International Journal of Advanced Research in Engineering and Technology (IJARET)
Volume 11, Issue 10, October 2020, pp. 535-544, Article ID: IJARET_11_10_057
Available online at http://www.iaeme.com/IJARET/issues.asp?JType=IJARET&VType=11&IType=10
ISSN Print: 0976-6480 and ISSN Online: 0976-6499
DOI: 10.34218/IJARET.11.10.2020.057

PIONEER APPROACH DATA DEDUPLICATION TO REMOVE REDUNDANT DATA FROM CLOUD STORAGE

A. Vijayakumar

Research Scholar, PG and Research Department of Computer Science, Govt. Arts and Science College, Lalgudi, Affliated to Bharathidasan University, Trichy, Tamilnadu, India.

Dr. A. Nisha Jebaseeli

Assistant Professor and Head, Department of Computer Science, Govt. Arts and Science College, Lalgudi, Affliated to Bharathidasan University, Trichy, Tamilnadu, India.

ABSTRACT

Cloud is the newest technology in the IT world to provide vast virtual storage, storage and retention of information. Cloud Storage is the leading service in the cloud; it is used by all users at all levels. A challenge is to prevent the store of unnecessary and duplicated data of various users by cloud storage providers. Discharge from storage data replication is a vital task. Cloud data duplicate makes cloud storage management incompatible. This paper recommends that redundant cloud store data be eradicated. The proposed solution uses file and block deduplication of data. For file level deductions, the data were reviewed. It separates blocks and deduplicates blocks if the data file is not duplicated. Deduplication for safe deduplication uses convergent key coding. A separate consumer with convergent encryption with the same key can encrypt the same data file to protect deductibility. Convergent key sharing is more attack prone Convergent key and other sensitive data are safely maintained; the paper introduces the Cloud-based Creating and Managing Keys (KGTMaaaS) service to remove duplicate data. This proposal removes redundant data from cloud storage effectively and saves unwelcome storage allotment, network bandwidth, and allows proper cloud storage.

Key words: Deduplication, Block-level deduplication, Convergent Encryption, File-level deduplication

Cite this Article: A. Vijayakumar and Dr. A. Nisha Jebaseeli, Determining Risk Factors in the Development of Road Traffic Infrastructure in PPP Form in Vietnam, *International Journal of Advanced Research in Engineering and Technology,* 11(10), 2020, pp. 535-544.
http://www.iaeme.com/IJARET/issues.asp?JType=IJARET&VType=11&IType=10

1. INTRODUCTION

The Cloud is an advanced computing technology that offers a service with computational resources. Most notably, storage is the cloud 's core service. Cloud includes a wide variety of service options, such as SaaS , PaaS, and IaaS. Users have now saved some cloud data [1].

The cloud is, however, a public space. Cloud storage can be used by anyone from any cloud service. In this situation, there is a chance that various independent users can upload and store the same data. The data submitted to the cloud is encrypted until it is stored in the cloud by encrypting the data.[2]. Multiple users use their key using traditional encryption and produce various encrypted data for the same original data. The encrypted data of users is processed in a distinct encrypted style in the storage. However, the content of the unique data is the same. In this case, the same information could be stored several times in the cloud and storage is allocated multiple times for the same data. It allows cloud storage to be wasted [3]. If the data is small, it doesn't matter much but the cloud is an infinite provider of storage resources. Many businesses and organizations use it for data storage. The International Data Corporation (IDC) analyzes it in the future and generates a study that indicates that data use around the world will rise to 50 trillion gigabytes by 2025 [4]. To effectively handle the data in the cloud, double storage in the cloud must be minimised. It is suggested that a well known technique called data deduplication be followed to prevent duplication in storage [5].

It supports multiple identical replicas of data in cloud storage. The paper suggests an approach to the deduplication of data using a cloud service to provide all authenticated users exclusively with key and token. The suggested solution uses both block and file-level deduplication methods. The solution suggested is the deduplication of the source stage, meaning that the data is deducted before being downloaded to the cloud. [6]. The deduplication process begins by checking the entire file for replication, along with user credits, by creating a token from the actual files. The token is initially tested to decide if similar data contents are already uploaded or not. If a Key Generation and Token Maintienung token is not present as a KGTMaaaS service, the data is split into small block sizes and the block size is 5 KB per block. For and block a token is created and replication tested. If there is no token in the KGTMaaS, all file blocks will be uploaded. There may not be one or two tokens in the KGTMaaS among the tokens blocks. In this case, the token of file blocks will not be balanced.The storage location connection is provided to users for the remaining blocks of the content of the file.

