Secure Data Sharing with Data Partitioning in Big Data

Mr. Shriniwas Patilbuwa Rasal[1]
[1]M.E. Computer Engineering Department, (Jaywantrao sawant collage of engineering, pune).
shriniwasrasal@gmail.com

Prof. Hingoliwala Hyder Ali[2]
[2]M.E. Computer Engineering Department, (Jaywantrao sawant collage of engineering, pune).

Abstract : Hadoop is a framework for the transformation analysis of very huge data. This paper presents an distributed approach for data storage with the help of Hadoop distributed file system (HDFS). This scheme overcomes the drawbacks of other data storage scheme, as it stores the data in distributed format. So, there is no chance of any loss of data. HDFS stores the data in the form of replica's, it is advantageous in case of failure of any node; user is able to easily recover from data loss unlike other storage system, if you loss then you cannot. We have implemented ID-Based Ring Signature Scheme to provide secure data sharing among the network, so that only authorized person have access to the data. System is became more attack prone with the help of Advanced Encryption Standard (AES). Even if attacker is successful in getting source data but it's unable to decode it.

Keywords: *HDFS, Data Sharing, ID based ring Signature.*

*****

1. INTRODUCTION

Sharing data among groups of organizations is necessary in web-based society and at the very core of business transactions. Data sharing is becoming important for many users and sometimes a crucial requirement, specially for businesses and organisations aiming to gain more profit. However, data sharing poses several problems including data misuse or abuse, privacy and uncontrolled propagation of data. Often organizations use legal documents (contracts) to regulate the terms and conditions under which they agree to share data among themselves. However, the constraints explained in such contracts remain inaccessible from the software infrastructure supporting the data sharing. Traditional legal contracts are written using natural language, from a computational point of view, it is complex, difficult to parse, and ambiguity prone. It is therefore more difficult to:

- Verify inconsistencies within a contract itself or between a contract
- Relate violations of the policies to the terms and conditions expressed in a contract.

Therefore, there is a gap between a traditional legal contract regulating the data sharing and the software infrastructure supporting it.

Figure 1.1: Data Sharing

Bigdata Scenario

A distributed system is a pool of autonomous compute nodes connected by swift networks that appear as a single workstation. In reality, solving complex problems involves division of problem into sub tasks and each of which is solved by one or more compute nodes which communicate with each other by message passing. The current inclination towards Big Data analytics has lead to such compute intensive tasks. Big Data, is used for a collection of data sets which are huge and complex and difficult to process using traditional data processing tools. The need for Big Data is to ensure high levels of data accessibility for business intelligence and big data analytics. This condition needs applications capable of distributed processing involving terabytes of information saved in a variety of file formats. Hadoop is an open source framework based on Java which supports the processing and storage of large amount of data sets in a distributed computing environment. Hadoop is a part of Apache project, which is sponsored by Apache Software Foundation. It makes it handle thousands of terabytes of data,

and to run applications on systems with thousands of commodity hardware nodes. Hadoop has a distributed file system i. e. HDFS which allows faster data transfer amongst nodes. It also allows systems to continue even if a node stops working. Hadoop is a well-known and a successful open source implementation of the MapReduce programming model in the distributed processing. The Hadoop runtime system with HDFS provides parallelism and concurrency to achieve system reliability. The major categories of machine roles in a Hadoop are Client machines, Master nodes and Slave nodes. The Master nodes supervise storing of data and running parallel computations on all that data using Map Reduce. The NameNode coordinates the data storage function in HDFS, while the JobTracker supervises and coordinates the parallel processing of data using Map Reduce. Slave Nodes are the vast majority of machines and do all the cloudy work of storing the data and running the computations. Each slave runs a DataNode and a TaskTracker daemon that communicates with and receives instructions from their master nodes. The TaskTracker daemon is a slave to the JobTracker likewise the DataNode daemon to the NameNode. HDFS file system is designed for storing large files with streaming data access patterns, running on clusters of commodity hardware. An HDFS cluster has two type of node operating in a master-slave pattern: A NameNode (Master) managing the file system namespace, file System tree and the metadata for all the files and directories in the tree and some number of DataNode (Workers) managing the data blocks of files. The HDFS is so large that replicas of files are constantly created to meet performance and availability requirements.

