Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 4, April 2015, pg.406 – 412

RESEARCH ARTICLE

A Secure Storage based on Rateless Erasure Code and Order-Preserving Encryption in Cloud Computing

J.Shyamala[1], A.Annie Jesus Suganthi Rani[2], M.Antony Vijaya[3]

[1]Computer Science and Engineering & P.B. Engineering College, India
[2]Computer Science and Engineering & Dr.Sivanthi Aditanar College of Engineering, India
[3]Computer Science and Engineering & Dr.Sivanthi Aditanar College of Engineering, India

---

Abstract-- Cloud computing enables highly scalable services to be easily consumed over the Internet on an as-needed basis. Cloud storage is a model of networked online storage where data is stored in virtualized pools of storage which are generally hosted by third parties. The first offered cloud service is moving data into the cloud: data owners let cloud service providers host their data on cloud servers and data consumers can access the data from the cloud servers. Storing data in a third party's cloud system causes serious concern on data confidentiality. In this paper, to ensure the correctness of users' data in the cloud, we propose Raptor coded data for the integration of storage correctness insurance and data error localization, i.e., the identification of misbehaving server(s). In order to provide strong confidentiality for messages in storage servers, a user can encrypt messages by a cryptographic method before applying an order-preserving encryption method to encode and store messages. Our system is highly distributed where storage servers independently encode and forward messages and key servers independently perform partial decryption. We analyze suitable parameter for more flexible adjustment between the number of storage servers and robustness. Our result shows that, our proposed model provides a secure storage for data in cloud.

Keywords: Rateless erasure correcting code, Order-preserving Encryption, distributed storage, error localization

---

I INTRODUCTION

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction. Several trends are opening up the era of Cloud Computing, which is an Internet-based development and use of computer technology. The ever cheaper and more powerful processors together with the software as a service (SaaS) computing architecture are transforming data centres into pools of

computing service on a huge scale. The increasing network bandwidth and reliable yet flexible network connections make it even possible that users can now subscribe high quality services from data and software that reside solely on remote data centres. 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. In this, focus on designing a cloud storage system for confidentiality. Data robustness is a major requirement for storage systems. One way to provide data robustness is to replicate a message [4]. Another way is to encode a message of k-symbols into a codeword of n-symbols by erasure coding. A decentralized erasure code is an erasure code that independently computes each codeword symbol for a message [5]. 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. Cloud computing inevitably poses new challenging security threats for number of reasons. Firstly, traditional cryptographic primitives for the purpose of data security protection cannot be directly adopted due to the users' loss control of data under Cloud Computing. Therefore, verification of correct data storage in the cloud must be conducted without explicit knowledge of the whole data. Secondly, Cloud Computing is not just a third party data warehouse.

The data stored in the cloud may be frequently updated by the users, including insertion, deletion, modification, appending, reordering, etc. To ensure storage correctness under dynamic data update is hence of paramount importance. However, this dynamic feature also makes traditional integrity insurance techniques futile and entails new solutions. Last but not the least, the deployment of Cloud Computing is powered by data centres running in a simultaneous, cooperated and distributed manner. Individual user's data is redundantly stored in multiple physical locations to further reduce the data integrity threats. Therefore, distributed protocols for storage correctness assurance will be of most importance in achieving a robust and secure cloud data storage system in the real world

In this paper, we propose an effective and flexible distributed scheme with an explicit dynamic data support to ensure the correctness of users' data in the cloud. We rely on rateless erasure correcting code (Raptor code) in the file distribution preparation to provide redundancies and guarantee the data dependability. This construction drastically reduces the communication and storage overhead as compared to the traditional replication-based file distribution techniques. By utilizing the homomorphic token with distributed verification of rateless erasure-coded data, our scheme achieves the storage correctness insurance as well as data error localization: Whenever data corruption has been detected during the storage correctness verification, our scheme can almost guarantee the simultaneous localization of data errors, i.e., the identification of the misbehaving server(s).

II PROBLEM STATEMENT

A. System model

Representative network architecture for cloud data storage is illustrated in Fig 1. Three different network entities can be identified as follows:

User: users, who have data to be stored in the cloud and rely on the cloud for data computation, consist of both individual consumers and organizations.

Cloud Service Provider (CSP): a CSP, who has significant resources and expertise in building and managing distributed cloud storage servers, owns and operates live Cloud Computing systems.

