Indian Journal of Science and Technology, *Vol 7(S4), 56–62, April 2014*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Authentication Service in Hadoop using One Time Pad

Nivethitha Somu*, A. Gangaa and V. S. Shankar Sriram

School of Computing, SASTRA University, Thanjavur, Tamil Nadu, India;
nivethithasomu@gmail.com, er.gangaacse@gmail.com, sriram@it.sastra.edu

Abstract

"Big Data" - voluminous and variety of data from different sources, which demands innovative processing and analysis for decision - making analysis. The data can be either in form of structured or unstructured data. Processing big data with the traditional processing tools and the present relational database management systems tends to be a difficult task. Parallel execution environment, like Hadoop is needed for processing voluminous data. For processing the data in a open framework like Hadoop we need a highly secure authentication system for restricting the access to the confidential business data that are processed. In this paper, a novel and a simple authentication model using one time pad algorithm that removes the communication of passwords between the servers is proposed. This model tends to enhance the security in Hadoop environment.

Keywords: Big Data, Hadoop and One Time Pad

1. Introduction

Look out at the technology we have in our world today, it simply states that everything revolves around data. We always have an attraction to rich media content. This appetite results in the generation o huge voluminous data and the rate of generation increases from a simple email to web links. These data needs to be processed in an innovative way to extract the useful data for decision - making. Organization need to know the value of this huge amount of data in order to improve their services to the customer.

Big Data – the word itself implies its meaning of huge data sets. These data are huge voluminous data, which needs to be processed for *business analytics*. For example, when you search for a product in an online site, at the bottom of the page there will be some products displayed as recent search. The recent search may contain products that you have searched in your previous visit or by some other person or the ones someone has bought along with your search item. Have you ever thought of how the recent searches are getting filtered out of all the products? Nothing too complex all these analysis can be made through processing big data. But for processing big data we need a secure parallel environment. Hadoop can be made as a processing environment for big data and all the data for business analytics can be processed in it.

Hadoop[1] is a scalable, open – source, java - based framework for distributed large scale processing. It scales from a single machine to multiple thousands of servers to set up an execution platform with local computation, storage and high availability. Hadoop comprises of Hadoop Distributed File System (HDFS) and MapReduce. HDFS consists of geographically dispersed Data Nodes where the user data resides. Initially, Hadoop had no security framework it assumes that the entire cluster, user and the environment were trusted. Even though it had some authorization controls like (file access permissions) a malicious user can easily impersonate a trusted used as the authentication were based on Password. Later on, Hadoop cluster moved on to private networks, where the

*Author for correspondence

users were given equal rights to access the data in the cluster. Equal access to all users enables the malicious user.

1. to read or modify the data in the other's cluster.
2. to suppress or kill the other job to complete his job earlier than the other to execute job from a malicious user as the data node does not enforce access control policies

Hadoop relies completely on Kerberos[2 3,8] for authentication between the client and the server. An encrypted token the authentication agent will be requested by the client. Using this, he can request for a particular service from the server.

Kerberos is ineffective against Password guessing attacks and does not provide multipart authentication.

2. Hadoop

Hadoop is a distributed framework with the distributed file system and MapReduce for processing the data in parallel on a large set of clusters. Hadoop comprises of two major components:

i. Hadoop Distributed File System (HDFS)
ii. MapReduce

2.1 HDFS

HDFS consists of server called NameNode and a set of DataNode. The NameNode stores the file directory, name space and metadata. The DataNode store files in terms of blocks identified by the block id.

In the NameNode user can perform operations like create and open file through the RPC (Remote Procedure Call) protocol. In the DataNode the user can read or write the data through the data-transfer protocols.

2.2 MapReduce

MapReduce is a framework for processing huge data sets in parallel. The DataNode acts as compute node to have the computation nearby the data processing node. Each node has a TaskTracker which performs map and reduce on the submitted job. JobTracker schedules to job submitted by the user on certain compute node.

The authentication between the user and the JobTracker is done through Kerberos using RPC. A submitted job should run with the user identity and permission, as of now there exist no authentication mechanism for MapReduce than Service Level Agreement (SLA). MapReduce stores the information about the executing and remaining jobs in HDFS.

User writes the configuration, input and the metadata for the input in to the home directory, which is fully under user control. Then the user passes on the directory location and the security information through the RPC to the JobTracker. These security information will be stored in Map and the delegation tokens will be put up with the NameNode. JobTracker will renew these tokens periodically, till the job completes.

