Hadoop Identity Authentication using Public Private Key Concept

Risha Tabassum[#1], Dr. Nidhi Tyagi[*2]

[#]*M.Tech Scholar, Department of Computer Science, MIET (Meerut), UPTU*

***Abstract*— *Protection from unauthorized access to data and information is notable challenge within data transmission process. One of the best known public key encryption algorithms is the RSA (Rivest, Shamir, Adleman) algorithm [3], which is based on the principles of number theory. Hadoop supports to authenticate its clients and users using Kerberos for security. This paper suggests the authentication mechanism of Kerberos protocol under HDFS and provide security to the communication channel with help of RSA. It modifies Kerberos protocol by using RSA public key encryption and data signature mechanism. It provides a more reliable and efficient identity authentication solution for HDFS.*

Key words*— *HDFS, Kerberos, Public key encryption, RSA.

I. Introduction

Hadoop was developed from GFS (Google File System) and MapReduce papers published by Google in 2003 and 2004 respectively[1,2]. It has been popular recently due to its highly scalable distributed programming or computing framework, it enables processing big data for data-intensive applications as well as many analytics. Hadoop is a framework of tools which supports running application on big data and it is implemented in java. It provide MapReduce programming architecture with a Hadoop distributed file system(HDFS), which has massive data processing capability with thousands of commodity hardware using map and reduce functions. Since Hadoop is executing in large cluster or may be in a public cloud service like Yahoo, Amazon, Google, etc. are such public cloud where many users can run their jobs using Elastic MapReduce and cloud storage that is used as Hadoop distributed file system then thee arises a need to implement the security of user data on storage or cluster. Encryption and decryption is key means for securing Hadoop file system(HDFS), where many DataNodes store file to HDFS and are transferred while executing MapReduce job. In today's era, internet now initiate huge amount of data every day. The volume of digital content on internet grows up to more than 2.7 ZB in 2012 which is up 48% from 2011 and now rocketing towards more than 10 ZB by 2015. In recent years, more than 70% of big data applications are running on Hadoop. The two layers of Hadoop are, Computation layer uses Map Reduce as its framework for providing computational capabilities. Distributed Storage layer provide storage for HDFS.

.

II. LITEATURE SURVEY

Hadoop supports to authenticate its clients and users using Kerberos for security. The Kerberos authentication system [Stei88, Mill87, Brya88] was introduced by MIT to meet the needs of Project Athena. It has since been adopted by a number of other organizations for their own purposes, and is being discussed as a possible standard. Kerberos has a number of limitations and weaknesses; a decision to adopt or reject it cannot properly be made without considering issues. Despite Kerberos’s many strengths, it has a number of limitations and some weaknesses. Some are due to specifics of the MIT environment; others represent deficiencies in the protocol design [4].

III. PRINCIPLE AND ARCHITECTURE OF HDFS

Principle- HDFS reads and writes file data by using stream. It’s especially suitable for the task whose data is only written once but read and analysed more than once. The client needs to communicate with NameNode to get the information of the DataNode’s position which it has file operations on. After that, the client can carry out operations of files.

Architecture- The files on Hadoop file system (HDFS) are split into different blocks and replicated with multiple DataNodes to ensure high data availability and durability to failure of execution of parallel application in Hadoop environment. Originally Hadoop clusters have two types of node operating as master-salve or master-worker pattern [5]. NameNode as a master and DataNodes are workers nodes of HDFS. Where data files are actually located in Hadoop is known as DataNode which only leads storage. However NameNode contains information about where the different file blocks are located but it is not persistent, when system starts block may changes one DataNode to another DataNode but it report to NameNode or client who submits the MapReduce job or owner of Data periodically [6]. The communication is in between DataNode and client NameNode only contains metadata.

For distributed storage, HDFS i.e hadoop distributed file system and for distributed processing, MapReduce paradigm has been provided by Hadoop [7]. Other components are Yarn and Hadoop Common.

