IJARETY

International Journal of Advanced Research in Education and TechnologY (IJARETY)

Volume 11, Issue 4, July-August 2024

Impact Factor: 7.394

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

International Journal of Advanced Research in Education and TechnologY(IJARETY)

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 7.394| A Bi-Monthly, Double-Blind Peer Reviewed & Referred Journal |

|| Volume 11, Issue 4, July-August 2024 ||

DOI:10.15680/IJARETY.2024.1104006

An Effective Secure Erasure Code based Storage System Integrity Check Scheme

Mrs.P.Vanitha, Ms.C.Srinithi

Assistant Professor, Department of Computer Applications (UG), Hindusthan College of Arts & Science, Coimbatore, Tamil Nadu, India

III BCA Student, Department of Computer Applications (UG), Hindusthan College of Arts & Science, Coimbatore, Tamil Nadu, India

ABSTRACT: Data integrity becomes a key challenge as erasure code-based storage solutions become more and more popular. In this paper, a robust integrity check scheme tailored to protect erasure code-based storage systems is presented. By identifying and fixing mistakes made during data transfer or storage and taking security into account, the suggested method improves the dependability of data storage. Using cryptographic techniques, the approach verifies the integrity of the encoded data to make sure it is uncorrupted and impervious to tampering. To attain efficiency and strong error detection capabilities, a well-crafted blend of hash functions and error-correcting codes is utilized. The suggested strategy, in contrast to conventional integrity check techniques, is designed to take into account the special features of erasure code-based storage systems, in which data is distributed among several nodes. The integrity check technique uses key management mechanisms to safeguard the cryptographic keys used during the verification process, hence improving security.

This keeps illegal access and manipulation at bay and guarantees that only authorized entities are able to carry out integrity checks. Moreover, the plan offers a way for cryptographic keys to be updated dynamically in order to reduce the possibility of key compromise. a thorough security examination of the suggested integrity check system, assessing how well it defends against different adversarial models and possible points of attack. Performance assessments also show how effective the system is in terms of computing overhead and storage needs, which makes it feasible to implement in widely dispersed large-scale storage environments. Through testing in erasure code-based storage configurations, the scheme's efficacy is confirmed, demonstrating its capacity to identify and fix problems while upholding a safe storage environment. The suggested integrity check scheme makes a significant improvement to the security of storage systems based on erasure codes by guaranteeing data confidentiality and integrity in the face of changing risks and difficulties.

I. INTRODUCTION

In our proposed system they address the problem of forwarding data to another user by storage servers directly under the command of the data owner. 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 attackes.

1.1 OBJECTIVE: A novel method called secured erasure code based algorithm for clouddata security in distributed storage system.

II. LITERATURE SURVEY

1. QoS Support for End Users of I/O-intensive ApplicationsUsing SharedStorageSystems.
Author: Xuechen Zhang ECE Department Wayne State Universities Trans.
Kei Davison Alamos National Laboratory Parallel and Distributed Systems, vol. 21, no. 11, pp. 1586-1594, Nov. 2010.

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 7.394| A Bi-Monthly, Double-Blind Peer Reviewed & Referred Journal |

|| Volume 11, Issue 4, July-August 2024 ||

DOI:10.15680/IJARETY.2024.1104006

The problem of constructing an erasure code for storage over a network when the data sources are distributed. Specifically, assume that there are n storage nodes with limited memory and k < n sources generating the data.want a data collector, who can appear anywhere in the network, to query any k storage nodes and be able to retrieve the data.

2. On the Effective Parallel Programming of Multi-core Processors.
Author: Prof.dr.ir. H.J. Sips Technische Universities Delft, promotor Prof.dr.ir. A.J.C. van GemundTechnische Universities Delft Prof.dr.ir. H.E. Bal. 7 December 2010.
Availability is a storage system property that is both highly desired and yet minimally engineered. While many systems provide mechanisms to improve availability– such as redundancy and failure recovery – how to best configure these mechanisms is typically left to the system manager. Unfortunately, few individuals have the skills to properly manage the trade-offs involved, let alone the time to adapt these decisions to changing conditions. Instead, most systems are configured statically and with only a cursory understanding of how the configuration will impact overall performance or availability. While this issue can be problematic even for individual storage arrays, it becomes increasingly important as systems are distributed – and absolutely critical for the wide area peer-to-peer storage infrastructures being explored. This paper describes the motivation, architecture and implementation for a new peer-to-peer storage system, called Total Recall that automates the task of availability management.

