An Effective Data Reduction in a Twin Cloud Environment using an Authorized De-dup. Technique with DARE Scheme

Rasika V. Gode
Department of Computer Engineering,
M.M.C.O.E., Pune, Savitribai Phule Pune University. Maharashtra, India

Rupali Dalvi
Department of Computer Engineering
M.M.C.O.E.,Pune, Savitribai Phule Pune University, Maharashtra, India

ABSTRACT

Data deduplication is the technique of reduction in the data which keeps only one physical copy and generates pointers to that copy for referencing other redundant data. To secure the confidentiality of sensitive data throughout deduplication, the convergent cryptography technique is employed that encrypts the info before uploading it onto the general public cloud.. In our paper, we present the deduplication-aware resemblance detection and elimination (DARE) scheme which supports authorization in twin cloud environment. This scheme uses a duplicate-adjacency information for resemblance detection where we have to consider any two data blocks to be similar only if their respective adjacent data blocks are duplicate. Our proposed system achieves deduplication on encrypted data with minimum overhead and also enhances the security by managing convergent keys. In addition our system also increases the security level by providing OTP validation technique to avoid unauthorized access to the cloud data.

Keywords

Data deduplication, delta compression, hybrid cloud, tag generation, performance evaluation, storage system.

1. INTRODUCTION

Now a days the cloud computing has widely attracted more and more attention. In the data storage to reduce the data copies we go for deduplication techniques.

The technique referred to as deduplication that is employed to reduces information by removing the duplicate copies of identical data and it's wide utilized in cloud storage to avoid wasting bandwidth and minimize the space for storing[1]. In cloud computing usually the users outsource their data to external cloud servers which may be the public cloud thus not secure and that data may contain some privacy information, such as personal photos, emails, etc. If there is no means for efficient protection, then it leads to severe confidentiality and privacy violations. Therefore it is very essential to encrypt the private data before storing them to the cloud. This issue in mobile cloud computing motivates to protect the confidentiality of sensitive data. One of the technique called convergent encryption is proposed to encrypt the data before storing it onto the cloud storage along with a support to de-duplication. Our technique solves the problem of authorized data de-duplication and provides the solution to detect and eliminate redundancy at minimum overhead.

Some necessary problems in data deduplication ar privacy and security for defense of data from insider or outsider offender. Different users uses their own secret key for encryption/decryption to achieve the confidentiality of sensitive data. For file uploading on cloud, users follows the following procedure: they generate convergent key first using SHA-256, then load it to the cloud in encrypted form. When deduplication found, Proof of ownership protocol is employed to convey the proof that the user conjointly owns a similar file. This protocol is employed to supply a licensed access. when authentication, server give a pointer to owner for accessing same file while not having to upload same file. once user need to transfer file he merely download the file from cloud that is in encrypted kind and decipher this file mistreatment convergent key[3]. we have a tendency to also present a Deduplication-Aware likeness detection and Elimination (DARE) scheme for information compression that removes redundancy at most level with minimum overhead[1].

In general, a block-level information deduplication scheme divides a data stream into multiple data blocks that are each uniquely known and duplicate-detected by a secure SHA-1 or MD5 hash signature also known as a fingerprint[5]. Storage systems then remove duplicates of data blocks and store only one copy of it to achieve the goal of space savings[2]. Even if the data deduplication has been wide deployed in storage systems for space savings, on the opposite hand the fingerprint-based deduplication approaches have some drawback: they usually fail to find the similar chunks that ar for the most part identical aside from some changed bytes, as a result of their secure hash digest are going to be totally different even just one byte of a data chunk was modified. Therefore, this is often one in all the largest challenge when applying data deduplication to cloud storage that have frequently changed data. It demands an effective and efficient way to eliminate redundancy among frequently changed and so similar data. An efficient approach to removing redundancy among similar information blocks is delta compression that has gained a giant attention in storage systems. For example, if block P2 is similar to block P1 (the base chunk), the delta compression technique calculates and then only stores the differences (i.e. delta factor) and mapping relation between P2 and P1. Thus, it's thought-about an efficient technique that effectively enhances the fingerprint-based deduplication approaches by detecting similar data missed by the latter. one among the most challenges facing the application of delta compression in deduplication systems is the way to detect the foremost similar candidates for delta compression accurately with low overheads.

From our observation of duplicate and similar data of backup streams stored in a very cloud storage, we discover that the non-duplicate blocks that are adjacent to duplicate ones might be thought-about sensible delta compression candidates in information deduplication systems. Hence we propose the technique of Duplicate- Adjacency based Resemblance Detection, or DupAdj for short. Exploiting this existing deduplication information (i.e., duplicate-adjacency) not only avoids the high overhead of super-feature computation but also reduces the size of index entries for resemblance detection.

