International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-6 Issue-5, June 2017

Secure Distrusted Model for Large Data in Cloud Storage Based on No-SQL Database

Fathima Mussarath, K. G Manjunath

Abstract*: Cloud based storage providers bring forth limitless storage capacity and ingress potential to store and retrieve large amount of information. These operations performed by several users lead to increase in the system load on cloud storage. Hence, in order to provide better quality of service to the users, the system has to consider numerous pre-requisites such as efficient management and storage of large files, efficient management of the space (reduce the wastage of storage space) and data protection. In this paper we propose a distributed cloud storage which provides architecture and algorithms to administer the problems of the cloud storage. A less complicated and fixed size metadata design is proposed which diminishes the space unpredictability of metadata. The solution also supports a secure de-duplication mechanism for cross-users, that reduce the operation cost and protects the privacy. The data has to be protected before being uploaded to the cloud storage. The solution makes use of the key-value store no-sql database thus providing distributed and scalable cloud storage for large information.

Index Terms*: Cloud Storage, No-SQL, Convergent Encryption, Scalable.

I. INTRODUCTION

The traditional storage services differ strikingly with the cloud storage services which provide the users with supple capacity and flexible access without the overhead of infrastructure management. Cloud based storage solutions serve a large number of clients with the storage limit for each client ranging from GB to TB of information. The cloud storage providers have the control over making the information available and accessible to the users without any bottleneck. Cloud storage serves large number of clients on several cases such as backing up of data, sharing the files with other users, uploading the data from different devices and lastly downloading the data. These operations performed by several users lead to increase in the system load on cloud storage. Hence, in order to provide better administration to the end-users, the system needs to consider numerous pre-requisites such as efficient management and storage of large files, efficient management of the space (reduce the wastage of storage space) and most importantly, data protection.

In conventional storage systems, there are many challenges faced in order to manage the files efficiently like scaling out the systems when the data increases, distributing and replicating the data across nodes to take care load balancing and fault tolerance. The present storage systems maintain a complicated metadata system. The extent of metadata is straight to the measure of the document. Also global de-duplication mechanism is not supported where multiple users store the same file content. Most importantly the files should be secured.

Cloud Storage service capabilities- Cloud Storage applications can implement several capabilities in order to optimize the storage space and increase the speed of transfers. These capabilities include chunking (i.e., splitting a large file in to number of configured size data units), bundling (transferring many small sized files as a single object), de-duplication (taking care of redundancy of data), compression and security.

The proposed storage system includes most of these capabilities to optimize the storage space and increase the speed of transfer. Unfortunately the cloud storage services are not free of cost. Hence it is required that the redundant data has to be removed. The capability of data de-duplication is attained by refraining from storing the same file multiple times. In cloud storage system, large files are split into multiple chunks and these chunk data is maintained in the metadata. As the size of metadata linearly increases with file, a new solution is proposed which reduce the wastage of space caused due to metadata.

No-SQL stands for Not Only SQL. No-SQL database doesn't have a predefined schema and doesn't have a declarative query language. This type of database is used when the user needs to handle unstructured and unpredictable data. The data store in No-SQL are much faster as it takes the advantages of scaling out where more nodes can be added to a Server in order to distribute the load. The advantages of no-sql databases are considered in the proposed storage system.Information protection is the major concern of the enterprise users. The end-users who have deposit the information on to the cloud can get reliable, obtainable, fault-tolerance and performance from service providers, but there is no guarantee that the service provider won't scan the stored information. The end-users require that the information outsourced on to the cloud should be secure.
The contributions of this paper are outlined as below:

Manuscript published on 30 June 2017.
***** Correspondence Author (s)
Fathima Mussarath, M.Tech Student, Computer Science and Engineering, Siddaganga Institute of Technology, Tumkur, Karnataka, India E-mail: fathima.mussarath03@gamil.com
K.G. Manjunath, Assistant Professor, Computer Science and Engineering, Siddaganga Institute of Technology, Tumkur, Karnataka, India E-mail: sitmanju@gmail.com

- The solution provides a lightweight metadata for the large information documents. The size of the metadata file is nearly same as that of the document.
- Give contiguous ids to the chunks that are generated after splitting the file. This makes it easier to distribute the information.
- De-duplication scheme on encrypted data that eliminates the duplicate documents across users.
- Overcome the limitations of convergent encryption and add more security to the encryption to avoid the attacks.

