DEDUPLICATION WITH KEYS AND CHUNKS IN HDFS STORAGE PROVIDERS

Indira.N[1], Karthika.S[2], Sowmya.S[3], Visudha.L[4]

[1]*Associate professor, Department of Computer science, Panimalar Engineering College, Tamil Nadu, India*
[2] *Student, Department of Computer science, Panimalar Engineering College, Tamil Nadu, India*
[3] *Student, Department of Computer science, Panimalar Engineering College, Tamil Nadu, India*
[4] *Student, Department of Computer science, Panimalar Engineering College, Tamil Nadu, India*

ABSTRACT

Deduplication characterizes the elimination of duplicate or redundant information and it removes the repetitive information before storing it. These techniques are widely employed for data backup, network minimization, and storage overhead. Long established deduplication schemes have restrictions on encrypted data and security. In the proposed system, new deduplication techniques are employed efficiently. Instead of possessing periphrasis of a sole content, deduplication banishes redundant data by keeping only one physical copy and broaching other redundant data to that copy. Each one can be expounded based on nonidentical granularities which may be either way a whole file or a data block. MD5 and 3DES algorithm strengthen the technique. The methodology proposed here is (POF) of the file. Deduplication can now, properly address the reliability and tag consistency problem in HDFS storage systems. The proposed system succeeded in reducing the cost and time of uploading and downloading with storage space.

INDEX TERMS– *HDFS: Hadoop Distributed File System, MD5: Message Digest, 3DES: Triple Data Encryption Standard, POF: Proof Of Ownership*.

---

1. INTRODUCTION

BIG DATA is a term for data sets that are so large or complex that traditional data processing applications are inadequate to deal with them. Challenges include analysis, capture, data curation, search, sharing, storage, and transfer, visualization, querying, and updating information privacy.

Hadoop is an open-source software framework for storing data and running applications on clusters of commodity hardware. It provides massive storage for any kind of data, enormous processing power and the ability to handle virtually limitless concurrent tasks or jobs. Hadoop makes it possible to run applications on systems with thousands of commodity hardware nodes, and to handle thousands of terabytes of data. **Hadoop Distributed File System (HDFS)**– the Java-based scalable system that stores data across multiple machines without prior organization. HDFS has a master slave architecture containing a single name node as a **master** and a number of data nodes as slaves**.** Metadata is a data that describes other data.

Working of HDFS: To store a file in this architecture, HDFS splits the file into fixed-size blocks (e.g., 64MB) and stores them on Data nodes. The mapping of blocks to data nodes is determined by the name node and also manages the file system's metadata and Namespace.

Today's commercial cloud storage accommodations, such as drop box, mozy, memo Pal, have been applying deduplication to utilize data to preserve maintenance cost. From a user's perspective, data outsourcing raises security and privacy concerns.

In this paper we tend to achieve incipient distributed de-duplication systems with higher reliability in which the data chunks are distributed across HDFS storage systems and reliable key management technique is employed for secure de-duplication. The proficiency which has been proposed to expunge the shortcomings of the existing deduplication concept is convergent encryption, proof of ownership and efficient key management schemes. Therefore the substrate deduplication is performed at both file level and block level and we define a HDFS Master machine to enhance the security and storage.

To summarize our contributions:

- Offers an **efficient key management** solution through the metadata manager.
- Preserves **confidentiality and privacy** against malicious storage providers by encrypting the chunks which are at random storage.
- Assures both **block-level** and **file-level deduplication.**
- Define typical operations such as editing and deleting of contents.

