© 2016 IJSRSET | Volume 2 | Issue 2 | Print ISSN : 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

Efficient Approach for Data Retrievability on Cloud Storage Systems

Priya A, Meena T, Devi M

Department of Computer Science and Engineering, Dhanalakshmi College of Engineering, Chennai, Tamilnadu, India

ABSTRACT

Cloud storage is a model of data storage in which the digital data is stored in logical pools. It allows users to store their data in a remote server to get rid of expensive local storage and management costs and then access data of interest anytime anywhere. We propose an enhanced dynamic proof of retrievability scheme supporting public audit ability and communication-efficient recovery from data corruptions. We split up the data into small data blocks and encode that data block using network coding. To eliminate the communication overhead for small data corruptions within a server, each data block is further encoded. Based on the encoded data blocks, we utilize tree structure to enforce the data sequence for dynamic operations, preventing the cloud service provider from manipulating data block to pass the integrity check in the dynamic scenario. We also analyze for the effectiveness of the proposed construction in defending against attacks during data retrievability.

Keywords: Data Retrievability, Providing Security, Creating Blocks, Network Encoding

I. INTRODUCTION

Cloud refers to a distinct IT environment that is designed for the purpose of remote provisioning scalable and measured IT resources. The term originated as a metaphor for the Internet which is, in essence, a network of networks providing remote access to a set of decentralized IT resources. Prior to cloud computing becoming its own formalized IT industry segment, the symbol of a cloud was commonly used to represent the Internet in a variety of specifications and mainstream documentation of Web-based architectures. This same symbol is now used to specifically represent the boundary of a cloud environment. It is important to distinguish the term "cloud" and the cloud symbol from the Internet. As a specific environment used to remotely provision IT resources, a cloud has a finite boundary. There are many individual clouds that are accessible via the Internet. Whereas the Internet provides open access to many Web-based IT resources, a cloud is typically privately owned and offers access to IT resources that is metered. Many approach has proposed implementing design diversity techniques to increase the reliability, availability and security of large-scale systems. However, none of them have explicitly linked the distribution of resources to risk and correlation between different candidate providers. The challenge would be to find an efficient and effective solution for investing in diversity while considering the risk and correlation between providers. Moreover, the dynamic nature of the cloud motivates the need for adaptation in that solution. Much of the Internet is dedicated to the access of content-based IT resources published via the World Wide Web. IT resources provided by cloud environments, on the other hand, are dedicated to supplying back-end processing capabilities and user-based access to these capabilities. Another key distinction is that it is not necessary for clouds to be Web-based even if they are commonly based on Internet protocols and technologies. Protocols refer to standards and methods that allow computers to communicate with each other in a pre-defined and structured manner. A

cloud can be based on the use of any protocols that allow for the remote access to its IT resources.

II. METHODS AND MATERIAL

A. System Architecture

In this section we describe about the process among the user, auditor and the admin,

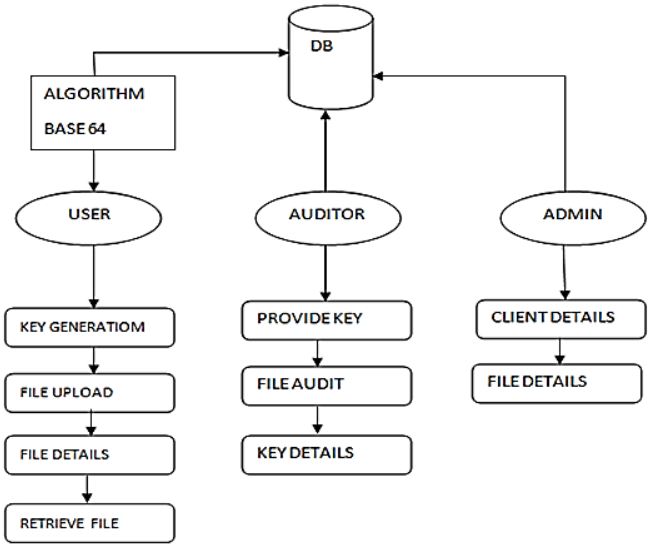

Figure 1. Architecture of the cloud process

B. A. System Description

Fig 1 shows the design of the cloud. As indicated this will undergo the following steps

1) Data integrity

Data integrity refers to maintaining and assuring the accuracy and consistency of data over its entire life-cycle, and is a critical aspect to the design, implementation and usage of any system which stores, processes, or retrieves data. Data integrity is the opposite of data corruption, which is a form of data loss.

