Volume 4, Issue 11, November 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

A Secure Data Forwarding Approach for Cloud Storage System

Sagar B*, Dhanraj, Deepu S R
Computer Science and Engineering
India

Abstract- 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 [2]. General encryption schemes protect data confidentiality, but also limit the functionality of the storage system because a few operations are supported over encrypted data. 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 [5]. Our method fully integrates encrypting, encoding, and forwarding. We analyses 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.

Keywords— cryptography, Encryption, Decryption, Network Attached Storage (NAS), Proxy-reencryption.

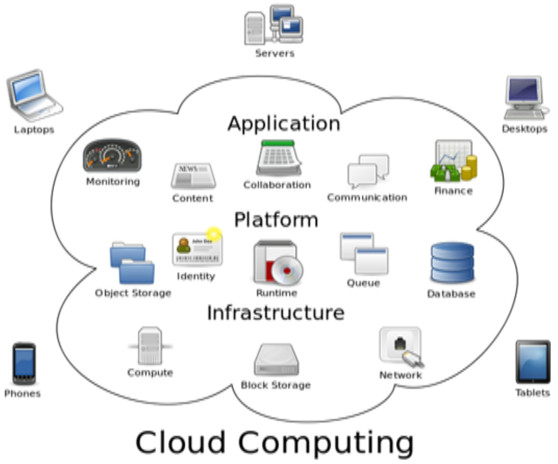
I. Introduction

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 [2]. 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. A decentralized erasure code is suitable for use in a distributed storage system [1]. 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. 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 code word symbols from storage servers, decode them, and then decrypt them by using cryptographic keys [5]. 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. 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 [3]. 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 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. A decentralized erasure code is suitable for use in a distributed storage system [1]. 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 [3]. 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 [7].

II. RELATED WORKS

Cloud storage is a model of networked enterprise storage where data is stored not only in the user's computer, but in virtualized pools of storage which are generally hosted by third parties it incorporate with three technologies to help with user authentication and storage management problems: Distributed Storage Systems, Proxy Re-Encryption Schemes, and Integrity Checking Functionality

The following figure shows the typical workflow in a Cloud Computing:



Distributed Storage System

At the early years, the Network-Attached Storage (NAS) and the Network File System (NFS) provide extra storage devices over the network such that a user can access the storage devices via network connection. Afterward, many improvements on scalability, robustness, efficiency, and security were proposed. A decentralized architecture for storage systems offers good scalability, because a storage server can join or leave without control of a central authority. To provide robustness against server failures, a simple method is to make replicas of each message and store them in different servers. However, this method is expensive as z replicas result in z times of expansion.

Erasure-code

In information theory, an erasure code is a forward error correction (FEC) code for the binary erasure channel, which transforms a message of *k* symbols into a longer message (code word) with *n* symbols such that the original message can be recovered from a subset of the *n* symbols. A decentralized erasure code is an erasure code that independently computes each code word symbol for a message. Thus, the encoding process for a message can be split into n parallel tasks of generating code word 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 code- word symbol for the received message symbols and stores.

Proxy Re-Encryption Schemes

Proxy re-encryption schemes are proposed by Mambo and Okamoto and Blaze et al. In a proxy re-encryption scheme, a proxy server can transfer a cipher text under a public key to a new one under another public key by using the re-encryption key [3]. The server does not know the plaintext during transformation. Ateniese et al. proposed some proxy re-encryption schemes and applied them to the sharing function of secure storage systems. In their work, 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. Our work further integrates encryption, re-encryption, and encoding such that storage robustness is strengthened. Type-based proxy re-encryption schemes proposed by Tang provide a better granularity on the granted right of a re-encryption key. A user can decide which type of messages and with whom he wants to share in this kind of proxy re-encryption schemes. Key-private proxy re-encryption schemes are proposed by Ateniese et al. In a key-private proxy re-encryption scheme, given a re-encryption key, a proxy server cannot determine the identity of the recipient. This kind of proxy re-encryption schemes provides higher privacy guarantee against proxy servers. Although most proxy re-encryption schemes use pairing operations, there exist proxy re-encryption schemes without pairing.

Integrity Checking Functionality

Another important functionality about cloud storage is the function of integrity checking. After a user stores data into the storage system, he no longer possesses the data at hand. The user may want to check whether the data are properly stored in storage servers. The concept of provable data possession and the notion of proof of storage are proposed. Later, public auditability of stored data is addressed in. Nevertheless all of them consider the messages in the clear text form.