2. RELATED WORK

"CloudDedup: secure deduplication with encrypted data for cloud storage ", Pasquale puzio[1], Refik Molva[2], Melek onen[3] [1], plans a framework which accomplishes secrecy and empowers block level deduplication in the meantime. Our framework is based on top of convergent encryption. We demonstrated that it merits performing block level deduplication rather than file level deduplication. Evades COF and LRI attacks (confirmation of record) (take in the rest of the data). **"A Hybrid cloud approach for secure authorized de-duplication", Jagadish[1], Dr.Suvurna Nandyal[2] [2],** Address the issue of authorized data duplication. Deals with hybrid cloud and thus possess the benefits of both the public and private cloud. The copy check tokens of documents are created by the private cloud server with private keys. **"Secure and constant cost public cloud storage auditing with De-duplication", Jiawel yuan[1], Shucheng yu[2] [3],** Outperforms existing POR and PDP schemes with deduplication**.** Consistent cost scheme that accomplishes secure public data integrity. **"Provable ownership of file in de-duplication cloud storage", Chao yang[1], jian ren[2], jianfeng ma[3] [4],** proposes a plan that can produce provable ownership for file [POF] and keep up a high discovery likelihood of the client misbehavior. Very proficient in lessening the weight of the client. **"Secure Deduplication with Efficient and Reliable Convergent Key Management" J.Li[1], X.Chen[2], M.Li[3], J.Li[4], P.Lee[5], and W.Lou[6], [5]** proposed dekey, a productive and solid management conspire for secure deduplication. They execute dekey utilizing the ramp secret sharing plan and show that it incurs little encoding/decoding overhead contrasted with the system transmission overhead in the general transfer/download operations. **"A Secure data deduplication scheme for cloud storage", J.Stanek[1], A.Sorniotti[2], E.Androulaki[3], and L.Kenel[4] [6],** private users outsource their data to cloud storage providers. Late data rupture episodes make end-to-end encryption an undeniably noticeable necessity. Data deduplication can be viable for popular data, while semantically secure encryption ensures disagreeable content. **"A reverse deduplication storage system optimized for reads to latest backups", C.Ng and P.Lee. Revdedup [7]** had present RevDedup, a de-duplication system designed for VM disk image backup in virtualization environments. RevDedup has several design goals: high storage efficiency, low memory usage, high backup performance, and high restore performance for latest backups. They extensively evaluate our RevDedup prototype using different workloads and validate our design goals. **"Secure Data Deduplication with Dynamic Ownership Management in Cloud Storage", Junbeom Hur[1], Dongyoung Koo[2], Youngjoo Shin[3], and Kyungtae Kang[4] [8],** a novel server-side deduplication scheme for encrypted data. It permits the cloud server to control access to outsourced data even notwithstanding when the proprietorship changes progressively by misusing randomized convergent encryption and secure possession gather key distribution. This counteracts data leakage exclusively to disavowed clients despite the fact that they beforehand possessed that information, additionally to a legitimate yet inquisitive distributed storage server. **"Enhanced Dynamic Whole File De-Duplication (DWFD) for Space Optimization in Private Cloud Storage Backup", M. Shyamala Devi[1], V. Vimal Khanna[2], A. Naveen Bhalaji[3] [9]**, provides dynamic space

optimization in private cloud storage backup as well as increase the throughput and de-duplication efficiency.**"Deduplication on Encrypted big data in cloud", Zhen yan[1], Wenxiu Ding[2] and Robert.H.Deng[3] [10],** De-duplicate encrypted data stored in cloud based on proxy re- encryption, Avoids encrypting of data while uploading thus saves bandwidth. Over came the brute force attack.

3. EXISTING METHODOLOGY

Existing systems have only been studied in a single-server setting [10] , [8]. De-duplication systems and Storage systems are predetermined by users and applications for higher reliability, especially in archival storage. Diverse clients may have similar information duplicates they should have their own arrangement of focalized keys so that no different clients can get to their documents. In particular, every client must affiliate an encoded convergent key with each block of its outsourced scrambled information duplicates, in order to later re-establish the information copies [1]. As an outcome the quantity of merged keys being presented directly scales with the number of blocks being put away. The gauge approach is untrustworthy, as it requires every client to dedicatedly ensure his own particular master key.

4. PROBLEMS IN THE EXISTING SYSTEM

The existing techniques were built over Single server system. Deduplication is not scalable with the enormous increasing cloud users. Efficient key management scheme was not maintained so as to manage the convergent keys generated with the increasing number of cloud users.Cost increases to the storage of content as well as for the keys storage. Security breaches as the technique is approached over a single server setting, if once hacked the info can be collected at a common server.

