Journal of Engineering Sciences
Vol 11, Issue 4 , April/2020
ISSN NO:0377-9254

An Effective Integrity Check Scheme for Secure Erasure Code-Based Storage Systems

[1]Husna Mazhar, [2]Sadiya Sadiya Sultana, [3]Syeda Mahnaaz Fatima, [4]Rafath Samrin

[1], [2], [3] UG Scholar, Department of Information Technology, ISL Engineering College, Hyderabad, India.
[4]HOD & Professor, Dept of Information Technology, ISL Engineering College, Hyderabad, India.

Objective:

We propose a novel method called secured erasure code based algorithm for cloud data security in distributed storage system.

ABSTRACT:

High-speed networks and ubiquitous Internet access become available to users for access anywhere at any time. Cloud computing is a concept that treats the resources on the Internet as a unified entity, a cloud. 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. Hosting companies operate large data centers, and people who require their data to be hosted buy or lease storage capacity from them.

The data center operators, in the background, virtualize the resources according to the requirements of the customer and expose them as storage pools, which the customers can themselves use to store files or data objects. Physically, the resource may span across multiple 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. A decentralized erasure code is suitable for use in a distributed storage system.

We construct a secure cloud storage system that supports the function of secure data forwarding by using an AES and Proxy re encryption. In this model initial phase owner will upload the data with AES Encryption. Next phase, inside of cloud again the data has divided into small pieces, for this process we will apply a dividing key. Data will place in different storage lactations. The information of data storage will monitor by a unique data distributors. If the valid user accessing the data cloud will retrieve the data as reversible manner.

Keywords: Cloud Management, Cloud Computing, AES Algorithm.

EXISTING SYSTEM:

In Existing System we use a straightforward integration method. In straightforward integration method 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. General encryption schemes protect data confidentiality, but also limit the functionality of the storage system because a few operations are supported over encrypted data. A decentralized architecture for storage systems offers good scalability, because a storage server can join or leave without control of a central authority.

DISADVATAGES:

- The user can perform more computation and communication traffic between the user and storage servers is high.
- The user has to manage his cryptographic keys otherwise the security has to be broken.
- The data storing and retrieving, it is hard for storage servers to directly support other functions.

PROPOSED SYSTEM:

In our proposed system we address the problem of forwarding data to another user by

Journal of Engineering Sciences

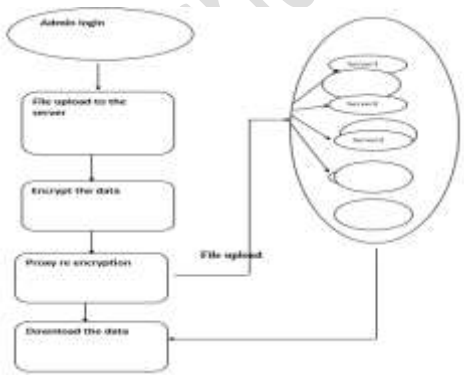
storage servers directly under the command of the data owner. We consider the system model that consists of distributed storage servers and key servers. Since storing cryptographic keys in a single device is risky, a user distributes his cryptographic key to key servers that shall perform cryptographic functions on behalf of the user. These key servers are highly protected by security mechanisms.

Here Storage system has allocates by different data container. Once owner uploads the data with AES encryption mechanism, system again takes the data and makes Secure Data segregation process. All the data pieces will be save in different location in cloud storage. Here public distributor monitors all the data and corresponding positions where it is saved. When a proper client asking the data, cloud system will provide the data in reversible manner. So our system will prevent our data from both Inside and Outside attackers.

ADVANTAGES:

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

System architecture:



Module

1. Registration
2. Sharing Data
3. Secure Cloud Storage
4. Proxy re-encryption
5. Data retrieval

Registration:

For the registration of user with identity ID the group manager randomly selects a number. Then the group manager adds into the group user list which will be used in the traceability phase. After the registration, user obtains a private key which will be used for group signature generation and file decryption.

