Scientific Journal of Impact Factor (SJIF): 4.14

e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 3, Issue 4, April -2016

DISTRIBUTED DEDUPLICATION SYSTEMS TO ACHIEVE RELIABILITY

Mitali Aher, Harshada Patil, Seema Karale, Sagar Suvarnkar, Prof. Lamikant Malphedwar

BE Scholar Computer Engineering D Y Patil school of Engg. Academy, Ambi, Pune
BE Scholar Computer Engineering D Y Patil school of Engg. Academy, Ambi, Pune
BE Scholar Computer Engineering D Y Patil school of Engg. Academy, Ambi, Pune
BE Scholar Computer Engineering D Y Patil school of Engg. Academy, Ambi, Pune
Asst. professer Computer Engineering D Y Patil school of Engg. Academy, Ambi, Pune

Abstract *— Information de-duplication is a system for putting off reproduction duplicates of information, and has been extensively utilized as a part of disbursed garage to diminish garage room and transfer information switch capacity. Be that as it can, there is one and simplest duplicate for each report put away in cloud no matter the opportunity that any such report is claimed by means of endless. For that reason, de-duplication framework complements outsource utilization whilst diminishing dependability. Furthermore, the test of safety for delicate facts likewise emerges when they're outsourced with the aid of clients to cloud. Looking forward to address the above security challenges, this paper makes the first endeavor to formalize the idea of disseminated strong de-duplication framework. We recommend new disseminated de-duplication frameworks with better unwavering nice in which the information pieces are circulated over distinct cloud servers. The safety stipulations of data privacy and label consistency are additionally executed with the aid of supplying a deterministic thriller sharing plan in conveyed outsource frameworks, in preference to utilizing focalized encryption as part of beyond de-duplication frameworks. Protection examination famous that our de-duplication frameworks are relaxed regarding the definitions determined in the proposed protection model. As a proof of concept, we actualize the proposed frameworks and exhibit that they brought about overhead is notably restricted in realistic conditions.*

Keywords*- De-duplication, distributed storage system, reliability, secret sharing.*

I. INTRODUCTION

There are two styles of de-duplication as a long way as the dimensions: (i) File level de-duplication, which finds redundancies between diverse files and uproots these redundancies to lessen restriction requests, and (ii) block stage de-duplication, which unearths and evacuates redundancies between data blocks. The file may be partitioned into littler altered length or variable-length blocks. Utilizing settled size squares disentangles the calculations of block limits, while utilizing variable length portions (e.g., taking into consideration Rabin fingerprinting) gives higher de-duplication effectiveness.

II. LITERATURE REVIEW

1.Optimizing Cauchy Reed-Solomon Codes for Fault-Tolerant Storage Applications
Author: James S. Plank

In this paper, we exhibit a change to Cauchy Reed-Solomon coding that depends on reformation the Cauchy distribution framework. We detail a calculation for creating great frameworks and after that assess the execution of encoding utilizing all behavior of Reed-Solomon coding, in addition to the best MDS codes from the writing. The enhancements over the first Cauchy Reed-Solomon codes are as much as 83% in practical situations, and normal around 10% over all cases that we tried.

2. Fast and Secure Laptop Backups with Encrypted De-duplication
Author: Paul Anderson and Le zhang

This paper describes an algorithm which takes advantage of the data which is common between users to increase the speed of backups, and reduce the storage requirements. This algorithm supports client-end per-user encryption which is necessary for confidential personal data. It also supports a unique feature which allows immediate detection of common subtrees, avoiding the need to query the backup system for every file. We describe a prototype implementation of this algorithm for Apple OS X, and present an analysis of the potential effectiveness, using real data obtained from a set of typical users. Finally, we discuss the use of this prototype in union with remote cloud storage, and present an analysis of the typical cost savings..

3. A Secure Data Deduplication Scheme for Cloud Storage
Author: Jan Stanek, Alessandro Sorniotti, Elli Androulaki, and Lukas Kencl

We show a clever thought that differentiates information as indicated by their common. In view of this thought, we plan an encryption plan that ensures semantic security for unlikable information and gives weaker security and better storage and transfer speed benefits for well known information. Along these lines, information deduplication can be effective for important information, whilst semantically secure encryption secures disliked substance.

4. Side channels in cloud services, the case of deduplication in cloud storage
Author: Danny Harnik
We study the privacy implications of cross-user deduplication. We display how deduplication can be used as a side channel which reveals information about the contents of files of other users. In a different scenario, deduplication can be used as a secret channel by which hateful software can communicate with its control center, regardless of any firewall settings at the attacked machine. Due to the high savings offered by crossuser deduplication, cloud storage providers are unlikely to stop using this technology. We therefore propose simple mechanisms that enable crossuser deduplication while greatly reducing the risk of data leakage.