5. PROPOSED WORK

To empower the de-duplication in distributed storage of data crosswise over HDFS, the idea of de-duplication is accomplished by the method of Proof of Ownership. The convergent keys are outsourced to slave machines safely and uphold both file and block level de-duplication. We bear confidentiality and security by implementing Triple DES algorithm. Cost efficiency is accomplished as numerous clients of the same data is quite recently alluded and not recently included. Erasing/Editing contents of shared document of a various client will permit erasing/altering the convergent key references and not the soul content in HDFS file storage. In the event that a FILE is found to have copy duplicates, the soul content in the hdfs master has recently alluded to the slave machines.

The proposed work has mainly four phases, **MASTERING FILE TO HDFS STORAGE PROVIDER; SEGMENTING THE FILE CHOSEN; KEY SHARING; HASH VALUE BASED DECRYPTION**

5.1 MASTERING FILE TO HDFS STORAGE PROVIDER

In this module a user is an entity who wants to outsource data storage to the HDFS Storage and access the data later. User registers to the HDFS storage with necessary information and login the page for uploading the file. User chooses the file and uploads to Storage where the HDFS store the file in rapid storage system and file level deduplication is checked. We tag the file by using md5 message-digest algorithm which is a cryptographic hash function that produces the required hash value.

***Fig 1**: initial phase*

5.2 SEGMENTING THE FILE CHOSEN

As the file is being uploaded to the cloud, the next step would be the segmentation of the file chosen and tag generation. Hence we produce united keys for each block split, to check block level de-duplication. Here we give a filename and password for file approval in future. The blocks are encrypted by Triple Data Encryption Standard (3DES) algorithm. The plain content is encoded triple times with convergent key, thus while decoding the original content it likewise requires a similar key to decode it again with the same convergent keys.

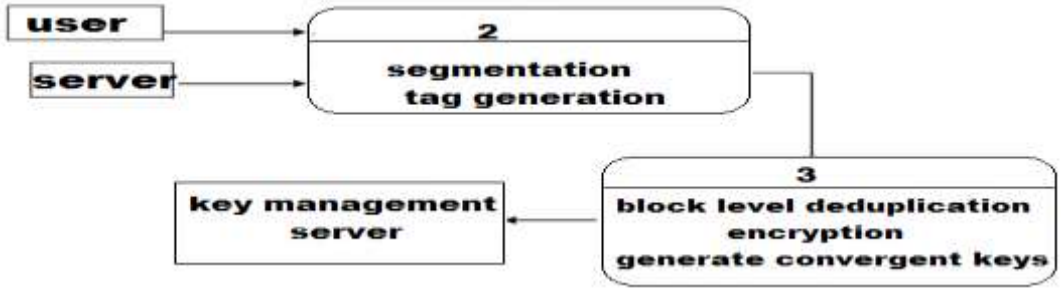


***Fig 2**: segmentation and tag generation of file*

5.3 KEY SHARING

After encryption the convergent keys are safely imparted with slave machines supplier to Key Management machines. Key administration slave checks copy duplicates of focalized (convergent) keys in KMCSP. Key Management slave keeps up Comma Separated Values (CSV) document to check evidence of verification and store keys safely. The diverse clients who share the common keys are alluded by their own particular proprietorship (proof of ownership). If User request for deletion, certainly he need to prove proof of ownership to delete own contents.

5.4. HASH VALUE BASED DECRYPTION

The last module where the client asks for downloading their soul content that is stored in HDFS storage. This download request needs proper possession check of the document and affirms existing tag of the user which is as of now generated by md5 algorithm. The proprietorship is confirmed with the unique tag. After verification, the original content is decrypted by requesting the Distributed HDFS storage. Where HDFS storage request the key management slave for keys to decrypt and finally the original content is downloaded. The delete request will delete only the reference of the content shared by the common users and not the whole content.

