VOL. 13, NO. 5, MARCH 2018 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

DEDUPLICATION IN CLOUD STORAGE USING HASHING TECHNIQUE FOR ENCRYPTED DATA

Vaishnavi Moorthy, Arpit Parwal and Udit Rout
Department of Computer Science Engineering, SRM University, Chennai, Tamil Nadu, India
E-Mail: arpitparwal@hotmail.com

ABSTRACT

Cloud storage services are widely used, due to which the volume of data on the cloud is very large. In order to avoid data redundancy and to make use of cloud storage efficiently, data deduplication is used which eliminates the storage of redundant data. Only one unique copy for each duplicate file uploaded in the cloud and the owners are referenced to that one file whenever they want to access. The objective of this paper is to automate the deduplication process while increasing confidentiality and security of the data owner and the data. The system uses user end encryption so that cloud services are not liable for data compromise. An effective comparison using hash value of different files helps in faster and secure comparison.

Keywords: encrypted data, data owner, cloud storage, hashing, cloud service provider, deduplication, encryption.

1. INTRODUCTION

The rise in the usage of cloud computing due to lower cost of operation, scalability and easier access has paved the way for wider use of cloud storage services. Cloud storage services helps user to secure its data from local hazard as well as improve portability. The same concept also helps in sharing data with different users and work on same project simultaneously. This leads to a major problem among Cloud Service Provider (CSP) of same data getting uploaded to their storage servers. This results in loss of storage memory and makes access to users slower. Compression techniques used previously have proved unsuccessful in maintaining the integrity of data being uploaded on the cloud storage servers. Thus, in order to make cloud storage more efficient data deduplication was proposed by Zheng Yan, *et al.* suggested deduplication based on ownership challenge and proxy re-encryption. This system used an extra external system that ensures data owner's verification and make cloud service provider as a proxy between the two. It included assumptions that the external party will remain free from any collusion with any other user or CSP itself [1]. This highly elevates the time of uploading a data and thus cannot be deployed for Big Data purposes. The intended system, let user uploads its data which gets encrypted on his machine as a hidden process and then get transferred to CSP's server. The data can then easily be compared with existing files by computing there hash values and comparing them. This removes third party involvement and helps in easier process flow between the two parties involved.

Deduplication has proved to achieve high cost savings, e.g., reducing up to 90-95% storage needs for backup applications, as in [2]. It has also proved space saving of up to 68% in standard file systems [3]. The new system will also provide access of data by customers who want to view the file by allowing data owners to approve their request using a secured One Time Password like service on mail.

This will ensure data owner privacy and their data integrity while improving data sharing among known colleagues or customers.

Current industrial solutions have achieved deduplication at the cost of extra resources or by complex process. In this paper, it is proposed a simple data flow model which keeps the architecture of our system very simple but at the same time it doesn't compromise on speed and integrity of both data and data owners.

The paper is organized into various sections viz. Section II gives the problem statement. Section III introduces the schemes used in the system to achieve the solution. Section IV describes about the data flow and architecture of proposed system. Section V elaborates inferences with security evaluations and time efficiency. Finally, future scope and advances is presented as conclusions.

2. PROBLEM STATEMENT

Current industrial Cloud Service Providers like Dropbox [4], Google Drive [5], Mozy [6], and other local CSPs provide deduplication by saving only once copy of file uploaded. However, this may not work if data owners encrypt their data using different encryption. This will lead to different cipher text resulting in reduced efficiency of data deduplication. DeDu [7] being the prime in deduplication is unable to segregate encrypted data.

a) Data privacy and security

The new system proposes a scheme to encrypt the data at the Data Owner machine itself. This way the data that is being transmitted to CSP will be encrypted. This leaves us with two major advantages over the previous techniques used viz 1) Data Owners are safeguarded with any data theft that may take place during transmission of data from Data owner machine to CSP's servers. 2) CSP has less liability as they are not the one who is encrypting data. This will help them in distracting attackers. The data stored in the cloud will be accessible only if they are

registered as owners of the data. This process will be handled by data right grant module which will challenge the requesting user to prove its ownership as claimed. Here there are only two actors involved in the scenario. One will be the group of users who have the data. They have been named as data owners. The other actor is the CSP. This is a many to one relation as there will be many data holders who will be uploading and downloading data to and from CSP respectively.