5. PSiOS: Bring Your Own Privacy & Security to iOS Devices
Author: Tim Werthmann1, Ralf Hund1, Lucas Davi
In this paper, we mean to address the open problem of preventing (not only detecting) privacy leaks and at the same time reduction security against runtime attacks on iOS. Compared to similar research work on the open Android, realizing such a system for the closed source iOS is highly involved.

III. PROPOSED SYSTEM

Four new secure deduplication frameworks are proposed to furnish effective deduplication with high dependability for file level and block level deduplication, individually. The conventional encryption techniques, is used to secure information privacy. In particular, information are split into sections by utilizing secure sharing plots and put away at diverse servers. Our proposed developments support both file level and block level deduplications.

IV. MATHEMATICAL MODEL

Let S be the Whole system which consists,
S= {I, P, O}
Where,
I-Input,
P- procedure,
O- Output.
I-{F,U}
F-Filesset of {F1, F2,….,FN}
U- No of Users{U1,U2,……,UN}

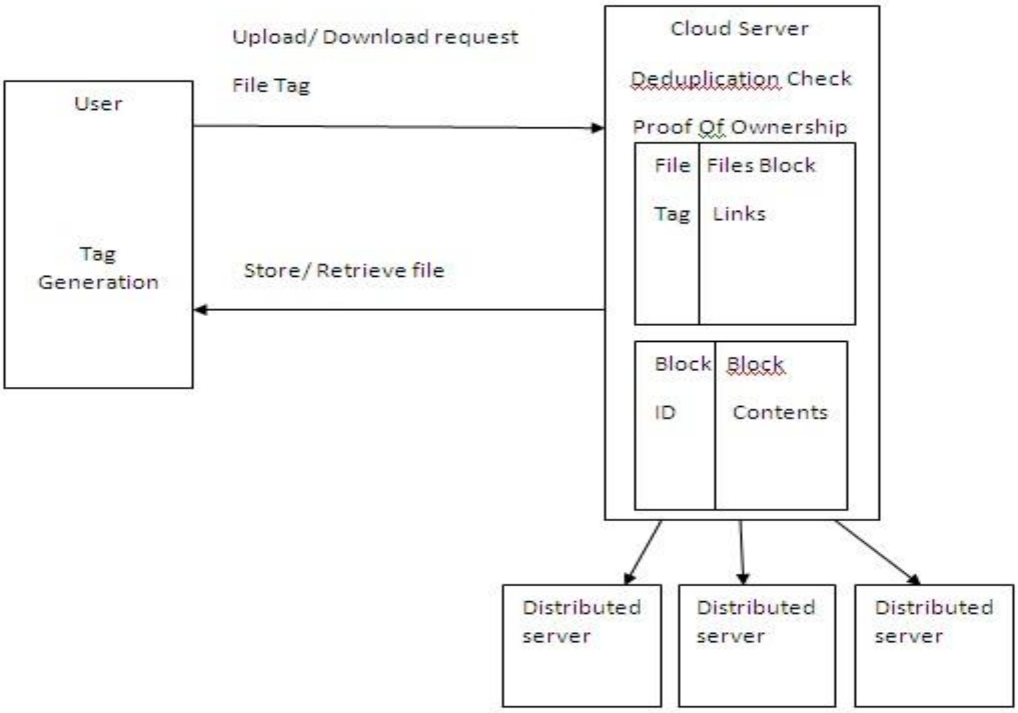
Procedure (P):
P= {POW, n , $^{\Phi}$, i, j ,m , k}.

Where,

1. POW - proof of ownership.
2. n - No of servers.

3. POW_B –Proof of ownership in blocks.
4. POW_F – Proof of ownership in files
5. Φ - tag.
6. i- Fragmentation.
7. j- No of server.
8. m-message
9. k- Key.

V. SYSTEM ARCHITECTURE



VI. IMPLEMENTATION

1. User Module:

New user first register on to the system. User login with entering correct username and assword. Then user perform uploading and downloading file by sending request to the cloud server. When user upload file then file level deduplication check. If already file exist then file deduplicated message shown to the user. Each time when uploading file then tag is generated by using SHA-1 algorithm i.e. message digest and send this digest to the cloud server for checking deduplicate files. If deduplicate occure then Call POW and response to the user from cloud server. If deduplicate not occure then AES algorithm used for encrypt the file AES-256 for key and send file tag to the cloud server. Encrypted file is send to Cloud Server divide this file into blocks then for each block message digest is stored at table with each encrypted block & block id check tag of each block if already exist or not if block tag is already exist then excess block id of that block & don’t store that encrypted block again & if block tag is not present then store that new block id, block tag & encrypted block. Share all blocks to distributed servers using Ramp Secrete Sharing Schemes if any failure occur then we recover that block from distributed servers. Recover block from distributed servers- using Ramp Secrete Sharing Schemes. For Downloading File first Decrypt Cloud Server file with AES algorithm by using Key from AES-256 algorithm on file- first merge all blocks of file then decrypt that file.