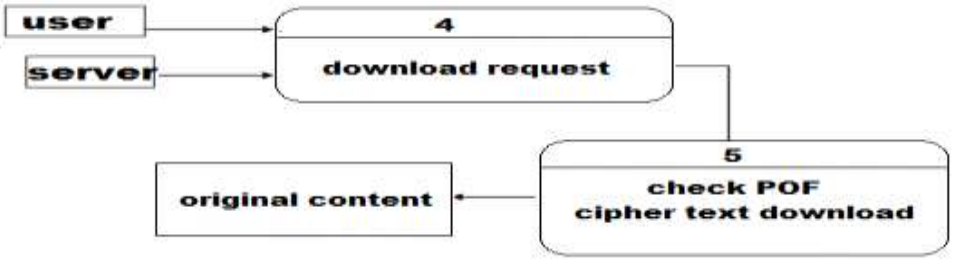


***Fig 3**: final phase- decrypting the file*

6. OVERALL ARCHITECTURE

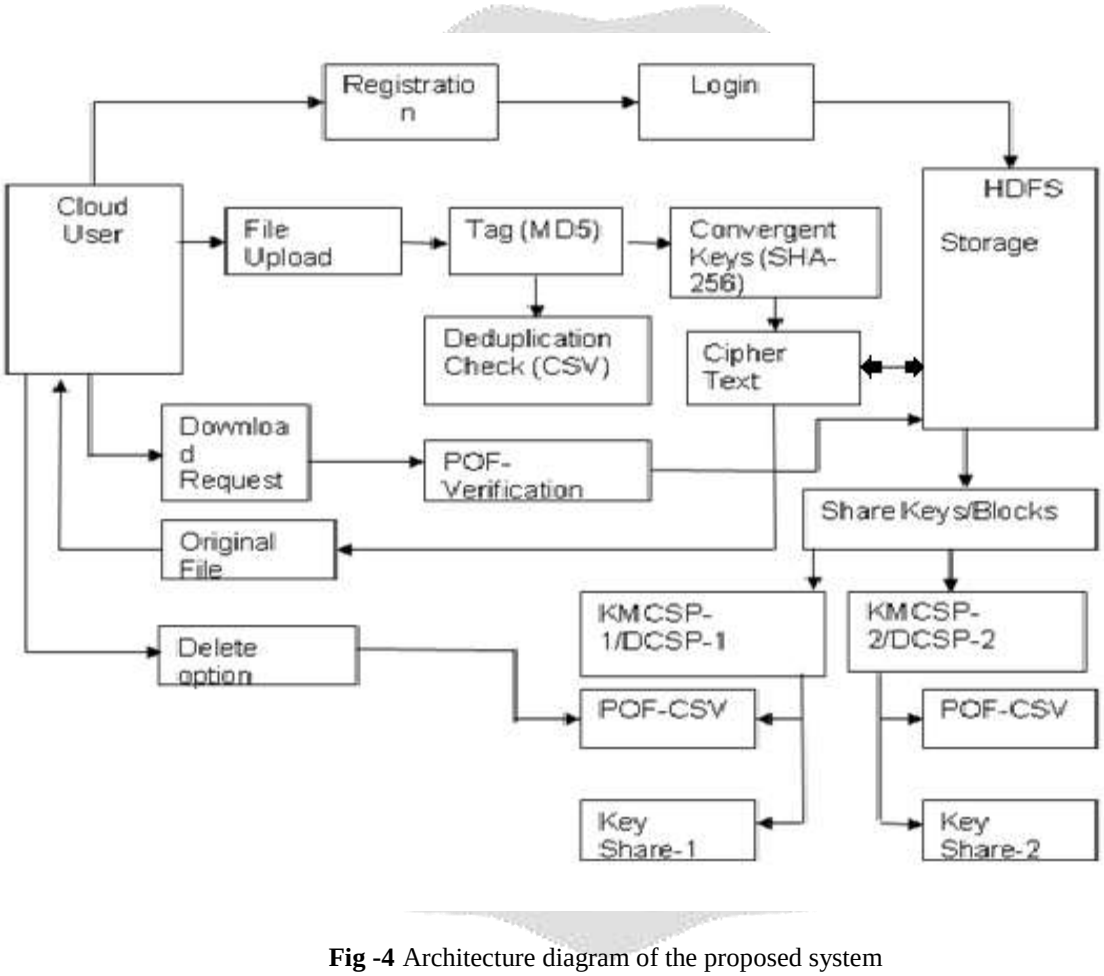
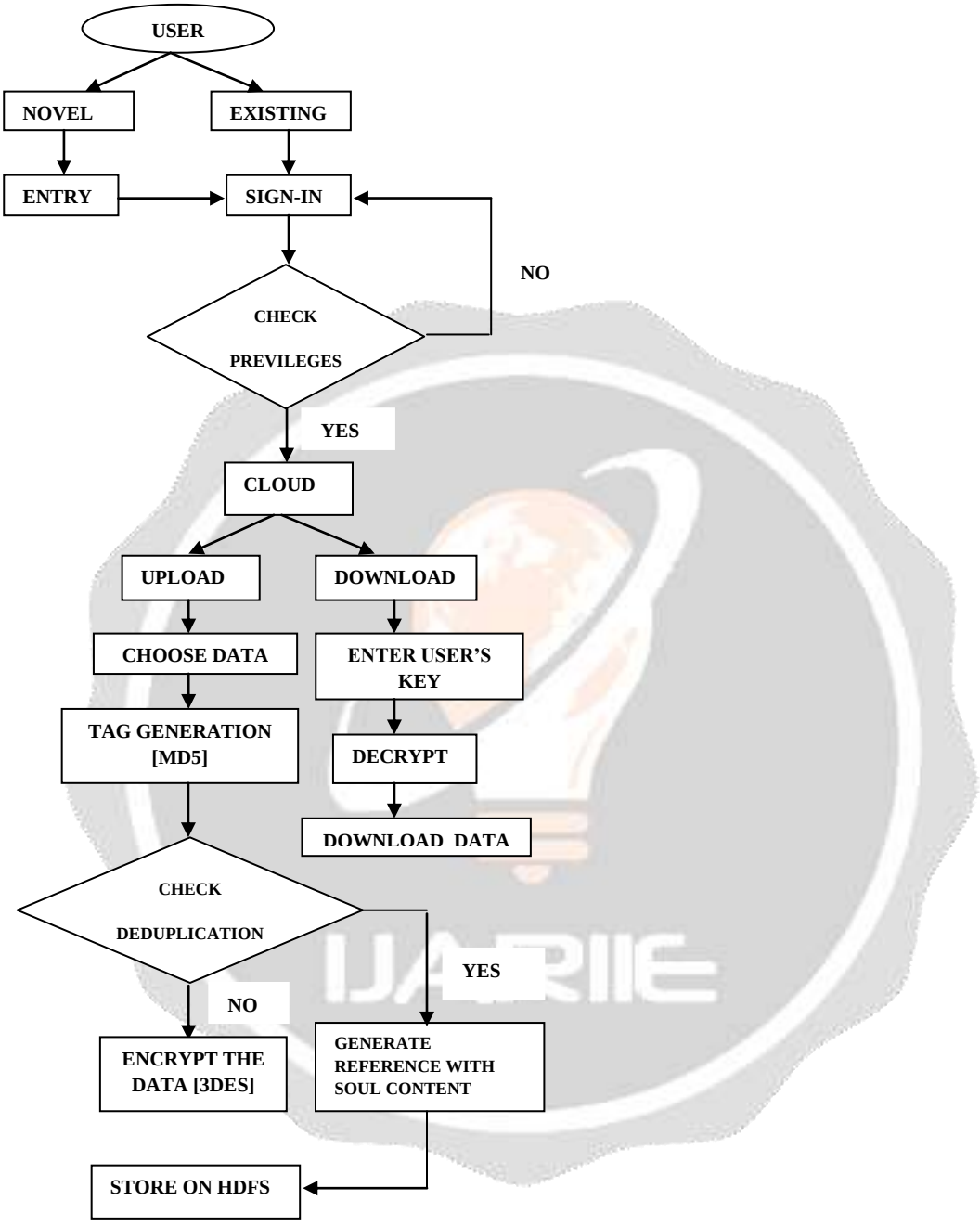


