ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

An Authorized De-dup Technique for Encrypted Data with DARE Scheme in a Twin Cloud Environment

Rasika V. Gode, Prof. Rupali Dalvi

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

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

ABSTRACT: The technique called Data deduplication (De-dup) is used to reduce the data storage space by removing the duplicate copies of exactly similar data. In this technique it preserves only one physical copy and generates pointers to that copy for referring other redundant data. De-dup technique does not supports traditional encryption, to secure the confidentiality of private data during deduplication, convergent encryption is the technique which is used to encrypt the data before uploading it onto the public cloud. The large scale data storage is facing one major problem of how to detect maximum redundancy and eliminate it at very low overheads in order to achieve data reduction. In our paper, we present a deduplication-aware resemblance identification and elimination(DARE) scheme with low-overhead. This scheme uses an existing duplicate-adjacency information for resemblance identification in data De-dup. The main basic idea to implement DARE is Duplicate-Adjacency based Resemblance Identification technique(DupAdj). In this we have to consider any two data blocks to be similar which are the candidates for delta compression, if and only if their respective adjacent data blocks are duplicate in a De-dup system. And then we use an improved super-feature approach for further enhancement. In existing systems, authorized De-dup technique is used only for in-house computers, in our proposed system, we can use an authorized De-dup technique in cloud storage also, again our system supports DARE scheme on encrypted data so that it will provide better data security along with minimum overhead. DARE only consumes about one fourth and one half respectively of the computation and indexing overheads required by the traditional super-feature approaches according to our experimental results based on synthetic backup datasets. It also detects 2 to 10 percent more redundancy and achieves a higher throughput, by using available duplicate-adjacency information for detecting resemblance.

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

I. 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 of deduplication(De-dup) which is used for storage space management by removing the duplicate data and it is widely used in cloud storage to save bandwidth and minimize the storage space. 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 cloud computing environment motivates to protect the confidentiality of private data. One of the technique called convergent encryption(CnvEn) is proposed to encrypt the data before storing it onto the cloud along with the support to de-duplication. Our technique solves the issue of authorized data de-duplication and provides the solution to identify and remove redundancy at minimum overhead. Some important issues in data De-dup systems are privacy and security for protection of data from insider or outsider attacker. Different users uses their own private 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-

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

256, then load it to the cloud in encrypted form. When deduplication found, Proof_of_ownership protocol is used to give the proof that the user also owns the same file. This protocol is used to provide an authorized access. After authentication, server provide a pointer to owner for accessing same file without needing to upload same file. When user want to download file he can get the file from cloud which is in encrypted form and then he has to decrypt this file using convergent key.

We also present a DARE scheme for data compression which removes redundancy at maximum level with minimum overhead by detecting and eliminating and eliminating resemblance in support of De-dup.

In general, a block-level data Dedup. scheme divides a data stream into number of data blocks that are each uniquely identified and duplicate-detected by a secure SHA-1 or MD5 hash signature also known as a fingerprint. Storage systems then remove duplicates of data blocks and store just the single copy of it to achieve the goal of space savings. Even if the data De-dup. has been widely used in storage systems for space savings, on the other hand the fingerprint-based deduplication approaches have some drawback: they often fail to detect the similar blocks, because their secure hash digest will be totally different even that are greatly identical except for a few modified bytes and even only one byte of a data block was changed. It is one of the biggest challenge
when applying data De-dup. to cloud storage that have frequently modified data. For e.g. data in backup systems. It demands an effective and efficient way to eliminate redundancy among frequently modified and thus similar data. An efficient approach for removing redundancy among similar data blocks is delta compression which has gained a big 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 is considered an efficient technique that effectively complements the fingerprint-based De-dup. approaches by detecting similar data missed by the latter. One of the major challenges facing the application of delta compression in De-dup. systems is how to detect the most similar candidates for delta compression accurately with minimum overheads.

From our observation of duplicate and similar data of backup streams stored in a cloud storage, we find that the non-duplicate blocks that are adjacent to duplicate ones could be considered good delta compression candidates in data De-dup systems[1]. Hence we propose the technique of Duplicate- Adjacency based Resemblance Detection, or DupAdj for short. Exploiting this existing De-dup information simply avoids the high overhead of super-feature computation along with reducing the size of index entries for identifying resemblance.

