key share SKa,i. Original code word symbols are retrieved by the key server and partial decryption is performed on them. The resulting code word is called partially decrypted code word symbol. Key server sends symbols and coefficients to user A. Finally, user A combines the partially decrypted codeword symbols to obtain the original message M. In second case, a user B retrieves a message forwarded to him. User B informs all key servers directly. Here the key servers retrieve re-encrypted code word symbols and perform partial decryption.[4]

2. ADVANTAGEOUS OF SYSTEM

- Convince. In Cloud computing you can access your data anywhere you can connect to the Internet.
- Data robustness. Provide data robustness in storage system by replicating a message such that each storage server stores a copy of the message.
- Data confidentiality. Integration of threshold proxy re-encryption scheme with a decentralized erasure code provides data confidentiality.
- Secure distributed system. Distributed storage scheme lets a user forward his data in the storage servers to another user without retrieving the data back.
- Protected key servers. A user distributes his cryptographic key to key servers that shall perform cryptographic functions on behalf of the user.

3. RESULT

- *User Login:*In login page user has to enter his user name and password to use the service of cloud storage system. If user has no account on cloud storage system then he has to registration for cloud account and his details will be stored in database. After login to account, his username and password will be verified, which are stored in database. If both matches then user will be allow to enter into the cloud storage system. This results in authenticate user entering the cloud system thus it increases the security of the cloud system.

Fig-3: user login page

- *Key Generation:* Each authenticated user can generate a secrete key along with file. And only with the help of that secrete key, user can store, forward or retrieve file. Secrete key only known to user. This secrete key must not be shared with other users.

Fig -4: Key Generation

- *Data Storage, Forward and Retrieval:*After key generation user's data stored in cloud server along with secrete key, then user is able to stored or forward his data. In forwarding process the cloud system may ask one question for that user. The user should answer the question. While user retrieves that file, again question may ask by cloud storage system. This improves the data confidentiality and data forwarding in secure way.

Fig -5: Data Storage, Forward and Retrieve

4. FUTURE SCOPE

The system develop in this work is fully functional and can be applied in practical situations. The future scope will be concerned with the security of users' passwords on the internet. Prevent users' passwords from being stolen by adversaries. We will be use virtual password, biometrics, and digital signature for login into account on cloud storage system with proxy re-encryption scheme.

5. CONCLUSIONS

The focus of this paper is to provide a well defined data storage, forwarding and retrieving between the user and cloud storage system. The cloud storage system is providing the security to the data by using proxy re-encryption scheme. The threshold proxy re-encryption scheme supports encoding, forwarding, and partial decryption operations in a distributed way. So by using the proxy re-encryption scheme we present secure cloud storage system that provides secure data storage, forwarding and retrieving functionality in decentralized system. For secure forwarding the data re-encryption is performed and then sent to cloud storage system. Each storage server in cloud system independently performs encoding and re-encryption and each key server independently perform partial decryption. When user send request to retrieve the data from storage system key server perform re-encryption key on demand for partial decryption.

6. REFERENCES

[1] www.google.co.in/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&ved=0CEwQFjAC&url=http%3A%2F%2Fwww.ijetae.com%2Ffiles%2FConference%2520ICISC-2013%2FIJETAE_ICISC_0113_79.pdf&ei=Q61FUsqHKMLYrQeEroCwAQ&usg=AFQjCNGhYtsjFkZEpN0VCO4YX50hqKK0egpublished

[2] https://www.google.co.in/url?sa=t&rct=j&q=&esrc=s&source=web&cd=9&ved=0CHMQFjAI&url=http%3A%2F%2Feprint.iacr.org%2F2011%2F668.pdf&ei=Q61FUsqHKMLYrQeEroCwAQ&usg=AFQjCNEiKm_hKVqgn3fL9xKeE2KFdgGE8g unpublished

[3] Bernd Grobauer, Tobias Walloschek, and ElmarStöcker*Siemens, "Understanding Cloud Computing Vulnerabilities*||, IEEE Journal On Computer And Reliability Socities, Vol. 9, Issue :2, pp. 50-57, March/April 2011 *references*

[4] Moritz Borgmann, Tobias Hahn, Michael Herfert, Thomas Kunz, Marcel Richter, Ursula Viebeg, Sven Vow´e Ed. Michael Waidner SIT Technical Reports" On the Security of Cloud Storage Services" SIT-TR-2012-001 March 2012 *references*

[5] Adya, W.J. Bolosky, M. Castro, G. Cermak, R. Chaiken, J.R. Douceur, J. Howell, J.R. Lorch, M. Theimer, and R. Wattenhofer, "Farsite: Federated, Available, and Reliable Storage for an Incompletely Trusted Environment," Proc. Fifth Symp. Operating System Design and Implementation (OSDI), pp. 1-14, 2002. *references*