2.3 Challenges in Designing Security Mechanism for Hadoop

Hadoop uses a different interaction pattern rather than the classic client – server interaction model where the client is authenticated and authorized for each operation through the Access Control List (ACL). Adding security to Hadoop seems to be a difficult task due to the following factors:

i. Scale of the system
ii. Hadoop is a distributed file system (File is partitioned and distributed throughout the cluster)
iii. Later job execution on a different node than the node where the user authenticated and job submission is done.
iv. Task from different users may be executed on a single node
v. Users can access the system through some workflow system.

2.4 Security Threats in Hadoop

The area of security breach in Hadoop are

i. Unauthorized user can access the HDFS file
ii. Unauthorized user can read/write the data block
iii. Unauthorized user can submit a job, change the priority, or delete the job in the queue.
iv. A running task can access the data of other task through operating system interfaces

Hadoop provides shared multi-tenant service to store sensitive data. Financial organizations using Hadoop started to put up their confidential data on Hadoop clusters. So, there comes a need for a strong authentication and authorization mechanism to protect the sensitive data. Hadoop does not follow any classic interaction model as the file system is partitioned and the data resides

in clusters at different points. One of the two situations can happen: job runs on another node different from the node where the user is authenticated or different set of jobs can run on a same node.

Some of the possible solutions can be

- Access control at the file system level.
- Access control checks at the beginning of read and write
- Secure way of user authentication

Authorization is the process of specifying the access **right to the resources** that the user can access. Without proper authentication service one cannot assure proper authorization. Password authentication is ineffective against[4].

- **Replay attack** - Invader copies the stream of communications in-between two parties and reproduces the same to one or more parties.
- **Stolen verifier attack** - *Stolen verifier attack* occur when the invader snips the Password verifier from the server and makes himself as an legitimate user.

The rest of the article is structured as follows; *Section* 2 briefs about Hadoop, *Section* 3 details about the existing authentication mechanism in Hadoop, *Section* 4 explains the proposed authentication mechanism using onetime pad, *Section* 5 expresses the performance analysis of the proposed mechanism *Section* 6 concludes the paper.

3. Existing System Architecture for Authentication Services

The system architecture plays a major role in the design of any authentication service. The existing architecture for authentication falls into four categories[5]. In each authentication model there seems to be some sort of loophole.

Figure 1 (i) consists of a single server, which stores the Password and is subjected to single point of vulnerability and offline dictionary attacks. Figure 1 (ii) consists of multiple servers where the user can communicate in parallel.

Communication bandwidth demand and synchronization issues are the major challenges.

Figure 1 (iii) consists of a gateway introduced between the user and the multi - server, creating a redundant layer for communication, thus reduces the system reliability.

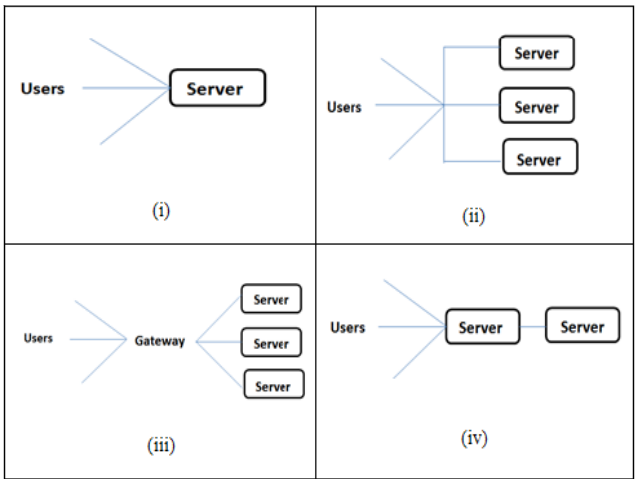

Figure 1. Existing Authentication Mechanism.

Figure 1 (iv) consists of two server for authentication, one which is made public to the user and the other is kept at the Backend. The Proposed architecture is similar to the two - server model in Figure 1 (iv).

The proposed architecture involves the action of two servers: **Registration Server** and **Backend Server** for authentication (Figure 2). This model is similar to the two – server model as the Registration Server remains fully visible to the user and the Backend Server remains hidden from the user. It reduces the communication and synchronization complexity. The Password encrypted using one time pad algorithm (**Cipher Text 1**) will be stored in the Registration Server. The Cipher Text 1 will be encrypted using mod 26 operation resulting in Cipher Text 2 which is stored in the Registration Server. Once again, the onetime pad key is encrypted with the password forming (**Cipher Text 3**) and is stored in the Back end Server along with the Username.