2) Data availability

Data availability is a term used by some computer storage manufacturers and Storage Service Providers (SSPs) to describe products and services that ensure that data continues to be available at a required level of performance in situations ranging from normal through "disastrous." In general, data availability is achieved through redundancy involving where the data is stored and how it can be reached.

3) Public Audit

The goal of Cloud Audit is to provide cloud service providers with a way to make their performance and security data readily available for potential customers. The specification provides a standard way to present and share detailed, automated statistics about performance and security.

4) Data Dynamic

Dynamic data or transactional data denotes information that is asynchronously changed as further updates to the information become available. The opposite of this is persistent data, which is data that is infrequently accessed and not likely to be modified.

5) Third Party Auditor

An entity, which has expertise and capabilities that clients do not have, is trusted to assess and expose risk of cloud storage services on behalf of the clients upon request. In the cloud paradigm, by putting the large data files on the remote servers, the clients can be relieved of the burden of storage and computation.

C. Related Works

Remote data integrity checks for public cloud storage have been investigated in various systems and security models. Considering the large size of the outsourced data and the owner's constrained resource capability, the cost to audit data integrity in the cloud environment could be formidable and expensive to the data owner. Therefore, it is preferable to allow an independent expertise-equipped TPA to check the data integrity on behalf of the data owners. Ateniese was the first to introduce the "Provable Data Possession (PDP)" model and proposed an integrity verification scheme for static data using RSA based homomorphic authenticators. At the same time, Juels et al proposed the "Proof of Retrievability (PoR)" model which is stronger than the PDP model in the sense that the system additionally guarantees the retrievability of outsourced data. Specifically, the authors proposed a spot-checking approach to guarantee possession of data files and

employed error-correcting coding technologies to ensure the retrievability. A limitation of their scheme is that the number of challenges is constrained. Shacham et al.utilized the homomorphic signatures in to design an improved PoR scheme. Although the scheme supported public auditability of static data using publicly verifiable homomorphic authenticators, how to perform data recovery was not explicitly discussed. To achieve strong data retrievability, Bowers proposed a data coding structure achieving the within-server redundancy and cross-server redundancy. Constructed their remote data checking schemes based on network coding which can save the communication cost of data recovery compared with erasure codes. In particular considered the cross-server redundancy as in a multiple server setting, where the cross-server coding was done using network coding instead of erasure codes in. Designed a secure cloud storage system using LT codes.

III. RESULTS AND DISCUSSION

A. Proposed System

We propose an enhanced dynamic proof of retrievability scheme supporting public audit ability and communication-efficient recovery from data corruptions. To this end, we split up the data into small data blocks and encode each data block individually using network coding. Network coding and erasure codes are adopted to encode data blocks to achieve within server and cross server data redundancy, tolerating data corruption. By combing range based 2-3 tree and improved version of aggregately signature based broadcast encryption, our construction can support efficient data dynamics while defending against data replay attack.