2. RELATED WORK

2.1 Authorized data de-duplication

Even if there are a lot of advantages of data de-duplication technique, this will not supports privacy and security of data in cloud storage. So there may be the possibility that users private data are susceptible to both the insider and outsider attacks. The traditional deduplication was unable to provide data confidentiality when applied on encrypted data. The traditional encryption requires different users to encrypt data with their own keys. Thus different cipher texts will generate from duplicate copies of different users and this makes de-duplication on encrypted data impossible. The convergent encryption is one in all the algorithmic program that is projected to encrypt information for confidentiality whereas creating de-duplication possible [6]. This system uses symmetric encryption, and therefore the secret's obtained by computing the cryptographic hash value of the content of the message. Once completion of key generation and encoding, users retain the keys to private cloud and then send the cipher text to the cloud. Since the encryption operation is deterministic and comes from the data content, identical information files generate an equivalent convergent key and thence an equivalent cipher text. A secure proof of ownership protocol is additionally needed to produce the proof that the user indeed owns. this can be all to stop unauthorized access. If the file duplicates are found, only the pointer to that file is stored in public cloud. After the proof submission by owner, who are having the subsequent files, doesn’t need to upload the same file[4]. The corresponding data users can download the encrypted files and also decrypt them by using their convergent keys. Therefore in cloud storage, convergent encryption allows the cloud to perform de-duplication on encrypted data and the proof of ownership prevents the unauthorized user to access the file and in this way, providing confidentiality along with authorization. The previous de-duplication systems cannot supports Differential authorized de-duplication check. With the authorized de-duplication system, each user issued a set of the privileges during system initialization[6]. The controlling task to deciding which type of user is allowed to perform the duplicate check and access the files is performed at the time of uploading each file to the cloud and is also bounded by the set of privileges. The user have to submit the file and their own privileges as inputs before sending the user duplication check request for the same file. If copy of the file exist and users privileges matched with the privileges stored in cloud, then and only then the user will get the pointer for the same file[7]. The detailed system architecture is shown in figure 1.

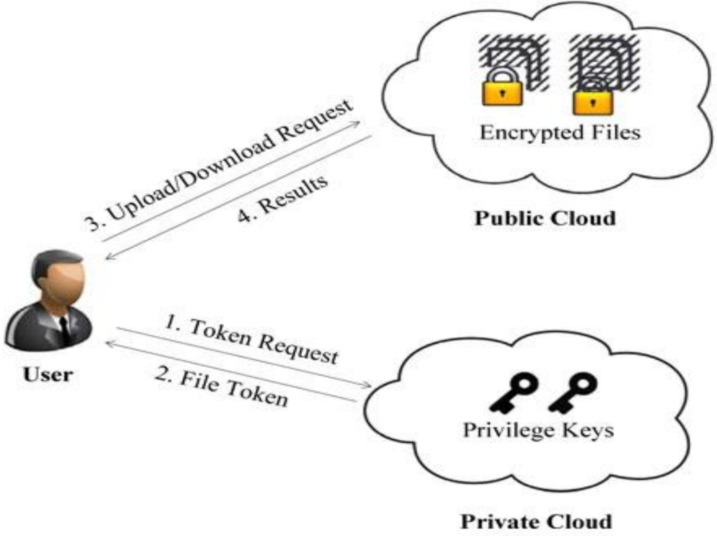


Fig.1. Architecture of authorized deduplication[3].

2.2 Resemblance Detection Based Data Reduction

Data deduplication is becoming increasingly popular in recent years. Specially in data-intensive storage systems as one of the most efficient data reduction approaches. Fingerprint based deduplication techniques eliminate duplicate chunks by checking their secure-fingerprints (i.e., SHA-1/ SHA-256 signatures), which has been widely used in commercial backup and archiving storage systems. Resemblance detection with delta compression [10], [11], one among the approach to data reduction in storage systems, was proposed over ten years past however was later overshadowed by fingerprint-based de-duplication because of the formers scalability issue. resemblance detection detects redundancy among similar information at the byte level whereas duplicate detection finds whole identical information at the chunk level. So generally the latter approach is much more scalable than the former in mass storage systems. REBL and DERD are 2 super - feature primarily based resemblance detection approaches for data reduction. They compute the features of the information stream (e.g., Rabin Fingerprints) and group features into super-features to capture the likeness of knowledge then delta compress the information. of these approaches for likeness detection needs high overheads of computation and categorisation. Shilane et al. projected a stream-informed delta compression (SIDC) approach utilized in a WAN surroundings for reducing similar information transmission and so fast information replication [9]. This approach is super-feature primarily based and enhances the block-level deduplication by solely police work resemblance among non-duplicate blocks within the cache that preserves the backup stream locality. It avoids the indexing. whereas the combined detection of duplicate and alikeness guarantees to attain a superior information reduction performance.

2.3 The concept of duplicate adjacency

The changed blocks could also be terribly like their previous versions in a very backup system whereas unmodified blocks can stay duplicates and are simply identified by the deduplication method. For those non-duplicate blocks that are location-adjacent to known duplicate knowledge blocks in a very deduplication system, it's intuitive and quite attainable that solely a couple of bytes of them are changed from the last backup, making them potentially excellent delta compression candidates.

Fig. 2 shows the situation of duplicate data blocks and their immediate non-duplicate neighbors. As mentioned above, our intuition is that the latter are highly likely to be similar and thus good delta compression candidates. Specifically, since blocks ’B3’ and ’B4’ are duplicates of chunks ’E3’ and ’E4’ in Fig. 2 respectively, their immediate neighbors, the block pairs ’B1’ and ’E1’, ’B2’ and ’E2’, and ’B5’ and ’E5’, are then considered good delta compression candidates, which is consistent with the aforementioned backup-stream locality.

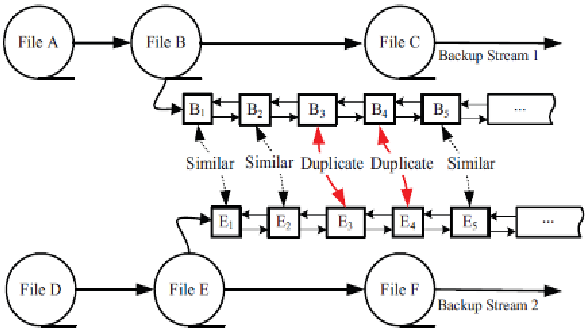


Fig. 2. A conceptual illustration of the duplicate adjacency. The non-duplicate chunks adjacent to duplicate ones are considered good delta compression candidates as they are potentially similar.

