ISSN: xxxx-xxxx (ONLINE)
DOI:

SECURE CONTENT DE-DUPLICATION UTILIZING EFFICIENT CONTENT DISCOVERY AND PRESERVING DE-DUPLICATION (ECDPD)

A. Mathew Branesh[1], S. Johnson[2] and F. Antony Xavier Bronson[3]
[1,2]Faculty of Management Studies, Dr. M.G.R. Educational and Research Institute, India
[3]Department of Biotechnology, Dr. M.G.R. Educational and Research Institute, India

Abstract

Cloud computing is a promising technology which is utilizing huge amount of data file storage with security. However, the content owner does not control data access for unauthorized clients and does not control data storage and usage of data. Some previous approaches data access control to help data de-duplication concurrently for cloud storage system. Current industrial de-duplication solutions can't handle encrypted data for cloud storage. The deduplication is vulnerable to brute-force attacks and can't flexibly support data access control. Data de-duplication is one of important data compression techniques for eliminating duplicate copies of repeating data, and has been widely used in cloud storage to reduce the amount of storage space and save bandwidth. To overcome above issues, an efficient content discovery and preserving De-duplication (ECDPD) algorithm is proposed for detecting client file level and block level of de-duplication in storing data files in the cloud storage system and support secure data access control at the similar time. The proposed system is protecting the confidentiality of sensitive data while supporting deduplication before data outsourcing. The system protects data security and attempt to formally address the problem of authorized data de-duplication. Based on Experimental evaluations, proposed ECDPD method reduces 3.802 milliseconds of DUT (Data Uploading Time) and 3.318 milliseconds of DDT (Data Downloading Time) compared than existing approaches.

Keywords:

Efficient Content Discovery and Preserving De-Duplication (ECDPD), Cloud Storage System, Data De-Duplication, Data Uploading Time, Data Downloading Time

1. INTRODUCTION

Data De-duplication is one of the specialized data compression mechanisms for eradicating the repeated data. It is utilized to enhance the storage and network utilization for reducing the number of bytes during data transfers in the network. In the De-duplication process, the transferred data file is split into number of blocks depends on dynamically specified bytes by the client and then matchless blocks of data file are discovered and stored during the examination process. In the examination, other blocks were contrasted to the previously stored data copy and whenever a match occurs, the matched block was substituted with a reference point to the already stored block. In the given data file, the similar byte of the blocks may occur various times, thus the amount of blocks storage and time can be reduced using this technique. The match frequency is calculated based on block size.

Various cloud services provide large volumes of file storage that maintain and control data file, which can contain files, texts, videos, photos, and personal health records, etc. however, an industrial data de-duplication resolutions does not handle the encrypted data for cloud storage system. The previous data de-duplication resolutions are susceptible to brute-force assaults in storage system. A previous data de-duplication solution does not flexibly to help data access control and revocation for authorized clients. An existing data de-duplication does not provide security for data in cloud storage system. The similar or different cloud users could store duplicated file data at the cloud storage server. While the cloud storage space is high, it utilized to waste networking assets, consumes excess power, and makes difficult data management

To overcome above issues, an efficient content discovery and preserving De-duplication (ECDPD) algorithm is proposed for detecting client file level and block level of de-duplication in storing data files in the cloud storage system and support secure data access control at the similar time. The proposed system is protecting the confidentiality of sensitive data while supporting de-duplication before data outsourcing. The system protects data security and attempt to formally address the problem of authorized data de-duplication. The method preventing the illegal utilization of data files accessing and make duplicate file data on cloud server to encrypt the data file before stored on cloud storage server. The proposed system is identifying the unique data block which is stored in the cloud. The paper contribution is following as:

- To design an efficient content discovery and preserving De-duplication (ECDPD) algorithm which detecting client file level and block level of de-duplication in storing data files in the cloud storage system
- To support secure data access control for authorized clients at the similar time.
- To prevent the illegal utilization of data files accessing and make duplicate file data on cloud storage server
- To reduce the overheads associated with the interactive duplication discovery and processing of data files
- To reduce the Data Uploading Time (DUT) in a sec and Data Downloading Time (DDT) in milliseconds contrast than previous methodologies.