It is also intended to guarantee the protection of data sent to the cloud by data replication. Convergent encryption can be used to safely deduplicate the data. Convergent encryption takes a coding key. The key is extracted from actual user data, called a convergent encrypton key. A single, identical data can use the same key to produce the same encrypted data with different users, based on current data using a Hash algorithm. The key should be exchanged by various key sharing strategies in order to use the same key for all users[7]. The cloud, however, is an open public environment; it is more likely to attack if the key is shared. The main generation of the user will also be exposed to the key. The paper proposes and implements a cloud service for producing and preserving the key in the cloud in order to avoid that scenario.Users need to upload the data, check for file level replication initially and not re-generate the key if the data file is already in the cloud. Only for the first time a key for a file or block is created. A single key is connected to each file or block of cloud storage. In the approach proposed, the following values are produced or retained, i.e., token of the FiTKN or FiBjTKN file, convergent key generator FiBjKCEK, converging key FiBjCEK encryption, encrypted data EDFiBj in the proposed approach for single block or file.

This paper aims to achieve a groundbreaking approach to solve the problem of data storage in the cloud world. The approach proposed implements the deduplication of file and

block data. This article introduces a cloud service as a KGTMaaS service to ensure the process of deduplication.

2. RELATED WORKS

Data deduplication is used to securely handle the cloud by preventing the same data replication. Data replication will be recognized using a mechanism for deduplication. Some deduplication methods are available in this area. This segment contains some of the associated cloud storage deduplication areas.

Periasamy et al. [8] have shown improved safe information deduplication of data recognition and replication avoidance approaches in order to enhance the discovery of cloud data coding file and data level deduplication. All files of cloud users have a key which indicates that this current approach is the master clave for encryption. The main is externalized in the cloud. This system aims to minimize the overheads incurred by joint identification and investigation procedures. The data unit to be processed on the cloud is found in this way. This approach decreases storage efficiency a little..

The result was concentrated on Gayathri Devi et al.[9] to resolve the difficulties produced by large data sections. The suggestion of the HAR solution, which means history conscious rewriting, emphasizes the concept of decreasing destruction. The solution proposed takes the previous backup storage material, which has made thin containers available and reduced them to the minimum. The data is split into pieces and every piece has a hazh code that is created with a hash code algorithm like MD5. The I d is used for the chunks of a particular file. DES will be used to encrypt, decrypt and generate a top secret file, when the user is set up by the data owner. DES is used to decrypt data. The amount of duplicate data determines how much the renovation output has developed.

Chandra et al . [10] suggested the deductible technique POD vs IDedup. The POD extends the deduplication method to achieve results. It uses Deduplication to improve the performance of simple storage systems in storage. This system utilizes the I / O path to define and protect the replicated storage query while storing data. By considering the selective deduplication approach, POD avoids the problem of data division. The data on the serious I / O route is used to deduplicate the I / O replication. In the meantime, cache management works in the proposed POD in order to optimize data reading on the I / O path, as well as to boost efficiency and save space.

Himshai Kamboj[11] released an application called the Dedup App, which runs on the deduplication service hardware. It has a user interface as an HDFS storage framework for rapid indexing as the front and back end. In the deduplication storage method, the authors discuss two problems. First, it is important to recognize in the storage duplicate data files. Second, the correct authentication scheme is maintained to establish data ownership. The fingerprints Method is used for the encryption and decryption of a file that is submitted for replication control and for AES 256. The system architecture proposed consists of several modules, each with a specific objective to recognize and protect duplicate data.

The DAC deduplication assisted cloud system has been suggested by Suzhen Wua etc. [12]. The paper considers the data as various bits. The proposed DAC seeks to eliminate redundant data file chunks. The proposed DAC distributes the chunks of the file with data deduplication to various storage providers to store file chunks. The pattern is used for all file chunks. It is the file deduplication reference character. Data are stored in the storage system depending on the data of the chunks of the file. The remaining sections are managed by the DAC architecture error code scheme. This is a deduplication of the goal stage, which makes the deduplication more management problems.