3. SYSTEM ARCHITECTURE

Our system will provide an authorized deduplication on encrypted data. The data is in the form of text file. The system effectively manages the entire storage space in a secure and authorized manner as well as the system enables to maximally detect and eliminate redundancy at very low overheads by implementing DARE scheme.

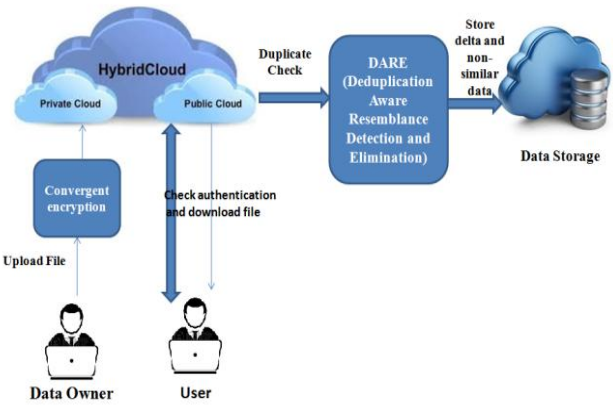

HybridCloud
Private Cloud
Public Cloud
Duplicate Check
DARE (Deduplication Aware Resemblance Detection and Elimination)
Store delta and non-similar data
Data Storage
Convergent encryption
Check authentication and download file
Upload File
Data Owner
User

Fig. 3 Architecture of An Effective Data Reduction in a Twin Cloud Environment Using an Authorized De-dup. Technique with DARE Scheme

Our proposed system is divided into three parts:

1) The authentication scheme for data users is used in a private cloud server to prevent system from attackers.
2) Convergent encryption is used to encrypt the data and perform duplicate check.
3) DARE approach is employed to maximally find and eliminate redundancy at minimum overheads.

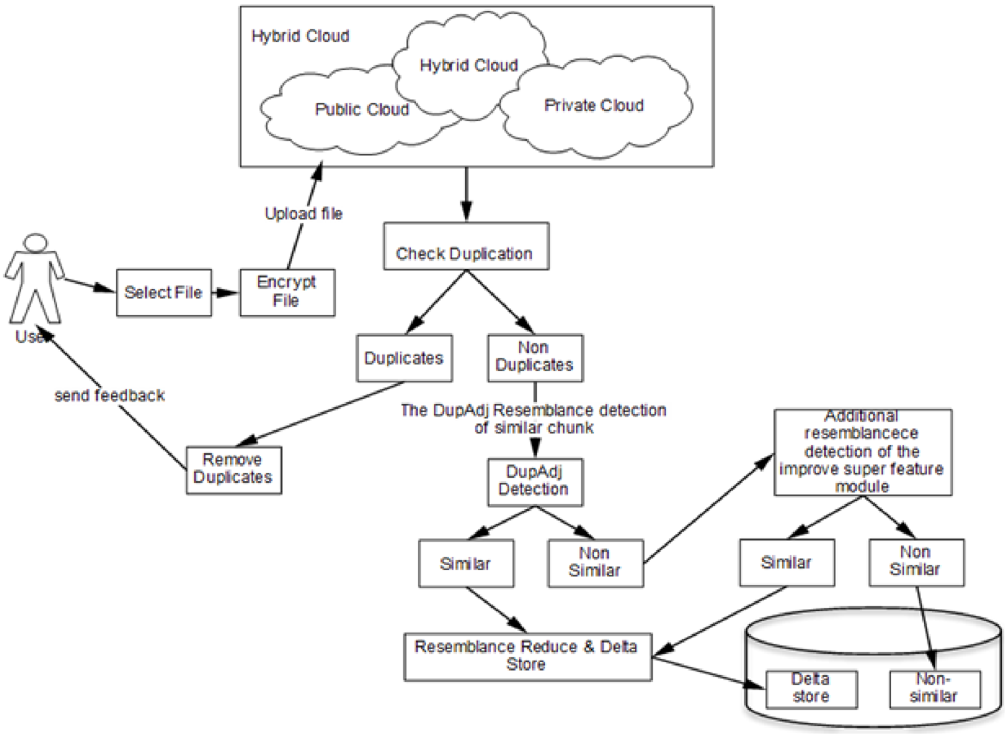


Fig. 4. The data reduction workflow of an authorized deduplication system for encrypted data with DARE scheme in a twin cloud environment

3.1 Secure Duplication with DARE scheme in a hybrid cloud environment:

The main issues within the cloud computing is de-duplication with differential privileges. the main aim of this paper is to resolve this drawback. For this we tend to go along with completely different variety of design, that has public cloud and private cloud i.e., Hybrid Cloud Architecture also called as twin cloud architecture. The private cloud performs the main role in hybrid cloud environment, that is involved as the substitution to allow data owners to perform de-duplication check securely with differentials privileges[6]. We implement the authorization by using OTP generation technique, where without permission of file owner, no one can access the file from private cloud. Here, the key is managed by private cloud and only the data which is in encrypted from stored on public cloud. Under the hybrid cloud design we have a tendency to propose differential duplication check separated by a brand new de-duplication system. A user solely with corresponding privilege on files has been allowed to perform de-duplication. we conjointly present DARE, a deduplication aware, minimum overhead resemblance detection and elimination technique for delta compression additionally of deduplication on cloud storage system.

