IJCSE International Journal of Computer Sciences and Engineering Open Access
Survey Paper Vol.-7, Special Issue, 6, March 2019 E-ISSN: 2347-2693

A Study on Secure Cloud server System using Proxy Re-Encryption Model

G. Praveen[1*], V. Ravikumar Chowdary[2]

[1,2]Dept. of Computer Science, RCR Institute of Management & Technology, S V Uuniversity, Tirupati, India

Corresponding Author: praveenganamaneni@gmail.com

DOI: https://doi.org/10.26438/ijcse/v7si6.99102 | Available online at: www.ijcseonline.org

***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.

Keywords: Architecture, Construction of Secure Cloud Storage Systems

I. INTRODUCTION

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.

II. ARCHITECTURE SCOPE OF THE PROJECT

Designing a cloud storage system for robustness, confidentiality and functionality. The proxy re-encryption scheme supports encoding operations over encrypted messages as well as forwarding operations over encoded and encrypted messages. 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.

The number of failure 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. This provides a tradeoff between the storage size and the tolerance threshold of failure servers.

A decentralized erasure code is an erasure code that independently computes each codeword symbol for a message. A decentralized erasure code is suitable for use in a distributed storage system.
A storage server failure is modeled as an erasure error of the stored codeword symbol.

We construct a secure cloud storage system that supports the function of secure data forwarding by using a threshold proxy re-encryption scheme. The encryption scheme supports decentralized erasure codes over encrypted messages and forwarding operations over encrypted and encoded messages. Our system is highly distributed where storage servers independently encode and forward messages and key servers independently perform partial decryption.

III. EXISTING SYSTEM

In Existing System we use a straightforward integration method. In straightforward integration method Storing data in a third party's cloud system causes serious concern on data confidentiality. 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. General encryption schemes protect data confidentiality, but

also limit the functionality of the storage system because a few operations are supported over encrypted data.
A decentralized architecture for storage systems offers good scalability, because a storage server can join or leave without control of a central authority.

DISADVATAGES OF EXISTING SYSTEM:

- 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 otherwise the security has to be broken.
- The data storing and retrieving, it is hard for storage servers to directly support other functions.

IV. PROPOSED SYSTEM

In our proposed systemwe 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.

ADVANTAGES OF PROPOSED SYSTEM:

- 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.
- More flexible adjustment between the number of storage servers and robustness.

V. CONSTRUCTION OF SECURE CLOUD STORAGE SYSTEMS

We use a threshold proxy re-encryption scheme with multiplicative homomorphism property. An encryption scheme is multiplicative homomorphic if it supports a group operation where E is the encryption function, D is the decryption function and (PK, SK) is a pair of public key and secret key.
Thus, a multiplicative homomorphic encryption scheme supports the encoding operation over encrypted messages. We convert a proxy re-encryption scheme with multiplicative homomorphic property into a threshold version. A secret key is shared to key servers with a threshold value t. To decrypt for a set of message symbols, each key server independently queries 2 storage servers and partially decrypts two encrypted code word symbols. As long as t key servers are available, code word symbols are obtained from the partially decrypted cipher texts.

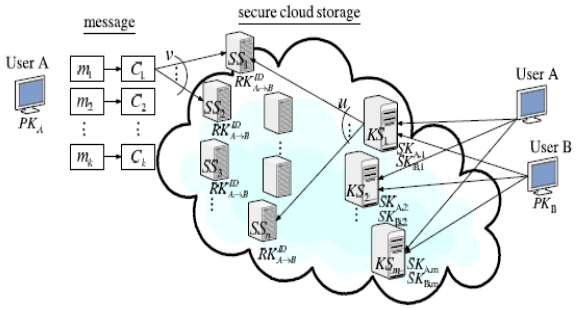

Fig.3. A Storage System with Random Linear Coding over exponents

5.1 A Secure Cloud Storage System with Secure Forwarding

We assume that there are n storage servers which store data and m key servers which own secret key shares and perform partial decryption. The owner shares the secret key x to m key servers with a threshold t. The storage process, re-encryption process and the retrieval process are described.