II. RELATED WORK

A. 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 the environment of cloud. So there may be the possibility that users private data are susceptible to attacks. The traditional De-dup. was unable to provide data confidentiality when applied on encrypted data. The traditional encryption requires different users to encrypt data with their own unique keys. Thus different cipher texts will generate from same data copies of different users and this makes De-dup. on encrypted data impossible[3]. The convergent encryption(CnvEn) is one of the algorithm which is proposed to encrypt data for confidentiality while making De-dup. feasible [6]. This technique uses symmetric encryption, and the key is generated by computing the cryptographic hash value of data copies. After completion of key generation and data encryption, users retain the keys to private cloud and then send the cipher text to the public cloud. Since the encryption algorithm is deterministic and is derived from the data content, identical data files generate the same convergent key and hence the same cipher text. A secure proof_of_ownership protocol is also required to provide the proof that the user indeed owns. This is all to prevent 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[6]. The corresponding data users can download the encrypted files and also decrypt them by using their convergent keys. Therefore in cloud storage, CnvEn. allows the cloud to perform De-dup. 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-dup 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 of deciding which type of user is allowed to perform the duplicate check and access the files is performed at the time of

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

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[3]. The detailed system architecture is shown in figure 1.

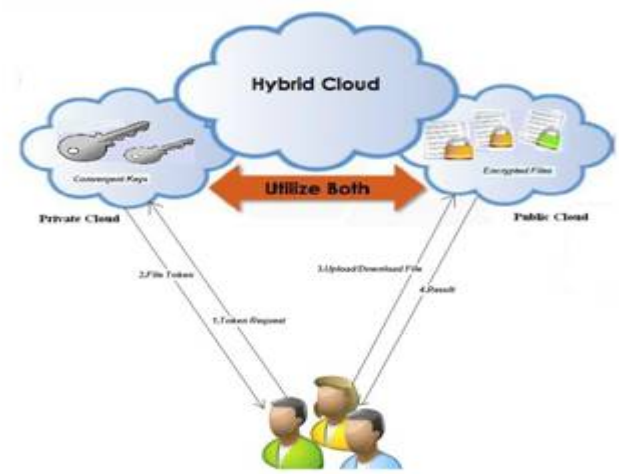

Fig. 1. Architecture of authorized deduplication[7].

B. Resemblance Detection Based Data Reduction

Data De-dup. is becoming increasingly popular in recent years. Specially in data-intensive storage systems as one of the most efficient data compression approaches. Fingerprint based De-dup techniques eliminate duplicate blocks by checking their secure-fingerprints (i.e., SHA-1/ SHA-256 signatures), which has been widely adopted in commercial backup storage systems.[1] Resemblance detection detects redundancy among similar data at the byte level while duplicate detection finds totally identical data at the block level, so in mass storage systems the latter approach is much more scalable than the former.

Resemblance detection with delta compression[12],[13], is one of the approach to data reduction, was proposed more than 10 years ago but was later overshadowed by
fingerprint-based deduplication due to the formers scalability issue. REBL[13] and DERD[12] are two super-feature-based resemblance detection approaches for data reduction. They calculates the features of the data stream (e.g., Rabin Fingerprints) and combine features into super-features to capture the resemblance of data and then delta compress the data. All these approaches for resemblance detection requires high overheads of computation and indexing. Shilane et al. proposed a stream-informed delta compression (SIDC) approach used in a WAN environment for reducing similar data transmission and thus accelerating data replication [11]. This approach is super-feature based and complements the block-level deduplication by only detecting resemblance among non-duplicate blocks in the cache that preserves the backup stream locality. It avoids the indexing which is very costly, at a limited loss of resemblance detection. While the combined detection of duplicate and resemblance promises to achieve a superior data reduction performance, challenges of maximum computation and indexing overheads stemming from resemblance detection remain[1].

B. The concept of duplicate adjacency

The modified blocks may be very similar to their previous versions in a backup system while unmodified blocks will remain duplicates and are easily identified by the De-dup process. For those non-duplicate blocks that are location-adjacent to known duplicate data blocks in a De-dup system, it is intuitive and quite possible that only a few bytes of them are modified from the last backup, making them potentially excellent delta compression candidates. Fig. 2 shows

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

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 excellent delta compression candidates. Specifically, since blocks 'B3' and 'B4' are duplicates of blocks '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

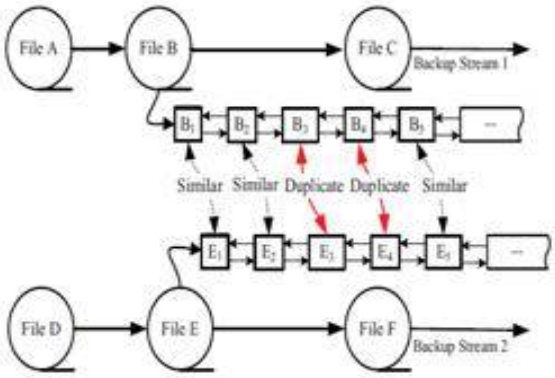

Fig. 2. A conceptual illustration of the duplicate adjacency. The non-duplicate chunks adjacent to duplicate ones are considered good delta compression candidates as they are potentially similar.[1]

considered excellent delta compression candidates, which is consistent with the aforementioned backup-stream locality.

