

A Data Deduplication Approach for Eliminating Duplicate File Upload over Cloud

Binita Thakkar¹, Dr. Blessy Thankachan²

¹Research Scholar, School of Computer and Systems Sciences

²Associate Professor, Jaipur National University, Jaipur, Rajasthan

ABSTRACT

Cloud computing is an emerging trend. Today cloud is one of the medium used for storing various files and data for easy access. The file stored should be secured. With security of file, ease of storage is also required. With this ease of storage, it is important that many copies of same files be not uploaded over cloud. This will just consume the storage space and duplicating files over cloud. To overcome this issue, data deduplication method is used. In this paper, a simple data deduplication method is proposed which will check whether file is available over cloud environment before uploading it.

Keywords: cloud, security, storage space, data deduplication, hash

1. INTRODUCTION

Data is an asset of importance in everyday life for every user in every field. One of the concern of important data is that they should be stored at a safe place and that they should be easy to access. In today's world, cloud environment is the best place where data can be stored easily and accessed easily. This is because cloud provides various services, applications, servers and storage for its customers[1]. This makes it more efficient, easy to use and reliable. As with security of data over cloud is major concern, managing cloud storage space is important. Many cryptographic algorithms like AES, DES and Blowfish can be used to secure files over cloud.

With the need for easy storage, it is important that same files be not uploaded over cloud. This will duplicate the files and just occupy more storage space over cloud. There is a need for eliminating this duplication over cloud. Data deduplication is one of the strategy to eliminate redundant data and file storage over cloud. That is there should be on single storage of file. Data deduplication will help identify if redundant data is being stored on cloud and eliminating file upload of the same. Data deduplication can be done at the source or at the target.

2. LITERATURE REVIEW OF RELATED WORK

B. Thakkar and B. Thankachan[2], made a comparison of cryptographic algorithms like AES, RSA, IDEA, Blowfish and DES over cloud in a survey. The comparison of cryptographic algorithms were based on the factors of encryption type, key size, block size, number of rounds used, execution time, memory usage and data encryption capacity. They concluded in their survey that symmetric algorithms are more efficient. On the basis on memory usage Blowfish was best and based on encryption time, AES and Blowfish were fast.

S. Chaudhary et al. [3]on the basis of efficiency made a comparative study of cryptographic algorithms like DH, RSA, AES, DES, 3DES and hybrid techniques like AES-RSA, RSA-AES-DS, dual RSA. They concluded in their study that hybrid techniques are average in efficiency but are more secure and symmetric algorithms are more efficient than asymmetric algorithms.

B. Thakkar and B. Thankachan [4], proposed a new technique of merging two rail fence and simple columnar transposition cipher techniques together. Initially using rail fence cipher encryption was done on plain text. The result was encrypted again using simple columnar cipher to obtain the output cipher. This output cipher was decrypted with simple columnar and then by rail fence to get back the plain text. To increase the level of security, double encryption strategy was applied.

N. Pachpor and P. Prasad[5], proposed a new Performance-Oriented Data (POD) deduplication technique and also made comparative analysis with other existing techniques. The new scheme removes duplicate files and duplicate data

in files to increase the performance and later applies encryption and stores data over cloud. The major goal was to increase the performance.

W. Kim and I. Lee[6], have discussed various ways of data deduplications sites and level. Issues regarding secure data deduplications like data encryption, dictionary attacks and poison attacks were put forward. The authors also discussed security techniques for achieving secure deduplication like use of convergent encryption and currently used secure deduplication systems like DupLESS, CloudDup and PerfectDup.

L. Suresh and M. Bharathi[7], made a study of various data deduplication techniques and issues of data deduplication on cloud storage. A new strategy was proposed that allows for fast data transfer from client to server, use of new hash algorithm and removal of duplicate data using block level deduplication.

J. Xiong et al.[8], proposed a new scheme of secure role re-encryption system (SRRS) with authorized deduplication. They made use of convergent key algorithm for privacy of data and role re-encryption algorithm to prevent deduplication. The management centre managed user roles and corresponding role keys by creating role authorized tree. The proposed scheme was efficient and effective.

A. Miri and F. Rashid [9], the authors have proposed a scheme of encoding text data and then compressing. At the client end, the data will first be encoded using Burrows Wheel Transform (BWT) encoding scheme. The encoded data then will be compressed using bzip2. The user will then upload file on cloud. At server end, the user file will be compared with the existing files. If a match is found, the file will be discarded, and if no match is found, server to store on cloud will accept the file. This scheme was secure and efficient as it saved bandwidth and storage.