In case of any vulnerability attack to the Backend Server, the authentication process is strong enough to withstand.

If an attacker steals the data from the Backend Server, he cannot retrieve the password as the process of decryption depends on the random key generated by the one - time pad algorithm. The random key is valid only for a particular session till the user logs out. Once the user logs out, a new random key replaces the old key and the entire process of encryption depends on the new random key. Our system is robust against replay, guessing and stolen verifier attack as a new random key will be generated for each login, so it is hard to decrypt the encrypted text.

4. Literature Survey

Kai Hwang et al.[4] proposed a integrated approach to protect clouds/datacenters using DDOS defense mechanisms, access control techniques and piracy prohibition methods. The proposed security mechanism is inconvenient to the acceptance of web scale computing in commercial world. Hamlen et al.[6] proposed a layered framework for security in cloud storage and data layers. A Secure Co Processor (SCP) is used for e client storage of encrypted data in cloud. High computation cost and limited memory become an inconvenient when implementing SCP.

George Kousiouris et al.[7] enhanced the security model of Hadoop by using virtual private networks. Eavesdropping and network attacks were prevented and every data node is authenticated. But the implementation of this approach is still in progress and not yet completed.

G.SudhaSadasivam et al.[5] proposed approaches for authentication service for Hadoop in cloud environment. They used the properties of triangle for authenticating the user. To enhance the security level of Hadoop cloud, dual servers were used. If one server is hacked and theta value is modified then user validation cannot be successful.

5. Proposed Architecture

The proposed approach provides authentication service by using one time pad and symmetric cipher cryptographic technique. This approach uses two - server model, with a Registration Server and a back end server. The whole process of authentication consists of two parts:

1. Registration Process
2. Authentication Process

During the registration process, the user enters his Username and Password. The Password is encrypted (**Cipher Text 1**) using one-time pad algorithm. Cipher Text 1 is again encrypted using mod 26 operation (**Cipher Text 2**) and stored in the Registration Server. Again, encrypt the onetime pad key using the Password which results in (**Cipher Text 3**) using symmetric cipher technique. Cipher Text 3 will be sent to the Backend Server to be stored along with the Username.

Next during the authentication process, after receiving the Username from the user, the Registration Server sends the Username to the user. The Backend Server sends the corresponding Cipher (**Cipher Text 3**) to the User via Registration Server. The user deciphers it using his Password and returns the key to Registration Server.

Registration Server decrypts Cipher Text 1 with the key returned by the User. Again encrypts the Password with same key and send the Cipher (**Cipher Text 4**) to the Backend Server.

The Backend server compares Cipher Text 4 with Cipher Text 3. If it matches, sends the Username to the Registration Server. The Registration Server compares the Username with the Username entered by the user. If it matches the user is authenticated. The random is valid only for one session. Once the user logs out, a new random key replaces the old one.

The various steps involved in one – time pad authentication scheme are as follows:

5.1 Registration Process

Figure 2 depicts the steps in Registration.

1. Using the Username and Password, the user logs onto the Registration Server.
2. Encrypt the Password using
   a. One time pad key
   b. Mod 26 operation
3. Store the resultant cipher (**Cipher Text 2**) in the Registration Server.
4. Encrypt the randomly generated one-time pad key using Password through symmetric cipher technique
5. Store the encrypted text (**Cipher Text 3**) along with Username to Backend Server

5.2 Authentication Process

Figure 3 depicts the steps in Authentication:

1. The user enters the Username to the Registration Server (**Fig 3**) Store the resultant cipher (**Cipher Text 2**) in the Registration Server.
2. Store the resultant cipher (**Cipher Text 2**) in the Registration Server.
3. The Registration Server sends the Username to the Backend Server.
4. The Backend Server returns the corresponding cipher (**Cipher Text 3**) of the appropriate Username to the Registration Server.
5. Send the cipher (**Cipher Text 3**) to user.

