International Journal of Applied Engineering and Technology ISSN: 2277-212X (Online)
An Online International Journal Available at http://www.cibtech.org/jet.htm
2013 Vol. 3 (3) July-September, pp.1-5/Elayaraja et al.
Research Article

CLOUD STORAGE SYSTEM WITH SECURE DATA FORWARDING

Elayaraja P.[1], Mayilvahanan P.[2] and *Muthukumaravel A.[3]
[1]Department of Computer Science, VELS University, Pallavaram, Chennai – 117
[2]Department of MCA, VELS University, Pallavaram, Chennai – 117
[3]Department of MCA, BHARATH University, Selaiyur, Chennai – 74
**Author for Correspondence*

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.

INTRODUCTION
Overview of the Project
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 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 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. 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-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. Our contributions assume that there are n distributed storage servers and m key servers in the cloud storage system. A message is divided into k blocks and represented as a vector of k symbols. Our contributions are as follows:

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.

We present a general setting for the parameters of our secure cloud storage system. In practical systems, the number of storage servers is much more than k. The sacrifice is to slightly increase the total copies of an encrypted message symbol sent tutorage servers. Nevertheless, the storage size in each storage server does not increase because each storage server stores an encoded result (a codeword symbol), which is a combination of encrypted message symbols.

I. Problem Definition

- Construction of Cloud Data Storage Module
- Data Encryption Module
- Data Forwarding Module
- Data Retrieval Module

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. In file forward processes contains the selected file name, E-mail address of the forwarder and enter the code to the forwarder. Now, another user can check his account properly and view the code forwarded from the previous user. Then the current user has 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.

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.

CONCLUSION

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.

REFERENCES

Adya A, Bolosky WJ, Castro M, Cermak G, Chaiken R, Douceur JR, Howell J, Lorch JR, Theimer M and Wattenhofer R (2002). "Farsite: Federated, Available, and Reliable Storage for an Incompletely Trusted Environment," *Proceedings of Fifth Symposium Operating System Design and Implementation (OSDI)* 1-14.

Adya, Bolosky WJ, Castro M, Cermak G, Chaiken R, Douceur JR, Howell J, Lorch JR, Theimer M and Wattenhofer R (2002). "Farsite: Federated, Available, and Reliable Storage for an Incompletely Trusted Environment," *Proceedings of the 5th Symposium on Operating System Design and Implementation (OSDI)* 1-14.

Amritha S and Saravana Kumar S (2013). "Secure Data Forwarding In Distributed Environment Using Cloud Storage System", *International Journal of Computational Engineering Research* (ijceronline.com) **3**(3) 267 – 271.

Ateniese G, Benson K and Hohenberger S (2009). "Key-Private Proxy Re-Encryption," *Proceedings of Topics in Cryptology (CT-RSA)* 279-294.

Ateniese G, Fu K, Green M and Hohenberger S (2006). "Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage," *ACM Transactions on Information and System Security* **9**(1) 1-30.

Ateniese G, Fu K, Green M and Hohenberger S (2006). "ImprovedProxy Re-Encryption Schemes with Applications to SecureDistributed Storage," *ACM Transactions on Information and System Security* **9**(1) 1-30, 2006.

Blaze M, Bleumer G and Strauss M (1998). "Divertible Protocols and Atomic Proxy Cryptography," *Proceedings of the International Conference on Theory and Application of Cryptographic Techniques (EUROCRYPT)* 127-144.

Blaze M, Bleumer G and Strauss M (1998). "Divertible Protocols and Atomic Proxy Cryptography," *Proceedings of International Conference on Theory and Application of Cryptographic Techniques (EUROCRYPT)* 127-144.

Brownbridge DR, Marshall LF and Randell B (1982). "The Newcastle Connection or Unixes of the World Unite!," *Software Practice and Experience* **12**(12) 1147-1162.

Dimakis AG, Prabhakaran V and Ramchandran K (2005). "Ubiquitous Access to Distributed Data in LargeScale Sensor Networks through Decentralized Erasure Codes," *Proceedings of Fourth International Conference on Symposium Information Processing in Sensor Networks (IPSN)* 111- 117.

Dimakis AG, Prabhakaran V and Ramchandran K (2006). "Decentralized Erasure Codes for Distributed Networked Storage," *IEEE Transactions of Information Theory* **52**(6) 2809-2816.

Dimakis G, Prabhakaran V and Ramchandran K (2006). "Decentralized Erasure Codes for Distributed Networked Storage," *IEEE Transactions of Information Theory* **52**(6) 2809-2816.

Druschel P and Rowstron A (2001). "PAST: A Large-Scale, Persistent Peer-to-Peer Storage Utility," *Proceedings of Eighth Workshop on Hot Topics in Operating System (HotOS VIII)* 75-80.

Haeberlen A, Mislove A and Druschel P (2005). "Glacier: Highly Durable, Decentralized Storage Despite Massive Correlated Failures," *Proceedings of Second Symposium Networked Systems Design and Implementation (NSDI)* 143-158.

Kubiatowicz J, Bindel D, Chen Y, Eaton P, Geels D, Gummadi R, Rhea S, Weatherspoon H, Weimer W, Wells C and Zhao B (2000). "Oceanstore: An Architecture for Global-Scale Persistent Storage," *Proceedings of Ninth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)* 190- 201.

Lin HY and Tzeng WG (2010). "A Secure Decentralized Erasure Code for Distributed Network Storage," *IEEE Transactions of Parallel and Distributed Systems* **21**(11) 1586-1594.

Mambo M and Okamoto E (1997). "Proxy Cryptosystems: Delegation of the Power to Decrypt Ciphertexts," IEICE Trans. *Fundamentals of Electronics, Communications and Computer Sciences* **E80-A**(1) 54- 63.

Shao J and Cao Z (2009). "CCA-Secure Proxy Re-Encryption without Pairings," *Proceedings of 12th International Conference on Practice and Theory in Public Key Cryptography (PKC)* 357-376.

Tang Q (2008). "Type-Based Proxy Re-Encryption and Its Construction," *Proceedings of Ninth International Conference on Cryptology in India: Progress in Cryptology (INDOCRYPT)* 130-144.

Wilcox-O'Hearn Z and Warner B (2008). "Tahoe: The Least-Authority Filesystem," *Proceedings of Fourth ACM International Workshop on Storage Security and Survivability (StorageSS)* 21-26.