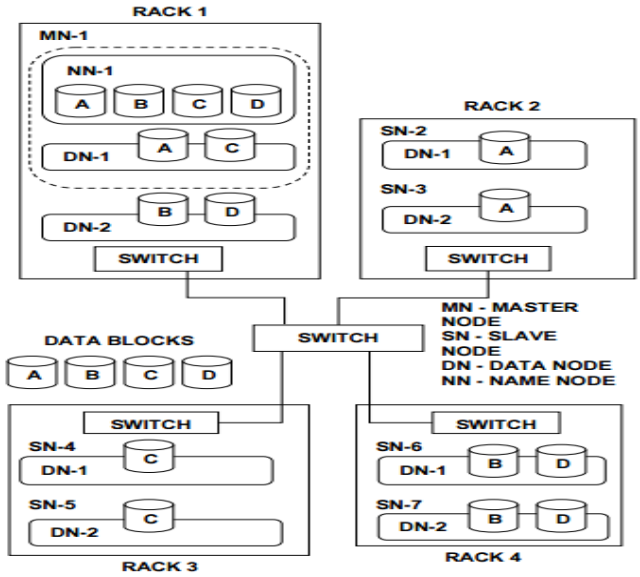

Figure 1.2: Hadoop Scenario

2. LITERATURE SURVEY

Keke Gai et al.[3] proposes a novel approach that can efficiently split the file and separately store the data in the distributed cloud servers, in which the data cannot be directly arrived by cloud service operators. The proposed approach is entitled as Security-Aware Efficient Distributed Storage (SAEDS) model, which is supported by the proposed algorithms, Secure Efficient Data Distributions (SED2) Algorithm and Efficient Data Conflation (EDCon) Algorithm. Our experimental analysis have assessed both security an efficiency performances.

Xinyi Huang et al.[4] shows Ring signature is a promising candidate to construct an anonymous and authentic data sharing system. It allows a data owner to authenticate his data which can be put into the cloud for storage or analysis purpose. Yet the certificate verification in the public key infrastructure (PKI) setting becomes a bottleneck for this solution to be scalable. Identity-based ring signature (ID-based), it removes the process of certificate verification, can be used instead. In this paper, further enhance the security of ID-based ring signature by providing forward security. This paper provide a concrete and efficient instantiation of our approach, prove its security and provide an implementation to show its practicality.

3. IMPLEMENTATION

a. Problem Statement

Data sharing with a large number of participants must take into account several issues, including efficiency, data storage, data integrity and privacy of data owner. The number of users in a data sharing system could be huge and a practical system must reduce the computation and communication cost as much as possible. Otherwise it would lead to a waste of energy and shortage of storage space.

b. Mathematical Implementation

Setup: On an unary string input 1^k where k is a security parameter, it produces the master secret key s and the common public parameters params, which include a description of a finite signature space and a description of a finite message space.

KeyGen: On an input of the signer's identity ID ϵ $\{0,1\}^*$ and the master secret key s, it outputs the signer's secret signing key S_{ID}. (The corresponding public verification key Q_{ID} can be computed easily by everyone.)

1. **Sign:** On input of a message m, a group of n users' identities $\bar{U}\{Di\}$, where
$1 \leq i \leq n$, and the secret keys of one members S_{IDs} , where $1 \leq s \leq n$; it outputs an ID-based ring signature σ on the message m.

2. **Verify:** On a ring signature σ, a message m and the group of signers' identities Ū{Di}, as the input, it outputs T for true or F for false. depending on whether σ is a valid signature signed by a certain member in the group Ū{Di} on a message m. It must satisfy the standard consistency constraint of IDbased ring signature scheme, i.e. σ = Sign(m ,Ū{Di}, S_{IDs}), and IDs ϵ Ū{Di}, we must have

Verify(σ ,m ,Ū{Di}) = T. A secure ID-based ring signature scheme should be unforgeability and signer-ambiguous.