DARE uses a likeness detection approach, DupAdj, that uses the duplicate-adjacency info for efficient resemblance detection in existing deduplication systems, and employs an improved super-feature approach to additional detecting likeness once the duplicate-adjacency info is lacking or limited. So, our contribution achieves root level deduplication and also implements an OTP validation scheme to avoid unauthorized access by using hybrid cloud environment where private cloud is responsible to manage the convergent key and public cloud holds the file in encrypted format.

3.2 System Modules

There are three entities define in our system :

1) Users
2) Private cloud
3) S-CSP in public cloud

1. Data Users: A user is associate entity that wishes to source information storage to the S-CSP and access the information later. User generate the key and store that key in private cloud. every file is protected by convergent encryption key and might access by solely licensed person. In our system user should have to be compelled to register in private cloud for storing token with several file that are store on public cloud..

2. Private Cloud: In general for providing more security instead of public cloud user can use the private cloud. User store the generated key in private cloud. At the time of file downloading, system first asks the key to download the file. User cannot store the secrete key internally. For providing proper protection to key we use private cloud. For providing correct protection to key we tend to use private cloud. private cloud solely store the convergent key with its individual file. once user wish to access the key he 1st check authority of user then solely give the key

3. Public Cloud: Public cloud entity is employed for the storage purpose. User upload the files in public cloud. Public cloud is similar as S-CSP. When the user want to download the files from public cloud, it will be ask the key which is generated or stored in private cloud. Only authorized user can access the file. In public cloud all files are stored in encrypted format.

4. ALGORITHMS USED

4.1 Convergent Encryption Technique

In data deduplication, data confidentiality is provided by convergent encryption algorithm. A user or owner of data generates a convergent key from each original data file and encrypts the file with the same key[8]. A tag for the file is derived by the user, such that the tag will be used to detect duplicates. Here, we assume that if two data files are the same, then their tags are also the same. The user first sends the tag to the server side for checking whether the duplicate copy of his file already exist or not. If duplicate is not found then both the encrypted data copy and its corresponding tag will be stored on server side [8].

Four primitive functions which defines the convergent encryp-tion scheme are as follows:

1) KeyGenCE(M) - > K maps a data copy M to a convergent key K, so it is the key generation algorithm;[6]
2) EncCE(K, M) - > C takes both the convergent key K and the data copy M as inputs and then outputs a ciphertext C, so it is the symmetric encryption algorithm.[6]
3) DecCE(K, C) - > M takes both the ciphertextC and the convergent key K as inputs and then outputs the original data copy M, so called as the decryption algorithm; [6]and
4) TagGen(M) - >T (M) maps the original data copy M and outputs a tag T (M), so T (M) is the tag generation algorithm.[6]

4.2 Proof of Ownership

The main purpose of proof of ownership (POW) protocol is to enable users to prove their ownership of data copies to the storage server. It is a protocol which is denoted by POW. The verifier derives a short value ϕ (M) from a data copy M. To prove the ownership of the data copy M, the prover needs to send ϕ to the verifier such that ϕ’=ϕ (M).[6]

PSEUDO CODE

```
Step1: Calculate the two convergent key values
Step2: Compare the two keys and files get accessed.
Step3: Apply de-duplication to eradicate the duplicate values.
Step4: If any other than the duplicates it will be checked once again and make the data unique.
Step5: That data will be unique and also more confidential the authorized can access and data is stored.
```

4.3 File Encryption

Data owner runs this algorithm. It encrypt the entire document and generates ciphertexts. For each document, this algorithm will create a delta a for its searchable encryption key ka. This algorithm outputs data ciphertext and keyword ciphertexts Ca on input of the owners public key pk and the file index i,

4.4 Encrypted Data Upload

The data holder encrypts its data using a randomly selected symmetric key DEK in order to ensure the security and privacy of data, and stores the encrypted data at CSP along with the token used for data duplication check only and only if data duplication check is negative. The data holder first encrypts DEK with pk AP and then passes the encrypted key to CSP.

4.5 Data Deduplication

Data duplication happens once information owner tries to store identical data that has been hold on already at the general public cloud storage. this can be checked by scrutiny the tags. If the comparison is positive, CSP contacts AP for deduplication by providing the token and also the knowledge key. The AP challenges knowledge possession, checks the eligibility of the information holder, so problems a re-encryption key which will convert the encrypted DEK to a type which will solely be decrypted by the eligible knowledge holder.

4.6. Delta Compression

To reduce data redundancy among similar chunks, Xdelta, an optimized delta compression algorithm, is adopted in DARE after a delta compression candidate is detected by DAREs resemblance detection. Only one-level delta compression for similar data is carried out by DARE as employed in DERD and SIDC[1]. In DARE, delta compression will not be applied to a chunk that has already been delta compressed to avoid recursive backward referencing. And DARE records the similarity degree as the ratio of compressed size/ original size after delta compression. For each of the resembling chunks detected in resemblance detection, DARE reads its base-chunk, then delta encodes their differences. In order to reduce disk reads, an LRU and locality-preserved cache is implemented here to prefetch the base-chunks in the form of data segments.

5. MATHEMATICAL MODEL

- **Input**: Input given to the system is: -Encrypted File in any format.
- **Output**: Whenever user wants to upload the file on cloud or store the file on secondary storage then we check or test the duplication and Resemblance Detection and Elimination or not.
- **Process:**

Step 1: Data owner Select File
Step 2: Encrypt File (For encryption we user AES or RSA).
Step 3: Upload file on cloud or store the file on secondary storage.
Step 4: CSP or Controller check the duplicate file available on cloud or secondary storage.
Step 5: If found then remove the duplication and maintain index.
Step 6: On non duplicate data CSP again check resemblance detection of similar chunk.
Step 7: If resemblance found then reduce it create delta store.
Step 8: Again on non similar data check resemblance detection.
Step 9: If resemblance found then reduce it store similar data in delta store.
Step 10: Finally non similar data stored into secondary storage or Cloud storage.