The key used to encrypt the data will be stored such that it is also not visible to user. The key will be securely put into use whenever the data holder wants to upload any data to the servers of CSP. The encryption will be done as soon as the user clicks 'upload'. This way we can ensure that the user is not able to view the key and at the same time is uploading his data securely to the servers of CSP.

b) Data sharing

Data sharing is an important aspect for any CSP. It provides its customers an advantage as the Data owner doesn't have to individually send file to the designated customers. Instead they can upload it to their cloud and allow other users to download it from there. This implementation needs special security arrangement in order to allow only genuine users to download the file.

It is suggested that the data customers get themselves registered in order to access the CSP's interface. There it can ask for the required file from the owner. The owner grants or rejects the customer request. If the data owner have allowed the data customer will get an access key. This will ensure that only designated customers are able to access the file.

Additional assumptions include: Data holder is providing the data uploaded without any further or different encryption from his side. If data holder is already encrypting data using his personal encryption, then that would result in generation of different cipher text even for the same data. Data users and CSP communicate through secure channel with each other. This way the CSP will be able to authenticate its users. Data theft will not be a problem here as the data is encrypted during communication. Thus, it reduces the liability of CSP when it comes to privacy issues.

3. ENHANCED SCHEME

To perform deduplication securely, we have following main aspects:

A. Data encryption & upload

The data to be uploaded needs to be encrypted before sending it to the server of CSP. This is to ensure that data integrity and privacy of user is maintained. Even the server where the data is stored is not able to view the contents of data. Data encryption takes place at the user point but the user is unaware of the key used to encrypt the data. The data is encrypted by using secure key Sk which is stored inside the uploader's machine but is inaccessible to the user.

Algorithm 1: Duplicate Data Check.

Data Owner 'u_i' login to the CSP interface using his credentials.

Data Owner 'u_i' chooses his file 'm' in the upload interface.

Once user clicks the upload button the machine starts encryption of the file 'm' and generates E (m).

This E (m) gets passed on the CSP servers using secure connections.

The cipher file E (m) is then computed for its hash value H (E (m)) and is compared with all other files already stored in the cloud. H (E (m_j)) where j= no. of files ∈ (1…N).

If H (E (m)) = H (E (m_j)), then the file is duplicate and, the file is discarded. The reference path of the uploaded file E (m) is set as same as path location as of E (m_j) in the index table.

The index table maintains record of all the files inserted. All the files are checked for its content by comparing its hash value to the file that is to be uploaded. The value for different data will therefore be different and for matching data the hash value will be same. This is necessary as the data size is too big and can't be checked for small chunks. The files need to be exactly same in content. Only then will be it having same hash value. Here the CSP is able to compare encrypted file

Data Owner(U_i)	CSP	CSP Storage
Credential Initiation U_i logins to the CSP portal selects the file'm' to be uploaded		
Encryption Computed E(m) is passed to CSP	**Hash Computation** Computer H(E(m))	
	Hash Comparison H(E(m)) compared to H($E(m_j)$)	
Comparison result is positive(File is duplicate)		
	H (E (m)) is not uploaded on CSP storage. Reference is added in index table	
Comparison result is negative(File not duplicate)		
		File E(m) is stored and reference is added for U_i

Figure-1. Data encryption and upload.

B. Sharing files and granting access

For a Data Customer C_i who wants to access a specific file m_i can do that by first registering himself in the CSP server as Data Customer. Next he can search for the desired file and can ask corresponding Data Owner U_i to grant him permission to view the file. The CSP sends the request on to the Data Owner portal, where he can readily accept or decline based on his choice.

