Selective Encryption and Component-Oriented De duplication for Cloud Data Computing

Amulya.R.Rao[1], Sushmitha Sen[2], Aishwarya M .H[3] , Sumithra B.R[4], Naveen Kumar M.R[5]

[1,2,3,4]Student, Information Science and Engineering VVCE, Mysuru, Karnataka, India
[5]Assistant Professor, Information Science and Engineering VVCE, Mysuru, Karnataka, India

***

Abstract - *Component–oriented de-duplication and selective encryption application is an effective and efficient data and memory optimizing technique and provides data security and decreases the cloud storage. It diminishes the overall encryption by using compression and security methods .Cloud computing gives unlimited storage resources to users. But the critical challenges in services of cloud storage is the management of ever increasing volume of data. To make data management ascendible in cloud computing, de-duplication has been well known method and has engaged more attention. Also recent years observed the trend of having advantage for cloud-based services for large climb content storage, processing and distribution. Data de-duplication is concentrating on data compression method for removing redundant copies of repeating information in storage. It optimizes memory. This method possesses effective storage utilization and can also be used for transfer of data through network to reduce the number of bytes that should be sent. Substitute of keeping redundant copies, de-duplication removes repeated data or information by having only one physical copy and involve with other redundant data or information to that copy.*

***Key Words*: Encryption, de-duplication, optimizes memory, Storage utilization, data compression**

1. INTRODUCTION

Cloud storage is a remote storage service, where users can upload and download their data anytime and anywhere. Data de-duplication, makes data holders to distribute a copy of similar data, it can be performed to overcome the consumption of storage space [9]. Component oriented de-duplication and selective encryption are well designed and well planned techniques used to de-duplicate redundant objects that are in files, that are in emails as well as in images using object-level components depending on their structures.

Data de-duplication brings advantages, solitude and security concerns for users' delicate data that are vulnerable to both inside and outside attacks. Traditional encryption, which gives data privacy, is conflicting with data de-duplication. This traditional encryption needs dissimilar users to encrypt their information by having their own keys. Hence, redundant copies of data of dissimilar users will guide to dissimilar cipher text, making de-duplication impossible. Therefore, convergent encryption has been preferred to impose data privacy while making de-duplication effective and efficient. Convergent encryption is processed by computing cryptographic hash value of the content of the data copy by using convergent key to encrypt and decrypt a data copy. Once the key generation is done data encryption for security, users keep the keys and send the cipher text to the cloud. Since, the encryption operation is settled and is extracted from data content, non-redundant data copies will be generated having the same convergent keys and also same cipher text. Hence, convergent security encryption allows the cloud to carry out de-duplication on the cipher text which minimizes usage of memory. Efficient non server side data minimizing methods are essential to save data on the path from a user to cloud servers or storage spaces. It in turn, expedites the data processing and transmission speed as well as reduces data vulnerability in the platform. Although traditional server-side data de-duplication methods dispose to attain high data reducing rate, as they require high organized overhead because of index processing and also data splitting, they cannot be directly used in capacity limited mobile devices. While, a simple file-level or a fixed-size block-level de-duplication (i.e., Dropbox) can be able to manage finite source device capacity. It cannot manufacture large data reduction rate.