The rest of the paper is organized as section 2 addresses the various literature that closest to proposed methodology. Section 3 explains the proposed methodology, implementation steps of proposed techniques with their features. Section 4 explores the implemented result with comparative analysis. Section 5 summarizes the overall work with the future outcome.

2. RELATED WORKS

Shobana et al. [1] developed a Cloud Computing Secure Framework (CCSF) which comprised four stages such as uniqueness management, interruption discovery, and prevention method, data de-duplication, and secure data cloud storage.

Kaaniche et al. [2] designed an OpenStack Swift that was a client-side de-duplication scheme for securely storing and allocation of outsourced information through the public cloud storage framework. Stanek et al. [3] evaluated an encryption scheme that guaranteed semantic protection stage for unpopular details offered lenient protection and enhanced storage ability and bandwidth benefits on relevant information. Akhila et al. [4] discussed a Data De-duplication approach altered as easy data storage optimization mechanism in secondary then generally adopted in primary storage with larger data storage areas like cloud storage region.

Harish et al. [5] developed a convergent encryption mechanism that utilized to overcome the data storage problems and to provide numerous protection mechanisms to particular data through verifying secret key. Thakar et al. [6] designed a hybrid cloud method that addressed a de-duplication occurring and supported authorized duplicate copy to validate in the hybrid cloud infrastructure. Shieh et al. [7] illustrated the concepts on de-duplication techniques and available de-duplication mechanisms such as pros and cons. It examined de-duplication mechanisms on the parameters like effectiveness, scalability, throughput, bandwidth ability and price.

Puzio et al. [8] developed a PerfectDedup for secure data de-duplication that taken into account the popularity of the data segments and leveraged the properties of entire hashing development to guaranteed de-duplication. However, it fails to accomplish block level de-duplication. Priyadharsini et al. [9] examined various methods of de-duplication to overcome the challenges. The de-duplication scheme diminished data storage demands in cloud computing environments with a significant quantity of VM floppy disk administrators. It fails to maintain data storage client demands for a huge amount of VM disk supervisors.

Devi et al. [10] illustrated dissimilar methods that have been utilized data de-duplication in cloud storage framework. It generated by compressing the data storage space needs for saving similar data. Harnik et al. [11] discussed an easy approach that permitted cross-client de-duplication while it reduced the thread of content leakage. It illustrated how de-duplication utilized as a side channel and disclose the information of the other clients. Bharat et al. [12] discussed an approved content de-duplication and it protected information protection through the procedure of comprising disparity privileges of the customers in the duplicate validation in cloud storage systems.

3. PROPOSED SYSTEM

The section represents the proposed scheme, Implementation pre-processing steps, and implemented methodology details. The proposed method is detecting client file level and block level of de-duplication in storing data files in the cloud storage system. It utilized to eradicate duplicate photocopy of repeated data file. The Fig.1 demonstrates the working model of the proposed method with implementation processing steps and mathematical estimation details. The pre-processing implementation steps are described below in details:

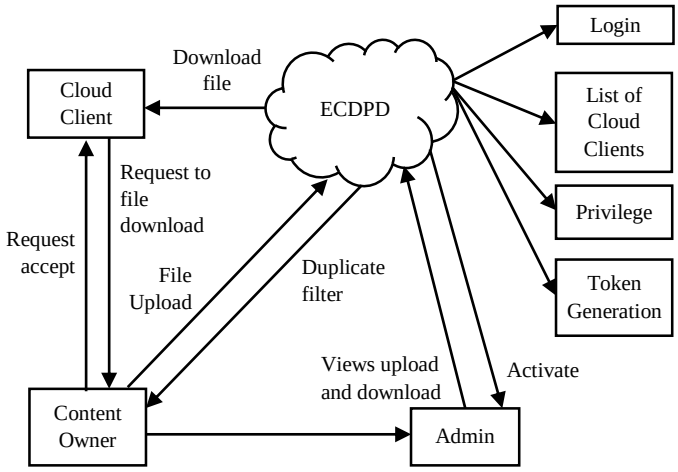

Fig.1. Working Model of the Proposed ECDPD Algorithm

3.1 CLOUD CLIENTS AUTHENTICATION

In the module, the Content owner makes utilization of cloud assets to store, retrieve and share data file with various cloud clients. A content owner can be either an individual or an enterprise. Content owner can validate the uploading data file and it can neglect or upload the data file. Content owner can view the de-duplicate file depend on cloud client can delete the redundant information. The data file can upload to the Cloud storage system from the content owner after that repeated file content upload is filtering de-duplication. An ECDPD algorithm applying the content owner side which utilizing filtering de-duplication.

3.2 CLOUD CLIENT

The cloud client can register their own details and get the secret key for authentication and the cloud client can download the content owner's uploaded data files. The cloud clients are able to access the file stored in the cloud storage and depends on their access rights which are approvals granted by the content owner, like access rights store the data file in the cloud storage system.

3.3 DUPLICATE CHECKING

The content owner uploaded files can be stored in cloud storage systems. Data deduplication is a specialized data file compression mechanism for eliminating duplicate photocopy of replicating data file. Interrelated synonymous terms are data compression and data storage system. The content owner demands to perform the client file level and block level deduplication before uploading data file. The data file is split into blocks, if no duplication is found and performs block level deduplication framework.

3.4 DATA DISTRIBUTION

Sharing and recovering is utilized in data distribution module. The data distribution is utilized for split and shared secret data. With sufficient data distribution is extracting and recovering the secret with the help of recovering method. Data distribution method partitions the secret data file into similar size of blocks which makes equal size of random blocks and transmits into easy language.

3.5 CLOUD CLIENT REVOCATION

Cloud client revocation is performed by the administrator through accessible of cloud client's revocation list and it based on which content owner can encrypt their data files and assurance the confidentiality against the revoked cloud clients. The administrator can alter the cloud client revocation record each day still no one has being revoked in the day.

3.6 EFFICIENT CONTENT DISCOVERY AND PRESERVING DE-DUPLICATION (ECDPD) ALGORITHM

Efficient content discovery and preserving De-duplication (ECDPD) algorithm is upgrading client file level and block level de-duplication with dependability and distributing data documents with safely for cloud client's storage frameworks. A content owner gets a master key from every original data. A duplicate and encodes the information duplicate with the master key. Furthermore, the client additionally infers a token for the information duplicate, such that the token will be utilized to identify duplicate copies. Here, we accept that the token accuracy property holds, i.e. if two information duplicates are the same, at that point their tokens are the same. To recognize duplicate copies, the client initially sends the token to the server side to verify if the indistinguishable duplicate has been already stored. Note that both the master key and the token are freely determined and the token cannot be utilized to conclude the master key and negotiation information privacy. Both the encoded information duplicate copy and its comparing token will be stored on the server side. Formally, an ECDPD plan can be characterized with four primitive capacities.

- *KeyGen*(*D*) → *K* - key generation method maps a data D to key K.
- *Enc*(*K*,*D*) → *C* - symmetric encryption method receives the input of both data copy *D* and master key *K*, then gives output cipher text *C*.
- *Decce*(*K*,*C*) → *D* - decrypting method receives the input of the key *K* and cipher text *C*, then provides the output of the original data file copy *D*.
- *TokGen*(*D*) → *T*(*D*) - tokens generating approach maps original data file copy *M* and provides output token *T*(*D*).