If the Data Owner U_i grants access then the CSP will generate a random key K_r and send it to C_i. This will ensure that only authentic customers are able to access the files. The reference of the key $K_{r\ and}$ the Data Customer C_i will be stored in the reference table against the file m_i

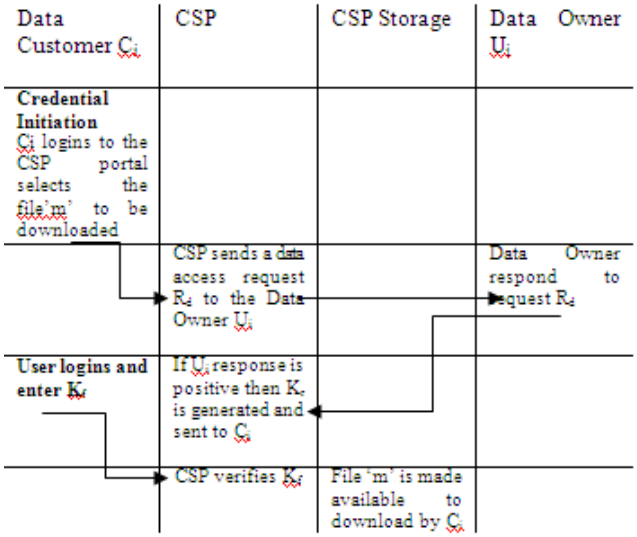

Figure-2. Sharing files and granting access.

C. Deletion of data

When data owner U_i requests for deletion of data from CSP, it will check the credentials and block access of the user U_i for the file. This way it ensures that other user using the file still can use it even if they were not the original user to upload the file. Deletion will not necessarily means deleting of files if there are multiple users for the same. If the request to delete the file is raised by any ordinary data holder, then his record will be deleted from the index table. Even for the users who have originally uploaded the data and are data owners will get their access blocked but the file will stay in the server as long as there are other users using the same file.

4. SYSTEM ARCHITECTURE

The main components of the proposed system are:

a) **Data owner:** the interface used by the client to use the cloud storage service.
b) **Cloud:** The server of the cloud service provider (CSP) where operations such as deduplication check using hashing is carried out and the data is stored.

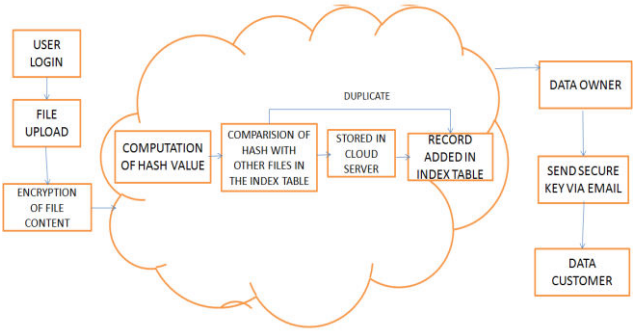

Figure-3. System architecture.

The end user after using the credentials logs into the interface on the CSP webpage. The data is uploaded into the cloud from the client machine by the end user of the CSP and once the data is encrypted and then the data is sent to the CSP server and deduplication is performed. Then, the data owners can view or edit the data but the data customers can only request permission to view the data through the privilege grant module and the random-access key provided to them by the CSP, once their request to view the document is accepted by the data owner.

a) Data flow

The end user interacts with the front end or the website of the cloud storage service provider. During, the upload process a \small application is downloaded on the client machine to encrypt the data with AES 128-bit encryption using a 15 bit private key. Once the data is encrypted the data is broken down into smaller chunks depending on the chunk size for the various data types, specified in the backend of the system. The hash values are the calculated for the data chunks using MD5 hashing technique. If the data chunk which is being uploaded has the same hash value as the data which is already exists in the cloud, then the status of the data is uploaded as duplicate in the index table which is present in CSP database and the location of the existing file is mentioned in the index table, whereas if the data chunk is not a duplicate then the status is set as original and the file upload function is run to upload the new data into the cloud, and the location of the new data is updated on the cloud along with a file id for reference in the table. The index table is only visible to the database administrator in case a manual operation must be carried out on the files on the database.

