ISSN **2321-743X**

International Journal of Research in

Engineering and Bioscience

Volume **6** Issue **3** (Pages 01- 06)

IJREB

Journal home page: www.ijreb.org

BIG-DATA STORAGE IN CLOUD WITH ENHANCED DATA SECURITY

Akhil P[1], Dipin S J[1], Kiran T Raj[1], C P Maheswaran[2]

[1]UG Scholar, [2]Assistant Professor, Department of Computer Science & Engineering, Noorul Islam Centre for Higher Education, Kumaracoil- 629 180, India.

ABSTRACT

One of the highest problems in Cloud data storage is regarding its security and the way it is stored on to the cloud. Cloud offers ahuge amount of storage space where the user can upload all their private data to the cloud server. The aim of this project is to add more than two hierarchal layers of security to avoid any kinds of security flaws to protect against data leaks and to add redundancy factor by using Big-Data for storage. Instead of encrypting the documents using a public key, each document uploaded will have its own password set by the user. The passwords are not stored anywhere in the cloud thus avoiding the data leak problem all together. Every file which is uploaded will get encrypted using AES 256 algorithm, thus adding the highest possible security. Data stored in the secured cloud account of a user and each user will have his own individual account. The main principle of the project is to use an obfuscation technique and big data storage to enhance the security and redundancy of cloud data storage. Obfuscation is used to transform the data into a new cypher text created using AES 256 algorithm while the use of big data will help to add the redundancy factor by backing up the data to different nodes. Thus, all the data's stored will be protected and the user can have Free State of mind that his data is not going anywhere even in case of the server being shut down.

INTRODUCTION

Cloud provides huge space for storing data for the user. It helps even small and large industries from spending or investing a huge amount of money for the storage server. Cloud storage is designed for Virtualized and distributed computing technology. It uses software that is provided by the cloud service provider. Storing data in the cloud is more efficient and safe than storing it on a local drive. But, the problem is that all data's are stored in a public cloud environment.

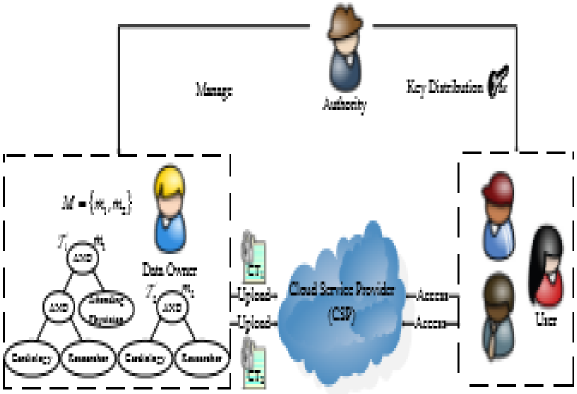

Fig. 1: An example of secure data sharing in cloud computing

Due to that security issues with storing data in the cloud is a huge factor. Storing data in the cloud without having good security protocol could lead to the data getting leaked or hacked or even the Cloud provider could illegally access the files that you stored in their public cloud storage environment. Sometimes the cloud server could get crashed you may lose all the data that you have stored on it. So, it's much to have a redundant backup for all the data you have stored in the cloud.

The main aim of the project is to add an additional layer of security to avoid security breaches and privacy and to add redundancy for all the files that are stored on the cloud storage. It is being done by using AES encryption for the files that are stored on the cloud system. Each file while uploading to the cloud will ask the user to set a password, using thispassword the files will be encrypted and will be uploaded to the server. While downloading the file, the user will have to use the same password which he used for encryption to decrypt the file and to download it. Since the password is not a public key and is a key set by the user the cloud provider will not be able to decrypt the files uploaded by the user thus, restricting the cloud providers from illegally accessing the data that is uploaded to the public cloud server by the user.