The pseudo code of ECDPD algorithm is explained below in details:

Pseudo Code for Proposed Algorithm

Input: Any document file

Output: visualize data uploading time, and data downloading time

Procedure

Start

Content owner authentication

Browse data file to upload the cloud storage server

Apply ECDPD algorithm

Encrypt the original content file with token creation

Upload data file to cloud storage server process

If duplicate not present

Upload the data file to cloud storage server

Visualize data uploading time

Else

Compute Duplicate file and cannot upload the data file to cloud storage server

End if

Cloud client authentication process

Admin accept the cloud client authentication

Request content owner to download file

Content owner accept the cloud client request

Token send to requested cloud client

If token is correct

Cloud client downloads the requested file from cloud storage server

Else

Token is incorrect

Failed to download requested file

End if

End

4. RESULTS AND DISCUSSIONS

4.1 EXPERIMENTAL SETUP

In the work is deployed on a laptop with Intel Dual Core Processor with 1GB memory, and Window 7 Ultimate system. Here, the proposed ECDPD method is implemented in Java programming environment utilizing Netbeans 8.0, Apache Tomcat and MYSQL 5.5 database. The proposed ECDPD algorithm is evaluated with 2MB, 4MB and 8MB data.

4.2 EXPERIMENTAL RESULT

In the phase, proposed efficient content discovery and preserving De-duplication (ECDPD) Algorithm represents a mathematical model as a well graphical view. The proposed ECDPD method is categorized in two parts where the first part elaborates mathematical equation of ECDPD methodology to design parameter for proposed approached evaluation. A second part represents the tabular and graphical result of ECDPD method according to various existing algorithms like a Leakage-Resilient (LR) [13], Randomized Convergent Encryption (RCE) [13] and Secure De-Duplication Scheme (SDS) [13]. These all methods are tested with different parameters like data uploading time, and data downloading time. In the method is estimated with every parameter with various kinds of data separately.

4.3 DATA UPLOADING TIME

In the section, proposed approach elaborates mathematical model for data uploading time in Eq.(1). In the step, ECDPD method calculates as uploading time with encryption of data owner content. Data uploading time (DUT) is calculated as:

$$DUT = T_{enc} + (T_{end} - T_{start}) \quad (1)$$

where,

T_{enc} = total time taken by the method to encrypt the content.

T_{end} is data uploading completion time, and

T_{start} is an initial time of data uploading process.

4.4 DATA DOWNLOADING TIME

In the section, the proposed method defines a mathematical model for data downloading time in Eq.(2). In step, ECDPD method computes as downloading time with decryption of data owner content based on file size. Data downloading time (DDT) is calculated as:

$$DDT = (T_{finished} - T_{processing})/(\text{FileSize}) + T_{decrypt} \quad (2)$$

where,

$T_{finished}$ = total time is taken by the method of downloading the content.

$T_{processing}$ is data processing to access and view the content and $T_{decrypt}$ is decryption time to download the content in original view based on file size.

The Table.1 describes Data Uploading Time (DUT) in a sec and Data Downloading Time (DDT) in milliseconds for 2MB, 4MB and 8MB dataset to perform efficient, portable and secure data de-duplication model in cloud computing environments. In the research work computes, the Data Uploading Time (in a sec), and Data Downloading Time (in milliseconds) along with different sizes of the dataset. Hence, the research work claims that Proposed ECDPD approach is the best protocol for overall aspects.

Table.1. Data Uploading Time (DUT) and Data Downloading Time (DDT) For 2MB, 4MB and 8MB Dataset

Learning Algorithm	2MB		4MB		8MB	
	DUT	DDT	DUT	DDT	DUT	DDT
LR	12.231	6.078	23.969	11.947	47.407	23.666
RCE	6.103	6.103	11.972	11.972	23.691	23.691
SDS	6.103	6.232	11.972	12.101	23.691	23.82
ECDPD	4.253	4.579	8.569	8.953	17.538	18.206

The Table.1 performs experimental result of ECDPD approach according to many existing approaches in a tabular format. In the method is computed with data uploading time and data downloading time. Here, in the method display that ECDPD algorithm provides best result compare than other existing methods on each aspect for overall databases.