3. Parallel Reed/Solomon Coding on Multicore Processors.
Author:Peter Sobs Institute of Computer Engineering University LuebeckLuebeck, Germany. 2010 EEE DOI 10.1109/SNAPI.2010.16
This paper sketches the design of PAST, a large-scale, Internet-based, global storage utility that provides scalability, high availability, persistence and security. PAST is a peer-to-peer Internet application and is entirely selforgaining. PAST nodes serve as access points for clients, participate in the routing of client requests, and contribute storage to the system. Nodes are not trusted, they may join the system at any time and may silently leave the system without warning. Yet, the system is able to provide strong assurances, efficient storage access, load balancing and scalability.

4. Privacy-preserving and Secure Distributed Storage Codes
Author: Nihar B. Shah, K. V. Rashmi, Kennan Ramchandran, Fellow, IEEE, and P. Vijay Kumar, Fellow, IEEE. 2011.
Storage outsourcing is a rising trend which prompts a number of interesting security issues, many of which have been extensively investigated in the past. However, Provable Data Possession (PDP) is a topic that has only recently appeared in the research literature. The main issue is how to frequently, efficiently and securely verify that a storage server is faithfully storing its client's(potentially very large) outsourced data. The storage server is assumed to be untrusted in terms of both security and reliability. (In other words, it might maliciously or accidentally erase hosted data; it might also relegate it to slow or off-line storage.) The problem is exacerbated by the client being a small computing device with limited resources.

5. Parallel Reed/Solomon Coding on Multicore Processors
Author: Peter Sobe Institute of Computer Engineering University of Luebeck,
Germany.2011,IEEE.
As data have been growing rapidly in data centers, Deduplication storage systems continuously face challenges in providing the corresponding throughputs and capacities necessary to move backup data within backup and recovery window times. One approach is to build a cluster Deduplication storage system with multiple Deduplicationstorage system nodes. The goal is to achieve scalable throughput and capacity using extremely high throughput (e.g. 1.5 GB/s) nodes, with a minimal loss of compression ratio. The key technical issue is to route data intelligently at an appropriate granularity.

III. PROPOSED SYSTEM

In our proposed system the problem of forwarding data to another user by storage servers directly under the command of the data owner. 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

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 7.394| A Bi-Monthly, Double-Blind Peer Reviewed & Referred Journal |

|| Volume 11, Issue 4, July-August 2024 ||

DOI:10.15680/IJARETY.2024.1104006

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 attackes.

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.

Primary goals:

There were five primary goals in the creation of the Java language:

- It should use the object-oriented programming methodology.
- It should allow the same program to be executed on multiple operating systems.
- It should contain built-in support for using computer networks.
- It should be designed to execute code from remote sources securely.
- It should be easy to use by selecting what were considered the good parts of other object-oriented languages

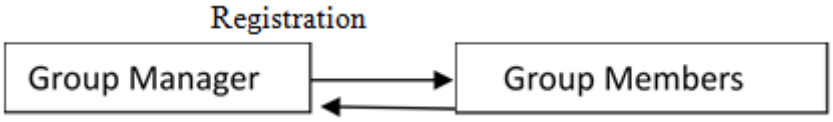
Description:

- An Entity Relationship Diagram (ERD) is a visual representation of different data using conventions that describe how these data are related to each other.
- An entity can be a person, place, event, or object that is relevant to a given system. Here the sender, receiver, etc are the entities
- A relationship describes how entities interact. Here the sender select video and reserve room so select is a relation..

IV. MODULE DESCRIPTION

4.1 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.



4.2 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 confidentialand selective way, with a fixed and small cipher text expansion, by distributing to each authorized user a single and small aggregate key.

4.3 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.