X. Tang et al. [10], both, user and CSP generate public-private key pairs. The user generates the hash value of file called convergent key, encrypts the file with the convergent key, and produces result C1. Before uploading, user generates query tag and authentication tag of cipher text for integrity checking. User also generates a rekey for original file using hash value and its own private key. CSP stores rekey and C1. CSP re-encrypts C1 with its public key. When user download file, it decrypt file using own secret key and check for its integrity. The objective here was to achieve data confidentiality, convergent key security and data integrity.

H. Aghili[11], made a comparative analysis of symmetric algorithms like DES, 3DES, AES and Blowfish. The analysis proved that Blowfish takes less time for execution. The author used Blowfish for encryption, applied IBM deduplication service on JPG image, and concluded that it is the best-suited algorithm with respect to security and time for processing.

A. Nair et al.[12], proposed three protocols for the process namely File uploading protocol, Integrity auditing protocol and Proof of ownership protocol. In first protocol, client generates hash for chunks of a file using SHA-1 algorithm, checks if hash already present on cloud and if found third protocol would be called. If no hash is found on cloud, client sends chunks to auditor, where auditor creates tags for chunks, encrypts the chunks using AES algorithm and compresses it using Deflate algorithm that is combination of Huffman coding and LZ77 and later sends the tags and chunks to cloud. Second protocol is used for verifying integrity, which is done by the cloud server. Client or auditor asks for verification or establishing proof and server verifies the same. Third protocol is used to identify the ownership of the file where server verifies the client. The authors applied the scheme on file of various sizes and achieved utilization of efficient space.

S. Sathe and N. Dongre[13], applied block level data deduplication strategy. The user registration is done before uploading of the file. The file is then fragmented into fixed size blocks. Before uploading the file, the hash value is generated using SHA-512 algorithm and this hash value is compared with already existing file fragments on cloud. If no match is found or the count of duplicate block is below a predefined threshold value, the file is uploaded on server. After the file is uploaded, it is encrypted using AES algorithm and stored. The file can be downloaded later if proper key attributes are provided. The server can also delete the file only after the owner is verified by using the policy based file assured deletion method.

J. Dave et al.[14], provide a new scheme of deduplication. The scheme allows for upload, download, delete and update of the file. To upload a file, sender generates its hash value using SHA-512 algorithm and requests server for file upload. Server identifies if hash already existed. If not, sender encrypts the file using AES-256 algorithm with randomly generated key and sends both, the encrypted file and hash value to server. To download or delete the file, sender sends hash value of file, server verifies if it is the owner of the file. If yes, it performs the said operation. To update a file, delete operation of old file is done followed by upload operation of new file. The objective of the authors was to achieve confidentiality of data, integrity of data, and security of key and prevent against poison attack.

L. Maragatharajan and L. Prequiet[15], encrypted data using proxy re-encryption method. The user chooses random symmetric key DEK, encrypts DEK with public key of authorized party, and passes encrypted key to CSP. CSP checks

for duplicate data with help of tokens and if found, communicates to authorized party. Scheme allows for uploading of file, checking for duplication, downloading file and deleting the file. This scheme provide high security and privacy.

K. Kim et al.[16] proposed a new client-side secure deduplication to prevent from poison attack using concept of tags and identifying various computational requirements. The study concluded that secure deduplication provides security from erasure attacks and duplicate faking, requires less network bandwidth and provides better server performance.

K. Akhila et al.[17], study of various data deduplication techniques like ClouDedup, DupLESS, HEDup, and SecDup was done. Algorithms were based on convergent key method. Authors stated that a good strategy for enhanced storage optimization technique could be used.

V. Radia and D. Dingh[18] made a study of data deduplication techniques: file level, block level, inline post process, source based and target based. The study concluded that source-based deduplication is best to optimize upload bandwidth and storage space over cloud. Distributed deduplication provides security and confidentiality. Both approaches together provides reliability.

J. Malhotra and J. Bakal[19], identified various challenges of deduplication, comparison of current deduplication techniques were made based on chunking method, metadata processing and throughput. Two Threshold Two Divisor algorithm with Switch Divisor and Two Threshold Two Divisor algorithm were used on different file sizes, analysis was made on time taken for deduplication, and it was identified that Two Threshold Two Divisor algorithm with Switch Divisor takes less time. The authors proposed that throughput could be achieved by using parallelized deduplication process.