B. Algorithm

1) BASE-64

Base64 is a generic term for a number of similar encoding schemes that encode binary data by treating it numerically and translating it into a base 64 representation. The Base64 term originates from a specific MIME content transfer encoding.Base64 encoding schemes are commonly used when there is a need to encode binary data that needs be stored and transferred over media that are designed to deal with textual data. This is to ensure that the data remains intact without modification during transport. Base64 is used commonly in a number of applications including email via MIME, and storing complex data in XML. Example- A quote snippet from Thomas Hobbes's Leviathan:"*Man is distinguished, not only by his reason, but* ..."represented as an ASCII byte sequence is encoded in MIME's Base64 scheme as follows: TWFuIGlzIGRpc3Rpbmd1aXNoZWQsIG5vdCBvbmx5IGJ5IGhpcyByZWFzb24sIGJ1dCAuLi4=

In the above quote the encoded value of *Man* is *TWFu*. Encoded in ASCII, *M*, *a*, *n* are stored as the bytes 77, 97, 110, which are 01001101, 01100001, 01101110 in base 2. These three bytes are joined together in a 24 bit buffer producing 010011010110000101101110. Packs of 6 bits (6 bits have a maximum of 64 different binary values) are converted into 4 numbers (24 = 4 * 6 bits) which are then converted to their corresponding values in Base64.

Text content	M	a	n
ASCII	77	97	110
Bit pattern	0 1 0 0 1 1 0 1	0 1 1 0 0 0 0 1	0 1 1 0 1 1 1 0
Index	19	22	5	46

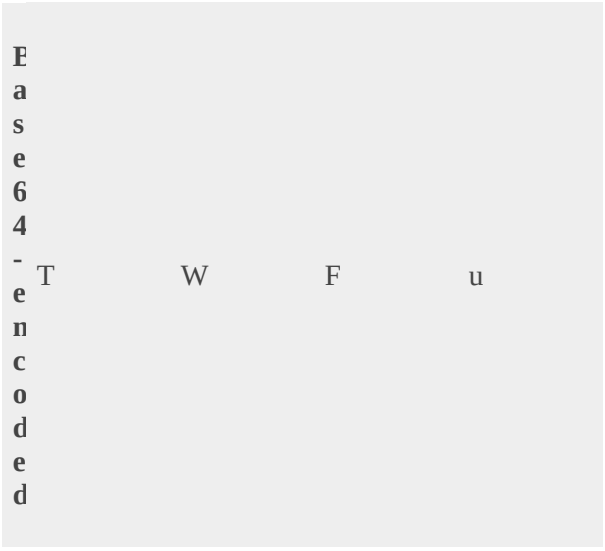

Figure 2. Encoded values using base-64

As Fig 2 illustrates, Base64 encoding converts 3 uncoded bytes (in this case, ASCII characters) into 4 encoded ASCII characters.

2) Network Encoding

Random linear network coding is a technique which can be used to improve a network's throughput, efficiency and scalability, as well as resilience to attacks and eavesdropping. Instead of simply relaying the packets of information they receive, the nodes of a network take several packets and combine them together for transmission. This can be used to attain the maximum possible information flow in a network.

It has been proven that linear coding is enough to achieve the upper bound in multicast problems with one or more sources. However linear coding is not sufficient in general (e.g. multisource, multilink with arbitrary demands), even for more general versions of linearity such as convolution theory and filter-bank coding. Finding optimal coding solutions for general network problems with arbitrary demands remains an open problem.