III. SYSTEM ARCHITECTURE

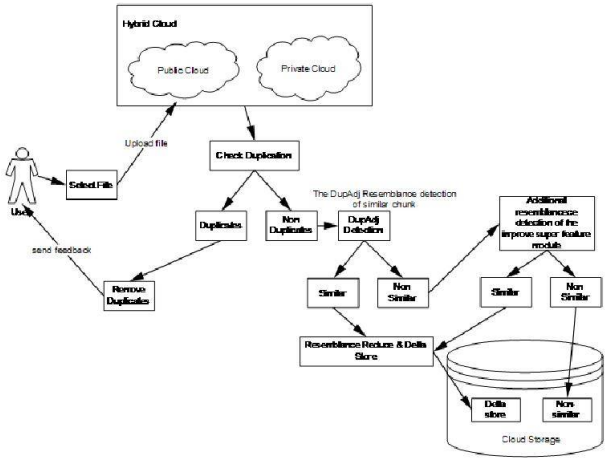

Fig. 3. The data reduction workflow of an authorized De-dup system for encrypted data with DARE scheme in a twin cloud environment.

The 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 identify and remove redundancy at minimum overheads by implementing DARE scheme.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

- Our proposed system consist of three parts :

1) The authentication scheme for data users is implemented in a private cloud server to prevent system from attackers.

2) CnvEn. algorithm is used to encrypt the data and perform duplicate check.

3) DARE approach is used to maximally identify and remove the redundancy at minimum overheads.

A. Secure Duplication with DARE scheme in a hybrid cloud environment:

The main problems in the cloud environment is data De-dup with differential privileges. The main aim of this paper is to solve this problem. For this we go with different type of architecture, which is having 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-dup check securely with differentials privileges[6]. Here, the key is managed by private cloud and only the data which is in encrypted from stored on public cloud. Under the twin cloud architecture we propose authorized duplication check separated by a new De-dup system. A user only with corresponding privilege on files has been allowed to perform download operation.

We also present DARE, a deduplication aware, minimum overhead resemblance identification and removal technique for delta compression in addition of De-dup on cloud storage system[1]. DARE uses a resemblance identification approach, DupAdj, which uses the duplicate-adjacency information already available in existing De-dup systems, for efficient resemblance removal and employs an improved super-feature approach to further detecting resemblance when the duplicate-adjacency information is lacking or limited.

C. System Modules

There are three entities define in our system :

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

- Data_Users: A data owner or user is an entity that wants to handover data storage to the third party storage provider and access the data later. User generate the key and store that key in private cloud. Each file is protected by CnvEn. key and can access by only authorized person. In our system user must need to register in private cloud for storing token with respective file which are store on public cloud. When he wants to access that file he access respective token from private cloud and then download his files from public cloud. Token consist of file data F and convergent key KF.

- Private_Cloud: Users give the preference to private_cloud in order to achieve more secrecy as compare to public 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 need not to store the secrete key internally. For providing proper protection to key we use private cloud. The Private_cloud simply store the convergent key with its respective file tag . When user want to access the key he first check authority of user and then only provide the key.

- Public_Cloud: Public_cloud entity is used for the storage purpose. User upload the files in public cloud. Public cloud is similar as S-CSP. When the user wants to download the files from public cloud, it will be ask the key which is generated or stored in private cloud. When the users key is match with files key at that time user can download the file, without key user can not access the file. Only authorized user can access the file. In public cloud all files are stored in encrypted format.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

IV. ALGORITHMS USED

A. Convergent Encryption Technique

In data deduplication, data confidentiality is provided by CnvEn algorithm. A user or owner of data generates a convergent key from each unique 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 identify duplicates. Here, we consider that if two data files are the same, then the tags of files are also 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 and its corresponding tag will be stored on server side [8].