Mathematical model contains five tuples –

$S= \{s, e, X, Y, \phi \}$

where the following conditions are satisfied-

s = Start of the program

1) Log in with webpage.
2) Load Text Files on cloud.

e = End of the program.

Retrieve the file from cloud storage system.

X = Input of the program.

Input should be any text file.

Y = Output of the program.

Φ = Success and failure conditions.

File will be first fragmented then it is encoded and the fragments are allocated. Finally when we request for text file downloading we get text file as output.

$X, Y \in U$

Let U be the Set of System.

U = { Client, F, S, T, M, D, R, DC }

Where,

Client, F, S, T, M, D,R,DC are the elements of the set.

Client = Data Owner, User

F = Fragmentation

S= Fragments encoded using Secret Sharing Scheme

T = Generate Tags for file blocks

M = Message Authentication Code for generating hash values.

D = Check for duplicate file or block

R = Detects resemblance by exploiting existing duplicate-adjacency information of a deduplication system.

DC = Delta compression module takes each of the resembling chunks detected in R, and reads its base-chunk, then delta encodes their differences.

- **Success Condition:** Successfully work keys aggregation, trapdoor generation, file splitting and stored into multi cloud. User gets result very fast according to their needs.
- **Failure Condition :**
  1) Huge database can lead to more time consumption to get the information
  2) Hardware failure.
  3) Software failure.
- **Space Complexity:** The space complexity depends on Presentation and visualization of discovered patterns. More the storage of data more is the space complexity.
- **Time Complexity:** Check No. of patterns available in the database = n. If (n>1) then retrieving of information can be time consuming.

Above mathematical model is NP Complete.

6. SYSTEM IMPLEMENTATION

We implement system with data deduplication,in which we model three entities as separate programs. To model the data users a Client program is used to carry out the file uploading/downloading process. A Private Server program is used to model the private cloud which manages the private keys and handles the file token computation. A Storage Server program is used to model the S-CSP which stores and deduplicates files. Followings are function calls used in system:

- FileTag(File) - It generates SHA-1 hash of the File as File Tag;
- DupCheckReq(Tag) - It requests the Storage Server for Duplicate Check of the file.
- FileEncrypt(File) - It uses 256-bit AES algorithm to encrypts the File with Convergent Encryption in cipher block chaining (CBC) mode, where the convergent key is from SHA-256 Hashing of the file;
- FileUploadReq(FileID, File, Token) – It uploads the File Data to the Storage Server only if the file is Unique and updates the File Token stored.
- FileStore(FileID, File, Token) - It stores the File on Disk and updates the Mapping.
- GenOTP(RandomFun) – It generates OTP which is send by owner of file to the other authorized user of same file.

7. EXPERIMENTAL RESULTS

We evaluate DARE in the key metrics of data reduction, data-reduction throughput and similarity degree.

Data reduction here is defined as the percentage of redundancy removed by deduplication and resemblance detection. Data-reduction throughput is measured by the rate at which datasets are processed, including deduplicating, detecting resemblance, and delta compressing. The similarity degree of resemblance-detected chunks is measured by the ratio of (compressed size) / (original size).

Proposed system work on low overheads means existing system only find those file which have 100% duplicate if file is 50% duplicate then existing system directly allocate the storage space for file in secondary storage, But in proposed work system reduce the 50% data of file using block and byte level duplication checking and maintain the index and store only 50% data on cloud storage which is not duplicate. Above result table and graph shows the Performance Mesurment and Comparitive analysis between Proposed and Existing System.

Following graph illustrates the percentage of found similer block in Existing System & Proposed System-

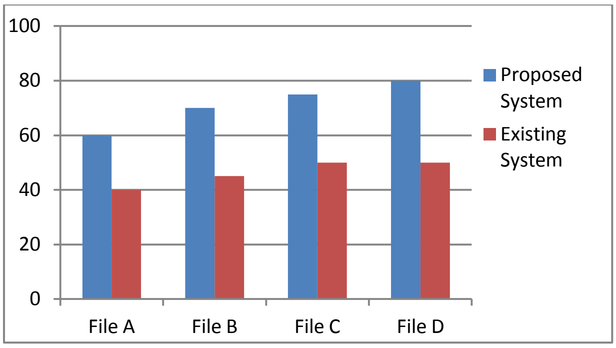


Fig. 5 Performance Mesurment and Comparitive analysis

Result table:
Percentage of found similer blocks in Existing System & Proposed System.

File\Method	Proposed System	Existing System
File A	60	40
File B	70	45
File C	75	50
File D	80	50

The screenshot shown in Fig. 6. indicates that the number of duplicate blocks found in our system is much more than that of existing simple deduplication systems. The analysis graph demonstrates that resemblance detection is very efficient in supplementing deduplication for data reduction.

Fig. 7 shows that DARE achieves the highest throughputs among all the resemblance detection enhanced data reduction approaches. The time required for processing each file block is shown is calculated and display in following screenshot. The screenshots of the execution process of the system are shown below. It illustrates the experimental results of the designed system. From these results we get the detailed information to check de-duplication and file uploading, fetching the signs using hashing algorithm and checking for duplication. Form the given detailed procedure of the system we confirmed that securely authorized deduplication at root level is successfully achieved with our approach. Output images are given below.

