JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Implementation on Data Duplication Avoidance and Data Integrity Analysis using AES and SHA Algorithm

[#1]Swati O. Rathod, [#2]Dr.V.S.Gulhane, [#3]Dr.A.P.Jadhao
DEPARTMENT OF COMPUTER ENGINEERING

Dr. Rajendra Gode Institute of Technology & Research Amravati.

Abstract : Secure data deduplication can significantly reduce the communication and storage overheads on server side services, and has potential applications in our big data-driven society. Existing data deduplication schemes are generally designed to the mobile flash storage application ensure the efficiency and data availability, but not both conditions. We are also not aware of any existing scheme that achieves accountability, in the sense of reducing duplicate information disclosure. In this system, we investigate proposed architecture, and propose an efficient and privacy-preserving big data deduplication in server side storage. Proposed structure achieves both privacy-preserving (AES encryption algorithm) and data availability. In addition, we take backup and recovery with accountability into consideration to offer better privacy assurances than existing schemes.

Keywords: AES, SHA, Data Privacy, Data Encryption, Deduplication Analysis.

I. INTRODUCTION

Our propose technique provides data security using data encryption in cloud environment. For effective usage of storage space, we provide de-duplication check at file level. We also provide new deduplication constructions supporting authorized duplicate check in cloud architecture, in which the duplicate-check is done at local cloud server. This avoids multiple transaction of file tag over network while checking de duplication. We introduce a relative addressing method in which data will check at entry level when user uploading phases. Nowadays, cloud computing is very important in the Information Technology. Cloud computing enables access to a shared pool of configurable computing resources like servers, storage and applications, etc. The storage services provided to users are though internet. There is chances of cloud disaster like problem in connection, performance, privacy & security, data management. To solve connection problem we can implement offline storage & sync mechanism. To improve performance, load balancing is being an important task for doing operations in cloud and so as de-duplication also. As cloud computing has been growing and many clients all over the world are demanding more services and better results, so load balancing is necessary. Load balancing assure efficient resource utilization to customers on their demand and build up the overall performance of cloud. Every increasing volume of back up data in cloud storage may be a vital challenge so we can use de-duplication mechanism for eliminating the duplicate data. Many algorithms have been developed for allocating client's requests to available remote nodes. The key idea behind this project is to develop a offline store & sync mechanism, dynamic load balancing algorithm based on de-duplication to balance the load across the storage nodes during the expansion of private cloud storage. Data is the key factor in the modern era. Starting from Food Items, Groceries and till the high end satellite and rocket mechanisms, datas play an important role. A separate study is being implemented in Grade Schools and Institutions on data analysis so as to cater to the ever growing era of data science and its applications. More than 2.5*10^18 bytes of information are produced consistently. More than 90% of the above information has been made just in the recent years alone. This number will arrive at 35 ZB in 2025, which has demonstrated to be too bigger than imagination and control. Data storage mediums ranged during the earlier days were selector n tube, Punched card , Punched tapes , Drum memory and then the IBM HDD. Depending on the type of data storage medium, the data storage type also changed and this digital era, we store data in the server with the specialized keys for security concerns. Use of cloud computing is increasing. Cloud optimization is increasing. Effective use of cloud resources is the want of this time, as redundant datas are stored in the cloud again and again. This causes inefficient data storage in cloud and also affects the upload bandwidth. Data security is the major criteria while accessing data from and to the server. We need to reduce the load on the server or cloud storage so as to make it free and perform hassle free data transactions. Removal of duplicate data from the cloud and provide and access to the files will be the prime address issue of this paper implementation. The paper aims to free space, bandwidth and storage in cloud. The suggested approach is to remove the redundant data, where every user has been assigned some access according to the duplication check & each user have their priority token. Hybrid cloud organization is deployed to accomplish the deduplication in cloud. Cloud processing gives versatile, minimal effort and area free administrations over the web. The administrations gave ranges from basic reinforcement administrations to distributed storage foundations. The quick development of information volumes has enormously expanded the interest for systems for space and transmission capacity [1]. Distributed storage administrations like Drop box, Google drive pick a deduplication procedure where

the cloud server stores just a solitary duplicate of repetitive information and makes connects to the duplicate as opposed to putting away genuine duplicates. The security of clients' information turns into another test. Consequently the clients encode the information before redistributing to the cloud. During that process we will be facing a problem of duplication. To solve that we have to perform data deduplication. There is a huge increment in the measure of information produced every day and in 2020 it is normal 44 zettabytes of information will be delivered. But storing and managing these large amounts of data is really a difficult task. Cloud computing offers a new way of service provisions by rearranging the resources over the internet. Cloud storage is the most popular among all the storage providers as cloud storage is the most efficient one. Data duplication occurs when the same data is being shared to the cloud storage by multiple users [2]. Data de-duplication keeps only one physical copy and eliminates multiple data copies. Through this consumption of resources will be reduced and saves the disk space and network bandwidth. Cloud user's upload their information. Security and privacy are the major issues though data deduplication promises lots of benefits. Data needs to be encrypted and store in the cloud which ensures security and user privacy[3].Let us consider three users user1, user2, user3 they are uploading some amount of data through this we will get an idea how deduplication results in.