Storage process:
To store k messages, the storage process is as follows:
- Message encryption. The owner encrypts all k messages via the data encryption key with the identifier ID for the set of messages M1, M2,….., Mk.
-Cipher text distribution. For each Ci, the owner randomly chooses v storage servers (with replacement) and sends each of them a copy of Ci.
-Decentralized Encoding. For all received cipher texts with the same message identifier hID, the storage server SSj groups them as Nj. The storage server SSj selects a random coefficient and forms a generator matrix of the decentralized erasure code.

Re-Encryption Process:
To forward a message, the re-encryption process is as follows:
-Key Recover. User A queries key servers for key shares. When at least t key servers respond, A recovers the first component a1 of the secret key SKA via the Key Recover algorithm.
-Key Re-Generation. User A computes the re-encryption key algorithm and securely sends the re-encryption key to each storage server. By using RKID, a storage server re-encrypts the original codeword symbol

Retrieval process:
To retrieve k messages, the retrieval process is as follows:
- Retrieval command. The owner sends a command to the m key servers with the message identifier hID.- Partial decryption. Each key server KSi randomly queries u storage servers with the message identifier hIDand obtains at most u stored data from the storage servers. The key server retrieves the re-encrypted codewordsymbols from the storage server and performs partial decryption. Then, the key server KSiperforms ShareDecon each received cipher text by its secret key share to obtain a decryption share of the cipher text.

Storage cost:
To store a message of k blocks, a storage server SSj stores a codeword symbol and the coefficient vector .The average cost for a message stored in a storage server is bits, which is dominated for a sufficiently large value of k. In practice, small co-efficients, reduce the storage cost in each storage server. In practice, small coefficients reduce the storage cost in each storage server

Computation cost:
We measure the computation cost by the number of pairing operations, modular exponentiation in G1 and G2, modular multiplications in G1 and G2, and arithmetic operations over GF(p). The cost is summarized in

Correctness:
There are two cases for correctness. The owner A correctly retrieves his message and user B correctly retrieves a message forwarded to him. The correctness of re-encryption and decryption for B can be seen. As long as at least k storage servers are available, a user can retrieve data with an overwhelming probability. Thus, our storage system tolerates n - k server failures.

The probability of a successful retrieval:
A successful retrieval is an event that a user successfully retrieves all k blocks of a message no matter whether the message is owned by him or forwarded to him. The randomness comes from the random selection of storage servers in the data storage phase, the random coefficients chosen by storage servers, and the random selection of key servers in the data retrieval phase.
The Computation Cost of Each Algorithm in Our Secure Cloud Storage System
- Pairing: a pairing computation of *e*.
- Exp1 and Exp2: a modular exponentiation computation in G1 and G2, respectively.
- Mult1 and Mult2: a modular multiplication computation in G1 and G2, respectively.
- Fp: an arithmetic operation in *GF (p).*
The difference between our system model and the one in is that our system model has key servers. A single user queries k distinct storage servers to retrieve the data. On the other hand, each key server in our system independently queries u storage servers. The use of distributed key servers increases the level of key protection but makes the analysis harder. Our generalization of parameter setting allows the number of storage servers be much greater than the number of blocks of a message. It gives a better flexibility for adjustment between the number of storage servers and robustness.
Security: The data confidentiality of our cloud storage system is guaranteed even if all storage servers, non target users, and up to (t-1) key servers are compromised by the attacker.

VI. METHODOLOGY USED:

In the proxy Re-encryption key the messages are first encrypted by the owner and then stored in a storage server. When a user wants to share 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, their system has data confidentiality and supports the data forwarding function.