The Fig.2 displays perform data uploading time (milliseconds) for all existing methods along with proposed efficient content discovery and preserving De-duplication (ECDPD) approach for 2MB, 4MB and 8MB dataset.

The Fig.3 presents data transportation time (s) for all existing methods along with proposed efficient content discovery and preserving De-duplication approach for 2MB, 4MB, and 8MB dataset.

According to Fig.2 and Fig.3 observations, the ECDPD technique is evaluated on data uploading time and data downloading time with previous classifiers. An ECDPD is the best algorithm for overall data such as 2MB, 4MB, and 8MB data set. The proposed ECDPD is evaluated with LR, RCE and SDS previous mechanisms on behalf of data uploading time and data downloading time. Regarding Data Uploading Time, RCE, and SDS are the nearest challengers to ECDPD system. However, the RCE fails to maintain unauthorized data access, and it fails to update tokens. The ECDPD cannot permit unauthorized access and updating security faults with tokens. It also fails to maintain less retrieving time. The proposed system maintains less retrieval time. To increase data set size for SDS process, it utilized high retrieval time. The proposed system cannot utilize high retrieval time for changing data set size. Behalf of Data Downloading Time, the LR is the closest existing competitor. But, it cannot be deployed during the data download stage without loss of functionality and effectiveness. The ECDPD can execute the data downloading stage without loss of functionality and effectiveness. It is also enhancing client file level and block level de-duplication with reliability and sharing data files with securely for cloud clients in cloud storage frameworks. The ECDPD reduces 3.802 data uploading time in milliseconds and 3.318 data downloading time in milliseconds. Finally, the paper declares the ECDPD algorithm performs best on each evaluation matrix and respective input constraints.

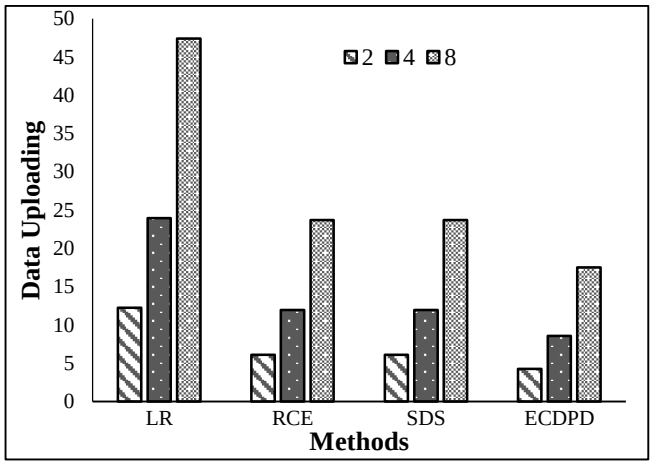

Fig.2. Data Uploading Time for 2MB, 4MB and 8MB dataset

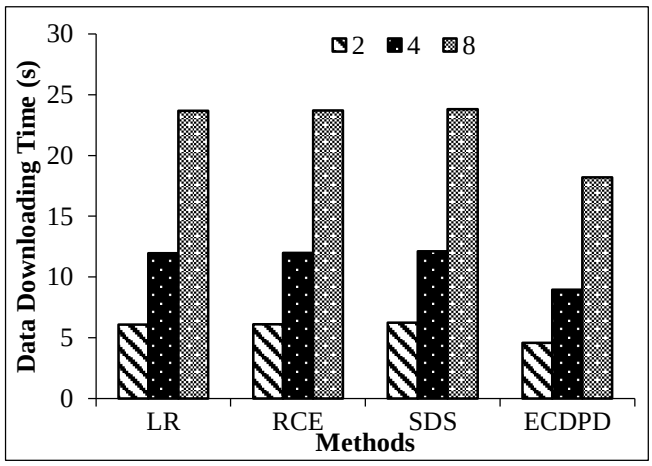

Fig.3. Data downloading time (s) for 2MB, 4MB and 8MB dataset

5. CONCLUSIONS

An efficient content discovery and preserving De-duplication (ECDPD) Algorithm is enhancing client file level and block level de-duplication with consistency and distributing content owner