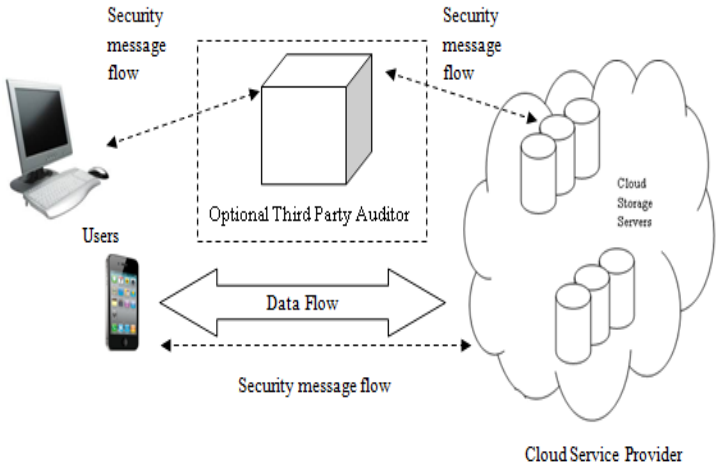

Fig.1 Cloud Storage Architecture

Third Party Auditor (TPA): an optional TPA, who has expertise and capabilities that users may not have, is trusted to assess and expose risk of cloud storage services on behalf of the users upon request.

In cloud data storage, a user stores his data through a CSP into a set of cloud servers, which are running in a simultaneous, cooperated and distributed manner. Data redundancy can be employed with technique of rateless erasure correcting code to further tolerate faults or server crash as user's data grows in size and importance. Thereafter, for application purposes, the user interacts with the cloud servers via CSP to access or retrieve his data.

As users no longer possess their data locally, it is of critical importance to assure users that their data are being correctly stored and maintained. That is, users should be equipped with security means so that they can make continuous correctness assurance of their stored data even without the existence of local copies. In case those users do not necessarily have the time, feasibility or resources to monitor their data, they can delegate the tasks to an optional trusted TPA of their respective choices.

III ENSURING CLOUD DATA STORAGE

In cloud data storage system, users store their data in the cloud and no longer possess the data locally. Thus, the correctness and availability of the data files being stored on the distributed cloud servers must be guaranteed. One of the key issues is to effectively detect any unauthorized data modification and corruption, possibly due to server compromise and/or random Byzantine failures. Besides, in the distributed case when such inconsistencies are successfully detected, to find which server the data error lies in is also of great significance, since it can be the first step to fast recover the storage errors.

To address these problems, our main scheme for ensuring cloud data storage is presented in this section. Order-preserving encryption method is briefly explained below. The first part of the section is file distribution across cloud servers. Then, the homomorphic token is introduced. The token computation function we are considering belongs to a family of universal hash function [12], chosen to preserve the homomorphic properties, which can be perfectly integrated with the verification of rateless erasure-coded data Subsequently, it is also shown how to derive a challenge response protocol for verifying the storage correctness as well as identifying misbehaving servers. Finally, the procedure for file retrieval and error recovery based on rateless erasure correcting code is outlined.

Order-preserving encryption

OPE is defined as a deterministic encryption scheme over the numerical values with the characteristic that, if the plain- texts x_1 and x_2 satisfy $x_1 < x_2$, then $E_K(x_1) < E_K(x_2)$, where $E_K(\cdot)$ denotes the OPE function with key K, is guaranteed. A simple OPE was presented[2]; Given that y numbers, $x_1 < \cdots < x_y$, are the possible plaintexts, the simplest OPE is to generate an array of y uniformly random numbers

$k_1 < \cdots < k_y$ as the key K. The encryption (decryption) is accomplished by searching in the key K for the cipher text (plaintext) corresponding to the plaintext (ciphertext); E .g. $E_K(x_i) = k_i,\ 1 \le i \le y$. A distinguishing feature of OPE is that its key also acts as possible ciphertexts. The above encryption reveals nothing but the numerical order of plaintexts because all the possible ciphertexts $k_1, \ldots, k_y$ are distributed uniformly over a specific range. Despite the leakage of numerical order, OPE is in fact provably secure[2] Albeit the above OPE scheme has the drawback of large key size, we keep such a simple form of OPE in mind for the ease of presentation throughout this paper. We particularly note that the keyed hash functions used in this paper are keyed-hash message authentication code (HMAC). Consider two parties sharing a secret key k. If the message m to be communicated is associated with $HMAC_k(m)$, the use of HMAC naturally guarantees the data authenticity and integrity. 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. Our system meets the requirements that storage servers independently perform encoding and re-encryption and key servers independently perform partial decryption.