IV. ANALYSIS

A. Problem Identification

i. Analysis of Kerberos protocol

At the initial stage of the Hadoop design, there was no authentication mechanism and assumes that the cluster was in a trusted domain. As the popularity of Hadoop application growing, security problems growing seriously as well. In order to deal with the security flaws in Hadoop, a third party security authentication mechanism based on Kerberos protocol was introduced in to make sure the credibility between communication nodes.

ii. HDF mechanism based on KERBEROS protocol

The latest version was Kerberos V5 protocol. The protocol can provide credible identity authentication mechanism in unsafe network for communication with the nodes. To set up an authentication center KDC (Key Distribution Centre) used to keep usernames, passwords and other information of clients, NameNodes and DataNodes in cluster and to offer services of identity authentication and authorization is the basic principle of Kerberos in HDFS environment. Two logically independent servers i.e. Authenticatin Server(AS)and Ticket Grant Server(TGS) together forms KDC.

In the cluster, firstly, any user who wants to apply for service needs to communicate with AS to get Ticket Grant Ticket (TGT). Secondly, it gets Ticket for service by communicating with TGS using TGT. Finally, the user communicates with the node that provides services to get services by Ticket.

The specific implementation processes of Kerberos protocol under HDFS are shown in figure.3.

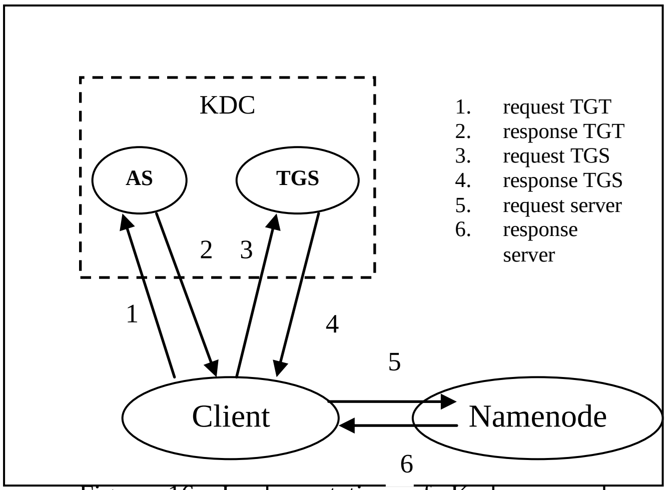


Figure 16: Implementation of Kerberos under HDFS

B. Security analysis of Kerberos protocol under HDFS

The introduction of Kerberos protocol solves the following security problems in original HDFS cluster.

1) Because of the dynamic scalability of Hadoop cluster, illegal user can disguise as a DataNode server and join to the cluster to receive data information from NameNode.
2) Illegal user disguises as authorized user by altering data package to request service sources
3) In an unsafe network environment, illegal user can intercept the exchange project of datagram and disturb the normal operation of NameNode or DataNode by replay attacks.

The Kerberos protocol provides identity authentication mechanism for HDFS, but there's still limitation.

The safety problems are as follows:

1) The problem of time synchronization in Hadoop cluster: In the process of Kerberos identity authentication, it's necessary to contrast timestamp to judge the authenticity of user's identity which requires the internal network of Hadoop cluster to have high ability of clock synchronization. Obviously it's difficult to achieve in Hadoop cluster that is composed of cheap commercial computers.
2) The security problem of KDC: Because of the Kerberos server stores all the passwords and other related information of clients, NameNodes and DataNodes. Once KDC is broken by a malicious user, it will cause a devastating blow to the entire Hadoop cluster.
3) The Problem of Dictionary Attack: In the process of Kerberos certification, AS server doesn't verify user's identity directly, but does it via the packet included TGT and encrypted by client secret key Kc which is postback information. Only the user knows Kc can get TGT, and then conduct subsequent authentication steps. If a malicious user collects a number of TGT information, user's password Kc is possibly cracked.
4) The problem of denial mechanism: Due to public key technology is not introduced in Kerberos protocol, so it does not provide digital signature for transmitting information, and cannot realize denial mechanism of information transmitting in authentication process.

C. Proposed Framework

In order to eliminate these limitations of Kerberos authentication mechanism, this work will make some appropriate changes on Kerberos protocol in the framework of Hadoop authentication mechanism. By bringing in asymmetric encryption for Kerberos protocol, it can make full use of the features of asymmetric key mechanism to solve the problems of Hadoop cluster authentication listed above.