Cryptography

Cryptography is the practice and study of techniques for secure communication in the presence of third parties (called adversaries). More generally, it is about constructing and analysing protocols that overcome the influence of adversaries and which are related to various aspect in information security such as data confidentiality data integrity authentication and non-repudiation. Modern cryptography is heavily based on mathematical theory and computer science practice; cryptographic algorithms are designed around computational hardness assumptions, making such algorithms hard to break in practice by any adversary. It is theoretically possible to break such a system but it is infeasible to do so by any known practical means. These schemes are therefore termed computationally secure.

III. PROPOSED SYSTEM

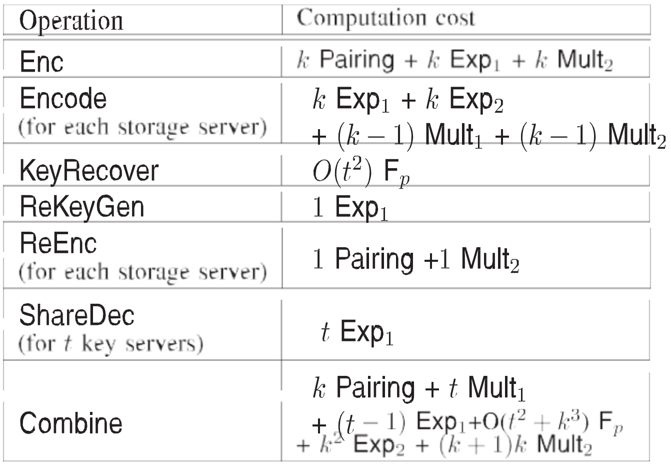
We address the problem of forwarding data to another user by storage servers directly under the command of the data owner. We consider the system model that consists of distributed storage servers and key servers. Since storing cryptographic keys in a single device is risky, a user distributes his cryptographic key to key servers that shall perform cryptographic functions on behalf of the user. To well fit the distributed structure of systems, we require that servers independently perform all operations. With this consideration, we propose a new 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.

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 independently perform partial decryption process.
- More flexible adjustment between the number of storage servers and robustness.

IV. RESULT ANALYSIS

Table 1 The Computation Cost of Each Algorithm in Secure Cloud Storage



Operation	Computation cost
Enc	k Pairing + k Exp$_1$ + k Mult$_2$
Encode (for each storage server)	k Exp$_1$ + k Exp$_2$ + $(k-1)$ Mult$_1$ + $(k-1)$ Mult$_2$
KeyRecover	$O(t^2)$ F$_p$
ReKeyGen	1 Exp$_1$
ReEnc (for each storage server)	1 Pairing +1 Mult$_2$
ShareDec (for t key servers)	t Exp$_1$
Combine	k Pairing + t Mult$_1$ + $(t-1)$ Exp$_1$+O(t^2+k^3) F$_p$ + k^2 Exp$_2$ + $(k+1)k$ Mult$_2$

Pairing: a pairing computation of $\tilde{e}$.

- Exp$_1$ and Exp$_2$: a modular exponentiation computation in $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively.
- Mult$_1$ and Mult$_2$: a modular multiplication computation in $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively.
- F$_p$: an arithmetic operation in $GF(p)$.

V. CONCLUSION

A cloud storage system consists of storage servers and key servers. Proxy re-encryption scheme supports encoding, forwarding, and partial decryption operations in a distributed way. To decrypt a message of k blocks that are encrypted and encoded to n code word symbols, each key server only has to partially decrypt two code word symbols in our system. By using the threshold proxy re-encryption scheme, we present a secure cloud storage system that provides secure data storage and secure data forwarding functionality in a decentralized structure. Moreover, each storage server independently performs encoding and re-encryption and each key server independently performs partial decryption

Our Key server performs the main role in our distributed storage system. This Key server performs the important role key management. But our proposed system doesn't provide any security over this Key Server. The attacker or intruder can attack the key server to get the secret key because there is no security provided to the Secret Key. So as a future work we focus on key server for giving more secure to our storage system.

REFERENCES

[1] Hsiao-Ying Lin and Wen-Guey Tzeng, "A Secure Erasure Code-Based Cloud Storage System with Secure Data Forwarding", IEEE Transactionson Parallel and Distributed Systems, vol. 23, no. 6, pp. 995-1003, June 2012.

[2] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing," Proc. IEEE 29th Int'l Conf. Computer Comm. (INFOCOM), pp. 525-533, 2010

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

[4] 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

[5] 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.

[6] 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.

[7] 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.

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

[9] Herbert Schildt, JAVA- Complete Reference, 7^{th} Edition, McGraw-Hill, 2006.