The solution uses the advantages of the No-SQL database to store the large documents and provide scalability and distributed environment.

II. BACKGROUND AND RELATED WORK

Drago et al [2].In storage pool such as Dropbox, one of the concerns with Dropbox is the metadata size. The measure of the metadata increments straightly with the measure of the original document. The meta-information of every document in dropbox includes a set of elements where each element has the data which includes the chunk size and the hash value of each chunk. So as the number of chunks increases the list also increases. Complications arise when the record size is huge. As the record size is huge, the metadata size also increases and not scalable. This leads to the space complexity of the metadata system to be O (n).

De-duplication [3, 4] eliminates the redundant data and disallows the data to get stored if already exists in the storage. If the same data is tried to upload, only one copy of the data gets stored in the storage. To utilize the de-duplication mechanism, it is required to take the benefit of cross-user de-duplication. There are several drawbacks related to de-duplication which include integrity and privacy. De-duplication can be performed on the client side or server side. When performed on client side, bandwidth is saved but can be a victim of side channel attack. When performed on the server, the side channel attacks are avoided but bandwidth is used.

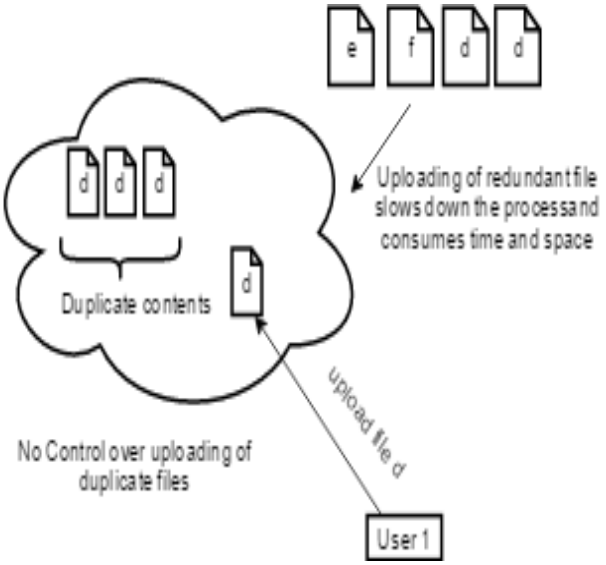

Figure 1: Redundant File Problem.

Convergent Encryption [5] provides the confidentiality of the data in de-duplication solution. De-duplication uses the cryptographic value of the documents and this value becomes the encryption key. This overcomes the problem of sharing the key among the users. The users need not interact with each other. But this scheme is susceptible to confirmation of file attack; learn the remaining information attack and dictionary attack. These attacks will help the attacker to retrieve the information saved by the end-user. In [6] a solution was proposed to salt the convergent key with a secret value, which adds randomness and uniqueness to the key for encryption. This solution overcomes the weakness of convergent encryption.

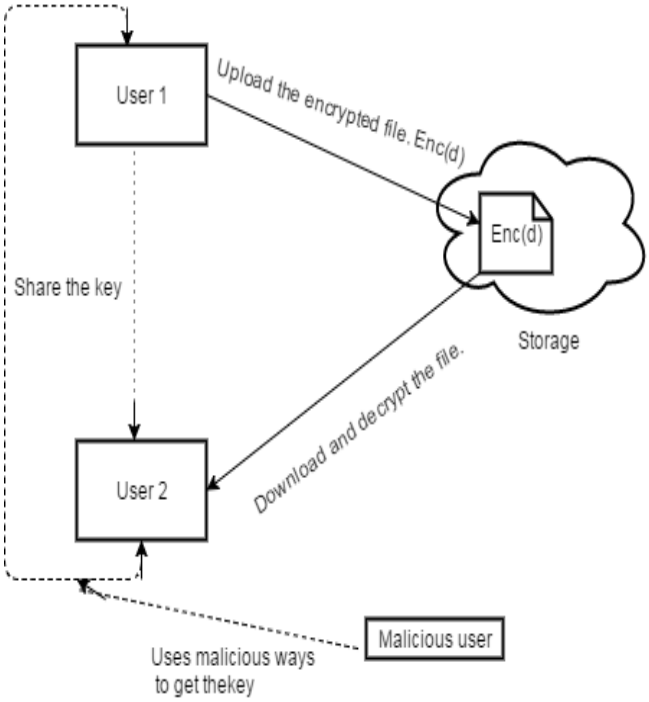

Figure 2: Security and key exchange in Cloud Storage

III. PROPOSED SOLUTION

The proposed distributed secure cloud storage system takes care of most of the issues in distributed cloud storage by using the no-sql data base. This is taken care by proposing a less complex settled size metadata outline, fast and simultaneous, distributed record input/output operations, information security to the documents and data de-duplication mechanism for static data.