1) *The improvement and implementation of Kerberos*

Asymmetrical encryption system is also called public key cryptosystem. It is composed of public key and private key that are generated by specific algorithm. The public key is public, while the private key which is the critical part of the asymmetrical encryption system is not open. In public key cryptosystem, data encrypted by public key can only be decrypted by private key. Similarly, data encrypted by private key must be decrypted by public key. Compared with symmetric key system, asymmetric key mechanism has longer key digits and separates public key and private key. As a result, it can provide more secure encryption service and is widely used in data encryption and data signature. Data signature provides verification of fingerprint level by relevant processing on protected data. It usually includes generating summary of data and encrypting the summary. Digital signature technology is based on public key cryptosystem: Before data transmission the sender uses HASH function to get a summary, and then uses the private key to encrypt the summary. The summery together with the original data is sent to the receiver. The receiver decrypts the signature information by the sender's public key, generates a summary of the original data by corresponding HASH function and contrasts the summery with the decrypted one. If there is no different between two summaries, the data will be received. Data signature is widely used in ensuring the integrity of information transmission, identity authentication of the sender and non-repudiation of electronic trading.

2) *RSA in HDFs*

Since in HDFs, there is a connection in order to communicate client with the server and server with the client as well. When the hadoop is initiated, HDFs loads all the files as per the command given. The implementation of improved Kerberos helps in solving the problem identified but what if the communication channel is not secure. This may leads to severe attacks which may result in losing of private data.

Thus, here, RSA is used in order to provide security to the communication channel by encrypting the channel between client and the server.

V. Experimental Setup and Results

It is quite expensive to build bigger servers with heavy configurations that handle large scale processing, but as an alternative, many commodity computers can be tied with single CPU, as a single functional distributed system and practically, the clustered machines can read the dataset in parallel and provide a much higher throughput. To carry out the experiment we have to create the environment of Hadoop on Windows XP and above. For this, we have installed latest version of Java or Netbeans IDE and develop a simulation in order to carry out the functioning of Hadoop on Windows platform. Also, RSA is implemented in order to provide security to the communication channel.

The framework of the proposed system is as follows:

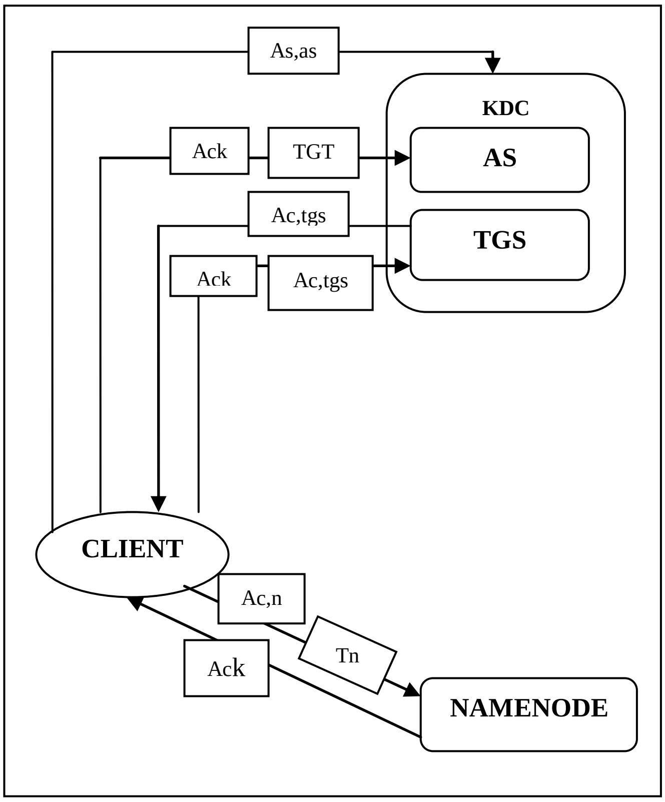


Figure 2: Improved Kerberos authentication process

A. *Client request namenode*

The specific processes of the request are shown in Figure 2, and the symbols used in certification process are as follows:

