Volume 3, Issue 8, August 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Robust Cloud Storage System through Erasure Codes and Timestamp Based Cooperation

K.Bindu Madhavi, *G.Kalpana, Dr. R.V.Krishnaiah

Department of CSE & JNTUH

India

Abstract -Cloud computing is a new model of computing while enables people to store hug amount of data besides providing other services in pay per use fashion. The data is stored in multiple servers. However, storing data in a cloud server causes security concerns among the users of cloud as the servers are treated untrusted and accessed through public network such as Internet. For security reasons encryption techniques can be used but that causes overhead on the system and performance gets degraded besides having restrictions on data dynamics. For this reason developing a secure storage system for cloud that supports data dynamics and provide fool proof security is a challenging task. Many techniques came into existence. Some techniques are related to cryptography while others are related to auditing and data integrity. But still a more secure storage system is desired. In [1] a threshold proxy re-encryption scheme and decentralized erasure codes are used to secure data of the cloud users. It makes use of storage and security servers for storing data and keys respectively. Though it is efficient in secure storage and provides facilities like retrieval and forwarding there is communication and consistency concern across servers. This paper proposes a timestamp based solution that extends the techniques proposed in [1] in order to make it more secure and robust to inherent inconsistencies. The empirical results reveal that the proposed timestamp based improvement over threshold proxy re-encryption is robust.

Keywords:

I. Introduction

Due to technological innovations such as virtualization and cloud computing it is possible to avail state of the art services without making capital investment. Those services can be availed through Internet in Pay per Use fashion. This has led to dramatic changes in the way data is stored and processed. The cloud has become a reality and there are many service providers who are providing the basic cloud services such as Infrastructure as a Service (IaaS), Software as a Service (SaaS), and Platform as a Service (PaaS). The new model of computing has made the world to see different way in storing and processing huge amount of business data. Cloud has become a viable technology due to virtualization. However, to store data in cloud servers cause security concerns as they are maintained by third parties. Due to this problem, the cloud usage might bereduced or not increased as expected. This paper focuses on improving the cloud storage system proposed in [1] further to make it more secure and robust besides ensuring confidentiality and secure data dynamics. As robustness of data is paramount important aspect in cloud storage systems, many systems came into existence. Some of them are described here. One technique is to make a replica of data in multiple servers. This technique is sound as there are many servers and even one server is sufficient in order to get data. Another way is to use erasure coding where given message is encoded into code words. Even when some servers fail in storage, the data is safe and secure. However, it makes a tradeoff between the threshold of failure servers and storage size. Thus a decentralized erasure code is very useful in cloud storage systems.

When it comes to data confidentiality, it is a serious security concern over cloud storage. Cryptographic methods were used in securing data in such cases. However, cryptographic methods make some restrictions of data storage and data dynamics. There are some problems in this approach. First of all there is most computation and communication traffic required between user and storage servers. The second problem is that the management of cryptographic keys. In this case when key server is compromised, security is lost. And the last problem is that servers in this case can’t support other functions such as forwarding data to other users which has to be done by the data owner himself.

This paper is based on the work done in [1] which addressed the problem of storing data, retrieving data and forwarding data securely. For this it makes use of two kinds of servers namely storage servers and key servers. As the name implies, the storage servers are meant for storing data while the key servers are meant for storing keys related to security. And the servers are distributed in nature. Storing keys in a single server or device is not safe. For this reason it provides provisions to distributed keys to multiple key servers which are highly protected through security mechanisms. The threshold proxy re-encryptionscheme is integrated with secure distributed storage and key servers. The encryption scheme also supports operations on messages which have been encrypted besides forwarding operation.

This paper improves the secure storage system presented in [1] to make it more robust. It brings about perfect cooperation among the storage and key servers using time-stamp based integration. The operations such as data storage and data forwarding perfectly even in the presence of communication problems among servers with the help of timestamp based scheme. The rest of the paper is structured into some sections. Section II reviews literature on cloud storage security. Section III provides the overview of the proposed system including a brief introduction to the concepts of [1]. Section IV provides details of implementation and evaluation while section V concludes this paper.

