Authentic and Anonymous Data Sharing with Data Partitioning in Big Data

Mr. Shriniwas Patilbuwa Rasal[1]
1M.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 : A 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 proposed 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, ID Based Ring Signature, SA-EDS, Data Sharing.*

*****

1. INTRODUCTION

The popularity and widespread use of Hadoop have brought great convenience for data sharing and collection. Not only can individuals acquire useful data more easily, sharing data with others can provide a number of benefits to our society as well. As a representative ex-ample, consumers in Smart Grid can obtain their energy usage data in a fine grained manner and are encouraged to share their personal energy usage data with others, e.g., by uploading the data to a third party plat- form such as Microsoft Hohm . From the collected data a statistical report is created, and one can compare their energy consumption with others (e.g., from the same block). This ability to access, analyze, and respond to much more precise and detailed data from all levels of the electric grid is critical to efficient energy usage. Due to its openness, data sharing is always deployed in a hostile environment and vulnerable to a number of security threats. Taking energy usage data sharing in Smart Grid as an example, there are several security goals a practical system must meet, including: Data Authenticity. In the situation of smart grid, the statistic energy usage data would be misleading if it is forged by adversaries. While this issue alone can be solved using well established cryptographic tools (e.g., message authentication code or digital signatures), one may encounter additional difficulties when other issues are taken into account, such as anonymity and efficiency; Anonymity. Energy usage data contains vast information of consumers, from which one can extract the number of persons in the home, the types of electric utilities used in a specific time period, etc. Thus, it is critical to protect the anonymity of consumers in such applications, and any failures to do so may lead to the reluctance from the consumers to share data with others; and Efficiency. The number of users in a data sharing sys- tem could be HUGE (imagine a smart grid with a country size), and a practical system must reduce the computation and communication cost as much as possible.

A distributed system is a pool of autonomous compute nodes [1] 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, [2] is termed for a collection of data sets which are large and complex and difficult to process using traditional data processing tools. The need for Big Data management 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. A Hadoop MapReduce cluster employs a masterslave architecture where one master node (JobTracker) manages a number of worker nodes (TaskTrackers). Hadoop launches a MapReduce by first splitting (logically) the input dataset into multiple data splits. Each map task is then scheduled to one TaskTracker node where the data split resides. A Task Scheduler is responsible for scheduling the execution of the tasks as far as possible in a data-local manner.

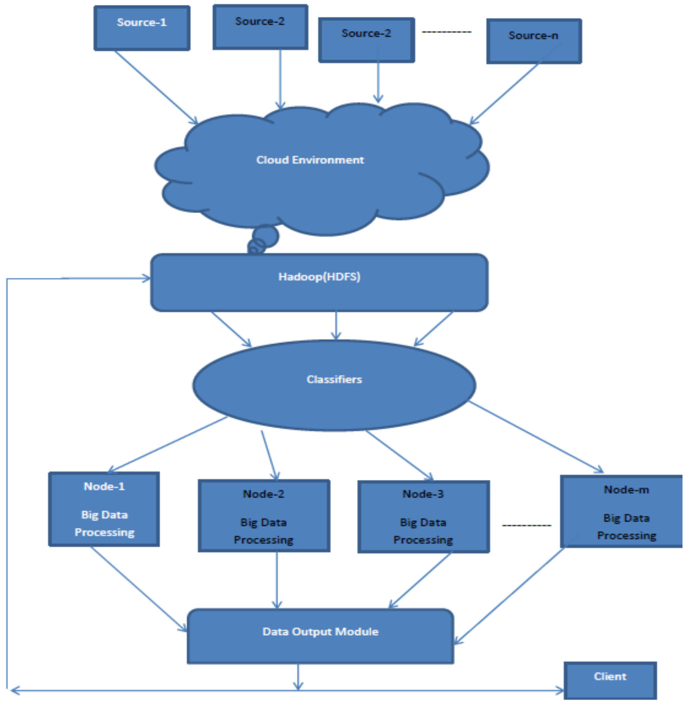

Figure 1.1: A simple Hadoop Network

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 reached by cloud service operators. The proposed scheme is entitled as Security-Aware Efficient Distributed Storage (SAEDS) model, which is mainly supported by the proposed algorithms, named Secure Efficient Data Distributions (SED2) Algorithm and Efficient Data Conflation (EDCon) Algorithm. Our experimental evaluations 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 anonymously authenticate his data which can be putinto the cloud for storage or analysis purpose. Yet the costly certificate verification in the traditional public key infrastructure (PKI) setting becomes a bottleneck for this solution to be scalable. Identity-based (ID-based) ring signature, which eliminates 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 scheme, prove its security and provide an implementation to show its practicality.