An encryption scheme is multiplicative homomorphic if it supports a group operation on encrypted plaintexts without decryption. The multiplicative homomorphic encryption scheme supports the encoding operation over encrypted messages. We then convert a proxy re-encryption scheme with multiplicative homomorphic property into a threshold version. A secret key is shared to key servers with a threshold value t. To decrypt for a set of k message symbols, each key server independently queries 2 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.
Given a proxy re-encryption key, the proxy can transform a cipher text of one user to a cipher text of the target user.Threshold decryption: By dividing the private key into several pieces of secret shares, all clients can work together to decrypt the cipher text – the output of the function.
MODULES:
- Construction of Cloud Data Storage Module
- Data Encryption Module
- Data Forwarding Module
- Data Retrieval Module

VII.MODULES DESCRIPTION:

Construction of Cloud Data Storage Module
In Admin Module the admin can login to give his username and password.Then the server setup method can be opened. In server setup process the admin first set the remote servers Ip-address for send that Ip-address to the receiver. Then the server can skip the process to activate or Dis-activate the process. For activating the process the storage server can display the Ip-address. For Dis-activating the process the

storage server cannot display the Ip-address. These details can be viewed by clicking the key server. The activated Ip-addresses are stored in available storage server. By clicking the available storage server button we can view the currently available Ip-addresses.

Data Encryption Module

In cloud login module the user can login his own details. If the user cannot have the account for that cloud system first the user can register his details for using and entering into the cloud system. The Registration process details are Username, E-mail, password, confirm password, date of birth, gender and also the location. After entering the registration process the details can be stored in database of the cloud system. Then the user has to login to give his corrected username and password the code has to be send his/her E-mail. Then the user will go to open his account and view the code that can be generated from the cloud system.

In Upload Module the new folder can be create for storing the files. In folder creation process the cloud system may ask one question for that user. The user should answer the question and must remember that answer for further usage. Then enter the folder name for create the folder for that user. In file upload process the user has to choose one file from browsing the system and enter the upload option. Now, the server from the cloud can give the encrypted form of the uploading file.

Data Forwarding Module

In forward module first we can see the storage details for the uploaded files. When click the storage details option we can see the file name, question, answer, folder name, forward value (true or false), forward E-mail. If the forward column display the forwarded value is true the user cannot forward to another person. If the forward column display the forwarded value is false the user can forward the file into another person.

Data Retrieval Module

In Download module contains the following details. There are username and file name. First, the server process can be run which means the server can be connected with its particular client. Now, the client has to download the file to download the file key. In file key downloading process the fields are username, filename, question, answer and the code. Now clicking the download option the client can view the encrypted key. Then using that key the client can view the file and use that file appropriately.

VIII.CONCLUSION

We consider a cloud storage system that consists of storage servers and key servers. Our systemprovides both the storage service and key management service. A secure cloud storage system is constructed using the threshold proxy re-encryption scheme that provides secure data storage and secure data forwarding functionality in a decentralized structure. The threshold proxy re-encryption scheme supports encoding, forwarding, and partial decryption operations in a distributed way.

REFERENCES

[1] J. Kubiatowicz, D. Bindel, Y. Chen, P. Eaton, D. Geels, An Architecture for Global-Scale Persistent Storage," Proc. Ninth Int'l Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS), pp. 190-201, 2000.

[2] P. Druschel and A. Rowstron, "PAST: A Large-Scale, Persistent Peer-to-Peer Storage Utility," Proc. Eighth Workshop Hot Topics in Operating System (HotOS VIII), pp. 75-80, 2001.

[3] A. 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.

[4] Haeberlen, A. Mislove, and P. Druschel, "Glacier: Highly Durable, Decentralized Storage Despite Massive Corr elated Failures," Proc. Second Symp.Networked Systems Design and Implementation (NSDI), pp. 143-158, 2005.

[5] A.G. Dimakis, V. Prabhakaran, and K. Ramchandran, "Decentralized Erasure Codes for Distributed Networked Storage," IEEE Trans. Information Theory, vol. 52, no. 6 pp. 2809-2816, June 2006.