1.1 LITERATURE SURVEY

Numerous data dividing methods have been offered to make better performance of data de-duplication. Traditional block-level de-duplication [1] technologies divide the data file into chunks of fixed or variable sizes. Since they achieve high de-duplication rates by providing the fine granularity dividing techniques, it has been used for additional resources or file systems such as venti [2] and also, for removing redundant network including Low Bandwidth File System (LBFS) [1]. However, as block-level de-duplication methods, especially variable-size ones, require large cost of processing, space-time (for example, the use of Rabin fingerprint matching [4] and of maintaining and tracking huge indexing and data splitting, it repeatedly runs on unique, fast and huge-capacity servers for in-line or cloud storage devices. An end user system of cloud-based repository is repeatedly limited in its processing capacity and memory space to carry out a successful data traditional de-duplication. Using data de-duplication on cipher texts is not optimal because cipher texts are different in nature even though the compatible plaintexts are the same. Therefore, the first way is to join a header. The header can let the cloud repository server be able to recognize two different cipher texts compatible to the similar plaintexts. Hence, the identical cipher texts that corresponds to the similar plaintexts were produced by dissimilar data keepers using dissimilar secret keys. There is another conflict: how can the cloud repository server protect a divided cipher text so that all the data keepers can decrypt and in which the corresponding plaintext cannot be known in cloud repository? [6] The second method is to give the similar plaintexts compatible to the similar cipher texts. In order to overcome the cloud storage server acquiring the plaintexts, the sensitive keys must be created from the plaintexts, and hence there is no random factor for the input of the algorithm. It means that the similar plaintexts create the similar sensitive keys, and also by using the similar sensitive keys to encrypt the similar plaintexts will produce similar cipher texts. Therefore, the cloud repository can identify duplicate cipher texts and just keeps one replica of them. Alternatively, the data keepers can decrypt the stored replica because duplicate ciphertexts are compatible to the similar sensitive key.[6]

1.2 EXISTING SYSTEM

File-level De-duplication:

When a file is uploaded freshly, it gets stored in Repository. When a second file gets uploaded it checks for the exact match. If there is a slight change in the similar file, it cannot identify the change. It is not a better optimization algorithm. But, it is a faster de-duplication method.

Block level De-duplication:

When a user uploads the information (file), it gets divided into blocks. Every block undergoes de-duplication algorithm. It has a limitation when blocks have too many objects (eg. images). If any changes are made in any one of the component, memory is again occupied. This method is better than file level de-duplication in memory optimization.

2. PROPOSED SYSTEM

Component-level de-duplication:

It deals with the components/objects (text files, images).It is a more optimized algorithm. For example, if second file's component (e.g.: image) is modified, only that modified component (edited image) occupies memory. While downloading, de-duplication algorithm regenerates the file and then downloads. It has many advantages.

(1) Efficient and successful data de-duplication at client-side. (2) Structured file such as MS DOCX, PPTX and PDF by exploiting object level. (3) It overcomes the overall encryption overhead.

3. ARCHITECHTURE

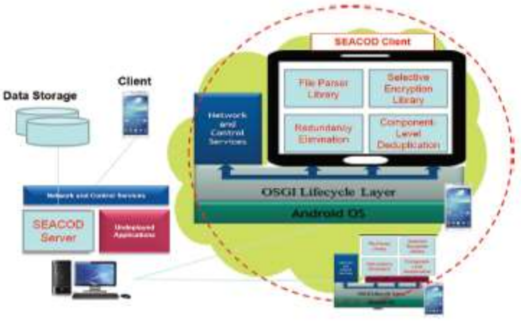

Fig-1: SEACOD Architecture

4. DESIGN AND IMPLEMENTATION

Implementation Modules

1. **File Parser Module**

- MS Word file (*.docx) which follows the Open XML format (**called Open XML**). Texts in a word file which uses document.xml object, and the images which comes under objects of media directory, while other directories contain meta-data objects
- Word file is parsed and decomposed into **smaller sized materials** based on file structure policies.

2. **Component-Level De-duplication Manager Module**

- First, it indexes the components by their **Hash Value** (**SHA256 Algorithm**)
- Without sending the data across the network, it uploads indexes for the cloud server to advance the similar data (i.e. checks the uniqueness of an index by checking the **Object Index Table**)
- Cloud server in return sends acknowledge on only the non-redundant component indexes

3. **Selective Encryption Module**

- Choose Components for Encryption
- Generates key for data encryption using the computed data index (Hash Value)
- Performs data encryption (Symmetric Encryption Algorithms – AES **- Rijndael Algorithm**)

4. **Redundant Data Elimination Module**

- Unique (non-redundant) encrypted data components are uploaded to cloud server for storage
- Cloud stores all the received data components & maintains an **Object Index Table**

- **Amazon Simple DB** will be used to carry file information's, object indexes, file owner, date & time etc.
- **Amazon S3** will be used to carry physical encrypted file objects (text/images)

5. File Download Module

- All encrypted data components (text/images) are downloaded from cloud server storage to client machine
- Encrypted data components are decrypted using respective cryptography keys. Word File is re composed with all decrypted components.