Four primitive functions which defines the convergent encryp-tion scheme are as follows[6]:

1) KeyGenCE(M) - > K maps a datacopy M to a convergent keyK, so it is the key generation algorithm.[6];
2) EncCE(K, M) - > C takes both the convergent keyK and the datacopy M as inputs and then outputs a ciphertext C, so it is the SymmetricEencryption 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 datacopy M and outputs a tag T (M), so T (M) is the TagGgeneration algorithm[6].

B. Proof Of Ownership

The main purpose of proof_of_ownership (POW) protocol is to enable users to prove their ownership of datacopies to the StorageServer. 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 ownershipof the data copy M, the prover needs to send ϕ to the verifier such that $\phi'=\phi$ (M).[6]
PSEUDO CODE
Step1: Calculate the two convergent keyvalues 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.

C. 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 generates ciphertext as an output and keyword ciphertexts Ca on input of the owners public key pk and the file index i,

D. 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.

E. Data Deduplication(De-dup)

Data De-dup occurs when the data owner tries to store the same data that has been stored already at the PublicCloud storage. This is checked by comparing the tags. If the comparison is positive, CSP contacts Authorized Party for De-dup by providing the token and the data key[6]. The Authorized Party challenges data ownership, checks the eligibility of the data holder, and then issues a re-encryption key that can convert the encrypted DEK to a form that can only be decrypted by the eligible data holder[3].

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

F. Delta Compression

To remove data redundancy among similar blocks, 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[11]. In DARE, delta compression will not be applied to a block 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 blocks identified in resemblance detection, DARE reads its base-block, then delta encodes their differences. In order to reduce disk reads, an LRU and locality-preserved cache is implemented here to prefetch the base-blocks in the form of data segments.

G. Improved Super-Feature Approach

Traditional super-feature approaches generate features by Rabin fingerprints and group these features into super-features to detect resemblance for data reduction. For example, Featurei of a chunk (length = N), is uniquely generated with a randomly predefined value pair mi and ni and ai and N Rabin fingerprints as used in content-defined chunking as follows:

$$\text{Feature}_i = \text{Max}_{j=1}^{N} \{(mi * \text{Rabin}_j + a_i) \bmod 2^{32}\}. \quad (1)$$

A super-feature of this chunk, SFeaturex, can then be calculated by several such features as follows:

$$\text{SFeature}_x = \text{Rabin}(\text{Feature}_{x*k}, \ldots\ldots, \text{Feature}_{x*k+k-1}). \quad (2)$$

For example, to generate two super-features with k=4 features
each, we must first generate eight features, namely, features 0...3 for SFeature1 and features 4...7 for SFeature2. For similar blocks that differ only in a tiny fraction of bytes, most of their features will be identical due to the random distribution of the blocks maximal-feature positions[1]. Thus two data blocks can be considered very similar if any one of their super-features matches. The state-of-the-art studies on delta compression and resemblance detection recommend the use of 4 or more features to generate a super-feature to minimize false positives of resemblance detection.

V. 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 identification and removal 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.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

Step 10: Finally non similar data stored into secondary storage or Cloud storage.

Mathematical model contains five tuples –

S= {s, e, X, Y, ϕ }

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 ϵ 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 CnvEn algorithm.

T = Generate Tags for file blocks

M = MAC (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 blocks detected in R, and reads its base-chunk, then delta encodes their differences.

A. Success Condition

Successfully work keys aggregation, trapdoor generation, file splitting and stored into multi cloud. User gets result very fast according to their needs.

B. Failure Condition

1) Maintaining indexing leads to more time consumption to get the proper file stored in cloud storage.
2) Hardware failure.
3) Software failure.

C. Space Complexity

The space complexity depends on Presentation and visualization of discovered patterns. More the storage of data more is the space complexity.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

D. 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.

VI. System Implementation

We implement system with data De-dup, in which we model three entities as separate programs. To implement the data users a Client program is used to perform the file uploading/downloading process. A PrivateServer program is used to build the private cloud that manages the private keys and handles the file token computation. A StorageServer program is used to build the S-CSP which stores and De-dup files. Followings are function calls used in system:

- File_Tag(File) - It generates hash of the File by using SHA-1 called as File Tag;
- Dup_Check_Req(Token) – It generates requests for StorageServer for Duplicate Check of the file.
- File_Encrypt(File) - It uses 256-bit AES algorithm to encrypts the File with CnvEn. in cipher block chaining (CBC) mode and the convergent key is obtained from SHA-256 Hashing of the file;
- File_Upload_Req (FileID, File, Token) – It uploads the File Data to the StorageServer only if the file is Unique and updates the File Token stored.
- File_Store(FileID, File, Token) - It stores the File on PublicCloud and updates the Mapping.