4.4 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.

4.5 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

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 7.394| A Bi-Monthly, Double-Blind Peer Reviewed & Referred Journal |

|| Volume 11, Issue 4, July-August 2024 ||

DOI:10.15680/IJARETY.2024.1104006

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.

4.6 Advanced Encryption Standard (AES):

The Advanced Encryption Standard (AES) is an encryption algorithm for securing sensitive but unclassified material by U.S. Government agencies and, as a likely consequence, may eventually become the de facto encryption standard for commercial transactions in the private sector. (Encryption for the US military and other classified communications is handled by separate, secret algorithms.)In January of 1997, a process was initiated by the National Institute of Standards and Technology (NIST), a unit of the U.S. Commerce Department, tofind a more robust replacement for the Data Encryption Standard (DES) and to a lesser degree Triple DES. The specification called for a symmetric algorithm (same key for encryption and decryption) using block encryption (see block cipher) of 128 bits in size, supporting key sizes of 128, 192 and 256 bits, as a minimum. The algorithm was required to be royalty-free for use worldwide andoffer security of a sufficient level to protect data for the next 20 to 30 years..The entire selection process was fully open to public scrutiny and comment, it being decided that full visibility would ensure the best possible analysis of the designs In August 1999, NIST selected five algorithms for more extensive analysis. These were:

- MARS, submitted by a large team from IBM Research
- RC6, submitted by RSA Security
- Rijndael, submitted by two Belgian cryptographers, Joan Daemen and Vincent Rijmen
- Serpent, submitted by Ross Andersen, Eli Biham and Lars Knudsen
- Twofish, submitted by a large team of researchers including Counterpane's respected cryptographer, Bruce Schneier

Implementations of all of the above were tested extensively in ANSI C and Java languages for speed and reliability in such measures as encryption and decryption speeds, key and algorithm set-up time and resistance to various attacks, both in hardware- and software-centric systems. Once again, detailed analysis was provided by the global cryptographic community (including some teams trying to break their own submissions). The end result was that on October 2, 2000, NIST announced that Rijndael had been selected as the proposed standard. On December 6, 2001, the Secretary of Commerce officially approved Federal Information Processing Standard (FIPS) 197, which specifies that all sensitive, unclassified documents will use Rijndael as the Advanced Encryption Standard.

AES is based on a design principle known as a Substitution permutation network. It is fast in both software and hardware. Unlike its predecessor, DES, AES does not use a Feistel network.AES has a fixed block size of 128 bits and a key size of 128, 192, or 256 bits, whereas Rijndael can be specified with block and key sizes in any multiple of 32 bits, with a minimum of 128 bits. The blocksize has a maximum of 256 bits, but the keysize has no theoreticalmaximum.AES operates on a 4×4 column-major order matrix of bytes, termed the state (versions of Rijndael with a larger block size have additional columns in the state). Most AES calculations are done in a special finite field.The AES cipher is specified as a number of repetitions of transformation rounds that convert the input plaintext into the final output of ciphertext. . Each round consists of several processing steps, including one that depends on the encryption key. A set of reverse rounds are applied to transform ciphertext back into the original plaintext using the same encryption key.

V. IMPLEMENTATION

The first step to implementing cloud computing services is determining the objective of implementing it; that is, the company must first figure out why they want to use cloud computing services in the first place. This includes investigating the weaknesses of the service they currently use, and setting a clear-cut goal for how data will be stored after implementing cloud computing.

The next step is planning out a beginning, middle, and final state of data storage for the company in terms of implementing cloud computing. After conducting a current state analysis on the IT service that is used at the time, it is then necessary to plan what the state of data storage and recovery will be during the implementation phase, and after. It is also important to decide which cloud model would work best based on the weaknesses of the current service provider outlined in Step 1. The final component of Step 2 is to plan how to sustain the new cloud computing model for long-term use.