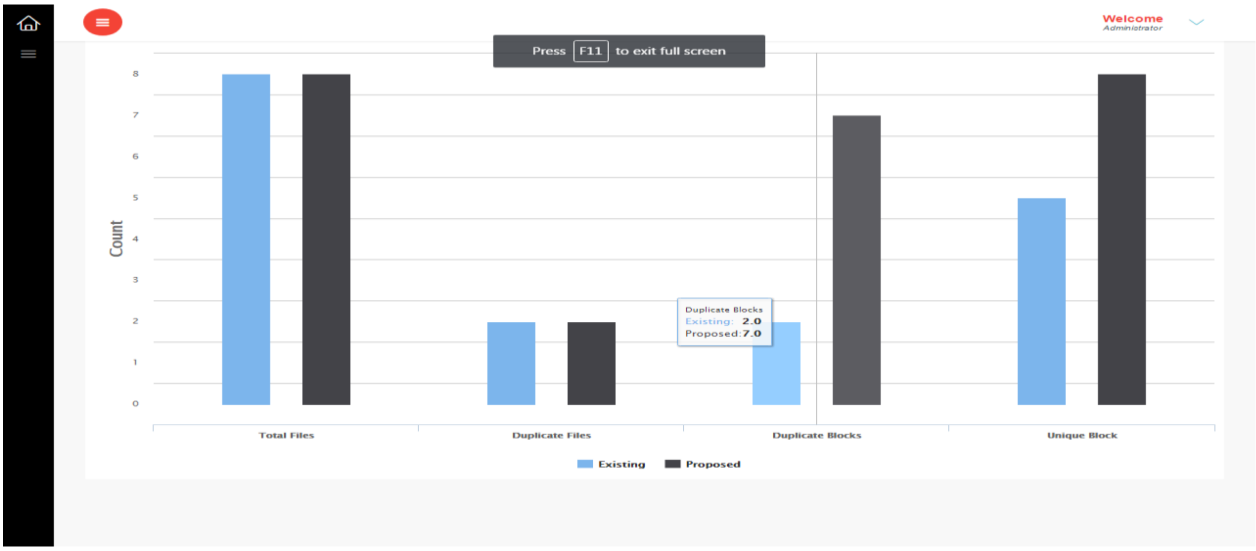
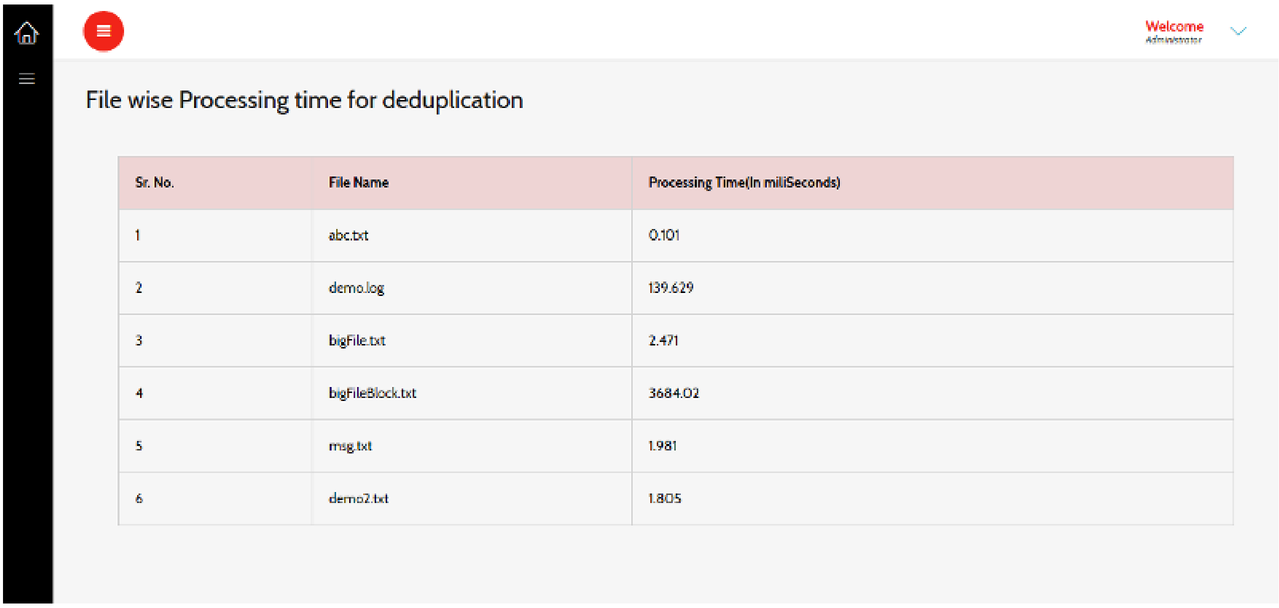


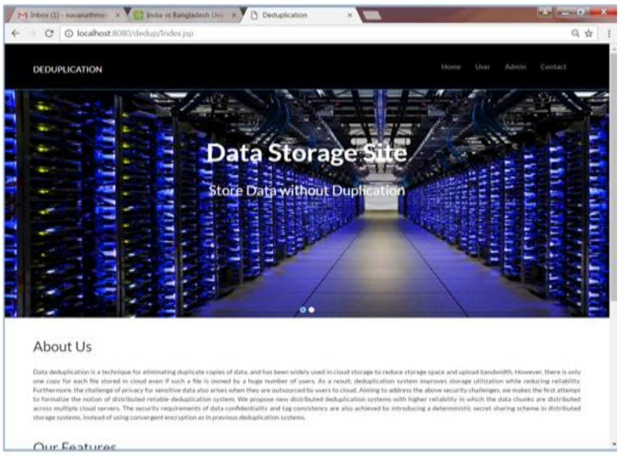
Fig**. 6. Analysis graph shows that more number of duplicate blocks are found by DARE as compare to existing dedupliction techniques**