files with securely for cloud clients in cloud storage frameworks. The method is protecting the illegal utilization of content owner files accessing and create duplicate copy of content owner file on a cloud storage server to encode the content owner file before storing on cloud storage server. The ECDPD algorithm minimizes 3.802 DUT (Data Uploading Time) in milliseconds and 3.802 DDT (Data Downloading Time). Finally, the paper announces the proposed ECDPD methodology performs best on each estimation matrix and particular input aspects.

REFERENCES

[1] R. Shobana, K.S. Shalini, S. Leelavathy and V. Sridevi, "De-Duplication of Data in Cloud", *International Journal of Chemical Sciences*, Vol. 14, No. 4, pp. 2933-2938, 2016.

[2] N. Kaaniche and M. Laurent, "A Secure Client-Side De-Duplication Scheme in Cloud Storage Environments", *Proceedings of IEEE International Conference on New Technologies, Mobility and Security*, pp. 1-7, 2014.

[3] J. Stanek, A. Sorniotti, E. Androulaki and L. Kencl, "A Secure Data De-Duplication Scheme for Cloud Storage", *Proceedings of International Conference on Financial Cryptography and Data Security*, pp. 99-118, 2014.

[4] K. Akhila, A. Ganesh and C. Sunitha, "A Study on De-Duplication Techniques over Encrypted Data", *Procedia Computer Science*, Vol. 87, pp. 38-43, 2016.

[5] B. Harish and K. Harshitha, "Data De-duplication In Cloud", *International Journal of Pure and Applied Mathematics*, Vol. 115, No. 8, pp. 353-358, 2017.

[6] M.P.D. Thakar and D.G. Harkut, "Hybrid Model for Authorized De-Duplication in Cloud", *International Journal of Emerging Trends and Technology in Computer Science*, Vol. 4, No. 1, pp. 147-151, 2015.

[7] F. Shieh, M.G. Arani and M. Shamsi, "De-Duplication Approaches in Cloud Computing Environment: A Survey", *International Journal of Computer Applications*, Vol. 120, No. 13, 2015.

[8] P. Puzio, R. Molva, M. Onen and S. Loureiro, "Perfect Dedup: Secure Data Deduplication", *Proceedings of International Conference on Data Privacy Management, and Security Assurance*, pp. 150-166, 2015.

[9] P. Priyadharsini, P. Dhamodran. And M.S. Kavitha, "A Survey On De-Duplication in Cloud Computing", *International Journal of Computer Science and Mobile Computing*, Vol. 3, No. 11, pp. 149-155, 2014.

[10] G.U. Devi and G. Supriya, "Encryption of Big Data in Cloud using De-duplication Technique", *Research Journal of Pharmaceutical Biological and Chemical Sciences*, Vol. 8, No. 3, pp. 1103-1108, 2017.

[11] D. Harnik, B. Pinkas and A. Shulman-Peleg, "Side Channels in Cloud Services: De-Duplication in Cloud Storage", *IEEE Security and Privacy*, Vol. 8, No. 6, pp. 40-47, 2010.

[12] S. Bharat and B.R. Mandre, "A Secured and Authorized Data De-Duplication in the Hybrid Cloud with Public Auditing", *International Journal of Computer Applications*, Vol. 120, No. 16, pp. 1-8, 2015.

[13] J. Hur, D. Koo, Y. Shin and K. Kang, "Secure Data De-Duplication with Dynamic Ownership Management in Cloud Storage", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 28, No. 11, pp. 3113-3125, 2016.