EFFICIENT APPROACH FOR DEDUPLICATION AND RESTORATION USING REFERENCES AND MULTIPLE DRIVES IN CLOUD STORAGE

[1]K. Sai Krishna, [2] C. Shoba Bindu, [3] P. Dileep Kumar Reddy
[1]M.Tech(CS), Department of CSE, JNTUACEA, Anantapur, India
[2]Professor, Department of CSE, JNTUACEA, Anantapur, India
[3]Asscoaite Professor, Department of CSE, SVCE, Karakambadi Road, Tirupati, India

Abstract: Cloud storage allows a user to store large quantities of data, which can be accessed remotely. Even though cloud storage allows user to store large amount of data at low cost, but the cost of maintenance is high. Data duplication is primary concern in cloud storage, in which multiple copies of same data is maintained which leads to storage and security issues. Data deduplication technique allows a user to eliminate duplicate copies of data and remains only a single copy of data. Attribute Based Encryption (ABE) is a technique in cloud which uses public key encryption type in which the cipher text and secret key of a user depends upon attributes. If a user uploads a file, that file will be encrypted with a certain access policy but not with user's credentials so that the duplicate files are not uploaded into the cloud storage, in which only a user will be allowed to store data in cloud storage, but if the file is lost it cannot be recovered. To overcome this problem and to achieve efficient deduplication anEfficient Deduplication and Restoration techniques using References and Multiple Drives (EDRRM) is proposed, EDRRM provide references of the file to remaining users and storing it in multiple drives.

IndexTerms **- Cloud storage, Data Deduplication, Storage System, Performance evaluation, Attribute Based Encryption (ABE).**

I. INTRODUCTION

Every day a large amount of data is generated and the data needs to be stored securely. Cloud storage allows a user to store large amounts of data which is maintained, managed, backed up remotely and made available to users over a network. To store large amount of data in cloud storage is of low cost but the cost of maintaining it is very high. In cloud storage different users of either same organization or different organizations may store same data which leads to duplication problems and security issues. To eliminate this problem deduplication [7, 8] technique is used, which allows us to eliminate multiple copies of a single file.

For a user to access his/her data in cloud storage, he/she has to provide credentials (or Attributes). Attribute based storage (ABS) supports secure deduplication and Attribute based encryption (ABE). ABE [9] is one of the techniques in cloud, it provides security by encrypting the data with access policies. ABE is one of public key encryption type in which the cipher text and secret key of a user depends upon attributes. If a user uploads a file, that file will be encrypted with a certain access policy but not with user's credentials so that the duplicate files are not uploaded into the cloud storage. [1] is built using a hybrid cloud architecture in which private cloud manages duplicate files and public cloud manages storage. In [1] the data will be encrypted with access policy rather than with user's attributes, by doing there are two advantages first even if different user's provided same data, after encryption the

encrypted cipher text will be same for every user and it is easy to identify and eliminate the duplicate data, second the first user can provide access policy rather than decryption key to the second user, which also provides confidentiality. But in this if a user has uploaded a file then that file can never be uploaded by another user which is not a good deduplication policy and if the original file is deleted that file will never be found or recovered.

In order to overcome the above mentioned problem we propose Efficient Deduplication and Restoration using References and Multiple Drives in cloud storage (EDRRM), in which first we store the data in multiple drives so that even if a data is lost in one drive we can recover it from another drive. Second, we will be providing reference to the second user who uses same file. By doing so we will achieve data deduplication as well as allow the second user to access the file.

II. RELATED WORK

In [1] if once a data provider stores a file into cloud storage we will to be able to store that file again into cloud storage even if it is from anther data provider because it is considered as a duplicated data file and deduplication technique will not allow it. So to overcome this issue these are the present available solutions

RevDedup [2, 3] an efficient deduplication method which allows the cloud storage to delete, restore and back up the files, while also maintaining high storage efficiency. It follows Reverse deduplication, the main purpose of this is that it will remove old back up data available in cloud storage and reduce latest back up files achieving deduplication. It stores the data by using chunking algorithms, to keep track of segments and chunks that are shared. The main drawback by using this is that we will not be able maintain the backup of the files, because if a chunk of a data is lost we will not be able to again access that data because it will become an invalid data.