2. Admin Module:(Cloud Server Module)

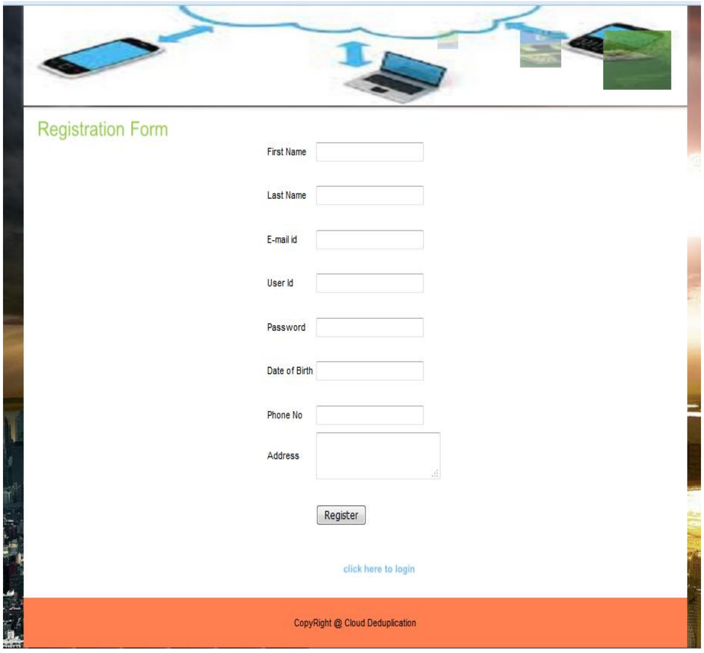
Admin means our cloud server first login to the system. Cloud server have all users uploaded files, all users deduplicated files, Blocks & share Servers name to recover blocks- using Ramp Secrete Sharing Schemes.

3. Three Distributed Servers Module:

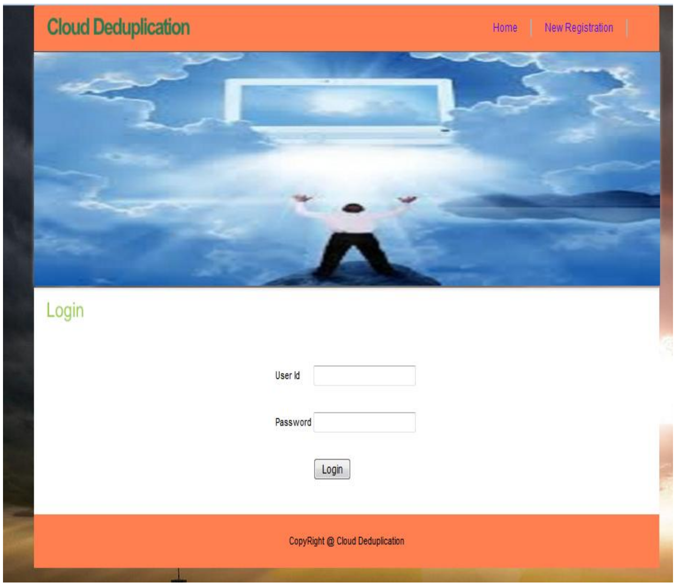
Distributed servers first login to the system. It store Blocks of the file and send block to cloud server if request comes from cloud server.

Impelementation:

1. **User Registration:**
   Here User enter their then details like name, email id, phone number, date of birth and address.