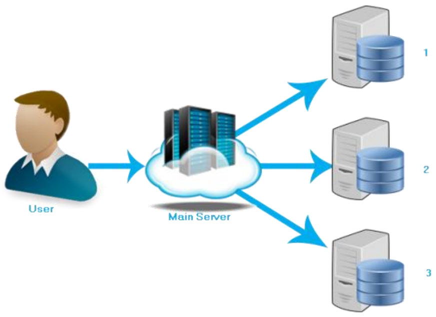

Fig.2**:** Data Storage and backup

RELATED WORKS

One of the highest problems in Cloud data storage is regarding its security. The work supported by the International Science and Technology Cooperation Program of China show that the proposed platform can satisfy the requirements of massive video data management. In SaveMe[2], every file is fully encrypted, erasure coded, and distributed. For better security and privacy, all data processes are performed on the client side, unlike the case of conventional cloud storage services. In our design data is divided into implicit, redundant blocks by which are stored on different designated cloud servers, can be used to reconstruct the original data file [3]. By making use of the attributes of CP-ABE, we can describe a user's credentials, and a party encrypting the data determines a policy who can decrypt [4]. In this paper, we propose an integrity check scheme for their system to enhance data robustness against storage server corruption, which returns tampered cipher texts [5]

PROPOSED SYSTEM

In the proposed in order to achieve secure, storage and access on outsource data in the cloud we exploit the technique of AES cryptography encryption to protect data files and proposed model has two part in the cloud storage server, Private data section and Shared data section. These two part of the cloud storage server makes the sharing of data easy and secure. User use the private data section to store his private data that is accessible to particular user only, whereas shared data section is used to store the data that needs to be shared among trusted users. This section is accessible to the particular user and his trusted users only. Data stored over cloud and flow through network in plain text format is a security threat. So, In our proposed model all the data stored in both section (Private data section, Shared data section) will be encrypted by using the AES cryptography Encryption.

The proposed system consists of two parts the front-end and the back-end. The user interacts with the web-based cloud application. Using the application, the user can create an account through which the user will be able to upload files and view them.Cloud-based web application consisting of a front-end developed using HTML, Bootstrap and different forms of JavaScript’s. It is developed in such a way the user interface is friendly for even someone who is new to this web application.Users interact with Cloud servers via main server to retrieve their data. They send the data to the main server and at the same time data are backup to multi-server. The proposed method includes some important security services such as authentication, encryption, decryption and compression and decompression in Cloud computing system. Users receive secret key generated by main server through registered email address that is used for security services. This mechanism solves the problem of handling big data and its sharing is secured by providing a common storage space.

The below shows the document or data the user uploads to the cloud storage and how the document or data is being encrypted and backed up to different server nodes. It also who the data being decrypted and when a file gets corrupted how the server recovers it.

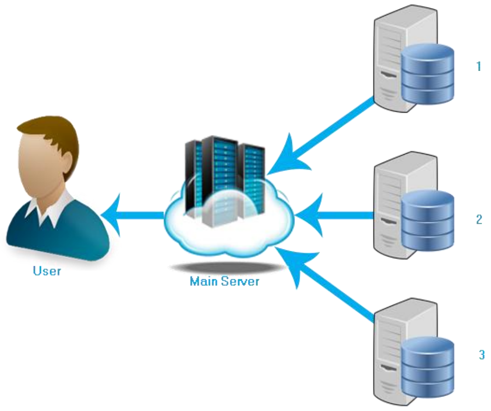

Fig.3: Data retrieval and recovery

The main aim of the project is to add an additional layer of security to avoid security breaches and privacy and to add redundancy for all the files that are stored on the cloud storage. It is being done by using AES encryption for the files that are stored on the cloud system. Each file while uploading to the cloud will ask the user to set a password, using this password the files will be encrypted and will be uploaded to the server. While downloading the file the user will have to again use the same password which he used for encryption to decrypt the file and todownload it. Since the password is not a public key and is a key set by the user the cloud provider will not be able to decrypt the files uploaded by the user thus, restricting the cloud providers from illegally accessing the data that is uploaded to the public cloud server by the user. The main scope of the project is store data in the cloud with more than one level of security, provide backup and avoid any kinds of data leaks. Data stored will be in a protected public cloud which uses Hadoop file system for storing large data and redundancy.