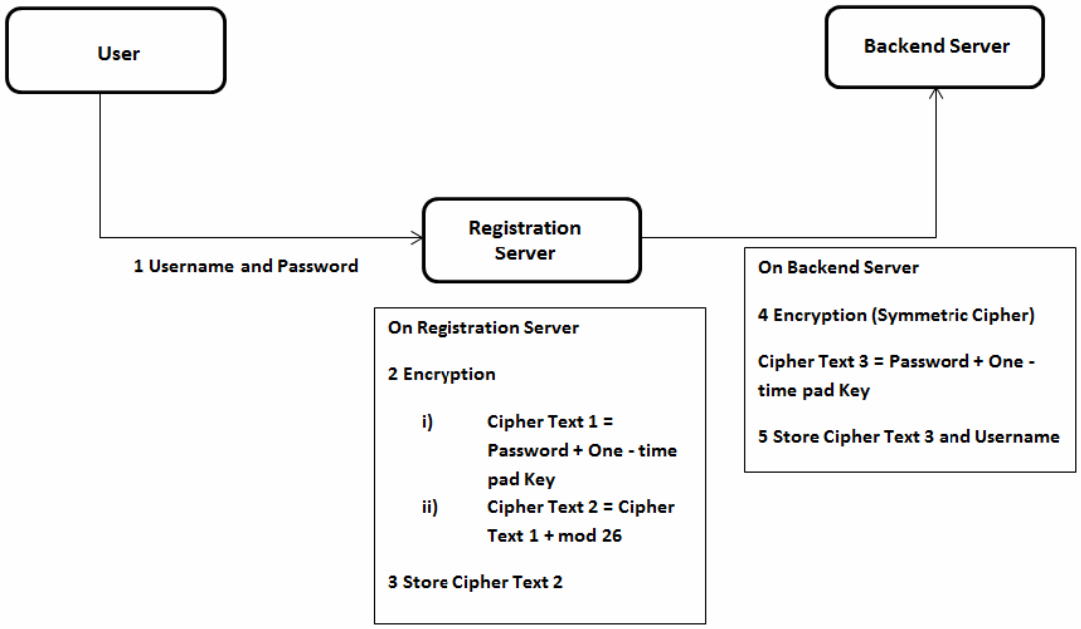

Figure 2. Steps in Registration

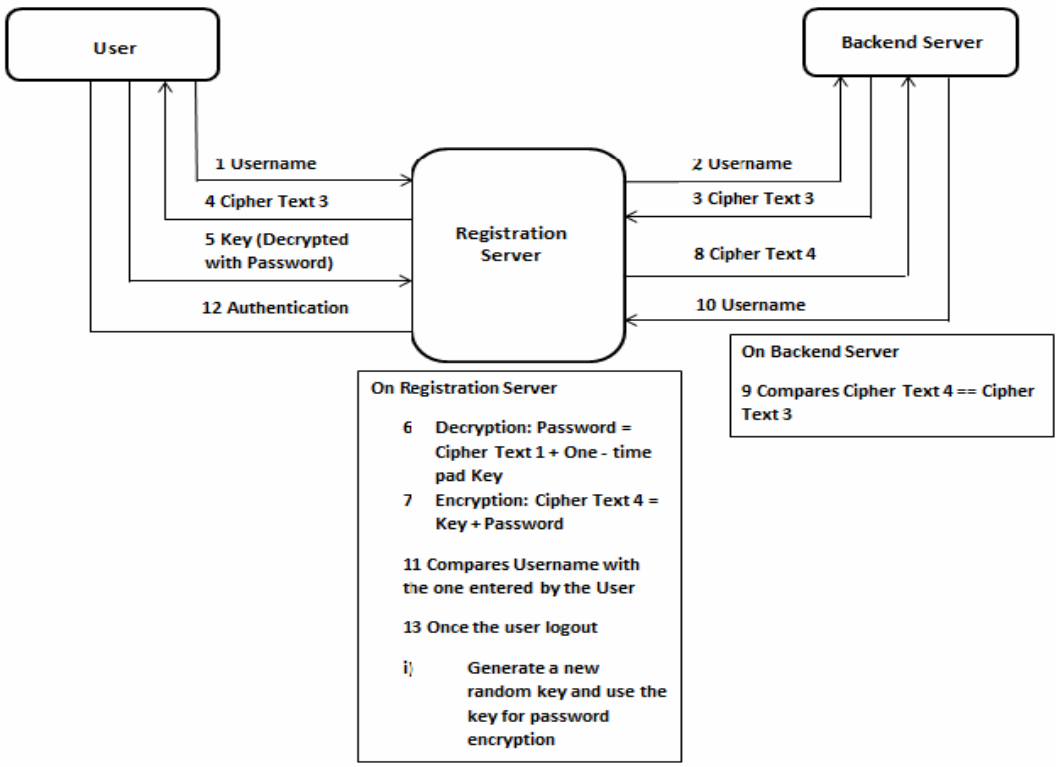

Figure 3. Steps in Authentication.