User1 uploads -----→ a, b, c

User2 uploads -----→d, a, b

User3 uploads -----→d, c, a

The similar kind of data has been uploaded which reduces storage and efficiency of resources [4]. By de-duplication only the files "a, b, c, d" will only be stored into the cloud. This eliminates the repeated data and only stores first unique instance of any data. Whenever the user tries to store the data which is already present in the stored in the cloud it only creates a pointer to the existing one rather than creating redundant data. In block level for each file or chunk a unique hash value will be generated using hashing algorithms like SHA. Whenever user needs to upload a file a hash value will be generated foe that and it will be compared with the existing hash values. If the hash value is not present then the hash value and the file will be stored else it will not store. But in this we will be facing a problem whenever the hash algorithm produces same hash values for different chunks of data then collision occurs. This hash collision leads to data loss [5].

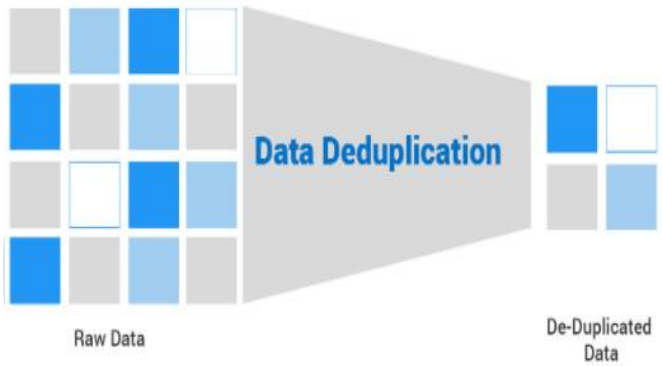


Fig 1. Simple Structure

II. PROPOSED DESIGN

The secure cloud data storage used for further access and the data is transferred via Internet connection and stored in a selected domain, but network and domain not under control of the clients at all. These problems not originate which user take the unauthorized access from the cloud storage is susceptible to security threats from both outside and inside of the cloud, during these process some data will loss from the clients may be hidden by the uncontrolled cloud servers to maintain the reputation. The most important parameter is that for clients are the data security which is less accessed is deliberately deleted by the servers to maintain the cost and space. This system considering the large data size of the outsourced data files uploaded by registered user and the clients' constrained resource capabilities, the first problem is as how can the client efficiently analysis the verifications based on the proposed algorithm even without the local copy of data files. Then we solve this above all problem using the detecting is secure de-duplication file on cloud storage. The remove increased data on cloud server by the cloud server provider stored at remote cloud servers accompany the rapid adoption of cloud services.

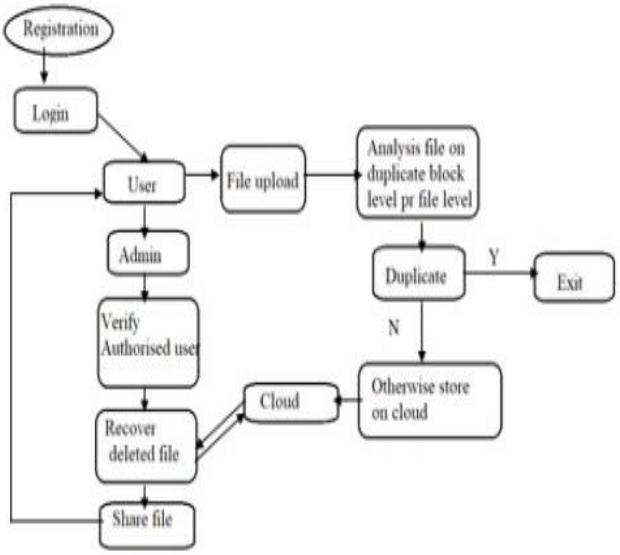


Fig. 2 Flow Chart of Methodology

III. ALGORITHM USED

SHA Algorithm:

Hashing is a sort of algorithm that takes information of any size and changes over it into information of settled size. The principle contrast between hashing and encryption is that a hash is irreversible. Hash capacities are utilized for hashing. A hash work is any capacity that can be utilized to delineate of subjective size to information of settled measure. The yield of the hash work is called hash codes, hash values, hash entireties, or hashes. A hash function should satisfy the following:

a) Two distinct messages ought not have a similar hash esteem. In this manner, the hash capacity ought to be safe against impact.

b) Given a hash esteem, it ought to be troublesome or for all intents and purposes difficult to create the relating message. In this manner, the hash capacity ought to have pre-picture protection.