3. DEDUPLICATION TYPES

Deduplication is a strategy of removing duplication data or file. There are various types of deduplication.

A. File-level deduplication

File-level deduplication as the name suggest works on file. It allows for comparing a file to be uploaded and stored on cloud with existing files on cloud with file attribute. If file is unique, it will be uploaded else discarded[18]. File-level cannot remove duplicate data in file. [7]

B. Block-level deduplication

Block-level deduplication works on blocks on file. In this, a file is divided into small chunks called as blocks. These blocks are checked for redundancy with stored chunks over cloud. The process takes more time[13]. Block-level can be applied over fixed chunks or variable chunks[18].

4. PROPOSED DEDUPLICATION TECHNIQUE

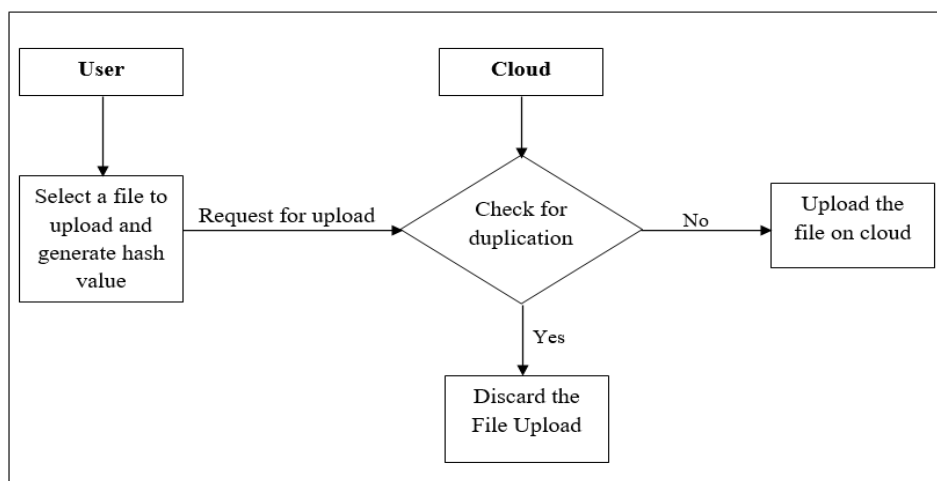


Figure 1.Process of Deduplication

The proposed deduplication technique works at file-level. The file type is restricted to .txt. There are two parties, user and cloud environment. The cloud environment here is a google drive. In this technique, a file to be uploaded will be selected by the user. The hash value of the file will be generated using MD5 algorithm. During upload, the hash value of file to be uploaded will be compared to the hash value of the files present on the cloud. If a match is found, the file will be refrained from upload. If no match found, the selected file will be uploaded successfully.

5. IMPLEMENTATION OF PROPOSED ALGORITHM

The proposed algorithm is implemented using NetBeans IDE in Java. The hardware configuration used is Intel(R) Core(TM) i5-processor with 8GB RAM. In this application, a user selects a file for upload and hash value of the file is generated. The text area shows the list of text files with their hash values already present in the cloud environment. As the user selects to upload, the hash value of the selected file is compared to the hash value of the files present on the cloud environment. If no match found, then the upload of file will be successful as shown in Fig. 2. If the match is found, the file upload will be unsuccessful as shown in Fig. 3.