Frederik Armknecht et al. [13] suggested a method to certify user data with a pattern of deduplication in cloud storage known as ClearBox. The ClearBox offer enables providers to recover the duplicate files through a fine-grained access control system. It advocates data security and even protests against intruders. The ClearBox has been introduced and assessed. Its result shows that for number users and number of files, the system scales continuously. It is one of the right systems for the verification and storage of data.

A protocol for safe data replication was suggested by Ashish Agarwala et al . [14]. This Protocol is used to secure the duplicate fake attack, and also to recognise the erasing attacks. The Protocol is a Double Integrity Convergent Encryption DICE. With the server and client side data integrity control process , the proposed DICE is enabled. The proposed DICE provides that a tag for honesty checks would be produced and downloaded to the server. By sending the tag on the server for integrity monitoring, the bandwidth for storage is reduced. When the user downloads the file from the server, the proposed DICE search for data integrity. By sending the tag instead of the actual data the device saves bandwidth.

Key management (KMS), focused on pairing cryptography, was introduced by Hyunsoo Kwon et al [15]. The authors believed that the scheme proposed was a safe and scalable deduction scheme. The KMS proposed divides the convergent encryption key into three keys and only permitted data owners can access it. The data owner shares the keys with a masking Id secretly. In the absence of the masked Id of key shares an opponent attempts to obtain key parts by tapping into the channels, the coils get the actual key. The approach proposed minimizes the user's workload in managing the key. This situation minimizes calculation time and overhead contact. Only by knowing the masked Id of key shares can the approved users collect the data. This approach raises problems which should be shared in the network several times. This makes adversaries constantly watch the channel for main shares and masked ID.

2.1. Problem Definition

Cloud storage provides space for storing multiple quantities of daily data. The assignment of storage and other management processes should be effective. Does cloud storage challenge the way duplicate data can be eliminated? yes. Cloud storage efficiency is impaired by the storage of duplicate data copies — the following reasons in the cloud to prevent data replication in the storage process .

• Unable to eradicate duplicate copies of data in a cloud, conventional cryptosystem.

• For the same file content users with different keys generate different encoded data.

• Alone deduplication at file level is sufficient to effectively deduplicate data.

• Clients have more pressure on managing the client-side convergent encryption key; it is harder for users.

• All users of the same data must be the same convergent encryption key, but it is vulnerable for security attack to share this key with all the users of data owners.

3. METHODLOGY

The research work proposed is focused to find that the file contents are either duplicated before they are sent to the cloud storage or not. The paper suggests a two-tier method for duplicate verification. The entire file is first tested as a file level deduplication for replication. Second, there are multiple blocks of files divided. The block is 5 KB, and deductions of block level are checked for all blocks. For each block file size, this research work takes default 5 KB into consideration.

For the first time, the suggested approach tests the entire file for a user. If you don't repeat the entire file, the file will split up into blocks by size. Both blocks are tested for cloud storage replication. If no block is used, then the blocks are encrypted and submitted to the cloud storage using convergent encryption. KGTMaaaS maintains tables metadata for each cloud-saved file.

When another user who has the same file verifies the file level, no further blocks verification is performed if the whole file is found to be replicated. Moreover, the current user is assigned logical links of all file blocks. This is not the whole file in the cloud. For the current user, the KGTMaaaS allows a new entry with the same data, which are initially entered on the cloud in the same file content. Cloud storage therefore includes one copy of the data. It reduces the undesirable memory allocation and saves bandwidth.

4. PROPOSED WORK SCENARIOS

The proposed deduplication of data operates with the data of the user to find the content's similarity in the database. Deduplication is carried out in three scenarios according to the proposed job. As a deduplication, the proposed work is given to the customer. Cloud service provider (CSPDUP) is managing and regulating it. The method for uploading and downloading (DWD) data is registered (REGU) to the users (U) by the method. The following parts define the scenarios for deduplication.