II. Related Work

This section provides review of literature pertaining to distributed storage systems, proxy re-encryption schemes and integrity checking functionalities. With respect to distributed storage systems different file systems were proposed. For instance Network File System [2] and Network – Attached Storage [3] came into existence initially. These systems exhibit decentralized storage facility for good scalability. Many techniques came into existence for making the data storage robust including replica management. Later on many improvements on them came into existence with features such as security, efficiency, robustness, and scalability [4], [5].

There was lot of research into erasure codes for secure storage in a distributed environment especially. The findings in [6], [7], [8], [9] and [10] are summarized here. When data owner wants to store data, the message in encoded as a codeword. The code world is nothing but a vector of symbols and each server stores a symbol. The failure of any storage server is represented as an erasure error. While retrieving the data, each server combines the data linearly and the final message is returned. Though the security and retrieval probability is higher, the communication cost is also higher. However, data confidentiality is still not guaranteed when attacker compromises storage servers. This issue was addressed by Lin and Tzeng [11] by implementing a secure erasure code based solution in a decentralized fashion which makes use of key servers as well in addition to storage servers.

In [12] and [13] proxy re-encryption schemes were proposed. In this scheme a cipher text can be transferred by a proxy server with a public key to new user through re-encryption process. While making transformations, the server does not know the plain text. With respect to sharing function Ateniese et al. in [14] proposed another such scheme where messages are encrypted and stored in servers. When user is willing to share his data, he has to send a re-encryption key to server. The encrypted message gets re-encrypted by the storage server. Thus it ensures data confidentiality. Tang [15] proposed a type-based proxy re-encryption scheme which provides better control over the right bestowed on re-encryption key. This allows users to choose the kind of message before applying the scheme. Another kind of re-encryption scheme was proposed in [16] by name key-private proxy re-encryption scheme where the server can't find the identity of the recipient. However, a common aspect in all the schemes is that they are without the concept of pairing [17]. With regard to integrity checking it is essential in cloud storage. When data is stored, retrieved, forwarded and modified, the data integrity has to be verified. There are many researches in this area. For instance in [18], [19] and [20] the notion of proof of storage is used while in [21] and [22] provable data procession is used. However, all these techniques treat the messages in plaintext. The proposed system in this paper improves the scheme presented [1] further to focus on communication concerns among the cloud servers for robust data storage, retrieval and forwarding.

III. The Proposed System

The proposed cloud storage system is meant for secure data storage, retrieval and forwarding. The encryption, erasure codes and proxy re-encryption and partial decryption concepts are used here besides the timestamp based solution to overcome the communication concerns across the servers. The proposed system is an enhancement over [1] whose broad outline is presented in figure 1. It shows schematic representation consisting of storage servers and key servers. The storage servers are meant for storing data while the key servers are meant for storing security keys.

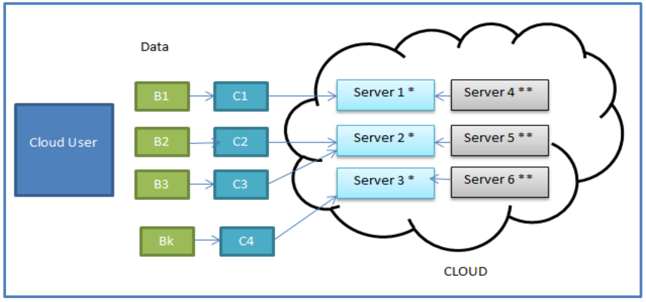

Fig. 1 – Schematic Representation of Proposed Storage System

As seen in fig. 1, it is evident that there are two types of servers. The servers who name ends with "*" are storage servers while the servers whose name ends with "**" are key servers. As the name implies, the storage servers are meant for storing data while the key servers are meant for storing security keys. Both the kinds of servers are distributed in nature in order to ensure confidentiality. The four phases of the proposed system are SETUP, data retrieval, data storage, data forwarding.