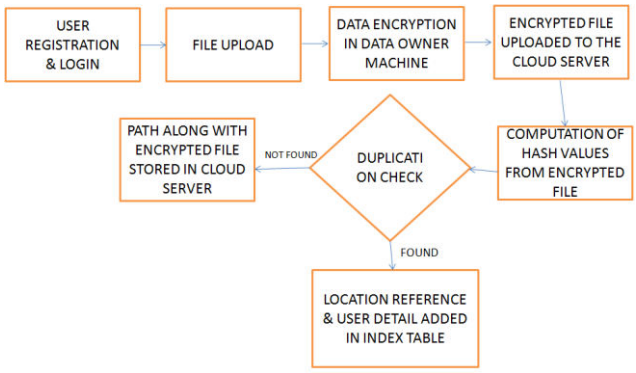

Figure-4. Data flow diagram.

5. RESULTS INFERENCE

a) Security analysis

The proposed schemes implement a secure approach to data deduplication. CSP doesn't have access to the plain text any time as the file is getting encrypted from the Data Owner side. The CSP computes Hash value of cipher text.

CSP and Users while functioning, without collusion guarantees that data is never compromised at the cloud storage. The data is being encrypted before reaching to CSP. Thus, CSP won't be able to know the actual data in 'm'. This is ensured as user won't share the actual data with the CSP.

CSP storage will have only the data that is being passed on to CSP. This ensures that only E (m) data is getting stored. In case CSP Storage is compromised, the attacker won't be able to get actual data as only Data Owner will be able to read it again using the CSP interface provided in his machine.

b) Data encryption and decryption efficiency

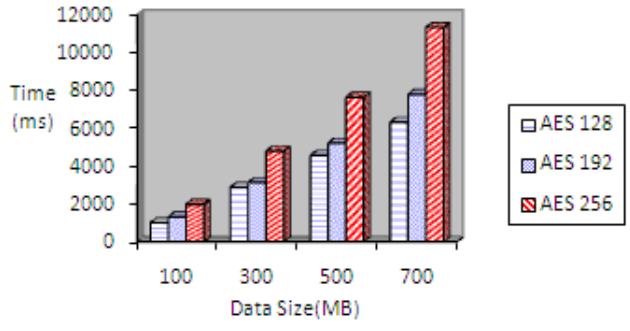

Figure-5. Period analysis of AES algorithm.

In this experiment, the time taken by various encryptions standard of AES is demonstrated. It was found that greater the bit size the encryption takes even more time to encrypt file.

Calculating the number of outputs that is expected out of different AES standard it was found that 2^128 different possibilities of output is there if AES 128 bit encryption is used. This using Birthday attack, a type of brute force attack will take 2^64 combinations to break. Achieving this amount of uniqueness and speed made AES our first choice majorly for getting lower encryption - decryption time as compared to others. This will highly reduce the amount of time spent in the entire process when huge data files are to be considered.

During the functional testing of the proposed system the following conversion of user's file 'm' into E (m) is shown.

Functional test of Data Customer requesting for data access rights and sending the Data Customer secret random key K_r is displayed. The Data Customer enters the key in CSP portal and is able to access the file.

6. CONCLUSIONS AND SCOPE

The system still has a scope of using a Secure Hashing Algorithm (SHA) and digital signature verification to check the authenticity of the data uploaded by the user. The system for data deduplication could be migrated to other platforms such as smartphones, Personal Computers (PCs), etc. This system cloud is used in other web services such as messaging services, web hosting, etc. to make them system more efficient and handle more data volumes. The proposed system can use more advanced version of secured hashing and encryption methods to perform this function.

Apart from these, the proposed system has the following added flexibility. The resolved system is efficient in implementing de duplication in big data files too. The hashing algorithm can be taken of greater bits like SHA1 and SHA2 in order to regulate collision resistance. This factor defines probability of finding same hash values for different data in files. The process that is put forth follows a very simple to understand and implement approach. This fact will result in wider acceptability of the model for use. This simplicity is achieved without compromising on privacy and data integrity. Deduplication of encrypted data is very useful in order to make a cloud service successful... In case of big data analytics and other industries where file sizes are huge; our proposed system will bring out simplicity in carrying of these operations. Further the functional implementation of our project brings out the practicality in it.