5. DATA FLOW DIAGRAMS

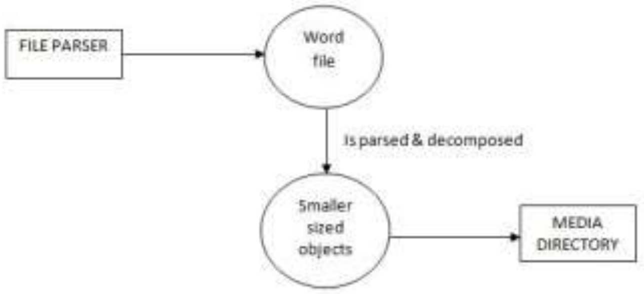

Fig-2: DFD for File parser

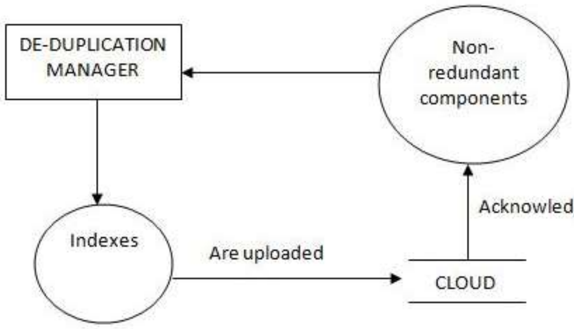

Fig-3: DFD for De duplication Manager

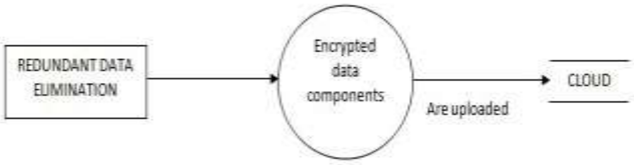

Fig-4: DFD for Redundant data elimination

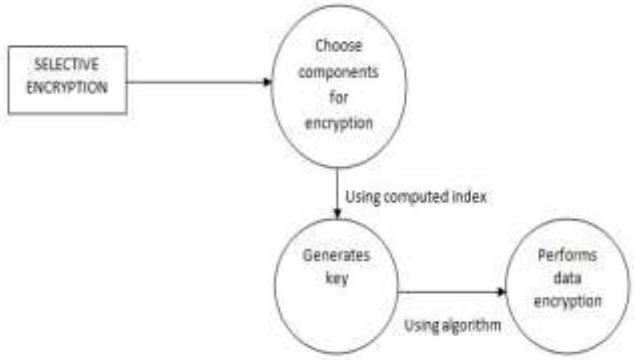

Fig-5: DFD for Selective Encryption

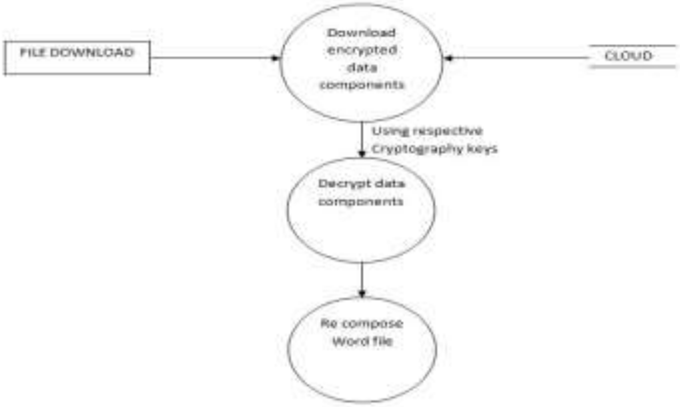
FILE DOWNLOAD
Download encrypted data components
CLOUD
Using respective Cryptography keys
Decrypt data components
Re compose Word file