Figure 1 shows the proposed method of uploading a file-level data deduplication to the cloud storage. In figure 1. duplication means the authentication process of the user's data file (UDFi) before uploading to the cloud. Users are initially signed into the deduplication scheme. Registered users are entitled to upload the data. A token is produced for the complete file (FiTKN) when data is send to the cloud. The FiTKN is tested for duplication with KGTMaaaS. FiTKN is not included in the KGTMaaas and DFU is a partition in the FiB1, FiF2, and FiBn blocks. For each FiBj, FiB1TKN, FiB2TKN, FiB3TKN, and up to FiBnTKN are produced. For replication in KGTMaaS each block token will be checked. If the block tokens do not match KGTMaaaS, it means that no file block is stored on the cloud. Each block is eventually encrypted and sent to the cloud.

Deduplication- Case 1: The data file blocks are first uploaded; no such file is kept in storage

Symmetrical convergence encryption to encrypt the file data is used and a key to encryption from the user's data is produced. Below, the notice explains file uploading and downloading encryption and decryption respectively.

$$E_D F_i B_j \rightarrow Enc(F_i B_j, F_i B_j CE_K) \quad \ldots \quad (1)$$

$$i \in [1,2,\ldots n] ; j \in [1,2,\ldots m]$$

$$F_i B_j \rightarrow Dec(E_D F_i B_j, F_i B_j CE_K) \quad \ldots \quad (2)$$

$$i \in [1,2,\ldots n] ; j \in [1,2,\ldots m]$$

For a file block, FiBjCEK is a convergent encoding key. The digestion of FiBj generates it. The FiBjCEK generation uses a FiBjKCEK key to generate FiBjCEK. The KGTMaaaS produces the FiBjKCEK, if the token FiBjTKN block does not fit KGTMaaaS tokens.

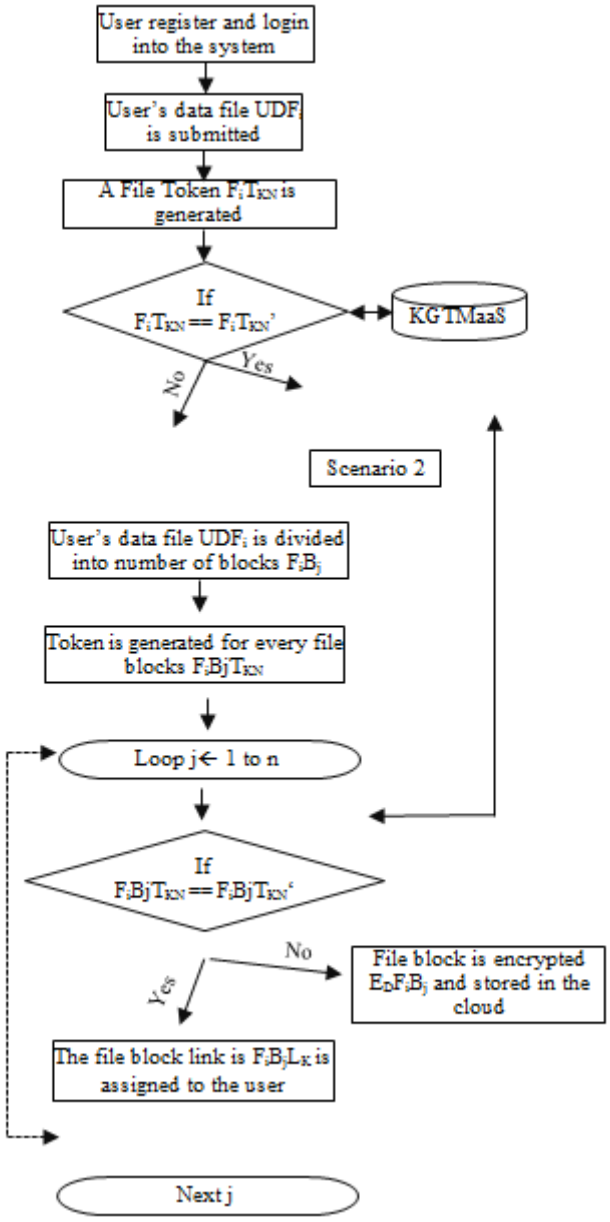


Figure 1 Deduplication Scenario-1: Steps in deduplicating a file when upload at the first time to cloud storage

Deduplication Case-2: For the next time, the whole file is already stored in the store, the block of a related data file will upload

In that case, a registered user can upload a file. A FiTKN token is made, and checked by KGTMaaaS. The FiTKN has tested the cloud storage for replication of FiTKN. This means that the whole file is already uploaded by any other users because it is already in the storage. For block-level file material, no further research is needed.