Setup

In this phase system parameters are set by system manager. Then a public-secret key pair is assigned to every user. User A sends his secret key to Key Server in such a way that the key is stored in key server and associated with that user.

Data Storage

In this phase data is saved to cloud by data owners or users. User A wants to save a file to cloud. Then he will break the file into multiple blocks and then each block is encrypted. The encrypted blocks are then sent to multiple storage servers. Once the cipher text is received, each storage server combines the cipher text linearly and converts that into a codeword symbols before being stored. The number of message blocks is known to storage servers in advance.

Data Forwarding

In this phase a user can forward his data to other user securely. User A encrypts data using his secret key and B's public keyin order to compute a re-encryption key. Then the re-encryption key is sent to all storage servers for further use. This key is used by storage servers to encrypt the code words stored in them. The encrypted data is sent to B. The re-encryption key in storage server helps it to process future forward requests from B. The code word symbol which has been re-encrypted is made up of all cipher texts encrypted using public key of B. Thus the codeword symbols are of two types now. They are original codeword which is intact and the re-encrypted codeword.

Data Retrieval

In this phase user A sends request to storage servers for data. The data which has been stored by the user or forwarded to him can be requested. Then user A also sends data retrieval request to key servers. After authenticating user, the key servers request storage servers for corresponding code words and does partial decryption on each and every codeword received. The decryption is done using the key share saved in key servers. Then the partially decrypted code words are combined in order to get original message. The user receives original content. More details on these four phases can be found in [1].

Fault Tolerance

When there is a failure of any storage server for any reason, a new storage server is added into the cloud and that will linearly combine all code words storage in all storage servers by requesting them. Afterwards, steps are taken in order to recover the failed server.

IV. Implementation and Evaluation

This section provides the implementation details. The basic implementation of setup, data storage, data forwarding and data retrieval are similar to [1]. However, the timestamp based cooperation among the servers is described here. Apart from the decentralized storage and security, the proposed system is making use of timestamp based coordination among servers in order to make the cloud storage system more robust. The two aspects considered for developing such scheme include fair request handling and system wide data consistency. Due to communication delays in servers, it may result in data inconstancies. This is overcome using the new timestamp based solution.

The time stamp based solution needs a global timestamp which is used by all storage and key servers for maintaining consistency. This is because the storage and retrieval operations involve multiple storage and key servers. To bring about robust cooperation among them, each operation is timestamped and the proposed scheme ensures that the individual operations are carried out in each server by monitoring the operations. When first server starts the storage operation, the distributed storage system has to ensure that the storage continues in all the servers as described in data storage section. The timestamp based monitoring helps in achieving this. The same is required while retrieving data as well. Data has to be collected from multiple servers.To ensure integrity and consistency the timestamp based retrieval process is used. The empirical results revealed that, the proposed system is more robust.

V. Conclusion

This paper presents a cloud storage security scheme based on timestamp based coordination which is an improvement over the scheme presented in [1] which proposed erasure codes and thresholdproxy re-encryption scheme for supporting secure storage with encoding, forwarding and decryption operations in a distributed fashion. The proposed scheme provides more robust cloud storage system. The storage servers and key servers are meant for storing data and security keys respectively. However, the communication concerns over the servers are addressed in this paper by using a timestamp based scheme which improves the threshold proxy re-encryption scheme. The results reveal that the proposed storage scheme for cloud is more robust.

References

[1] Hsiao-Ying Lin, Member, IEEE, and Wen-Guey Tzeng, Member, IEEE, "A Secure Erasure Code-Based Cloud Storage System with Secure Data Forwarding", IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 23, NO. 6, JUNE 2012.

[2] R. Sandberg, D. Goldberg, S. Kleiman, D. Walsh, and B. Lyon, "Design and Implementation of the Sun Network Filesystem," Proc. USENIX Assoc. Conf., 1985.

[3] D.R. Brownbridge, L.F. Marshall, and B. Randell, "The Newcastle Connection or Unixes of the World Unite!," Software Practice and Experience, vol. 12, no. 12, pp. 1147-1162, 1982.

[4] 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.