File wise Processing time for deduplication

Sr. No.	File Name	Processing Time(In miliSeconds)
1	abc.txt	0.101
2	demo.log	139.629
3	bigFile.txt	2.471
4	bigFileBlock.txt	3684.02
5	msg.txt	1.981
6	demo2.txt	1.805

Fig. 7 Throughputs of resemblance detection enhanced data reduction approaches (i.e., deduplication + delta compression) on the sample backup dataset



Data Storage Site

Store Data without Duplication

About Us

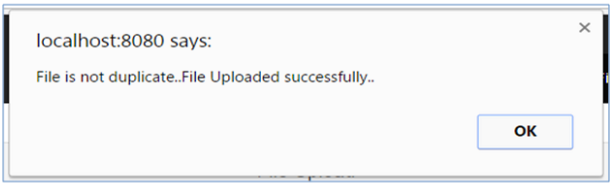
Data deduplication is a technique for eliminating duplicate copies of data, and has been widely used in cloud storage to reduce storage space and upload bandwidth. However, there is only one copy for each file stored in cloud even if such a file is owned by a huge number of users. As a result, deduplication system improves storage utilization while reducing reliability. Furthermore, the challenge of privacy for sensitive data also arises when they are outsourced by users to cloud. Aiming to address the above security challenges, we makes the first attempt to formalize the notion of distributed reliable deduplication system. We propose new distributed deduplication systems with higher reliability in which the data chunks are distributed across multiple cloud servers. The security requirements of data confidentiality and tag consistency are also achieved by introducing a deterministic secret sharing scheme in distributed storage systems, instead of using convergent encryption as in previous deduplication systems.

Our Features

Fig.8 Home screen of the deduplication system.

User or owner selects the file to be uploaded on the cloud. After selecting the file for uploading, the system checks for deduplication. There are four possibilities-

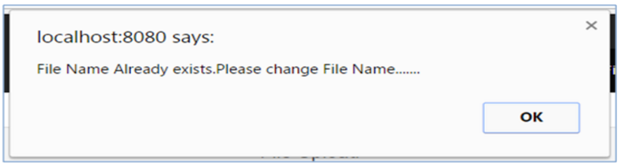
1. The duplicates are not found.
2. The file with same name may exist.
3. The file with duplicate contents may exist.
4. The file with some similarity may exist.



localhost:8080 says:

File is not duplicate..File Uploaded successfully..

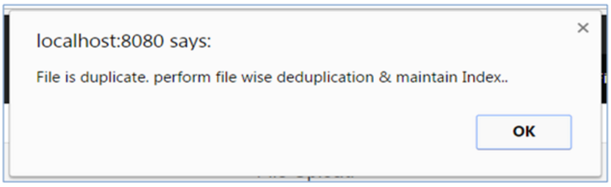
Fig. 9 File is upload if no duplicate file exist.



localhost:8080 says:

File Name Already exists.Please change File Name.......

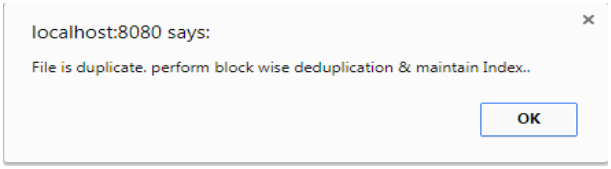
Fig. 10 Popup message if the file with existing file name is selected for uploading.



localhost:8080 says:

File is duplicate. perform file wise deduplication & maintain Index..

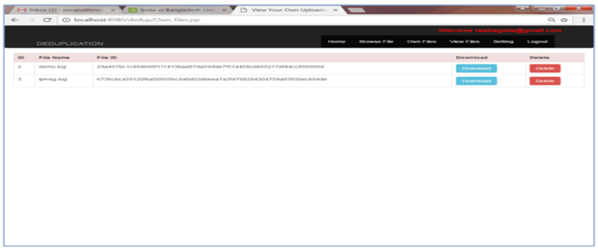
Fig. 11 Popup message if the selected file contents are found duplicate with existing file contents. Only pointer is stored.



localhost:8080 says:

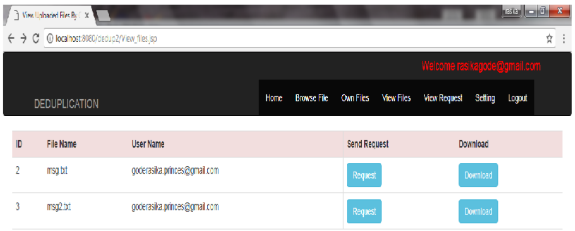
File is duplicate. perform block wise deduplication & maintain Index..

Fig.12 Popup message after detecting the resemblance and removing it by DARE scheme



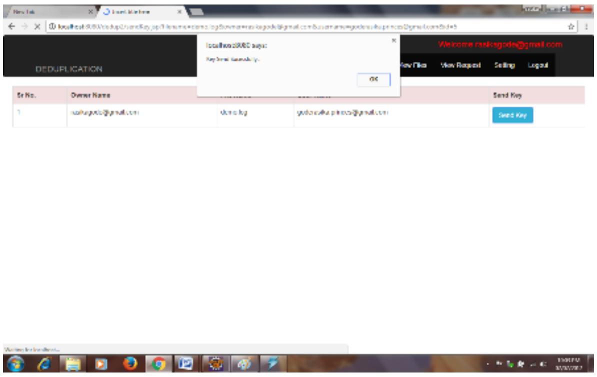
ID	File Name	File ID	Download	Delete
2	demo.log	29a457bc1c804b98f111413baa87da594de7f514458c6b55277df44cc858956d	Download	Delete
3	ipmsg.log	6736cbca39120fea509356c9a0d03dbeea7a3f47882843047 59a83930ecb94de	Download	Delete

Fig. 13 File owner can download and delete their files.