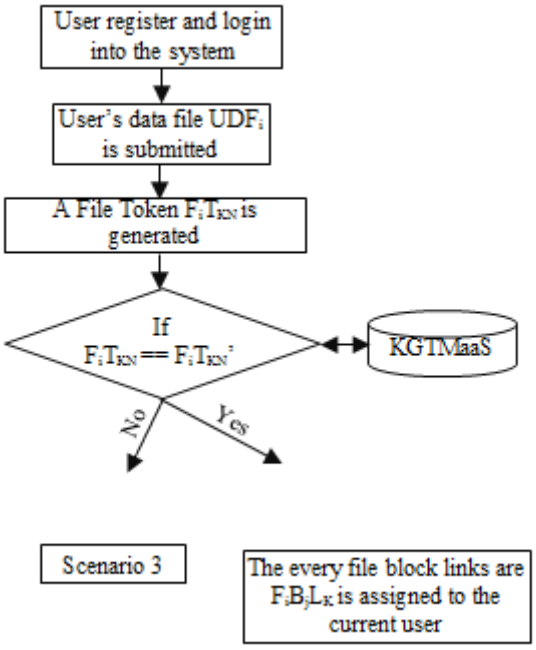


Figure 2 Deduplication Scenario-2: Steps in deduplicating a file when upload at the second time to cloud storage by different user

If you have a file already in cloud storage, it is registered FITKN in the KGTMaaaS. The file is not re-uploaded in this case. The logical relation to the current user of each file block is instead registered. In this case no data will be uploaded to the cloud. The process you mention when you reload the same file is represented in Figure 2. But the file is not sent to the cloud.

Deduplication Case-3: 1 to 2 blocks are not similar; partial file content has been saved in stock; file content has been partially changed

In this example, the file is already saved by a user in the cloud Storage in certain situations. The same file, but it is changed by some other user or can be sent to the cloud by the same user. In this case, the checking of file-level tokens alone identifies that this file does not exist. The truth is, however, that part of a file's content is already stored in cloud storage. Therefore, the same partial content does not have to be stored again. The method suggested in this case defines the modified partial file contents by means of block deductions. A file is split into blocks of numbers as specified

A size block is 5 KB. If you notice that the file level token is new, the file is subdivided into block size (FiB1, FiB2, FiB3 and FiBn). Each block (FiB1TKN, FiB2TKN, FiB3TKN, and FiBnTKN) is a token block file. A token already stored in the KGTMaaaS is verifying every block token. If a token file block is matched, the particular block of data contents is known in the cloud storage already. The matched token content for a file block is not then uploaded into the cloud. If the block token of a file does not suit KGTMaaaS tokens at the same time. The portion of the block content of the file is not stored in the cloud. The solution suggested uploads the part of the information not already stored in cloud storage. Consider for example a file containing content,