VII. 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 De-dup and resemblance detection. Data-reduction throughput is measured by the rate at which datasets are processed, including deduplicating, identifying resemblance, and delta compressing. The similarity degree of resemblance identified blocks is measured by the ratio of (compressed size) / (original size).

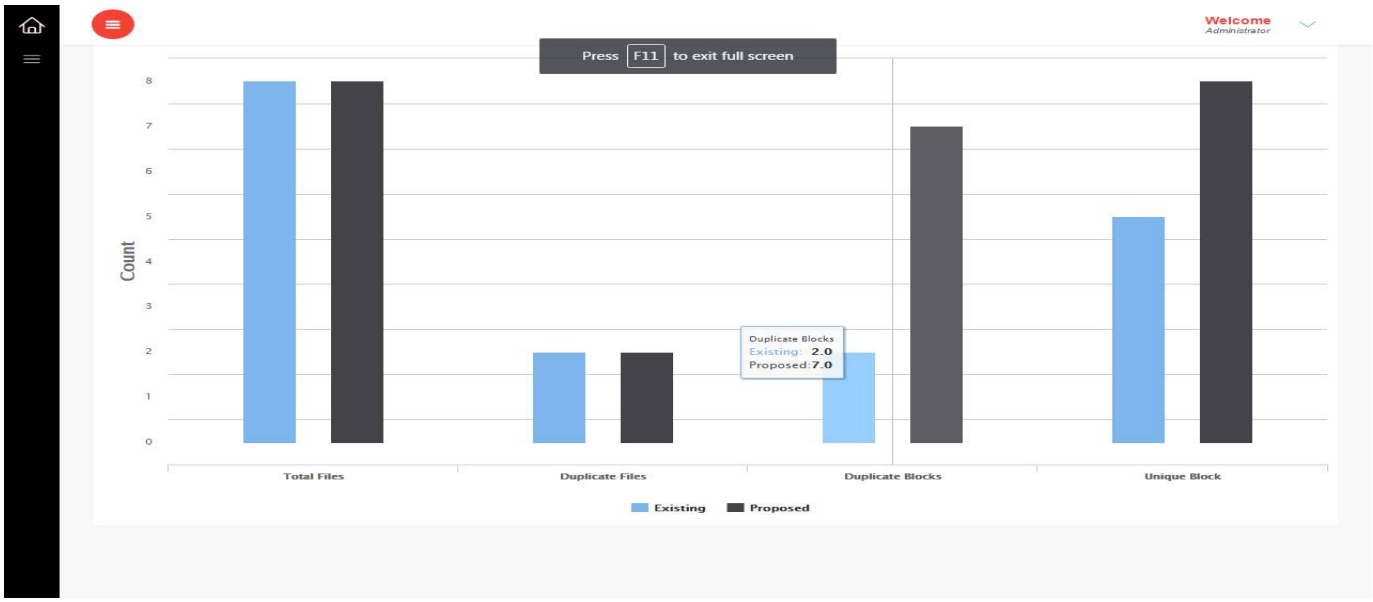

Fig. 4. Analysis graph shows that more number of duplicate blocks are found by DARE as compare to existing dedupliction techniques.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

The screenshot shown in Fig. 4. indicates that the number of duplicate blocks found in our system is much more than that of existing simple De-dup systems. The analysis graph demonstrates that resemblance identification is very efficient in supplementing De-dup for data reduction.

Fig. 5 shows that DARE achieves the highest throughputs among all the resemblance identification enhanced data reduction approaches compared running on both RAID structured HDDs and the SSD. DARE achieve lower throughputs than the SiLo/D approach, which is because SiLo/D only does deduplication (i.e., Rabin-based chunking and SHA1-based fingerprinting) while DARE involves more computation tasks and I/Os by delta compression (i.e., resemblance detection, reading base chunks, and delta encoding). Cached SF-4F has the lowest throughput because it incurs the largest computation overhead for resemblance detection. It is noteworthy that DAREs average data-reduction throughput on RAID, at 50 MB/s, is much lower than DAREs average throughput of 85 MB/s on SSD. The root cause of RAIDs inferior data-reduction performance (in Fig. 5 a) mainly lies in the random reads of the base-chunks.