The system architecture can be defined as a conceptual model that defines the structure, behavior and other system views. The proposed solution consists of three components.

User/Client Application – It gives interfaces to the client/end user in order to perform operations such as data upload, download and sharing of files with encryption and decryption tasks.

Server Application – It consists of the API which has the algorithms for uploading downloading the files with encryptions and decryption techniques and takes care of data de-duplication and access control for the data.

No-SQL data store – It serves as the data store for all the data that is transmitted by the end-user.

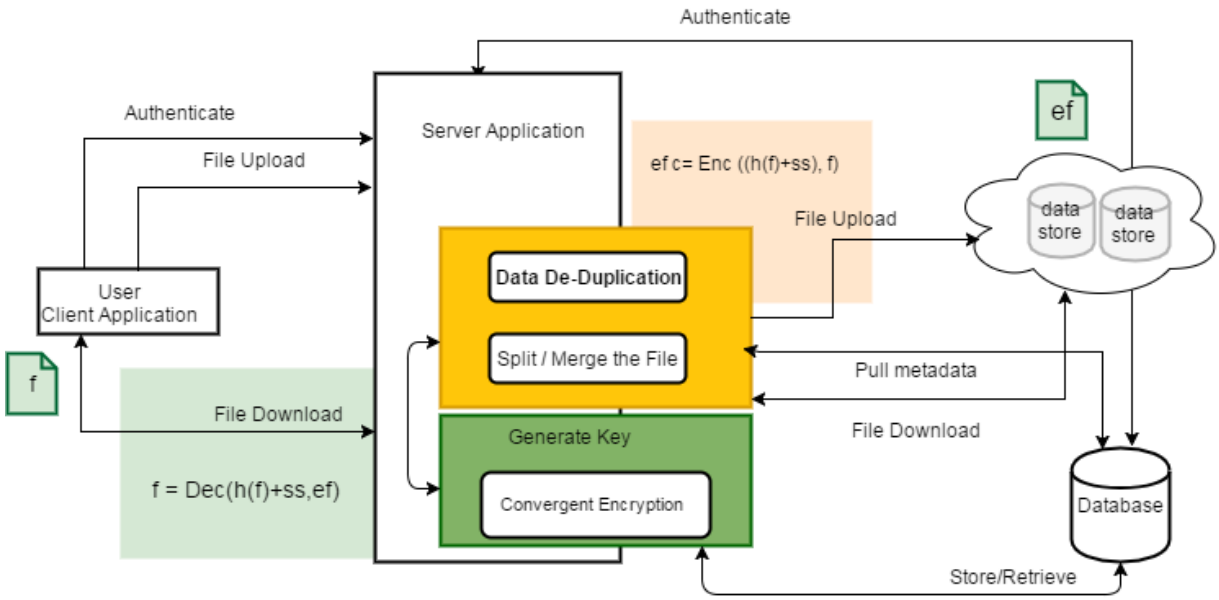

Figure: 3. System Architecture

A. Chunk Storage:

Chunk is the essential part of the proposed cloud storage system. A chunk is nothing but a fragment of information that is generated from a document. When the client wants to store the record in cloud storage, if the size of the record is more than the pre-configured size then the record is divided into collection of chunks. Apart from the last chunk, all the other chunks generated have equivalent size. The last chunk will be less than or equal to the preconfigured size. Then a unique identification is generated for the record and the first chunk generated. The subsequent chunks get their unique IDs.

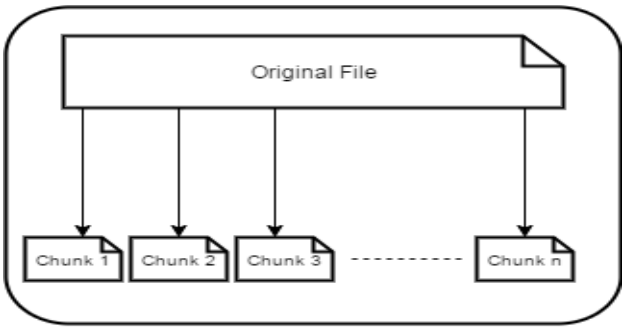

Figure 4: File split into small chunks.