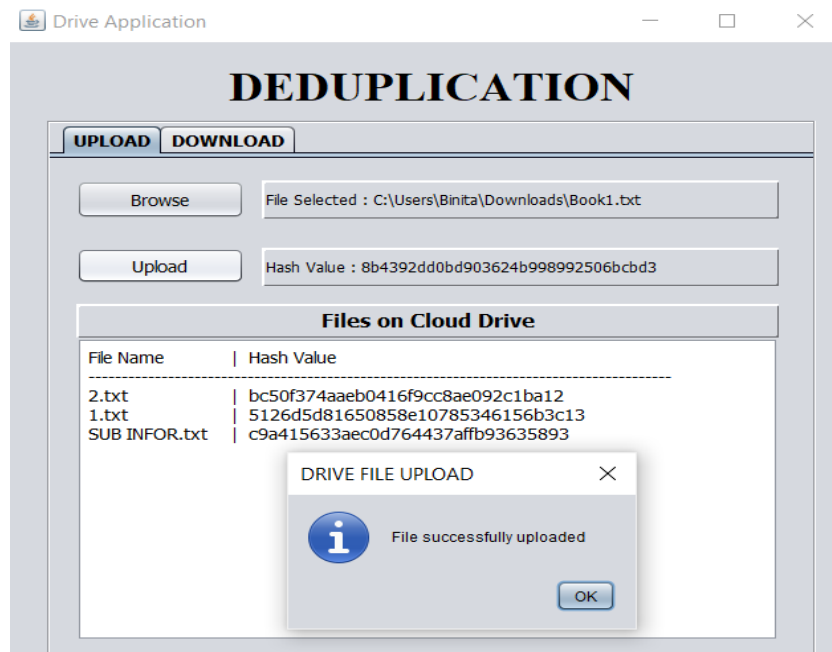


Figure 2. Implementation of Proposed Deduplication Strategy for Successful File Upload

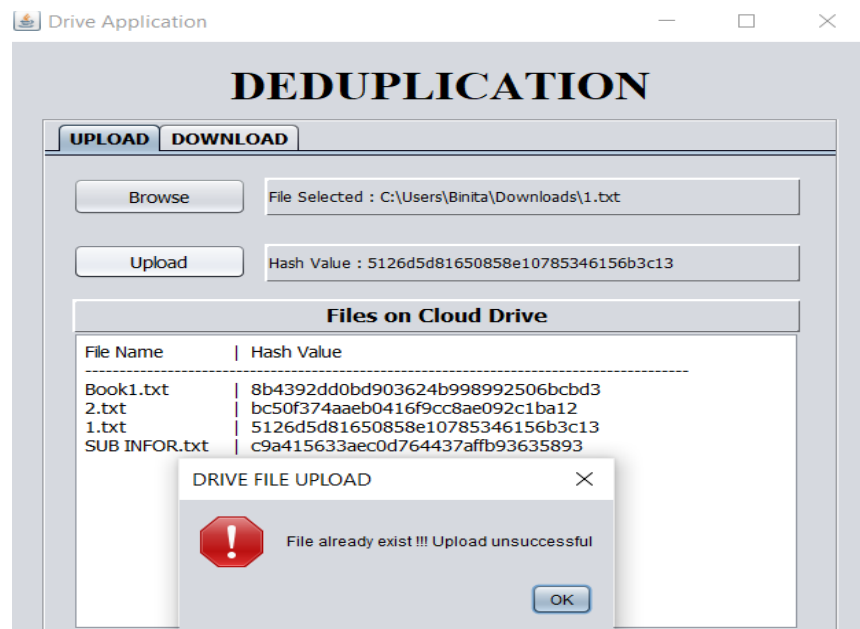


Figure 3. Implementation of Proposed Deduplication Strategy for Unsuccessful File Upload

CONCLUSION

Cloud plays an important role for security files and storage of files. It provides reliable and efficient ways in access of all file. To minimize the duplication of same files over cloud, a simple strategy of deduplication was proposed that calculates the hash value of file to be uploaded and matches with hash value of any file stored over cloud. The

application developed works only text files which is restricted to this research. The overall idea was to minimize duplication over cloud. The hash algorithm used was MD5. For future scope, the same strategy can be applied using other hash algorithms like SHA-1 and SHA-512 over various file types.