Cost Aware cloud Back-up system (CAB) [5]first connects multiple devices as a single device, which allows the user to store data from multiple connected devices so that the cost of storing the data into cloud will be reduced by a large margin and maintaining data deduplication and increasing data transfer rate. Here the data will be divided into multiple blocks and is distributed among the devices which are connected and is uploaded to cloud storage because a file of small size costs much less to store in cloud storage than to store a large file. By doing this we can save the data at a relatively cheaper cost, but when uploading data if a device got disconnected due to some problem we will not be able to store it successfully, we have to start the process again which leads to bigger costs. If a data from device is lost then the complete data becomes invalid data

ABE [1, 6, 10] technique highly used in cloud computing, in which data providers can store and share their data to another data provider or user with particular credentials. Deduplication is a technique which helps in eliminating unnecessary copies of a file in order to maintain network bandwidth and save storage space. To apply deduplication technique in ABE we will use hybrid cloud in which deduplication and computation is managed by private cloud whereas storage issues are managed by public cloud. The trap door key and cipher text are provided by private cloud, in which cipher text of one access policy can be transformed to cipher text of same using different access policies without knowing original text. It checks the validity of the file as soon as it receives the storage request if it is a valid and not a duplicate file, file will be sent to public cloud for storing it, if not it is discarded. The main drawback of this is that if a file is lost from cloud storage it is permanently lost we can't again access it.

DARE [4, 9] is a low overhead deduplication detection system which performs deduplication operation at chunk level. DARE uses duplicate adjacency technique and delta compression, Duplicate adjacency technique main object is to find highly similar data chunks in a which helps in finding the similar data chunks available and deleting them thus allowing it to gain high data processing speed. Delta compression approach removes redundancy between two similar data chunks and stores only the difference between them while mapping two similar data chunks. The main drawback of this is that it only considers the highly similar data chunks and eliminate them but if the data chunks has low similarity it will not consider them which can lead to data duplication problem.

III. PRELIMINARIES

Symmetric encryption

A symmetric algorithm consists of both encryption and decryption algorithms. Encryption algorithm takes message space M and key K as input and generates cipher text CT as output.

$$Adv_{S\varepsilon,A}^{IND-CPA}(\lambda) = \Pr\left[b' = b \;\middle|\; \begin{array}{c} K \leftarrow K; b \leftarrow \{0,1\} \\ (m_0, m_1, st) \leftarrow A_1(1^\lambda) \\ CT^* \leftarrow s\varepsilon.Enc(K, m_b) \\ b' \leftarrow A_2(pup, m_0, m_1, st, CT^*) \end{array}\right] - 1/2$$

The above equation depicts both the encryption and decryption performed.

IV. EFFICIENT DEDUPLICATION AND RESTORATION USING REFERENCES AND MULTIPLE DRIVES (EDRRM)

4.1 Architecture of EDRRM

Our EDRRM scheme mainly depends upon the components such as Attribute Authority (AA), Data Provider (DP), User, Cloud Storage. When a data provider uploads or stores data into the cloud storage it will be first encrypted by using access policies and then check the encrypted file whether it is already uploaded by another data provider, if it is uploaded by another one it will only take reference of that file and store the reference under the new data provider. User can only access the available files in cloud storage by getting permission from data provider and get key from attribute authority. The cloud storage has both private cloud and public cloud, private cloud will encrypt the data provided by DP using access policies and check if it is duplicate or not if it is a duplicate file it will save a reference for that file under that data provider, Attribute authority is responsible for either giving or denying file access which a user has requested, it also generates key to facilitate an authorized user to download a file. Reference Index in our cloud storage contains references of already available files uploaded into cloud storage by different data provider so that the new data provider can also have access that file, and it is managed by public cloud. If a particular file is lost we will maintain the backup of that file in multiple drives so that we can easily restore the file.