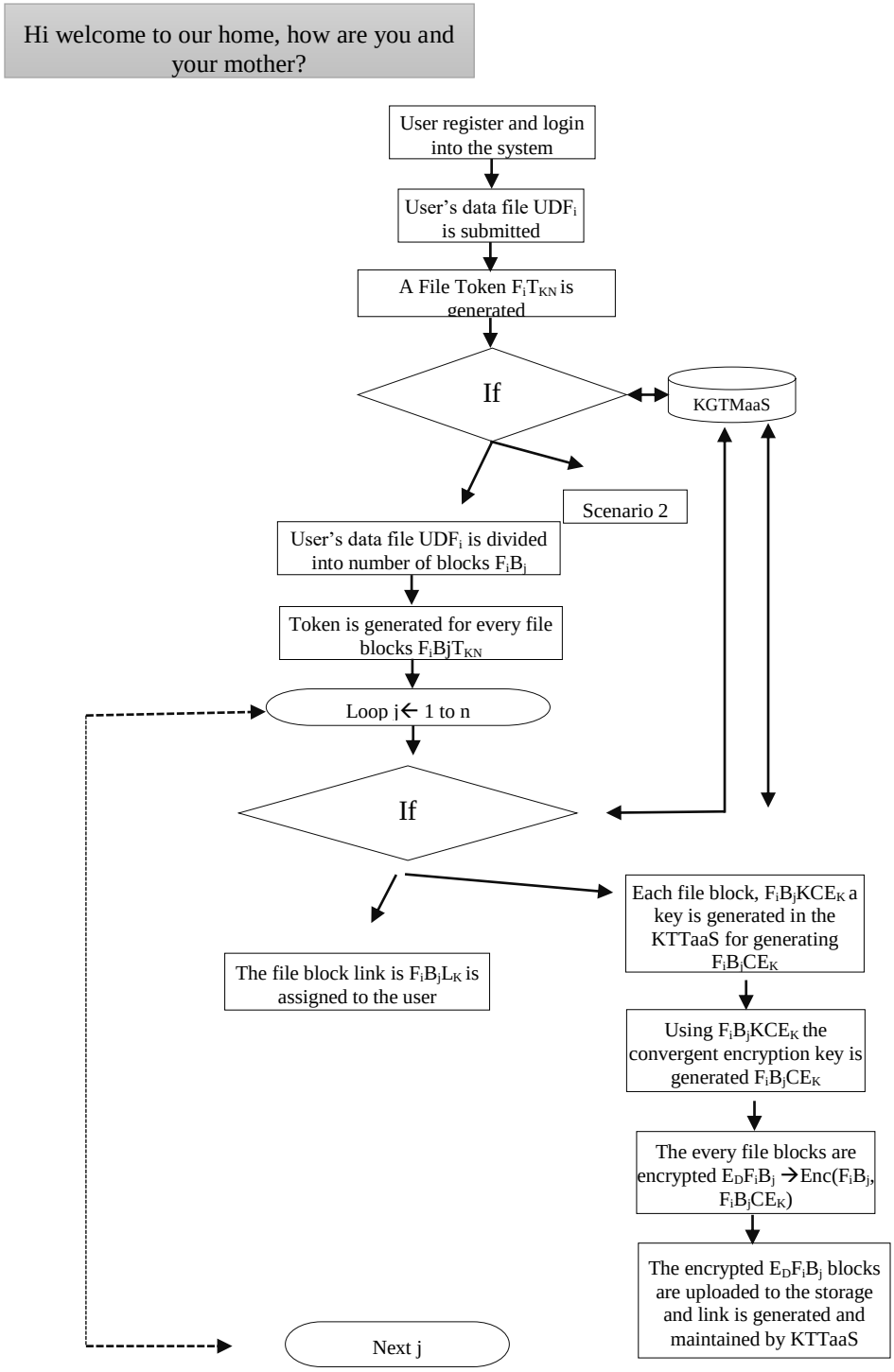
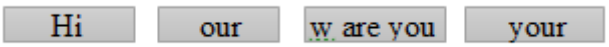


Figure 3 Deduplication Scenario-3: Steps in deduplicating a portion of the file

This content is already uploaded in the cloud storage, considered. Consider that F1TKN = "hwtohayaym" is the token of the file. The file block is split into 13 blocks for comprehension. The content of the file blocks is,



Token of each file block is,

$F_1B_1T_{KN}$←Hwt;

$F_1B_2T_{KN}$←ohh;

$F_1B_3T_{KN}$←waya;

$F_1B_4T_{KN}$←ym;

The same file content is now uploaded to cloud storage with the same modifications from another user. The file material, for example,

Hi welcome to our home, how are you

The deduplication of the file-level files is an F2TKN← Hwtohhayayf file token for the above file. File-level deduplication means that a file-level token does not fit the previous file token because it's new file-level token. The file is broken down into several blocks, as below. Based on the proposed approach.

Hi | our | w are you | your

The file block tokens are,

$F_1B_1T_{KN}$←Hwt;

$F_1B_2T_{KN}$←ohh;

$F_1B_3T_{KN}$←waya;

$F_2B_4T_{KN}$←yf;

The token results are shown, the first three toks are similar and the last not matched. The token results Therefore, in the first three blocks of data, the planned solution is not saved and considers that a last blocks of data are stored in cloud storage. This file is seen as another file in the proposed solution. However, it takes content from the file already stored, and the updated file content for the last block. Figure 3 indicates the partially updated data material deduplication.