3. Concepts And Implementation

A. System Definition: Before the data are stored, the input data are partitioned into functional units. Our approach is designed to divide the sensitive data into encrypted parts for distributed storage in HDFS systems. The main problem addressed by SA-EDS is to avoid Data Center reaching the data without reducing the efficiency performance.

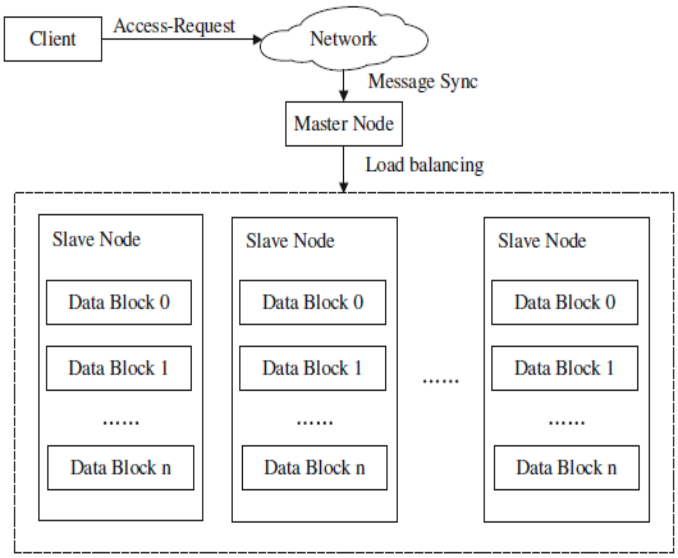

Figure 3.1: Storage Architecture

As shown in the below figure, when the user uploads file to the HDFS system then first that file get encrypted by using SA-EDS(Security-Aware Efficient Distributed Storage) Algorithm. The main purpose of SA-EDS is to avoid plaintext storage of data. On the other hand, while downloading the file from HDFS system. First user need to send request to the data center. Data center will analyze the user authentication for granting permission to him. Once the permission has been granted, user is now has an access to that file. This module shows the data security through trusted data center.

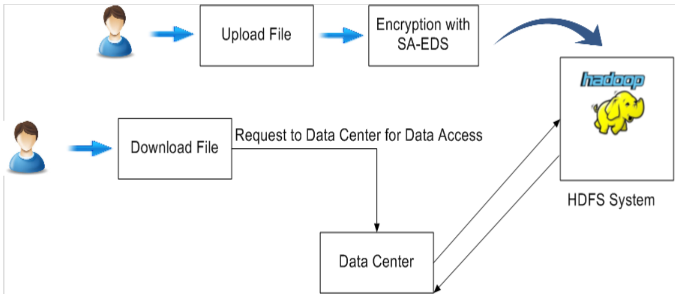

Figure 3.2: System Architecture

B. Algorithm:

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: $\alpha = C \bigoplus K$; $\beta = R \bigoplus K$.

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

```
Algorithm : Secure Efficient Data Distributions (SED2) Algorithm
Require: $D$, $C$
Ensure: $\alpha$, $\beta$
Input $D$, $C$
Initialize $R \leftarrow 0$, $\alpha \leftarrow 0$, $\beta \leftarrow 0$
/* C is a random binary that is shorter than D */
Randomly generate a key $K$
FOR $\forall$ input data packets
    IF $D \neq C$ && $C \neq 0$
        DO $R \leftarrow D - C$
        $\alpha \leftarrow C \bigoplus K$
        $\beta \leftarrow R \bigoplus K$
    ENDIF
ENDFOR
Output $\alpha$, $\beta$
```

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*.

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*.

```
Algorithm : Efficient Data Conflation (EDCon) Algorithm
Require: $\alpha$, $\beta$, K
Ensure: $D$
Input $\alpha$, $\beta$, K
Initialize $\gamma \leftarrow 0$, $\gamma' \leftarrow 0$, $D \leftarrow 0$
/* User receives $\alpha$, $\beta$ from separate cloud servers*/
$\gamma \leftarrow \alpha \bigoplus K$
$\gamma' \leftarrow \beta \bigoplus K$
$D \leftarrow \gamma + \gamma'$
Output D
```

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

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

4) Output D.

4. Expected Result

The system will became more secure due to SA-EDS algorithm. While uploading file to the HDFS system, the proposed 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.

CONCLUSION

In this architecture we proposed authentic data sharing and secure distributed big data storage in cloud computing Architecture for protecting Big Data in Cloud Computing

Environment. Map Reduce framework has used to find the number of users who used to logged into the cloud data center. Proposed framework protects the mapping of various data elements to each provider using implemented architecture. Though this proposed approach requires high implementation effort, it provides valuable information for cloud computing environment that can have high impact on the next generation systems. Our future work is to extend the proposed secure distributed big data storage in cloud computing Architecture 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.