Generate SHA Sample Key key="da39a3ee5e6b4b0d3255bfef95601890afd80709

AES Algorithm:

For encryption of data:-

START
Step 1- U=Upload (file), he input is consider as Text, is being converted to 128 bit plain text.
Step 2- R= Read (input file),
Step 3- K=Key generation (file)
e.g= key=123456;
Step 4- E=Encrypt(file, key), encode the upcoming file
Step 5- C=Convert (file),
If(encrypt), then file convert plain to cipher text
Split (file1, file2);
Stored (file)
Else, file not encrypted
Step 6- D=Decrypt (file), decode the file
if (decode), then file convert cipher text to plain
Combine (file1, file2);
Else, file not decoded
Step 7- Download file
END

IV. RESULT

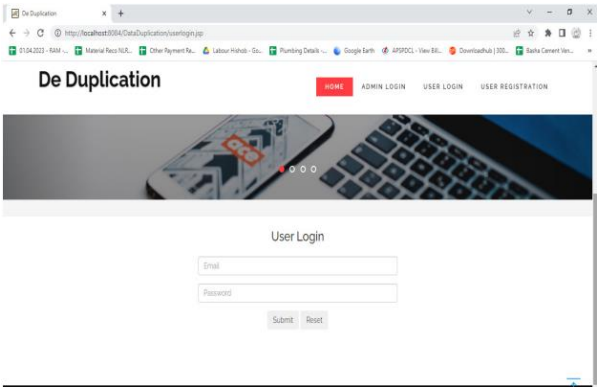


Fig 3. User Login

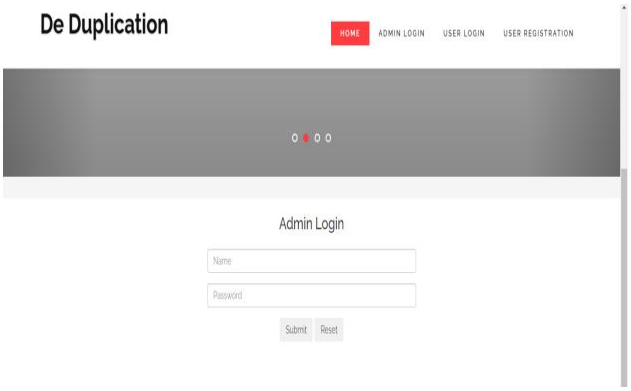
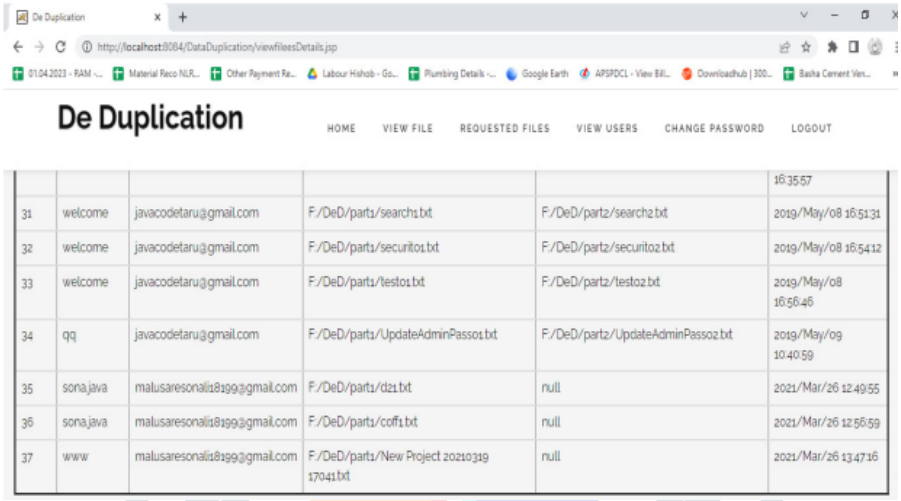