A. File distribution preparation

It is well known that rateless erasure-correcting code (raptor code) may be used to tolerate multiple failures in distributed storage systems. In cloud data storage, we rely on this technique to disperse the data file F redundantly across a set of n= m+k distributed servers. Raptor codes pre-encode the source symbols using a fixed length block code, and then encode these new symbols with an LT code. The main advantage is that, for correct decoding.

B. Challenge token pre-computation

In order to achieve assurance of data storage correctness and data error localization simultaneously, our scheme entirely relies on the pre-computed verification tokens. The main idea is as follows: before file distribution the user pre-computes a certain number of short verification tokens on individual vector G(j) (j ∈ {1, . . . , n}), each token covering a random subset of data blocks. Upon receiving challenge, each cloud server computes a short "signature" over the specified blocks and returns them to the user. The values of these signatures should match the corresponding tokens pre-computed by the user. In our case here, the user stores them locally to obviate the need for encryption and lower the bandwidth overhead during dynamic data operation.

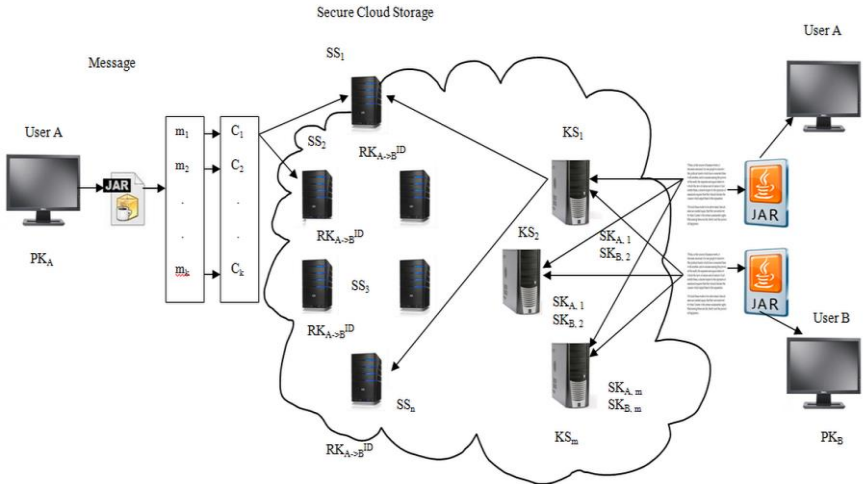

Fig. 2 System Overview

C. Correctness verification and error localization

Error localization is a key prerequisite for eliminating errors in storage systems. However, many previous schemes do not explicitly consider the problem of data error localization, thus only provide binary results for the storage verification. Our scheme outperforms those by integrating the correctness verification and error localization in our challenge-response protocol: the response values from servers for each challenge not only determine the correctness of the distributed storage, but also contain information to locate potential data error(s).

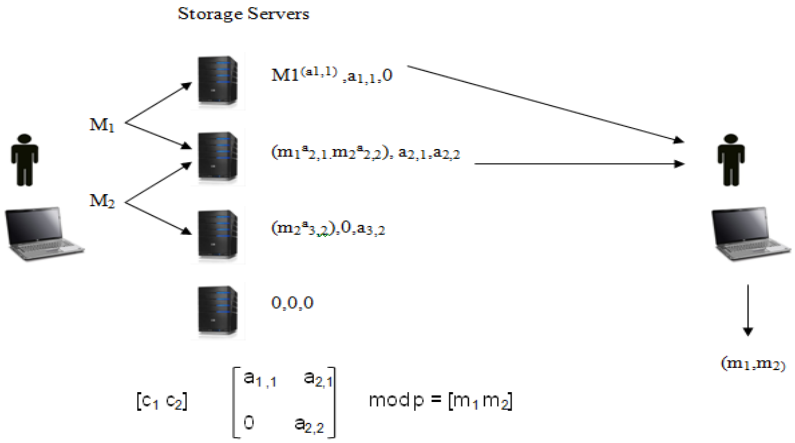

Fig. 3 Erasure code based matrix

D. File retrieval and error recovery

Since our layout of file matrix is systematic, the user can reconstruct the original file by downloading the data vectors from the first m servers, assuming that they return the correct response values. Notice that our verification scheme is based on random spot-checking, so the storage correctness assurance is a probabilistic one. However, by choosing system parameters (e.g., r,l,t) appropriately and conducting enough times of verification, we can guarantee the successful file retrieval with high probability. On the other hand, whenever the data corruption is detected, the comparison of pre-computed tokens and received response values can guarantee the identification of misbehaving server(s) with high probability.