6. User deciphers the text using his Password and returns the key to Registration Server.
7. Registration Server uses the key to decrypt the Cipher Text 1 stored during registration process.
8. Encrypt the Password with the key (**Cipher Text 4**) by using symmetric cipher technique.
9. Send the encrypted text (**Cipher Text 4**) to the Backend Server.
10. The Backend Server compares the cipher (**Cipher Text 4**) with the stored cipher (**Cipher Text 3**).
11. If both the Cipher matches, the Backend Server returns the corresponding Username to the Registration Server.
12. The Registration Server checks the Username with the Username entered by the user.
13. If it matches, then the user will be authenticated.

6. Result Analysis

This section deals with the security analysis of the one - time pad Hadoop authentication service. The major extortions faced by the authentication scheme in Hadoop and the onetime pad authentication scheme solution to overcome the attacks are detailed below.

1. **Replay-attack:** User credentials are sniffed by the attacker when the user communicates with the server. To overawe this, the user has to randomly change the credentials. Our authentication scheme stores the onetime pad key as an encrypted text using the Password in the Backend Server. When a replay attack occurs, the hacker cannot retrieve the key easily as it is encrypted with the Password and the key stored is valid only for that session.

2. **Guessing attack:** The adversary contacts the server with randomly generated user credentials. By choosing the Password with maximum character, the probability of guessing attack can be lowered. The proposed authentication scheme uses a randomly generated key for each session to encrypt the Password. Even if the adversary guesses the Password, he may not be able to attack the server for the subsequent session.

3. **Stolen verifier attack**: In general, the password verifier is stored in the server rather than the original password. Adversary can steal the verifier from the server and act as a legitimate user. As the proposed system stores the encrypted Password with the randomly generated key for the session, even if the hacker has the key he cannot reveal the Password from it as the key will be valid for the current session.

The entire security and the simplicity of the system rely on the random key generation. Our approach provides more security than the existing system through generating the new random key for each session. The complexity analysis of the proposed algorithm in terms of communication and computation is given in Figure 4 (Registration Process) and Figure 5 (Authentication Process. The complexity analysis is done based on the number of bits and the communication rounds.

	User → Registration Server	**Registration Server → Backend Server**
Communication	{UN} + {PWD}	{KEY} + {PWD}
Computation	O(n)	O(n)
Communication Rounds	1	1

Figure 4. Complexity analysis of registration process

	User → Registration Server	**Registration Server → Backend Server**	**Backend Server → Registration Server**	**Registration Server → User**
Communication	{UN} + {KEY}	{UN} + {CT 4}	{CH 3} + {UN}	{CH 4}
Computation	O(n)	O(n)	O(n)	O(n)
Communication Rounds	1	1	1	1

Figure 5. Complexity analysis of authentication process.

7. Conclusions

The proposed authentication scheme for Hadoop is based on the encryption mechanism using onetime pad key. A random key is used to encrypt the password for secure transmission between the two servers (Registration Server and Backend Server). It makes the Hadoop environment more secure as the new random key for encryption is generated for each login. It also reduces the possibility of an adversary to reveal / decrypt the cipher stored in the server as in involves the knowledge about the random key valid for that session. The randomly generated key enhances the security of the authentication scheme in a Hadoop environment.

8. References

1. Turkington G. Hadoop Beginner's Guide. PACKT Publishing; 2013.
2. White T. Hadoop Definitive guide OReilly, 2009.
3. Owen O Malley. Integrating Kerberos into Apache Hadoop Kerberos. Conference 2010, 2010 26–27 Oct, MIT, USA.
4. Hwang K, Kulkarni S, Hu Y. Cloud security with virtualized defense and reputation - based trust management. Eighth IEEE International Conference on Pervasive Intelligence and Computing, (PICom2009). 2009; Chengdu, China. p. 717–22.
5. SudhaSadasivam G, AnithaKumari K, Rubika S. A novel authentication service for hadoop in cloud environment. IEEE International Conference on Cloud Computing in Emerging Markets (CCEM). 2012. p.16.
6. Hamlen K, Kantarcioglu M, Khan L, Thuraisingham B. Security issues for cloud computing. International Journal of Information Security and Privacy. 2010 Apr–Jun; 4(2): 39-51.
7. Kousiouris G, Vafiadis G, Varvarigou T. A frontend, Hadoopbased data management service for efficient federated clouds. IEEE Third International Conference on Cloud Computing Technology and Science (CloudCom). 2011; p. 511–16.
8. Lin CL, Hwang T. A password authentication scheme with secure password updating. Comput Secur. 2003; 22(1): 68-72.