ID	File Name	User Name	Send Request	Download
2	msg.txt	goderasika.princes@gmail.com	Request	Download
3	msg2.txt	goderasika.princes@gmail.com	Request	Download

Fig. 14 User can send request to file owner for downloading their files

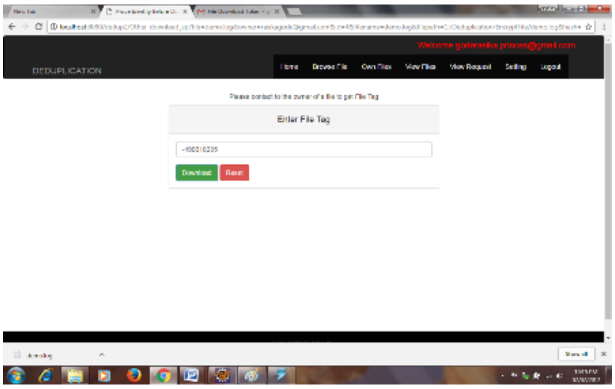


localhost:8080 says:

Key Send Successfully...

Sr No.	Owner Name			Send Key
1	rasikagode@gmail.com	demo.log	goderasika.princes@gmail.com	Send Key

Fig. 15 Owner sends the key as OTP to the authorized user to download the file



Please contact to the owner of a file to get File Tag

Enter File Tag

-490316335

Fig. 16 User get the OTP and download the file successfully

8. CONCLUSION

In this paper to support stronger security, we present an advanced authorized deduplication scheme by encrypting the Text file with convergent keys in a twin cloud environment. In this way, the users without taking owner's permission cannot access the files of other users. Furthermore, any unauthorized users cannot decrypt the cipher text even collude with the S-CSP. In this manner we achieves the authorization. Also we implement DARE scheme which is very effective in determining and eliminating redundancies at maximum level and with very low overheads. The system effectively manages the storage space in a secure and authorized manner. And the system enables to maximally detect and eliminate redundancy at very low overheads supporting the secure data storage on public cloud.

9. ACKNOWLEDGEMENT

We express our deepest gratitude to the college authorities for technical guidance and infrastructure. Lastly we wish to thank the researchers and reviewers for their contributions because of which we could complete this work.

10. REFERENCES

[1] Wen Xia, Member, IEEE, Hong Jiang, Fellow, IEEE, Dan Feng, Member, IEEE, and Lei Tian, Senior Member, IEEE, DARE: A Deduplication-Aware Resemblance Detection and Elimination Scheme for Data Reduc-tion with Low Overheads, IEEE TRANSACTIONS ON COMPUTERS, VOL. 65, NO. 6, JUNE 2016.

[2] Zheng Yan, Senior Member, IEEE, Wenxiu Ding, Xixun Yu, Haiqi Zhu, and Robert H. Deng, Fellow, IEEE, Deduplication on Encrypted Big Data in Cloud, IEEE TRANSACTIONS ON BIG DATA, VOL. 2, NO. 2, APRIL-JUNE 2016.

[3] Jin Li, Yan Kit Li, Xiaofeng Chen, Patrick P. C. Lee, Wenjing Lou, A Hybrid Cloud Approach for Secure Authorized De-duplication IEEE Transactions on Parallel and Distributed Systems: Volume:26. No. 5, MAY 2015.

[4] J. Li, X. Chen, M. Li, J. Li, P. Lee, andW. Lou. Secure deduplication with efficient and reliable convergent key management. In IEEE Transactions on Parallel and Distributed Systems, 2013.

[5] A. Venish and K. Siva Sankar, Study of Chunking Algorithm in Data Deduplication Proceeding of the International Conference on Soft Computing System, ICSCS , Volume 2.

[6] JadapalliNandini, Rami reddyNavateja Reddy Implementation De-duplication System with Authorized Users International Research Journal of Engineering and Technology (IRJET), volume-2, Issue 3, June -15.

[7] Backialakshmi. N Manikandan. M Secured Authorized De-Duplication in Distributed System IJIRST International Jour-nal for Innovative Research in Science and Technology Volume 1 Issue 9 February 2015.

[8] Rajashree Shivshankar Walunj, 2, Deepali Anil Lande, 3, Nilam Shrikrushna Pansare, Secured Authorized Deduplication Based Hybrid Cloud, The International Journal Of Engineering And Science (IJES), Volume 3 , Issue 11.2014

[9] P. Shilane, M. Huang, G. Wallace, and W. Hsu, WAN optimized replication of backup datasets using stream-informed delta compression, in Proc. 10th USENIX Conf. File Storage Technol., Feb. 2012, pp. 4964.

[10] F. Douglis and A. Iyengar, "Application-specific delta-encoding via resemblance detection," in Proc. USENIX Annu. Tech. Conf., General Track, Jun. 2003, pp. 113–126.

[11] P. Kulkarni, F. Douglis, J. D. LaVoie, and J. M. Tracey, "Redundancy elimination within large collections of files," in Proc. USENIX Annu. Tech. Conf., Jun. 2012, pp. 59–7.