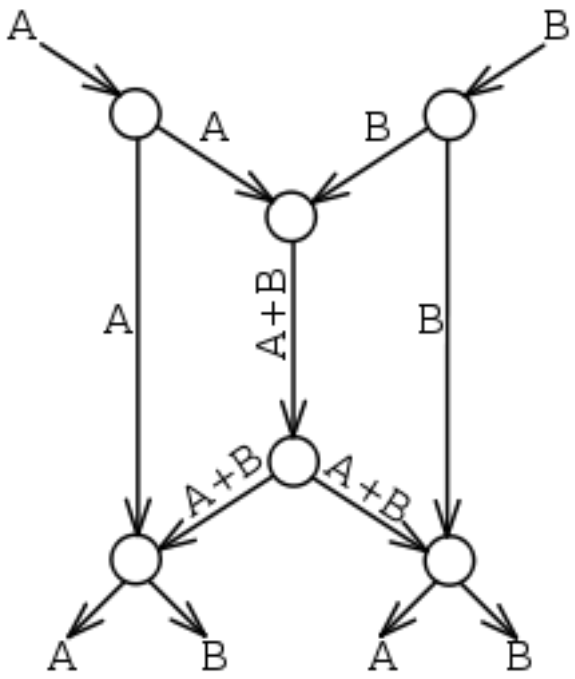

Figure 3. Butterfly Network

The butterfly network is often used to illustrate how linear network coding can outperform routing. Two source nodes (at the top of the picture) have information A and B that must be transmitted to the two destination nodes (at the bottom), which each want to know both A and B. Each edge can carry only a single value (we can think of an edge transmitting a bit in each time slot).If only routing were allowed, then the central link would be only able to carry A or B, but not both. Suppose we send A through the center; then the left destination would receive A twice and not know B at all. Sending B poses a similar problem for the right destination. We say that routing is insufficient because no routing scheme can transmit both A and B simultaneously to both destinations.

Using a simple code, as shown, A and B can be transmitted to both destinations simultaneously by sending the sum of the symbols through the center – in other words, we encode A and B using the formula "A+B". The left destination receives A and A + B, and can calculate B by subtracting the two values. Similarly, the right destination will receive B and A + B, and will also be able to determine both A and B.A similar concept has been used to encode stereophonic sound, where there is a "left" signal and a "right" signal. The two analog signals are "added" together, and the "sum" is subsequently used to recover the original signals.

IV. CONCLUSION

We have done this in order to improve data reliability and availability. Our inter coding and outer coding of outsourced data enables efficient recovery when data corruption occurs. Using trusted Third Party Auditor (TPA) for data audit report and data audit delegation. Reduce server hacks or Byzantine failure to maintain reputation. There is possible increase in security by sending key to data owner to upload and retrive files. When one server is corrupted, the original data can be recovered by simply copying the entire data from one of the healthy servers.

V. REFERENCES

[1] P. Mell and T. Grance, "Draft NIST working definition of cloud compu-ting," Referenced on June. 3rd, 2009 Online at http://csrc.nist.gov/groups/SNS/cloud-computing/index. html, 2009.

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoi-ca, and M. Zaharia, "Above the clouds: A berkeley view of cloud computing," University of California, Berkeley, Tech. Rep. UCB-EECS-2009-28, Feb 2009.

[3] M. Arrington, "Gmail disaster: Reports of mass email deletions," Online at http://www.techcrunch.com/2006/ 12/28/gmail-disasterreports- of- mass- email- deletions/, December 2006.

[4] J. Kincaid, "MediaMax/TheLinkup Closes Its Doors," Online at http://www.techcrunch.com/2008/07/10/ mediamaxthelinkup- closes- its- doors/, July 2008.

[5] A. L. Ferrara, M. Greeny, S. Hohen-berger, and M. Pedersen, "Practical short signature batch verifica-tion," in Proceedings of CT-RSA, volume 5473 of LNCS. Springer-Verlag, 2009, pp. 309– 324.

[6] M. A. Shah, R. Swaminathan, and M. Baker, "Privacy- preserving audit and extraction of digital contents," Cryptol- ogy ePrint Archive,Report 2008.

[7] C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in Proc. of CCS'09, 2009, pp. 213– 222.

[8] 104th United States Congress, "Health Insurance Portability and Accountability Act of 1996 (HIPPA)," Online at http:// aspe.hhs.gov/ admnsimp/pl104191.htm, 1996.

[9] R. C. Merkle, "Protocols for public key cryptosystems," in Proc. of IEEE Symposium on Security and Privacy, Los Alamitos, CA, USA, 1980.

[10] Y. Dodis, S. P. Vadhan, and D. Wichs, "Proofs of retrievability via hardness amplification," in TCC, 2009, pp. 109–127.

[11] Cloud Security Alliance, "Se-curity guidance for critical areas of focus in cloud computing," 2009, http:// www. cloudsecurityalliance.org.