B. Metadata

In the existing distributed storage, the metadata will individually increment with the measure of record. It compromises of elements where every element has the data such as the measure of the chunk and unique hash value of the chunk. This linear increase in the metadata of file gets complicated when the file are large. The proposed cloud storage system provides a solution to this where the size of the metadata is independent of the size of the file. How much ever the file size the metadata size remains fixed. The solution includes by storing the unique id of the beginning chunk and the count of the chunks estimated by the record. Whenever the end user uploads the document on to the storage pool, the solution stores the file details, which include the document name, document size, estimated hash value, start chunk id of the chunk, number of chunks. This makes the metadata of the file to be of constant size.

C. Cloud Storage mechanism with de-duplication:

Uploading: The uploading mechanism is as shown in the figure 5. The proposed method actualized on server-side. By utilizing hash capacity like SHA2 it is easy to distinguish redundant copy of records in the entire in the system. Whenever a user chooses a document to upload, the server estimates the hash value of the contents of the record. This information along with the document name and document size is sent to the server. This hash value is checked against the records in the storage. If the unique has value exits, which implies both the documents are same. The file is referenced to the existing record.

Downloading: Downloading process is a one strategy where the client gets the information. In the proposed framework the downloading processes happens as follows: end-user chooses a file for download. This request is processed by the object store layer, where the metadata information of the file is fetched. The chunk id and number of chunks give the number of files to be downloaded. These files are merged and then decrypted with the encryption key. The downloading method is shown in the figure 6.

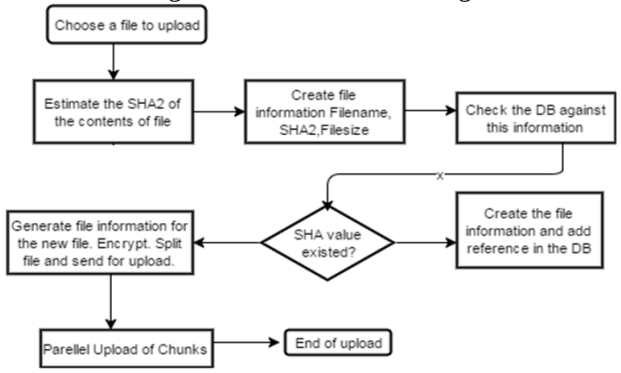

Figure: 5. Upload algorithm.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

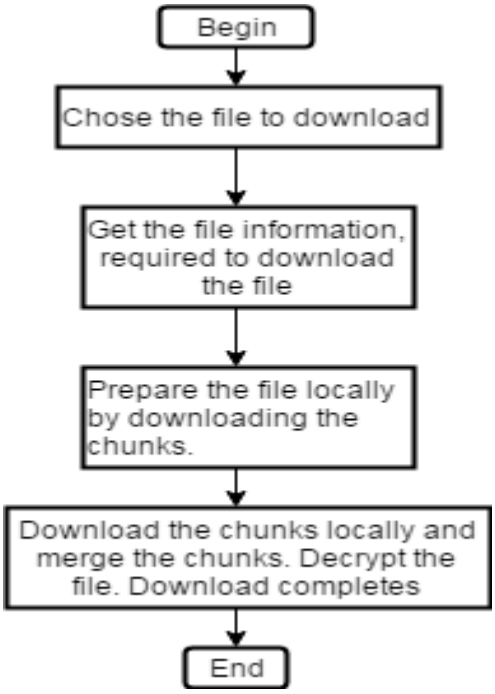

Figure: 6. Downloading Algorithm

D. Secure transmission of files:

To protect the encryption of the resource, a secret value is generated for each document that will be uploaded. When the user chooses to upload a document, a convergent key is generated by the server. The key is a combination of the hash value of the document (h(f)) and a secret value (ss).

$$GenKey(\ h(f) + ss) \rightarrow Key$$

The generated key is used to encrypt the document. The proposed solution overcomes the drawback of the conventional convergent encryption by salting a secret value to the hash of the document. This increases the randomness and uniqueness to the secret key so that the encryption is not determined. By doing this the Confirmation of file and known plain text attack can be overruled. Even if the attacker is aware of the plaintext then the attacker cannot infer the content of the encrypted document. The document is encrypted with the key generated to produce a cipher text as shown below.

$$Encrypt(\ key, file) \rightarrow ef$$

When the end-user requires downloading the documents, the control key is used to decrypt the convergent key and this convergent key is used to decrypt the document. Once the decryption process is completed the file gets downloaded.

$$Decrypt(key, ef) \rightarrow f$$

The convergent encryption is based on the AES 256 algorithm and the hashing function is based on the SHA256 algorithm. The AES algorithm uses a static initialization vector which is obtained from the secret value.