The files will be uploaded through Hadoop and it will be uploaded to the Hadoop DFS. By using Hadoop DFS, we can use its built-in functions to add redundancy to the files that are uploaded. HDFS file system will provide high throughput for accessing the data that is stored on to it. HBase and Hive are used to store the entire user database. It is more secure and fails proof compared to other database solutions available. HBase and Hive will help protect all the user details from leaking or restricting access to any third-party developers. It helps protect the user details from reaching the wrong hands.

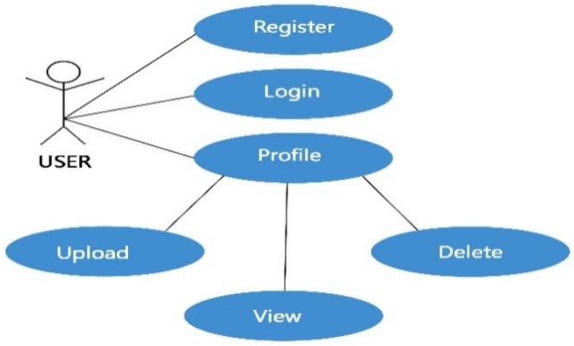

Fig.4: Dataflow Diagram

The presented storage architecture can support multiple data models, including all kinds of relational data and nonrelational heterogeneous data called NoSQL data, by dividing nodes in cloud storage centre into several clusters, each of which stores data with special model such as key value model and document model. Furthermore, the architecture provides users with unified storage interface and query interface.

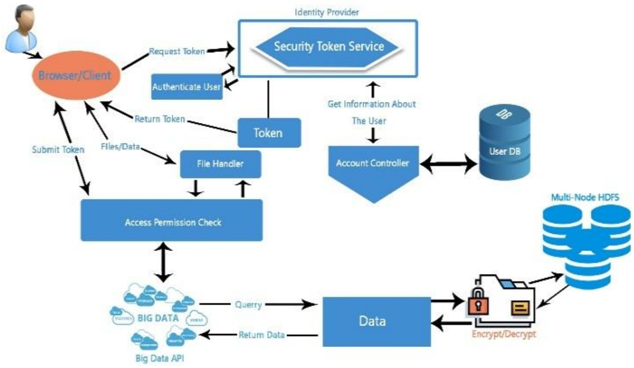

Fig. 5: System Architecture Diagram.

A. HADOOP

Hadoop is an open source, Java-based programming framework that supports the processing and storage of extremely large data sets in a distributed computing environment. It is part of the Apache project sponsored by the Apache Software Foundation.

B. HDFS

Apache HDFS or Hadoop Distributed File System is a block-structured file system where each file is divided into blocks of a pre-determined size. These blocks are stored across a cluster of one or several machines. Apache Hadoop HDFS Architecture follows a Master/Slave Architecture, where a cluster comprises of a single NameNode (Master node) and all the other nodes are DataNodes (Slave nodes).NameNode is a very highly available server that manages the File System Namespace and controls access to files by clients. Node is a very highly available server that manages the File System Namespace and controls access to files by clients. The HDFS architecture is built in such a way that the user data never resides on the NameNode. The data resides on DataNodes only.

C. HBASE

HBase is a column-oriented NoSQL database HBase has three major components i.e., HMaster Server, HBase Region Server, Regions and Zookeeper. HBase HMaster performs DDL operations and assigns regions to the Region servers as you can see in the above image. The META table is a special HBase catalog table. It maintains a list of all the Regions Servers in the HBase storage system, as you can see in the above image.

D.APACHE SPARKS

Apache Spark has a well-defined and layered architecture where all the spark components and layers are loosely coupled and integrated with various extensions and libraries. Apache Spark Architecture is based on two main abstractions-

- Resilient Distributed Datasets (RDD)
- Directed Acyclic Graph (DAG)

A spark cluster has a single Master and any number of Slaves/Workers. The driver and the executors run their individual Java processes and users can run them on the same horizontal spark cluster or on separate machines i.e. in a vertical spark cluster or in mixed machine configuration.