Fig-6: DFD for File Downloading

6. SEQUENCE DIAGRAM

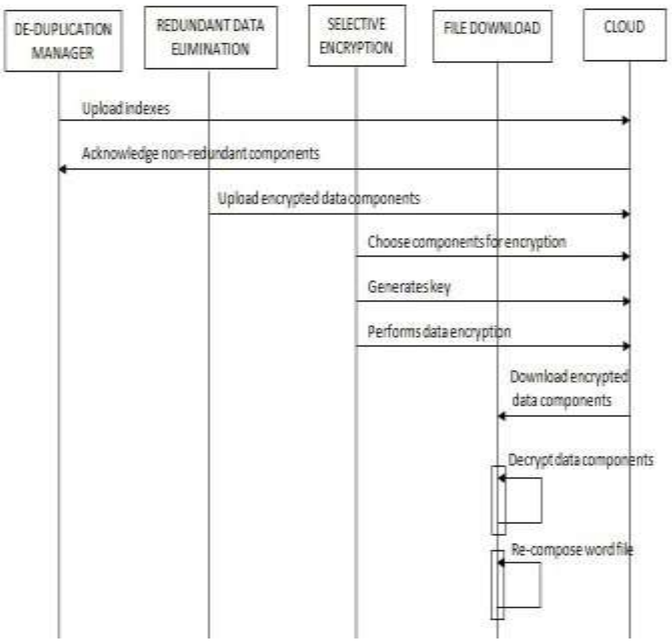
DE-DUPLICATION MANAGER
REDUNDANT DATA ELIMINATION
SELECTIVE ENCRYPTION
FILE DOWNLOAD
CLOUD
Upload indexes
Acknowledge non-redundant components
Upload encrypted data components
Choose components for encryption
Generates key
Performs data encryption
Download encrypted data components
Decrypt data components
Re-compose word file

Fig -7: sequence diagram

7. DISCUSSION

As for the future work, we plan to extend our prototype system to incorporate with the video files and also support other Electronic Health Record (EHR) files such as digital imaging and communications in medicine (DICOM) format. [9] We also want to develop an intelligent mechanism to keep away the unnecessary data exchanges by exploring the collaborating members' data processing and transfer capability and existing data components. We will further recognize the distinct encryption methods and procedures for the SEACOD framework. [9]

8. CONCLUSION

We have presented a paper on Component-Oriented Deduplication and selective encryption application that achieves effective data reduction, efficient encryption, and data-oriented collaboration control for resource intensive mission-oriented cloud computing services. Specifically, (1) we built an effective software framework for smartphones to eliminate redundancies in structured files by exploiting object-level components; (2) we designed effective methods to overcome the overall encryption overhead on the devices by selectively applying encryption methods based on the decomposed data types; and (3) we developed an intelligent mechanism to overcome the unnecessary data exchanges by exploring the collaborating data members' of data processing and transfer capability and existing data components.

REFERENCES

[1] A. Muthitacharoen, B. Chen, and D. Mazieres, "A low-bandwidth network file system," in *SOSP '01 Proceedings of the eighteenth ACM symposium on Operating systems principles*, vol. 35. ACM, Dec. 2001, pp. 174–187.

[2] S. Quinlan and S. Dorward, "Venti: A new approach to archival storage," in *Proceeding of the USENIX Conference on File and Storage Technologies(FAST)*, vol. 4, Jan. 2002.

[3] B. Zhu, K. Li, and H. Patterson, "Avoiding the disk bottleneck in the data domain deduplication file system," in *Proceeding of the USENIX Conference on File and Stroage Technologies(FAST)*, vol. 18, 2008.

[4] M. O. Rabin, "Fingerprinting by random polynomials," Harvard University, Tech. Rep. Report TR-15-81, 1981.

[5] A Secure Client Side Deduplication Scheme in Cloud Storage Environments-Nesrine Kaaniche, Maryline Laurent

[6] Encrypted Data Deduplication in Cloud Storage-Chun-I Fan,Shi-Yuan Huang,Wen-Che Hsu

[7] A Hybrid Cloud Approach for Secure Authorized Deduplication-Jin Li, Yan Kit Li, Xiaofeng Chen, Patrick P. C. Lee, Wenjing Lou

[8] HEDup: Secure Deduplication with Homomorphic Encryption-Rodel Miguel, Khin Mi Mi Aung

[9] Selective Encryption and Component-Oriented Deduplication for Mobile Cloud Data Computing , Sejun Song Baek-Young Choi Daehee Kim