Future work includes testing with more secure encryption and hash algorithm. Securing CSP database where index tables are stored is an option that could be explored for the system.

REFERENCES

[1] Zheng yan, Wenxiu Ding, Xi Xun Yu, Haiqi Zhu,and Robert H. Deng. Deduplication on Encrypted Big Data in Cloud. IEEE Computer Society.

[2] Opendedup.http://opendup.org

[3] D.T. Meyer and W.J. Bolosky. 2012. A study of Practical Deduplication. ACM Transactions on Storage. 7(4): 1-20, doi: 10:1145/2078861.2078864

[4] Dropbox. A File-Storage and Sharing Service. http://www.dropbox.com Google Drive, http://drive.google.com

[5] Mozy. Mozy: A File-Storage and Sharing Service: http://mozy.com

[6] Z.Sun, J.Shen and J.M. Yong. 2011. DeDu: Buiding a Duplication Storage System over Cloud Computing. Proceedings of IEEE International Conference on Cloud Computer Supported Cooperative Work in Design. pp. 348-355, doi:10.109/CSCWD.2011.5960097.

[7] J. Li, Y.K. Li, X.F. Chen, P.P.C. Lee, and W.J. Lou. 2015. A Hybrid Cloud Approach for Secure Authorized Deduplication. IEEE Transactions on Parallel and Distributed Systems. 26(5): 1206-1216, doi:10.1109/TPDS.2014.2318320.

[8] P. Meye, P. Raipin, F. Tronel, and E. Anceaume. 2014. A Secure Two-Phase Data Deduplication Scheme. Proceedings of

[9] HPCC/CSS/ICESS. pp. 802-809, doi:10.1109/HPCC.2014.134.

[10] J. Paulo and J. Pereira. 2014. A Survey and Classification of Storage Deduplication Systems. ACM Computing Surveys. 47(1): 1-30, 2014, doi:10.1109/HPCC.2014.134.

[11] Y.-K. Li, M. Xu, C.-H. Ng and P.P.C. Lee. 2014. Efficient Hybrid Inline and Out-of-Line Deduplication for Backup Storage. ACM Transactions on Storage. 11(1): 2: 1-2: 21, doi: 10.1145/2641572.

[12] M. Fu, D. Feng, Y. Hua, X.B. He, Z.N. Chen, W. Xia, F.T. Huang and Q. Liu. 2014. Accelerating Restore and Garbage Collection in Deduplication-Based Backup Systems via Exploiting Historical Information. Proceedings of USENIX Annual Technical Conference. pp. 181-192.

[13] M. Kaczmarczyk, M. Barczynski, W. Kilian, and C. Dubnicki. 2012. Reducing Impact of Data Fragmentation Caused by In-Line Deduplication. Proceedings of the 5th Annual International Systems and Storage Conference. pp. 15: 1-15: 12, doi:10.1145/2367589.2367600.

[14] M. Lillibridge, K. Eshghi, and D. Bhagwat. 2013. Improving Restore Speed for Backup Systems that Use Inline Chunk-Based Deduplication. Proceedings of USENIX Conference on File and Storae Technologies. pp. 183-198.

[15] L.J. Gao. 2015. Game Theoretic Analysis on Acceptance of a Cloud Data Access Control Scheme Based on Reputation. Master thesis, Xidian University.

[16] Z. Yan, X.Y. Li, M.J. Wang, and A.V. Vasilakos. 2015. Flexible Data Access Control Based on Trust and Reputation in Cloud Computing. IEEE Transactions on Cloud Computing. doi:10.1109/TCC.2015.2469662.

[17] C. Yang, J. Ren, and J.F. Ma. 2013. Provable Ownership of File in Deduplication Cloud Storage. IEEE Global Communications Conference. pp. 695-700, 2013, doi:10.1109/GLOCOM.2013.6831153.