Sharing Data:

The canonical application is data sharing. The public auditing property is especially useful when we expect the delegation to be efficient and flexible. The schemes enable a content provider to share her data in a confidential and selective way, with a fixed and small ciphertext expansion, by distributing to each authorized user a single and small aggregate key.

Secure Cloud Storage:

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. A decentralized erasure code is suitable for use in a distributed storage system.

Proxy re-encryption:

Proxy re-encryption schemes are crypto systems which allow third parties (proxies) to alter a cipher text which has been encrypted for one user, so that it may be decrypted by another user. By using proxy re-encryption technique the encrypted data (cipher text) in the cloud is again altered by the user. It provides highly secured information stored in the cloud. Every user will have a public key and private key. Public key of every user is known to everyone but private key is known only the particular user.

Data retrieval:

Reports and data are the two primary forms of the retrieved data from servers. There are some overlaps between them, but queries generally select a relatively small portion of the server, while reports show larger amounts of data. Queries also present the data in a standard format and usually display it on the monitor; whereas reports allow formatting of the output however you like and is normally retrieved.

Screenshot:

Fig 1: in this it shows the home page.

Fig 2: in this it shows the login of the admin

Fig 3: in this it shows user registration.

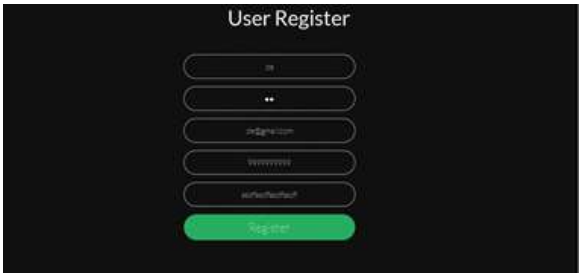


Fig 4: in this it shows to upload a file

Conclusion:

Erasure codes are promising for improving the reliability of the storage system due to its space efficiency compared to the replication methods. Traditional erasure codes split data into equalized data blocks and encode strips in different data blocks. This brings heavy repairing traffic when clients read parts of the data, since most strips read for repairing are not in the expected blocks. This paper proposes a novel discrete data dividing method to completely avoid this problem. The key idea is to encode strips from the same data block. We could see that for repairing failed blocks, the strips to be read are either in the same data block with corrupted strips or from the encoded strips. Therefore, no data is wasted. We design and implement this data layout into a HDFS-like storage system. Experiments over a small-scale testbed shows that the proposed discrete data divided method avoids downloading data blocks that are not needed for clients during the repairing operations.

REFERENCES

[1] James S. Plank, Erasure Codes for Storage Systems A Brief
Primer, *USENIX .login*, Vol. 38 No. 6, 2013.

[2] Hsing-bung Chen, Ben McClelland, et al., An Innovative Parallel Cloud Storage System using OpenStack's Swift Object Store and Transformative Parallel I/O Approach, *Los Alamos National Lab Science Highlights*, 2013.

[3] Corentin Debains, Gael Alloyer, Evaluzation, Evaluation of Erasure-coding libraries on Parallel Systems, 2010.

[4] Peter Sobe, Parallel Reed/Solomon Coding on Multicore Processors, in *Proceedings of International Workshop on Storage Network Architecture and parallel I/O*, 2010.

[5] Babak Behzad, Improving parallel I/O auto tuning with performance modeling, in *Proceedings of ACM International Symposium on High-performance Parallel and Distributed Computing (HPDC)*, 2014.
[6] Hsing-bung Chen, parEC – A Parallel and Scalable of erasure coding support in Cloud Object Storage Systems, Los Alamos National Lab.
[7] A. Varbanescu , On the Effective Parallel Programming of Multi-core Processors, Ph.D Thesis, Technische Universiteit Delft , 2010.
[8] William Gropp Ewing Lusk, Anthony Skjellum, Using MPI:Portable Parallel Programming with the Message-Passing Interface, The MIT Press, 2014.

Journal of Engineering Sciences