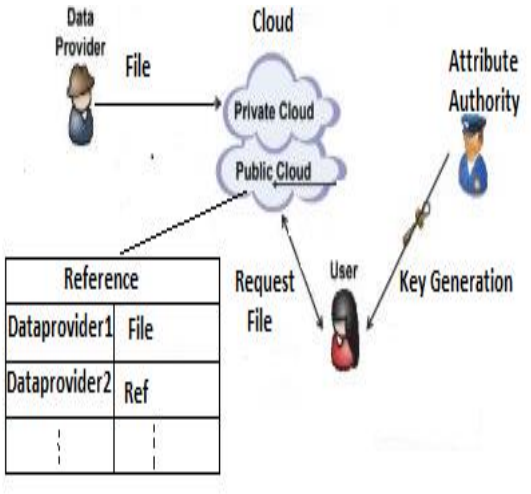

Fig 1: Architecture of EDRRM

4.2 Algorithm

Setup (1^{λ}) → (pupa, mk): The setup phase is mainly responsible for generating master key. This takes security parameter as input and gives master key (mk) and public parameters (pupa) as output. This action is performed by Attribute Authority.

Encrypt (pupa, M, AP) → (TD_A, CT): The encrypt phase is mainly responsible for encrypting the data and send it to private cloud. It takes public parameters, message and access policy as input and generates trapdoor key (TK) with label as an output. The TK cannot be disclosed. This action is performed by the Data Provider and is sent to the private cloud.

Validity Check (pupa, CT) → 1/0: This phase is mainly responsible to check whether the file is a duplicated file or not. If it is a duplicate file it will give 0, if not it gives 1 as output. This action is performed by Private Cloud.

Re-encrypt (pupa, TD_A, (L, ct)) → (L, ct'): This phase is mainly responsible re-encrypting the data provided by data provider. It takes trapdoor key and access policy as an input and gives cipher text as output which is sent to public cloud. This action is performed by Private cloud.

Key Generation (pupa, mk, AS) → Sk_A : This phase is mainly responsible for providing key to user. If a user requests access to a file, if he has permissions to access that file he will be given a private key with attribute set to download it else he will be denied access. This action is performed by Attribute Authority.

Decrypt (pupa, (L, ct), AS) → M: This phase is mainly used to decrypt the downloaded data by using the key generated by attribute authority and secret key of user to obtain the original file. This action is performed by User.

V. PERFORMANCE EVALUATION:

In EDRRM performance will be evaluated based on Restoration throughput. DARE, Dedupe uses logical content to be divided into data chunks and will be saved in different locations, mainly DARE uses delta compression to store data which should lead to faster restoration of data but as it takes data chunks the data restoration speed will decrease if there is any interruption while performing restoration operation.

EDRRM stores data only under one user and give that data reference to other users by doing this we will be able to able to achieve deduplication and even if that data is lost, it can be restored from another drive where that data is stored.

The fig 2 shows the restoration throughput of EDRRM is high when compared with DARE and Dedupe as we are using only References instead of using complete data to restore the data.