[18] T.Y. Wu, J.S. Pan, and C. F. Lin. 2014. Improving Accessing Efficiency of Cloud Storage Using Deduplication and Feedback Schemes. IEEE Systems Journal. 8(1): 208-218, doi:10.1109/JSYST.2013.2256715.

[19] C. Fan, S.Y. Huang, and W.C. Hsu. 2012. Hybrid Data Deduplication in Cloud Environment. 2012 International Conference on Information Security and Intelligence Control (ISIC). pp. 174 -177, doi:10.1109/ISIC.2012.6449734.

[20] J.W. Yuan and S.C. Yu. 2013. Secure and Constant Cost Public Cloud Storage Auditing with Deduplication. IEEE 2013 International Conference on Communications and Network Security. pp. 145-153, doi:10.1109/CNS.2013.6682702.

[21] N. Kaaniche and M. Laurent. 2014. A Secure Client Side Deduplication Scheme in Cloud Storage Environments. 2014 6th International Conference on New Technologies, Mobility and Security (NTMS), pp. 1-7, doi:10.1109/NTMS.2014.6814002.

[22] Z. Yan, W.X. Ding and H.Q. Zhu. 2015. Manage Encrypted Data Storage with Deduplication in Cloud. Proceedings of ICA3PP2015, Zhangjiajie, China.

[23] Z. Yan, X.Y. Li and R. Kantola. 2015. Controlling Cloud Data Access Based on Reputation. Mobile Networks and Applications, Springer. doi: 10.1007/s11036-015-0591-6.

[24] T.T. Wu, W.C. Dou, C.H. Hu, and J.J. Chen. 2014. Service Mining for Trusted Service Composition in Cross-Cloud Environment. IEEE Systems Journal. 99: 1-12, doi:10.1109/JSYST.2014.2361841.

[25] C. Liu, C. Yang, X.Y. Zhang and J.J. Chen. 2015. External Integrity Verification for Outsourced Big Data in Cloud and IoT: A Big Picture. Future Generation Computer Systems. 49: 58-67.

[26] N.X. Xiong, A.V. Vasilakos, L.T. Yang, L.Y. Song, Y. Pan, R. Kannan, and Y.S. Li. 2009. Comparative Analysis of Quality of Service and Memory Usage for Adaptive Failure Detectors in Healthcare Systems. IEEE Journal on Selected Areas in Communications. 27(4): 495-509, doi:10.1109/JSAC.2009.090512.

[27] Y.Z. Zhou, Y.X. Zhang, H. Liu, N.X. Xiong, and A.V. Vasilakos. 2014. A Bare-Metal and Asymmetric Partitioning Approach to Client Virtualization. IEEE Transactions on Services Computing. 7(1): 40-53, doi:10.1109/TSC.2012.32.

[28] W.K. Ng, Y. Wen and H. Zhu. 2012. Private data deduplication protocols in cloud storage. Proceedings of the 27th Annual ACM Symposium on Applied Computing. pp. 441-446.

[29] C.W. Tsai, C.F. Lai, H.C. Chao, and A.V. Vasilakos. 2015. Big data analytics: a survey. Journal of Big Data. 2(1): 1-32, doi: 10.1186/s40537-015-0030-3.

[30] L.F. Wei, H.J. Zhu, Z.F. Cao, X.L. Dong, W.W. Jia, Y.L. Chen, and A.V. Vasilakos. 2014. Security and privacy for storage and computation in cloud computing. Information Sciences. 258: 371-386, doi:10.1016/j.ins.2013.04.028.

[31] M. Ali, S.U. Khan, and A.V. Vasilakos. 2015. Security in cloud computing: Opportunities and challenges. Information Sciences. 305: 357-383, doi:10.1016/j.ins.2015.01.025.

[32] M. Ali, R. Dhamotharan, E. Khan, S.U. Khan, A.V. Vasilakos, K.Q. Li, and A.Y. Zomaya. SeDaSC: secure data sharing in clouds. IEEE Systems Journal. vol.