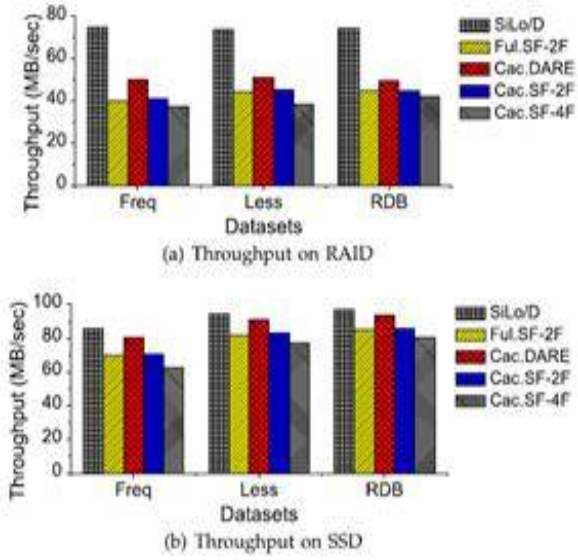

Fig. 5. Throughputs of four resemblance detection enhanced data reduction approaches (i.e., deduplication + delta compression) on the three backup datasets.[1]

The deduplication-only approach (i.e., SiLo/D) is also shown for comparison. due to the lack or limitation of accessing locality. This is the reason why DARE on SSD achieves a similar throughput to the deduplication-only approach (averaged 91 MB/s on SSD and 74 MB/s on RAID) while reducing more redundant data to be stored. In general, DARE achieves a superior performance of both throughput and data reduction efficiency among all the resemblance identification enhanced data reduction approaches. Furthermore and importantly, our analysis of the execution times by the four data-reduction steps based on data suggests that the average throughputs of resemblance detection on non-duplicate data of DARE, SF 2F, and SF-4F are 154, 60, and 30 MB/s respectively. The throughput results of SF-2F and SF-4F are consistent with the results. The main contributor to DAREs much higher resemblance detection throughput is its DupAdj based resemblance identification scheme. This suggests that DARE is shifting the bottleneck of data reduction from Step 2 to the other steps (e.g., Step 1 or 3), and the performance impact of this shifting will likely be much more pronounced when larger indexing cache is used and/or when the emerging storage class memory (SCM) devices (e.g., Nand Flash, PCM, STTRAM, etc.) replace or supplement HDDs.

ISSN(Online): 2319-8753
ISSN (Print): 2347-6710

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 6, June 2017

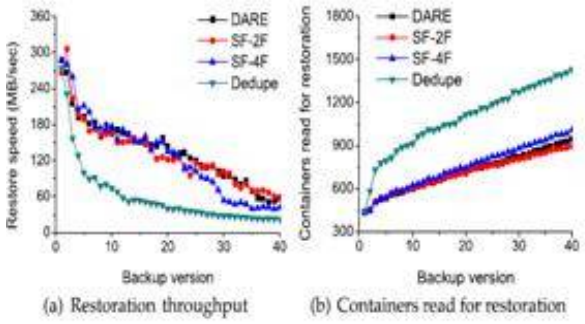

Fig. 6. Data-restore performance as a function of the backup version on the Linux dataset with an LRU cache of size 256 MB.[1]

VIII. CONCLUSION

In this paper to support stronger security, we are presenting an advanced authorized De-dup 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 powerful in identifying and eliminating redundancies at maximum level and with minimum overheads. The system effectively manages the storage space in a secure and authorized manner. And the system enables to maximally identify and remove redundancy with minimum overheads supporting the secure data storage on public cloud.

ACKNOWLEDGMENT

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.

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] Bo Mao, Member, IEEE, Hong Jiang, Fellow, IEEE, Suzhen Wu, Member, IEEE, andLei Tian, Senior Member, IEEE Leveraging Data Deduplication to Improve the Performance of Primary Storage Systems in the Cloud IEEE TRANSACTIONS ON COMPUTERS, VOL. 65, NO. 6, JUNE 2016
[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] B. Zhu, K. Li, and R. H. Patterson, Avoiding the disk bottleneck in the data domain deduplication file system, in Proc. 6th USENIX Conf. File Storage Technol., Feb. 2008, vol. 8, pp. 114.
[10] D. Meister, J. Kaiser, and A. Brinkmann, Block locality caching for data deduplication, in Proc. 6th Int. Syst. Storage Conf., 2013, pp. 112.
[11] 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.
[12] 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.
[13] 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. 59–72.