Fig -4 Architecture diagram of the proposed system

Fig -4 gives an out picture of the proposed system. The cloud user uploads the original file, at the initial phase a tag is generated to the entire file using MD5 algorithm. The convergent keys are generated respectively and the file gets uploaded to the HDFS storage as cipher text. As the tag is generated for each file, the de-duplication check(both block level and file level) is done using CSV. If a copy is found for the file uploaded, only the reference is generated with the soul content. The references are stored into the slave machines, and the hdfs master will have the soul content as cipher text. As on with the download request, PROOF OF VERIFICATION [POF] for each authenticated user is verified in the HDFS storage. The HDFS storage supports back with the cipher text to the authorized user. With the keys generated the cipher text is decrypted and original file is given to the cloud user. Regarding the delete and edit option, only the references are deleted and edited and not the soul content.

7. FLOW DIAGRAM OF PROPOSED SYSTEM



8. RESULTS AND DISCUSSIONS

Our system was tested for efficiency in terms of computation and communication cost requirements and it was found that our proposed method incurred very less computation and communication costs. Further, the uploading of cloud user's file to the HDFS storage was encrypted using the 3DES method which required very less cost for setting up and is considered the most efficient for our application since any attempt to hack the system is very expensive and is thus avoided. The Proof of ownership scheme was also reliable to authenticate each cloud user, and the tag generation of each file was generated using MD5 algorithm. Thus our system is found to radiate high performance and efficiency.

9. CONCLUSION

In this project, the new conveyed de duplication frame works with file level and fine grained blocked level data deduplication, support with higher unwavering quality in which the data lumps are appropriately aslant over HDFS storage, and reliable key administration is enforced in secure deduplication and the security of labels consistency and honesty were accomplished**.**

10. REFERENCES

[1] Pasquale puzio[1,] Refik Molva[2], Melek onen[3]: **Cloud Dedup - secure deduplication with encrypted data for cloud storage**. In IEEE 5th International Conference, Dec 2013.

[2] Jagadish[1], Dr.Suvurna Nandyal[2] : **A Hybrid cloud approach for secure authorized de-duplication**. Published in International Journal of Science and Research (IJSR), 2013.

[3] Jiawel Yuan[1], Shucheng yu[2]: **Secure and constant cost public cloud storage auditing with Deduplication .** In IEEE Conference, published in communication and network security, 2013.

[4] Chao Yang[1], Jianren[2], Jianfengma[3:] **Provable ownership of file in deduplication cloud storage.** Published in Global Communication Conference, Dec 2013.

[5] J.Li[1], X.Chen[2], M.Li[3], J.Li[4], P.Lee[5], and W.Lou[6]: **Secure Deduplication with Efficient and Reliable Convergent Key Management.** In IEEE Transactions on parallel and Distributed systems, 2013.

[6] J.Stanek[1], A.Sorniotti[2], E.Androulaki[3], and L.Kenel[4]: **A Secure data deduplication scheme for cloud storage.** In Technical Report, 2013.

[7] C.Ng and P.Lee. Revdedup: **A reverse deduplication storage system optimized for reads to latest backups.** In Proc of APSYS, Apr 2013.

[8] Junbeom Hur[1], Dongyoung Koo[2], Youngjoo Shin[3], and Kyungtae Kang[4]: **"Secure Data Deduplication with Dynamic Ownership Management in Cloud Storage",** In IEEE Transactions on Knowledge and Data Engineering, June 2016.

[9] M. Shyamala Devi[1], V. Vimal Khanna[2], A. Naveen Bhalaji[3]: **"Enhanced Dynamic Whole File De-Duplication (DWFD) for Space Optimization in Private Cloud Storage Backup", in** International Journal of Machine Learning and Computing, April 2014

[10] Zhen Yan[1], Wenxiu Ding[2], Robert.H.Deng[3]: **De-duplication on encrypted big data in cloud .** In IEEE transaction on big data Vol.2, No.2, April – June 2016.