Fig 4. Admin Login



					16:35:57
31	welcome	javacodetaru@gmail.com	F:/DeD/part1/search1.txt	F:/DeD/part2/search2.txt	2019/May/08 16:51:31
32	welcome	javacodetaru@gmail.com	F:/DeD/part1/securito1.txt	F:/DeD/part2/securito2.txt	2019/May/08 16:54:12
33	welcome	javacodetaru@gmail.com	F:/DeD/part1/testo1.txt	F:/DeD/part2/testo2.txt	2019/May/08 16:56:46
34	qq	javacodetaru@gmail.com	F:/DeD/part1/UpdateAdminPasso1.txt	F:/DeD/part2/UpdateAdminPasso2.txt	2019/May/09 10:40:59
35	sonajava	malusaresonali18199@gmail.com	F:/DeD/part1/d21.txt	null	2021/Mar/26 12:49:55
36	sonajava	malusaresonali18199@gmail.com	F:/DeD/part1/coff1.txt	null	2021/Mar/26 12:56:59
37	www	malusaresonali18199@gmail.com	F:/DeD/part1/New Project 20210319 17041.txt	null	2021/Mar/26 13:47:16

Fig 5. File Details

V. ACKNOWLEDGEMENT

I wish to express my profound thanks to all who helped us directly or indirectly in making this paper. Finally, I wish to thank to all our friends and well- wishers who supported us in completing this paper successfully I am heartily thankful to my project guide for his valuable guidance and inspiration. In spite of their busy schedules they devoted their self and took keen and personal interest in giving us constant encouragement and timely suggestion. Without the full support and cheerful encouragement of my guide, the paper would not have been completed on time.

VI. CONCLUSION

We implemented our deduplication systems using the Encryption and Hashing algorithm scheme and demonstrated that it overhead compared to the network transmission over-head in regular upload/download operations.

VII. FUTURE SCOPE

One of the promising future works is to introduce the efficient user revocation mechanism on top of proposed anonymous ABE. Supporting user revocation is an important issue in the real application, and this is one of the greatest challenges in the application of ABE schemes. Making this scheme compatible with existing ABE schemes, support efficient user revocation.

VIII. REFERENCES

[1] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, “Provable data possession at untrusted stores,” in Proceedings of the 14th ACM Conference on Computer and Communications Security, ser. CCS ’07. New York, NY, USA: ACM, 2007, pp. 598– 609.

[2] G. Ateniese, R. Di Pietro, L. V. Mancini, and G. Tsudik, “Scalable and efficient provable data possession,” in Proceedings of the 4th International Conference on Security and Privacy in Communication Netowrks, ser. SecureComm ’08. New York, NY, USA: ACM, 2008, pp. 9:1– 9:10.

[3] C. Erway, A. K¨upc¸ ¨u, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in Proceedings of the 16th ACM Conference on Computer and Communications Security, ser. CCS '09. New York, NY, USA: ACM, 2009, pp. 213– 222.

[4] F. Seb´e, J. Domingo-Ferrer, A. Martinez-Balleste, Y. Deswarte, and J.-J. Quisquater, "Efficient remote data possession checking in critical information infrastructures," IEEE Trans. on Knowl. and Data Eng., vol. 20, no. 8, pp. 1034–1038, 2008.

[5] H. Wang, "Proxy provable data possession in public clouds," IEEE Transactions on Services Computing, vol. 6, no. 4, pp. 551–559, 2013.

[6] Y. Zhu, H. Hu, G.-J. Ahn, and M. Yu, "Cooperative provable data possession for integrity verification in multicloud storage," IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 12, pp. 2231– 2244, 2012.

[7] H. Shacham and B. Waters, "Compact proofs of retrievability," in Proceedings of the 14th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology, ser. ASIACRYPT '08. Springer Berlin Heidelberg, 2008, pp. 90– 107.

[8] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in Computer Security – ESORICS 2009, M. Backes and P. Ning, Eds., vol. 5789. Springer Berlin Heidelberg, 2009, pp. 355–370.

[9] J. Xu and E.-C. Chang, "Towards efficient proofs of retrievability," in Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security, ser. ASIACCS '12. New York, NY, USA: ACM, 2012, pp. 79– 80. Data Duplication Avoidance and Cloud Computing Security Using AES and SHA Algorithm Dr. Rajendra Gode Institude of Technology & Research, Amravati Page 75.

[10] E. Stefanov, M. van Dijk, A. Juels, and A. Oprea, "Iris: A scalable cloud file system with efficient integrity checks," in Proceedings of the 28th Annual Computer Security Applications Conference, ser. ACSAC '12. New York, NY, USA: ACM, 2012, pp. 229– 238.

[11] M. Azraoui, K. Elkhiyaoui, R. Molva, and M. O¨ nen, "Stealthguard: Proofs of retrievability with hidden watchdogs," in Computer Security - ESORICS 2014, ser. Lecture Notes in Computer Science, M. Kutyłowski and J. Vaidya, Eds., vol. 8712. Springer International Publishing, 2014, pp. 239–256.