c. System Architecture

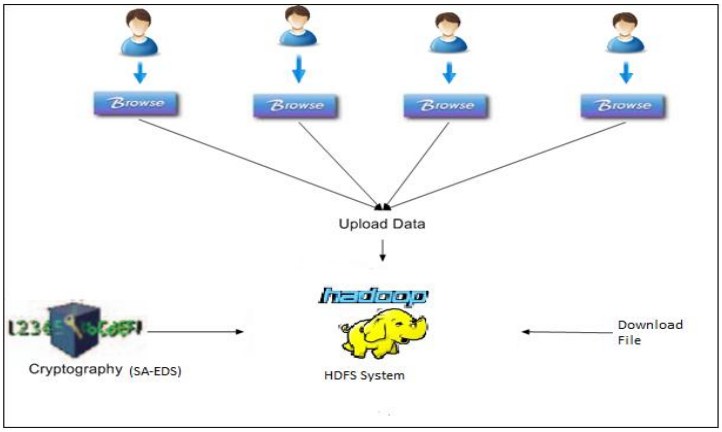

Figure 3.1: System Architecture

d. Algorithmic Implementation

1. ID-Based Ring Signature Algorithm:

- Setup: On input an unary string 1^{λ} where λ is a security parameter, the algorithm outputs a master secret key msk for the third party PKG (Private Key Generator) and a list of system parameters param that includes λ and the descriptions of a user secret key space D, a message space M as well as a signature space ψ .
- Extract: On input a list param of system parameters, an identity IDi ϵ {0,1}* for a user and the master secret key msk, the algorithm outputs the user's secret key ski ϵ D such that the secret key is valid for time t = 0. In this paper, we denote time as non-negative integers. When we say identity IDi corresponds to user secret key ski;0 or vice versa, we mean the pair (IDi,ski;0) is an input-output pair of Extract with respect to param and msk. Update. On input a user secret key ski;t for a time period t, the algorithm outputs a new user secret key ski;t+1 for the time period t + 1.
- Sign: On input a list param of system parameters, a time period t, a group size n of length polynomial in λ , a set Ł = {IDi ϵ {0.1}* l i ϵ [1,n] } of n user identities, a message m ϵ M, and a secret key $sk_{\pi,t}$ ϵ D , π ϵ [1, n] for time period t, the algorithm outputs a signature ρϵψ.
- Verify. On input a list param of system parameters, a time period t,a group size n of length polynomial in λ,a set Ł = {IDi ϵ {0.1}* l i ϵ [1,n] }of n user identities, a message m ϵ M, a signature ρϵψ, it outputs either valid or invalid.

2. Security-Aware Efficient Distributed Storage (SA-EDS):

Secure Efficient Data Distributions (SED2) Algorithm :

The inputs of this algorithm include the *Data* (D), a random split binary parameter *C*. The length of C is shorter than D. The outputs include two separate encrypted data α and β.

1) Input data packet D and C. Data C needs to be a nonempty set that is shorter than D. C should not be as same as D.

2) Create and initialize a few dataset, R, α, and β; assign 0 value to each of them.

3) Randomly generate a key *K* that is stored at the user's special register for the purpose of encryption and decryption. We calculate the value of R by (D-C), then execute two XOR operations to obtain the data value stored in the HDFS. The data in the remote storage are denoted to α and β. We use the following formulas to gain α and β: $\alpha = C \oplus K$; $\beta = R \oplus K$.

4) Output α and β and separately store them in the different datanode.

Efficient Data Conflation (EDCon) Algorithm:

EDCon algorithm is designed to enable users to gain the information by converging two data components from different datanodes. Inputs of this algorithm include two data components from datanode α, β, and *K*. Output is user's original data *D*.

1) Input the data, α and β, that are acquired from different cloud servers. The user gains the key *K* from the special register. Initialize a few dataset γ, γ', and *D*.

2) Do the XOR operation to both α and β by using *K*. Assign the value to γ and γ', respectively. $\gamma \leftarrow \alpha \oplus K$, $\gamma' \leftarrow \alpha \oplus K$

3) Sum up γ and γ' and assign the summation to *D*, as $D = \gamma + \gamma'$.

4) Output D.

4. RESULT ANALYSIS

The system will became more secure due to SA-EDS algorithm. While uploading file to the HDFS system, the implemented system will split the data according to nodes in such a way that even Hadoop administrator is unable to plaintext the split content so it proves more data security. Only authorized person will access the file data, so it will increase the data security. Ultimately, it increases data integrity.

As shown in below graph, the system is checked with various file size and in comparison it will take less time. Similarly, System implementation also provides durability for file downloading.