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 7.394| A Bi-Monthly, Double-Blind Peer Reviewed & Referred Journal |

|| Volume 11, Issue 4, July-August 2024 ||

DOI:10.15680/IJARETY.2024.1104006

The third step of cloud computing implementation is actually implementing the new service provider. Ways of carrying out this step vary depending on if the company selects a public, private, or hybrid cloud model.
The fourth step of implementing a new cloud service provider is initiating a maintenance model for long-term use. This includes the business side (building value metrics and making sure it holds enough data) and the IT side (spotting and fixing any system bugs or failures, and ensuring that security and feature updates are up to date). This final step is an ongoing process, as bugs must be repaired as they pop up, and requirements for the cloud model could often change. Part of the last step is to continue to enhance the cloud model in order to cater to changing requirements. The managed service provider also has to continually focus on ongoing security enhancements, resilience improvements, and capacity planning to ensure the highest level of protection and operational efficiency for their clients' systems.

VI. CONCLUSION

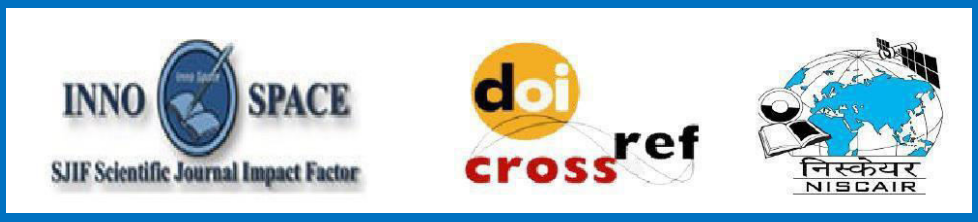
Erasure codes are promising for improving the reliability of the storage system due to its space efficiency compared to the replication methods. Traditional erasure codes split data into equalsized data blocks and encode strips in different data blocks. This brings heavy repairing traffic when clients 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 test bed shows that the proposed discrete data divided method avoids downloading data blocks that are not needed for clients during the repairing operations.

VII. FUTURE WORK

As a response, erasure coding as an alternative to backup has emerged as a method of protecting against drive failure. Raid just does not cut it in the age of high-capacity HDDs. The larger a disk's capacity, the greater the chance of bit error. And when a disk fails, the Raid rebuild process begins, during which there is no protection against a second (or third) mechanism failure. So not only has the risk of failure during normal operation grown with capacity, it is much higher during Rai rebuild, too. Also, rebuild times were once measured in minutes or hours, but disk transfer rates have not kept pace with the rate of disk capacity expansion, so large Raid rebuilds can now take days or even longer.

REFERENCES

[1] James S. Plank, Erasure Codes for Storage Systems A BriefPrimer, USENIX .login, Vol. 38 No. 6, 2013.
[2] Hsing-bung Chen, Ben McClelland, et al., An InnovativeParallel Cloud Storage System using OpenStack's Swift ObjectStore and Transformative Parallel I/O Approach, Los AlamosNational Lab Science Highlights, 2013.
[3] Corentin Debains, Gael Alloyer, Evaluzation, Evaluation ofErasure-coding libraries on Parallel Systems, 2010.
[4] Peter Sobe, Parallel Reed/Solomon Coding on MulticoreProcessors, in Proceedings of International Workshop onStorage Network Architecture and parallel I/O, 2010.
[5] Babak Behzad, Improving parallel I/O auto tuning withperformance modeling, in Proceedings of ACM InternationalSymposium on High-performance Parallel and Distributed
Computing (HPDC), 2014.
[6] Hsing-bung Chen, parEC – A Parallel and Scalable of erasurecoding support in Cloud Object Storage Systems, Los AlamosNational Lab.
[7] A. Varbanescu , On the Effective Parallel Programming ofMulti-core Processors, Ph.D Thesis, Technische UniversiteitDelft , 2010.
[8] William Gropp Ewing Lusk, Anthony Skjellum, Using MPI:Portable Parallel Programming with the Message-PassingInterface, The MIT Press, 2014.
[9] Hsing-bung Chen, Parallel Workload Benchmark on HybridStorage EcoSystem, Los Alamos national Lab.



International Journal of Advanced Research in Education and Technology

ISSN: 2394-2975 **Impact Factor: 7.394**

www.ijarety.in editor.ijarety@gmail.com