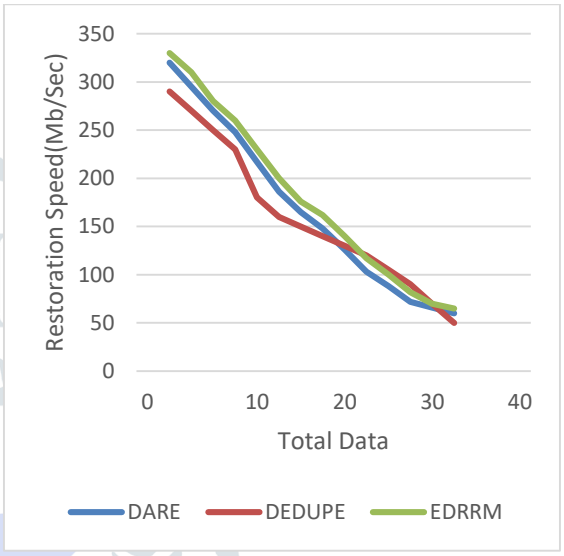

Fig 2: Restoration Throughput

VI. CONCLUSION

This paper proposes EDRRM scheme to achieve deduplication and recover any lost data in cloud storage. When users upload files into cloud storage, it is encrypted with access policies. If the encrypted file of different users is same, then the scheme stores only the original file of first user who uploaded it first and the file reference of the first user is provided to the authenticated users who uploaded it later. As we are sharing only the reference of a file but not a complete copy of the file, deduplication is achieved. Multiple drives are used to store data of all users to maintain backup. If there is data loss due to any reason in cloud storage EDRRM restores it at high speed, because there is only one complete file and the remaining are all references of that file which improves restoration speed.

VII. REFERENCES

[1] H. Cui, Robert H. Dong, Y. Li and G. Wu "Attribute-Based Storage Supporting Secure Deduplication of Encrypted Data in Cloud." IEEE Transactions on Big Data, Volume: PP, Issue: 99. Available: http:// 10.1109/TBDATA.2017.2656120

[2] Z. Yan, W. Ding, X. Yu, H. Zhu, and R. H. Deng, "Deduplication on encrypted big data in cloud," IEEE Trans. Big Data, vol. 2, no. 2, pp. 138–150, 2016. [Online]. Available: http://dx.doi.org/10.1109/TBDATA.

[3] Anand Bhalerao, Ambika Pawar, "A Survey: On Data Deduplication for Efficiently Utilizing Cloud Storage for Big Data Backups" International Conference on Trends in Electronics and Informatics ICEI 2017, pp. 933-938. Available: http:// 10.1109/ICOEI.2017.8300844.

[4] Wen Xia; Hong Jiang; Dan Feng; Lei Tian" DARE: A Deduplication-Aware Resemblance Detection and Elimination Scheme for Data Reduction with Low Overheads" IEEE Transactions on Computers Year: 2016, Volume:65, Issue: 6 Pages: 1692 – 1705. Available: http://10.1109/TC.2015.2456015.

[5] Yingwu Zhu; Justin Masui. "Backing Up Your Data to the Cloud: Want to Pay Less?" 2013 42nd International Conference on Parallel Processing Year: 2013 Pages: 409 – 418. Available: http:// 10.1109/ICPP.2013.50.

[6] L. Cheung and C. C. Newport, "Provably secure cipher text policy ABE," in Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007, Alexandria, Virginia, USA, October 28-31, 2007. ACM, 2007, pp. 456–465. Available: http:// 10.1109/CloudCom.2007.45.

[7] J. Stanek, A. Sorniotti, E. Androulaki, and L. Kencl, "A secure data deduplication scheme for cloud storage," in Financial Cryptography and Data Security - 18th International Conference, FC 2014, Christ Church, Barbados, March 3-7, 2014, Revised Selected Papers, ser. Lecture Notes in Computer Science, vol. 8437. Springer, 2014, pp. 99–118. Available: http:// 10.1109/PST.2014.6297923.

[8] M. Lillibridge, K. Eshghi, and D. Bhagwat, "Improving restore speed for backup systems that use inline chunk-based deduplication," in the 11th USENIX Conference on File and Storage Technologies (FAST). San Jose, CA, USA: USENIX Association, February 2013, pp. 183–197. Available: http:// 10.1109/MASCOTS.2013.46.

[9]Achieving Efficient and Privacy-Preserving Cross-Domain Big Data Deduplication in Cloud: DOI 10.1109/TBDATA.2017.2721444, IEEE Transactions on Big Data.

[10] Z. Yan, W. Ding, X. Yu, H. Zhu, and R. H. Deng, "Deduplication on encrypted big data in cloud," IEEE Trans. Big Data, vol. 2, no. 2, pp. 138–150, 2016. [Online]. Available: http://dx.doi.org/10.1109/TBDATA.