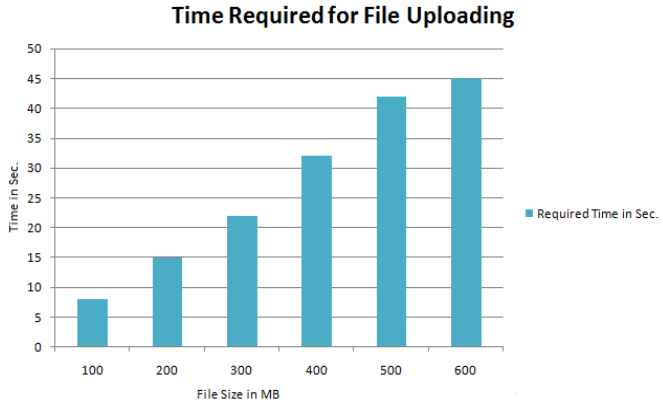

Figure 4.1: Time Required for File Uploading

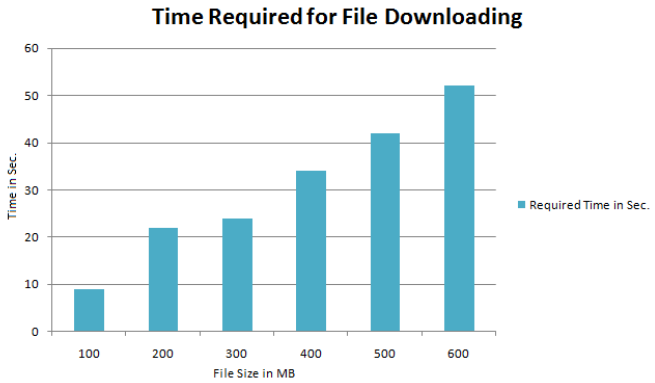

Figure 4.2: Time Required for File Downloading

CONCLUSION

In this architecture we implemented authentic data sharing and secure distributed big data storage architecture for protecting Big Data. Map Reduce framework has used to find the number of users who used to log into the data center. This framework protects the mapping of various data elements to each provider using implemented architecture. Though this approach requires high implementation effort, it provides valuable information that can have high impact on the next generation systems. Our future work is to extend the secure distributed big data storage for real time processing of streaming data.

REFERENCES

[1] Chang, F., Dean, J., Ghemawat, S., Hsieh, W. C., Wallach, D. A., Burrows, M., & Gruber, R. E. (2008). Bigtable: A distributed storage system for structured data. ACM Transactions on Computer Systems (TOCS), 26(2), 4.

[2] Agrawal, Divyakant, Sudipto Das, and Amr El Abbadi. "Big data and cloud computing: current state and future opportunities." In Proceedings of the 14th International Conference on Extending Database Technology, pp. 530-533. ACM, 2011.

[3] Keke Gai, Meikang Qiu, Hui Zhao," Security-Aware Efficient Mass Distributed Storage Approach for Cloud Systems in Big Data" IEEE 2nd International Conference on Big Data Security on Cloud, IEEE International Conference on High Performance and Smart Computing, IEEE International Conference on Intelligent Data and Security, 2016.

[4] Xinyi Huang, Joseph K. Liu, Shaohua Tang, Yang Xiang, Kaitai Liang, Li Xu, Jianying Zhou ,"Cost-Effective Authentic and Anonymous Data Sharing with Forward Security", IEEE TRANSACTIONS ON COMPUTERS VOL: 64 NO: 6 YEAR 2015

[5] K. Gai and S. Li. Towards cloud computing: a literature review on cloud computing and its development trends. In *2012 Fourth Int'l Conf. on Multimedia Information Networking and Security*, pages 142–146, Nanjing, China, 2012.

[6] M. Qiu, H. Li, and E. Sha. Heterogeneous real-time embedded software optimization considering hardware platform. In *Proceedings of the 2009 ACM symposium on Applied Computing*, pages 1637– 1641. ACM, 2009.

[7] M. Qiu, E. Sha, M. Liu, M. Lin, S. Hua, and L. Yang. Energy minimization with loop fusion and multi-functional-unit scheduling for multidimensional DSP. *J. of Parallel and Distributed Computing*, 68(4):443–455, 2008.

[8] K. Gai and A. Steenkamp. A feasibility study of Platform-as-a- Service using cloud computing for a global service organization. *Journal of Information System Applied Research*, 7:28–42, 2014.

[9] C. Wang, Q. Wang, K. Ren, N. Cao, and W. Lou. Toward secure and dependable storage services in cloud computing. *IEEE Trans. On Services Computing*, 5(2):220–232, 2012.

[10] Y. Li, W. Dai, Z. Ming, and M. Qiu. Privacy protection for preventing data over-collection in smart city. *IEEE Transactions on Computers*,PP:1, 2015.