INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATIONS AND ROBOTICS
ISSN 2320-7345

A SECURE CLOUD STORAGE SYSTEM WITH SECURE DATA FORWARDING

P.Raja Kaushik[1], G.Praveen Babu[2]

[1]*Masters in Computer Networks and Information Security, SIT, JNTU, Hyderabad, INDIA*
*E-mail:*rajakaushik23@yahoo.com

[2] *Associate professor (CSE), SIT, JNTU, Hyderabad, INDIA E-mail:* pravbob@jntuh.ac.in

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 and to reduce data management costs. However, we must provide security guarantees for the outsourced data. We design and implement a secure cloud storage system that provides robustness, confidentiality, functionality for the services it provides. File access policies are includes in the design for secure access to the data stored. To achieve such security goals, a set of cryptographic key operations that are maintained by a separate key server(s) or manager(s). 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.

Keywords*:* Decentralized erasure code, proxy re-encryption, threshold cryptography, secure storage system, access control, cloud storage

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. 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 as a large scale distributed storage system that consists of many independent storage servers. 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 storage server failure corresponds to an erasure error of the codeword symbol. As

Vol.2 Issue 8, Pg.: 167-171
August 2014

long as 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 trade off 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. 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. After the message symbols are sent to storage servers, each storage server independently computes a codeword symbol for the received message symbols and stores it. This finishes the encoding and storing process. The recovery process is the same. 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.

There are three problems in the above straightforward integration of encryption and encoding. First, the user has to do most computation and the communication traffic between the user and storage servers is high. Second, the user has to manage his cryptographic keys. If the user's device of storing the keys is lost or compromised, the security is broken. Finally, besides data storing and retrieving, it is hard for storage servers to directly support other functions. For example, storage servers cannot directly forward a user's messages to another one. The owner of messages has to retrieve, decode, decrypt and then forward them to another user. In this paper, 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. To well fit the distributed structure of systems, we require that servers independently perform all operations.

With this consideration, 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. 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.

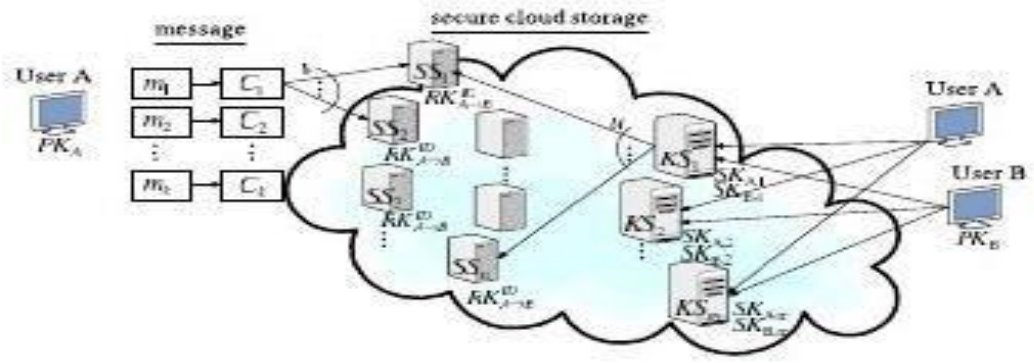

Fig.1.A general system model

2. Scope of the project:

Designing a cloud storage system for confidentiality, functionality and robustness. The proxy re-encryption scheme supports encoding operations over encrypted messages as well as forwarding operations over encoded and encrypted messages. Data robustness is provided by replicate a message such that each Storage server stores a copy of the message. It is highly 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 that can be recovered from codeword symbols stored in the available storage servers by the decoding process. This provides a relation 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 modelled 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. This system is highly distributed in nature where storage servers independently encode and forward messages and key servers independently perform partial decryption.

3. Methodology used:

3.1 Proxy Re-Encryption Scheme with Multiplicative Homomorphism Property:

The owner of the data first encrypts the data in the proxy re-encryption. For this we use proxy re-encryption key. Then the owner stores the data in the storage servers. To store the data that is stored in the storage servers we send re-encryption key t the storage servers. Here re-encryption of encrypted messages are done by the storage servers for the authorized user, by which two level f encryption is done to provide high level of security.