EXPERIMENT AND PERFORMANCE EVALUATION

In this section, we evaluate our proposed storage architecture for big data by real-world experiments and give a comprehensive performance analysis. We first describe our experiment setup, followed by the experimental results.

A. Experiment

Setup of our experimental environment consists of 12 distributed nodes as cloud nodes, each of which has a 2.8GHz core, 1GB memory and 250GB hard drive. We choose three data models to evaluate the presented architecture, including keyvalue model, extensible record model and structured spatiotemporal model. So, we divide the cloud nodes into four clusters and setup three database management systems, Redis for storing the data of key-value model, HBase for storing the data of extensible record model and Oracle for storing the data of structured spatiotemporal model. In our experiment, all of the data have two backups. In our experiment, there are three kinds of data, the spatiotemporal data collected from the digital home lab including three sets of data: temperature, humidity, and carbon dioxide concentration, the semi-structured data, that is HTML texts, collected from massive web pages as the instance of extensible record model and the log data of Linux system, which are unstructured, as the example of key-value model. We evaluate the performance of our storage architecture from two aspects: data loading and query processing. We compare CloST with two production systems that are widely used to store big data: Redis and HBase.

B. Performance of Data Loading

To evaluate the data loading performance of our storage architecture, we use 2 datasets, each of which includes three kinds of data mentioned above. The size of each dataset is 10GB and 20GB. As is shown in Figure 4.1, the data loading speed of our architecture is up to 5 times faster than the other systems.

C. Performance of Data Backup

In our storage architecture, we divide the cloud nodes into four clusters to make the communication cost during data backup and data loading smallest. To evaluate the performance of our method, we compare our algorithm with the random division method randomly dividing cloud nodes into four clusters, each of which have the same number of cloud nodes.

CONCLUSION

The main goal is to securely store and manage big data on Cloud that is not controlled by the owner of data. We exploit the mechanism of encryption and compression of data using secret key at the main server while uploading to the Cloud storage servers. This mechanism solves the problem of handling big data and its sharing is secured by providing a common storage space. Moreover, two times authentication improves the security and ensures that only legitimate users can access the data. The proposed mechanism of compressing and decompressing of data helps to backup and recover the data from different Cloud storage servers in feasible time. Using a private key that is given by the user to secure the data in cloud helps reduce the problem of data security since the key is not stored in cloud. Using more than one node in HDFS for backup helps to add the redundancy factor the data stored.

REFERENCES

[1] “A Distributed Video Management Cloud Platform Using Hadoop”, Published in IEEE Access Volume 3 on December 11, 2015,Xin Liu, Dehai Zhao, Liang Xu, Weishan Zhang, Jijun Yin, And Xiufeng Chen

[2] “SaveMe: Client-Side Aggregation of Cloud Storage”, IEEE Transactions on Consumer Electronics, Volume 61, October 2015,Gyuwon Song, Suhyun Kim, and Dongmahn Seo.

[3] “HASG: Security and efficient frame for accessing cloud storage”, Volume: 15,Published in Journal of Systems Engineering and Electronics,12 February 2018,Shenling Liu, Chunyuan Zhang, Yujiao Chen.

[4] “NC-MACPABE: Non-centred Multi-Authority Proxy re-encryption based on CP-ABE for cloud storage systems”, Published in the Journal of Systems Engineering and Electronics, 03 March 2016, XU Xiao-long, ZHANG Qi-tong, ZHOU Jing-lan

[5] “An Effective Integrity Check Scheme for Secure Erasure Code-Based Storage Systems”, IEEE Transactions on Reliability, Volume: 64, Issue: 3, 24 April 2015, Shiuan-Tzuo Shen, Hsiao-Ying Lin, Wen-Guey Tzeng.

[6] “Measuring Scale-up and Scale-out Hadoop with Remote and Local File Systems and Selecting the Best Platform”, IEEE Transactions on Parallel and Distributed Systems, Zhuozhao Li, Student Member and Haiying Shen, Senior Member, IEEE.