IV. RESULTS AND DISCUSSION

A. Metadata:

Dropbox[2] is the most widely used cloud storage system which accounts for a volume equivalent to one third of the YouTube traffic at campus networks. The size of metadata file in the Dropbox increases as the original file size increases. As the file gets larger, the metadata file also gets larger as it contains the list of hash values estimated for each chunk. In the proposed solution the metadata file has a fixed size irrespective of the original file size.

B. De-Duplication:

The comparison done for the de-duplication mechanism with other cloud storages shows that Dropbox supports de-duplication for single user, where as the proposed solution supports the global de-duplication.

Table: 1. Comparison of De-duplication

De-duplication	Drop Box	Proposed Solution
Per User	Yes	Yes
Multi-User	No	Yes

C. Security

The encryption key acts like a secret key to the contents of the document. The encryption key is generated by estimating the hash value of the contents of the document. In the proposed solution the encrypted key is generated by combining the unique has value of the contents of the document and a salt/secret value. This newly generated key is used to encrypt the document. This solution overcomes the confirmation-of-file attack and learn -the- remaining-information attack.

V. CONCLUSION

The proposed Cloud storage is based on the No-SQL data store. The metadata of the file stored on this storage is fixed size and does not increase with the file size. A large file is split in multiple small chunks before it is being uploaded increasing the performance. By using the advantages of the Amazon S3 data store, it is easy to scale the system and distribute the data. The cloud storage uses the sha-256 has function to find the redundant files and hence achieve the de-duplication. A secure de-duplication mechanism is implemented which allows the de-duplication to occur on the encrypted data. This overcomes the existing attacks of confirmation of file, dictionary attack and LRI attack.

REFERENCES

1. Amazon Simple Storage Service, http://aws.amazon.com/s3, 2015.
2. I.Drago, M. Mellia, M. M Munafo, A. Sperotto, R. Sadre, and A. Pras. "Inside dropbox: understanding personal cloud storage services." In Proceeding of the 2012 ACM conference on Internet measurement conference, pages 481-494, ACM, 2012.
3. J. Li. Secure deduplication with efficient and reliable convergent key management. IEEE Transaction On Parallel And Distributed System, 25(6):1615–1625, jun 2014.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

4. F. Rashid. A secure data deduplication framework for cloud environments. 2012 Tenth Annual International Conference on Privacy, Security and Trust, 978-1-4673-2326-0(12):81–87, 2012.
5. P. Puzio. Cloudedup: Secure de-duplication with encrypted data for cloud storage. 2013 IEEE International Conference on Cloud Computing Technology and Science, 978-0-7695-5095-4(13):363–370, 2013.
6. D. Perttula. Drew perttula and attacks on convergent encryption. https://tahoe-lafs.org/hacktahoelafs/drew perttula.html, mar 2008.
7. I. Drago, E. Bocchi, M. Mellia, H. Slatman, and A. Pras. Benchmarking personal cloud storage. In Proceedings of the 2013 conference on Internet measurement conference, pages 205–212. ACM, 2013.
8. P. FIPS. 197: the official aes standard. Figure2: Working scheme withfour LFSRs and their IV generation LFSR1 LFSR, 2, 2001.
9. F. PUB. Secure hash standard (shs). 2012.
10. J. Stanek, A. Sorniotti, E. Androulaki, and L. Kencl. A secure data deduplication scheme for cloud storage. 2014.

Fathima Mussarath, pursing M.Tech in Department of Computer Science and Technology at Siddaganga Institute of Technology, Tumkur, Karnataka.

K.G Manjunath is working as an Assistant Professor (Senior) in the Department of Computer Science and Engineering at Siddaganga Institute of Technology, Tumkur, Karnataka, since 2002. He has taught subjects like software engineering, computer networks, and database management system to UG, PG students. He is pursuing his PhD under the guidance of Dr. N. Jaisankar and is a Professor of the School of Computer Science and Engineering, VIT University, Vellore, since February 2011.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)