A group operation with encrypted plain text without decryption is done in encryption scheme which is multiplicative homomorphism in nature. Encoding operation over encrypted message is supported by the encrypted messages is supported by the encryption scheme. Now we convert proxy re-encryption scheme into threshold version.

With the threshold value ‘t’ a secret key is shared to the key servers. To decrypt for a set of k messages symbols, each key server independently share 2 storage servers and decrypts two encrypted codeword symbols to decrypted for a set of ‘k’ message symbols. For ‘t’ key servers, ‘k’ codeword symbols are obtained from partially decrypted cipher texts.

3.2 Construction of Cloud Data Storage Servers:

The admin gives his login credentials, and then setup method starts. First remote server ip- address setup process starts and the ip-address are stored in storage servers. Before that storage servers are enabled and number 0f storage servers and number of key servers are made equal for the further functionality. Then activation of ip-address are done. If activated then it displays ip-address, for dis-activation it does not shows any message. We can also check for all number of currently available ip-addresses.

3.3 Data Encryption Process:

The registration f user is done the credentials like username, password, email, gender, location are taken. Then these details are stored in the storage servers. Now the user login starts , user provides his login credentials like username and password if the given credentials matched with the data stored in the servers the user gets access to the system.

Now the user can access to storage servers to store the data. first user has to create a folder to store the data which is being uploaded. User can also rename the file accordingly. 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 .The user uploads the file which is stored in the local machine by using this system. All the uploaded files are stored in storage servers. Now all kind of further operations are performed like splitting of data, forward on the data.

3.4 Data Forwarding Process:

In forward, first we can see the storage details for the uploaded files. When click the storage details option we can see the folder name, question, answer, file 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. In file forward processes contains the selected E-mail address and file name of the forwarder and enter the code to the forwarder. Another user can check his account properly and view the code forwarded from the previous user. Then the current user login to the cloud system and to check the receive details. In receive details the forwarded file is present then the user will go to the download process.

3.5 Data Retrieval Process:

Username and filename is given importance in the download process. First server process starts running. Which means the server can be connected with its particular client. To download the file key the client has to download the file. In file key downloading process the fields are filename, username, question, answer and the code. The client can view the encrypted key by clicking the download button. Client can view that file using that key and the n use the file appropriately.

4. Conclusion

In this paper, we design cloud storage system with multiple storage servers and multiple key servers. We attach proxy re-encryption scheme and erasure code together. This cloud storage system not only supports robust and secure, data storage and retrieval, but also lets a user forward his data in the storage servers to another user without retrieving the data back. Encoding, partial decryption operations, data forwarding is supported by threshold proxy re-encryption scheme. A message of p blocks is encrypted, encoded, decrypted to n codeword symbols. In our system two codeword symbols partially decrypted by each key server. Secure data storing, data forwarding, decrypting structure functionalities are provided in our system using threshold proxy re-encryption, partial decryption done by key servers and re-encryption, encoding done by storage servers independently. 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.

REFERENCES

[1] R. Perlman, "File System Design with Assured Delete," Proc. Network and Distributed System Security Symp. ISOC (NDSS), 2007.
[2] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data," Proc. 13th ACM Conf. Computer and Comm. Security (CCS), 2006.
[3] H. Abu-Libdeh, L. Princehouse, and H. Weatherspoon, "RACS: A Case for Cloud Storage Diversity," Proc. ACM First ACM Symp. Cloud Computing (SoCC).
[4] 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.
[5] T. Dierks and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2," RFC 5246

A Brief Author Biography

P. Raja Kaushik – Received B.Tech degree in the stream of Information technology from SICET. And presently working toward M.Tech degree in the stream of Computer science information security from School of information technology, JNTUH, Hyderabad.

G. Praveen Babu – Received M.Tech degree and currently working as a associate professor of computers science and engineering , School of information technology, JNTUH, Hyderabad .