IV DYNAMIC DATA OPERATION SUPPORT

In this section, we will show how our scheme can explicitly and efficiently handle dynamic data operations such as update, delete, append and insert operations for cloud data storage for verification token construction.

A. Update operations

Due to the linear property of Reed-Solomon code, a user can perform the update operation and generate the updated parity blocks by using Δfij only, without involving any other unchanged blocks. The data update operation inevitably affects some or all of the remaining verification tokens, after preparation of update information to maintain the same storage correctness assurance.

B. Delete operations

Sometimes, after being stored in the cloud, certain data blocks may need to be deleted. The delete operation we are considering is a general one, in which user replaces the data block with zero or some special reserved data symbol.

C. Append operations

In some cases, the user may want to increase the size of his stored data by adding blocks at the end of the data file, which we refer as data append. We anticipate that the most frequent append operation in cloud data storage is bulk append, in which the user needs to upload a large number of blocks (not a single block) at one time.

D. Insert operations

An insert operation to the data file refers to an append operation at the desired index position while maintaining the same data block structure for the whole data file, i.e., inserting a block corresponds to shifting all blocks starting with index by one slot. Thus, an insert operation may affect many rows in the logical data file matrix, and a substantial number of computations are required to renumber all the subsequent blocks as well as recomputed the challenge response tokens [6].

V CONCLUSION

Cloud computing is a new paradigm in which users can store their data. However, security and privacy issues impose strong barrier for users' adoption of Cloud systems and Cloud services. In this paper, we investigated the problem of data security in cloud data storage, which is essentially a distributed storage system. To ensure the correctness of users' data in cloud data storage, we proposed an effective and flexible distributed scheme with explicit dynamic data support. We rely on rateless erasure correcting code in the file distribution preparation to provide redundancy parity vectors and guarantee the data dependability. The Order-preserving encryption(OPE) scheme supports encoding, forwarding, and partial decryption operations in a distributed way. We particularly note that the keyed hash functions used in this paper are keyed-hash message authentication code (HMAC) for sharing a secret key k. If the message m to be communicated is associated with $HMAC_k(m)$, the use of HMAC naturally guarantees the data authenticity and integrity. By utilizing the homomorphic token with distributed verification of rateless erasure coded data, our scheme achieves the integration of storage correctness insurance and data error localization, i.e., whenever data corruption has been detected during the storage correctness verification across the distributed servers, we can almost guarantee the simultaneous identification of the misbehaving server(s).

ACKNOWLEDGEMENTS

The authors would like to thank THE ALMIGHTY for showering his blessings throughout their life. They thank all the anonymous reviewers for their valuable feedback.

REFERENCES

[1] Amazon.com, "Amazon Web Services (AWS)," Online at http://aws. amazon.com, 2008.

[2] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Order preserv- ing encryption for numeric data," in *Proc. ACM SIGMOD*, 2004, pp. 63–574.

[3] Amin Shokrollahi, "Raptor Codes", IEEE Transactions on Information Theory, Vol. 52, No. 6, June 2006.

[4] 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.

[5] 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.

[6] A.Juels and J. Burton S. Kaliski, "PORs: Proofs of Retrievability for Large Files," Proc. of CCS '07, pp. 584–597, 2007.

[7] H. Shacham and B. Waters, "Compact Proofs of Retrievability," Proc. of Asiacrypt '08, Dec. 2008.

[8] Cong Wang, Qian Wang, Kui Ren, Ning Cao, and Wenjing Lou ,"Toward Secure and Dependable Storage Services in Cloud Computing", IEEE Transactions On Services Computing, Vol. 5, No. 2, AprilJune 2012.

[9] Hsiao-Ying Lin, and Wen-Guey Tzeng,, "A Secure Erasure Code-Based Cloud Storage System with Secure Data Forwarding," IEEE Trans. Parallel andDistributed Systems, vol. 23, no. 06, June. 2012

[10] T. S. J. Schwarz and E. L. Miller, "Store,Forget, and Check: Using Algebraic Signatures to Check Remotely Administered Storage," Proc.of ICDCS '06, pp. 12–12, 2006.

[11] M. Lillibridge, S. Elnikety, A. Birrell, M. Burrows, and M. Isard, "A Cooperative Internet Backup Scheme," Proc. of the 2003 USENIX Annual Technical Conference (General Track), pp. 29–41, 2003.

[12] C. Wang, Q. Wang, K. Ren, and W. Lou,"Ensuring Data Storage Security in Cloud Computing," Proc. 17th Int'l Workshop Quality of Service (IWQoS '09), pp. 1-9, July 2009.