REFERENCES

- [1] P. Mell and T. Grance, "The NIST Definition of Cloud Computing (draft)," *NIST Spec. 800.145*, p. 7, 2011.
- [2] B. Thakkar and B. Thankachan, "A Survey for Comparative Analysis of various Cryptographic Algorithms used to Secure Data on Cloud," *Int. J. Eng. Res. Technol.*, vol. V9, no. 08, pp. 753–756, 2020, doi: 10.17577/ijertv9is080328.
- [3] S. Chaudhary, F. Suthar, and N. K. Joshi, "Comparative Study Between Cryptographic and Hybrid Techniques for Implementation of Security in Cloud Computing," pp. 127–135, 2020, doi: 10.1007/978-981-13-8253-6_12.
- [4] B. Thakkar and B. Thankachan, "A Multilevel Approach of Transposition Ciphers for Data Security over Cloud," *GIS Sci. J.*, vol. 8, no. 5, pp. 1732–1738, 2021.
- [5] N. N. Pachpor and P. S. Prasad, "Securing the Data Deduplication to Improve the Performance of Systems in the Cloud Infrastructure," in *Performance Management of Integrated Systems and its Applications in Software Engineering*, Springer Singapore, 2020, pp. 43–58.
- [6] W. Bin Kim and I. Y. Lee, "Overview of Data Deduplication Technology in a Cloud Storage Environment," in *Lecture Notes in Electrical Engineering*, 2020, vol. 536 LNEE, pp. 465–470, doi: 10.1007/978-981-13-9341-9_80.
- [7] L. S. and M. A. Bharathi, "Analysis of Block-Level Data Deduplication on Cloud Storage," *Ambient Commun. Comput. Syst.*, vol. 904, no. July, pp. 401–409, 2019, doi: 10.1007/978-981-13-5934-7.
- [8] J. Xiong, Y. Zhang, S. Tang, X. Liu, and Z. Yao, "Secure Encrypted Data with Authorized Deduplication in Cloud," *IEEE Access*, vol. 7, pp. 75090–75104, 2019, doi: 10.1109/ACCESS.2019.2920998.
- [9] A. Miri and F. Rashid, "Secure Textual Data Deduplication Scheme Based on Data Encoding and Compression," in *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference, IEMCON 2019*, 2019, pp. 207–211, doi: 10.1109/IEMCON.2019.8936222.
- [10] X. Tang, L. Zhou, Y. Huang, and C. C. Chang, "Efficient Cross-User Deduplication of Encrypted Data Through Re-Encryption," in *Proceedings - 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications and 12th IEEE International Conference on Big Data Science and Engineering, Trustcom/BigDataSE 2018*, 2019, pp. 897–904, doi: 10.1109/TrustCom/BigDataSE.2018.00128.
- [11] H. Aghili, "Improving Security Using Blow Fish Algorithm on Deduplication Cloud Storage," *Fundam. Res. Electr. Eng.*, vol. 480, pp. 723–731, 2019, doi: 10.1007/978-981-10-8672-4.
- [12] R. P. J. and P. S. L. K. Arya S. Nair, B. Radhakrishnan, "Secure Data Deduplication and Efficient Storage Utilization in Cloud Servers Using Encryption, Compression and Integrity Auditing," *Int. Conf. Soft Comput. Syst.*, vol. 837, pp. 326–334, 2018, doi: 10.1007/978-981-13-1936-5.
- [13] S. C. Sathe and N. M. Dongre, "Block level based data deduplication and assured deletion in cloud," in *Proceedings of the International Conference on Smart Systems and Inventive Technology, ICSSIT 2018*, 2018, no. Icssit, pp. 406–409, doi: 10.1109/ICSSIT.2018.8748482.
- [14] J. Dave, S. Saharan, P. Faruki, V. Laxmi, and M. S. Gaur, "Secure random encryption for deduplicated storage," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2017, vol. 10717 LNCS, pp. 164–176, doi: 10.1007/978-3-319-72598-7_10.
- [15] M. Maragatharajan and L. Prequiet, "Removal of duplicate data from encrypted cloud storage," in *Proceedings of the 2017 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing, INCOS 2017*, 2017, vol. 2018-Febru, pp. 1–5, doi: 10.1109/ITCOSP.2017.8303134.
- [16] K. Kim, T. Y. Youn, N. S. Jho, and K. Y. Chang, "Client-side deduplication to enhance security and reduce communication costs," *ETRI J.*, vol. 39, no. 1, pp. 116–123, 2017, doi: 10.4218/etrij.17.0116.0039.
- [17] K. Akhila, A. Ganesh, and C. Sunitha, "A Study on Deduplication Techniques over Encrypted Data," in *Procedia Computer Science*, 2016, vol. 87, pp. 38–43, doi: 10.1016/j.procs.2016.05.123.
- [18] V. S. R. and D. K. Singh, "Secure Deduplication Techniques: A Study," *Int. J. Comput. Appl.*, vol. 137, no. 8, pp. 41–43, 2016, doi: 10.5120/ijca2016908874.
- [19] J. Malhotra and J. Bakal, "A survey and comparative study of data deduplication techniques," *2015 Int. Conf. Pervasive Comput. Adv. Commun. Technol. Appl. Soc. ICPC 2015*, vol. 00, no. c, pp. 0–4, 2015, doi: 10.1109/PERVASIVE.2015.7087116.