5. CONCLUSION

Data storage replication creates various operational challenges for cloud providers and cloud users. The cloud storage is also in critical conditions of management. This paper's proposed solution allows both block and file-level deduplication to be used in a groundbreaking way. Incorporating deduplication of blocks and file levels makes identification of duplicate data more effective. It's a deduplication at the source stage. Until it is loaded into the cloud , data is deduplicated. In the paper a cloud service was introduced called KGTMaaAS Key Generation and Token Maintenance. The approach proposed highly protects the replication of cloud-specified data.

REFERENCES

[1] B. Mahalakshmi and G. Suseendran, An Analysis of Cloud Computing Issues on Data Integrity, Privacy and Its Current Solutions, Data Management, Analytics and Innovation, Advances in Intelligent Systems and Computing, Springer Nature Singapore, 2019, pp. 467-481.

[2] MS.C.Kamatchi, R.Pooja, S.Serishma, R.Vanitha, Data Deduplication Security with Dynamic Ownership Management, International Journal of Computer Science Trends and Technology, Volume 5, Issue 2, Mar-Apr 2017, pp.252-256.

[3] Uma G, Jayasimman L 2018 Survey on Data Deduplication Techniques used for Efficient Management of Cloud Storage International Journal of Scientific Research in Computer Science Applications and Management Studies 7 2018 pp 163-170.

[4] D. T. Meyer and W. J. Bolosky, "A Study of Practical Deduplication," Trans Storage, vol. 7, no. 4, pp. 14:1–14:20, 2012.

[5] Nishant N. Pachpor and Prakash S. Prasad Securing the Data Deduplication to Improve the Performance of Systems in the Cloud Infrastructure, Performance Management of Integrated Systems and its Applications in Software Engineering, Asset Analytics, springer, 2020, pp.43-58

[6] Yukun Zhou, Dan Feng, Wen Xia, Min Fu, and Yu Xiao, DARM: A Deduplication-Aware Redundancy Management Approach for Reliable-Enhanced Storage Systems, Springer Nature Switzerland, ICA3PP, LNCS 11335, 2018, pp. 445–461.

[7] Jin Li, Xiaofeng Chen, Mingqiang Li, Jingwei Li, Patrick P.C. Lee, and Wenjing Lou, Secure Deduplication with Efficient and Reliable Convergent Key Management, IEEE Transactions on Parallel and Distributed Systems, 2014, pp 1615-1625.

[8] J. K. Periasamy and B. Latha, An enhanced secure content deduplication identification and prevention (ESCDIP) algorithm in a cloud environment, January 2019Springer-Verlag London Ltd., 2019, PP. 1-10.

[9] K. Gayathri Devi, S. Raksha, and Kavitha Sooda, Enhancing Restore Speed of In-line Deduplication Cloud-Based Backup, Systems by Minimizing Fragmentation, Smart Intelligent Computing, and Applications, Springer Nature Singapore Pte Ltd, 2020, pp. 9-21.

[10] Hemanth Chandra N, Sahana D. Gowda, Secure and Efficient Client and Server Side Data Deduplication to Reduce Storage in Remote Cloud Computing Systems, International e-Journal For Technology And Research, Volume 1, Issue 5, May 2017, pp 1-8.

[11] Himshai Kamboj, Bharati Sinha, DEDUP: Deduplication system for Encrypted Data in Cloud, IEEE International Conference on Computing, Communication and Automation, 2017, pp. 795-800.

[12] Suzhen Wua, Kuan-Ching Li c, Bo Maob, and Minghong Liao, DAC: Improving storage availability with Deduplication-Assisted Cloud-of-Clouds, Elsevier Future Generation Computer Systems, Volume 74, September 2017, pp. 190-198.

[13] Frederik Armknecht, Jens-Matthias Bohli, Ghassan O. Karame, and Franck Youssef, Transparent Data Deduplication in the Cloud, ACM Conference on Computer and Communications Security, ISBN: 978-1-4503-3832-5, 2015, pp. 886-900

[14] Ashish Agarwala, Priyanka Singh, and Pradeep K. Atrey 2017 DICE: A Dual Integrity Convergent Encryption Protocol for Client-Side Secure Data Deduplication, IEEE International Conference on Systems, Man, and Cybernetics (SMC) pp 2176-2181.

[15] Hyunsoo Kwon, Changhee Hahn, Dongyoung Koo, and Junbeom Hur, Scalable and Reliable Key Management for Secure Deduplication in Cloud Storage, IEEE International Conference on Cloud Computing, 2017, pp.391-398.