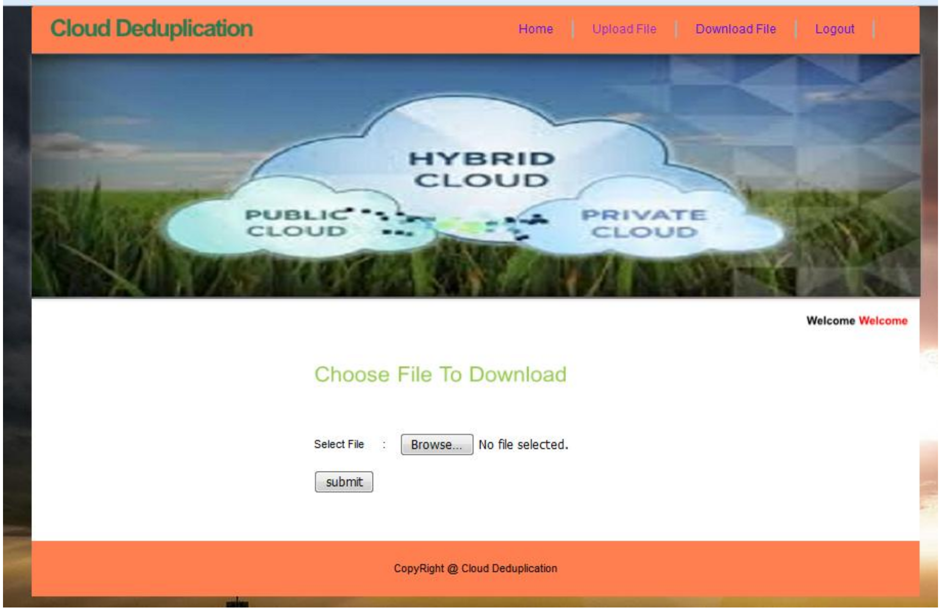


2. **User Login:**
   User enter in the system when entering correct User ID and password.



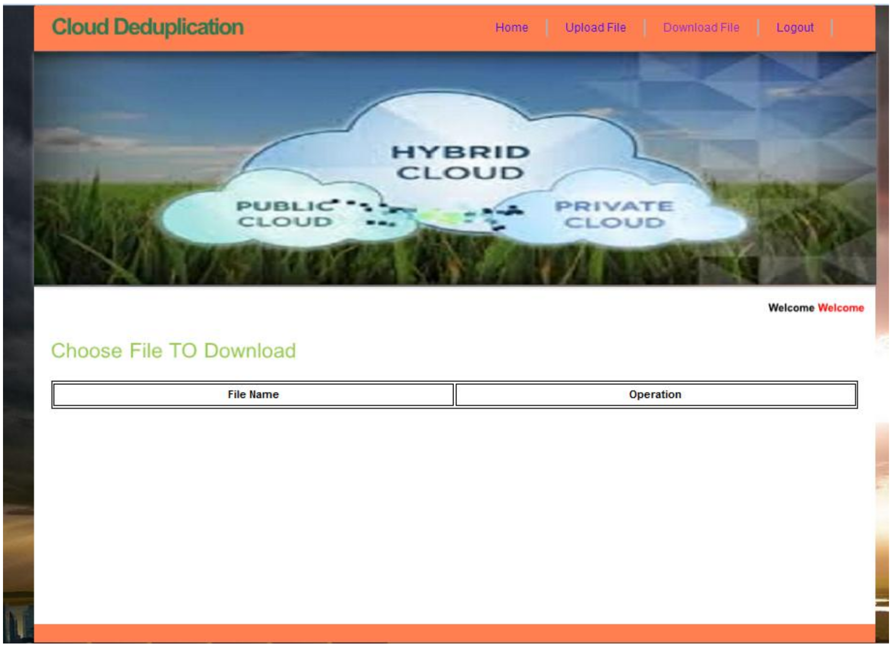
3. Upload File:

User select file from any location on the user system and upload file.



4. Download file:

User Download file from the cloud server.



Choose File TO Download

File Name	Operation

VII. CONCLUSION

We targeting the problem of comparing if an un trusted server shops a customer's information. We presented a model for provable data possession (PDP), wherein it is eye-catching to limit the file piece receives to, the calculation at the server, and the purchaser–server correspondence. Our answers for PDP fit this model: They reason a low (or even regular) overhead on the server and oblige a bit, regular degree of correspondence according to project. Key components of our plans are the backing for spot checking, which ensures that the plans stay mild weight, and the homomorphism capable labels, which allow to affirm records possession while not having access to the genuine facts file. We likewise define the idea of hearty inspecting, which coordinates remote data checking (RDC) with forward mistake amending codes to moderate discretionarily little file basements and advocate a non unique alternate for adding energy to any spot checking-based RDC plan. Examinations show that our plans make it all the way down to earth to test ownership of large information sets. Beyond plans that do not permit checking out aren't commonsense whilst PDP is applied to illustrate ownership of lots of facts, as they force a significant I/O and computational weigh on the server.

REFERENCES

[1] J. S. Plank and L. Xu, “Optimizing Cauchy Reed-solomon Codes for fault-tolerant network storage applications,” in NCA-06: 5th IEEE International Symposium on Network Computing Applications, Cambridge, MA, July 2006.

[2] P. Anderson and L. Zhang, “Fast and secure laptop backups with encrypted de-duplication,” in Proc. of USENIX LISA, 2010.

[3] J. Stanek, A. Sorniotti, E. Androulaki, and L. Kencl, “A secure data deduplication scheme for cloud storage,” in Technical Report, 2013.

[4] D. Harnik, B. Pinkas, and A. Shulman-Peleg, “Side channels in cloud services: Deduplication in cloud storage.” IEEE Security & Privacy, vol. 8, no. 6, pp. 40–47, 2010.

[5] J. Xu, E.-C. Chang, and J. Zhou, “Weak leakage-resilient client-side deduplication of encrypted data in cloud storage,” in ASIACCS, 2013, pp. 195–206.