[5] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu, “Plutus: Scalable Secure File Sharing on Untrusted Storage,” Proc. Second USENIX Conf. File and Storage Technologies (FAST), pp. 29- 42, 2003.

[6] Z. Wilcox-O’Hearn and B. Warner, “Tahoe: The Least-Authority Filesystem,” Proc. Fourth ACM Int’l Workshop Storage Security and Survivability (StorageSS), pp. 21-26, 2008.

[7] S.C. Rhea, P.R. Eaton, D. Geels, H. Weatherspoon, B.Y. Zhao, and J. Kubiatowicz, “Pond: The Oceanstore Prototype,” Proc. Second USENIX Conf. File and Storage Technologies (FAST), pp. 1-14, 2003.

[8] 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.

[9] A.G. Dimakis, V. Prabhakaran, and K. Ramchandran, “Ubiquitous Access to Distributed Data in Large-Scale Sensor Networks through Decentralized Erasure Codes,” Proc. Fourth Int’l Symp. Information Processing in Sensor Networks (IPSN), pp. 111- 117, 2005.

[10] 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.

[11] H.-Y. Lin and W.-G. Tzeng, “A Secure Decentralized Erasure Code for Distributed Network Storage,” IEEE Trans. Parallel and Distributed Systems, vol. 21, no. 11, pp. 1586-1594, Nov. 2010.

[12] 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.

[13] M. Blaze, G. Bleumer, and M. Strauss, “Divertible Protocols and Atomic Proxy Cryptography,” Proc. Int’l Conf. Theory and Applicationof Cryptographic Techniques (EUROCRYPT), pp. 127-144, 1998.

[14] 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.

[15] 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.

[16] G. Ateniese, K. Benson, and S. Hohenberger, “Key-Private Proxy Re-Encryption,” Proc. Topics in Cryptology (CT-RSA), pp. 279-294, 2009.

[17] J. Shao and Z. Cao, “CCA-Secure Proxy Re-Encryption without Pairings,” Proc. 12th Int’l Conf. Practice and Theory in Public Key Cryptography (PKC), pp. 357-376, 2009.

[18] H. Shacham and B. Waters, “Compact Proofs of Retrievability,” Proc. 14th Int’l Conf. Theory and Application of Cryptology and Information Security (ASIACRYPT), pp. 90-107, 2008.

[19] G. Ateniese, S. Kamara, and J. Katz, “Proofs of Storage from Homomorphic Identification Protocols,” Proc. 15th Int’l Conf. Theory and Application of Cryptology and Information Security (ASIACRYPT), pp. 319-333, 2009.

[20] K.D. Bowers, A. Juels, and A. Oprea, “HAIL: A High-Availability and Integrity Layer for Cloud Storage,” Proc. 16th ACM Conf. Computer and Comm. Security (CCS), pp. 187-198, 2009.

[21] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, “Provable Data Possession at Untrusted Stores,” Proc. 14th ACM Conf. Computer and Comm. Security (CCS), pp. 598-609, 2007.

[22] G. Ateniese, R.D. Pietro, L.V. Mancini, and G. Tsudik, “Scalable and Efficient Provable Data Possession,” Proc. Fourth Int’l Conf. Security and Privacy in Comm. Netowrks (SecureComm), pp. 1-10, 2008.

AUTHORS

K.Bindu Madhavi is student of DRK Institute of Science and Technology, Hyderabad, AP, INDIA. She has received B.Tech in Biotechnology, M.Tech Degree in computer science and engineering. Her main research interest includes data mining and Cloud computing.

Kalpana Gudikandula is working as Associate Professor at DRK Institute of Science & Technology, Ranga Reddy, and Andhra Pradesh, India. She has received M.Tech Degree in Computer Science. His Main Interest includes Cloud Computing, Software Engineering.

Dr.R.V.Krishnaiah (Ph.D) is working as Principal at DRK INSTITUTE OF SCINCE & TECHNOLOGY, Hyderabad, AP, INDIA. He has received M.Tech Degree EIE and CSE. His main research interest includes Data Mining, Software Engineering.