Client: Service requester

KDC: Key Distribute Center

AS: Authentication Server

TGS: Ticket Grant Server

NameNode: The server in HDFS named NameNode

Kc,as: The symmetrical key used between Client and AS, the same as Kc, tgs and Kc, n
KcPr: Private key of client, the same as KasPr, KtgsPr and KnPr
KcPu: Public key of client, the same as KasPu, KtgsPu and KnPu
IDc: The identity of client, the same as IDn
Timestamp: Timestamp
Random: Random number
Ac,tgs: The identification between AS and TGS, the same as Ac,n
Ack: Acknowledgement
TGT: Ticket Grant Ticket
Tn: The ticket can access NameNode
SKcPr {}: Sign the data by KcPr, the same as SKasPr{}, SKasPr{}, KtgsPr{}and SKnPr{}

(1) Client Request AS

{Ac,as}
Ac,as=KasPu{SKcPr{IDc,IDtgs,Kc,as,Random,Timestamp}}

Client signs IDc, IDtgs, Kc,as, Random and Timestamp by the private key KcPr, and then encrypts them by the public key of AS, finally generates the packet of {Ac,as}.It is sent to AS as the credential for requesting TGT.

(2) AS Response Client

{Ack,TGT}
Ack= Kc,as {IDc,Random }
TGT=KtgsPu{SKasPr{IDc,Random,Lifetime }}

AS uses its private key KasPr to decrypt Ac,as, client‟s public key KcPu to verify the integrity of Ac,as and tests the validity of timestamp to prevent the replay attacks. Only if the above processes are successfully validated, AS generates TGT and Ack for Client. The processes of generating TGT and Ack are as follows:
Firstly, to sign the information in TGT (including IDc, Random,Lifetime) by AS‟ private key KasPr and to encrypt them by TGS‟ public key. Afterwards, to encrypt IDc and Random included in Ack by Kc,as. Finally, to combine TGT and Ack as postback packet of {Ack,TGT}. Description: Kc,as, IDc and Random included in Ack are all from the initial decrypted message Ac,as.

(3) Client Request TGS

{Ac,tgs,TGT}
Ac,tgs=KtgsPu{SKcPr{IDc,IDn,Kc,tgs,Random, Timestamp}

Client uses symmetric key Kc,as saved locally to decrypt Ack and to compare the required IDc and Random with the native duplicate in order to prove the correctness of the message. After the above processes are successfully validated, client then generates symmetric key Kc,tgs and packet sent to TGS server. The packet is composed of Ac,tgs and TGT. Ac,tgs includes IDc,IDn,Kc,tgs,Random and Timestamp that are signed by clients private key KcPr and encrypted by TGS’ public key KtgsPu.

(4) TGS Response Client

{Ack,Tn}
Ack= Kc,tgs {IDc,Random}
Tn=KnPu{SKtgsPr{IDc ,Lifetime}}

Firstly, TGS decrypts the packet of Ac,tgs by its private key KtgsPr, verifies signing messages of Ac,tgs by client‟s public key KcPu and tests the effectiveness of timestamp to prevent replay attacks. ThenTGS uses its private key KtgsPr to decrypt TGT, AS’ public key KasPu to verify the signing information of TGT and checks whether the lifetime of TGT is in force. If the above processes are successfully validated, TGS then generates ACK and Tn for client to access NameNode. The steps of the generation of ACK and Tn are as follows:
Firstly, to sign IDc and lifetime included in Tn by TGS private key KnPr and encrypt them by NameNode’ s public key. Secondly, to encrypt IDc and Random included in Ack by Ac,tgs.
Finally, to combine TGT and Ack as postback packet of {Ack,Tn}.

(5) Client Request NameNode

{Ac,n,Tn}
Ac,n=
KnPr{SKcPr{IDc,Kc,n,Random,Timestamp}}

Client uses symmetric key Kc,tgs saved locally to decrypt Ack and to compare the required IDc and Random with the native duplicate in order to prove the correctness of the message. After the above processes are successfully validated, first of all, client generates symmetric key Kc,n to prepare for data communication with NameNode after connection establishment. Afterwards, client signs IDc, Random and Timestamp by its private key KcPr and encrypts them by the public key of NameNode KnPu to generate Ac,tgs from client to TGS. Finally, client packages Ac,tgs and Tn to form packet
of {Ac,n,Tn}sent to NameNode.

(6) NameNode Response Client

Kc,n{Random}

NameNode decrypts Ac,tgs by its own private key KnPr, verifies signing messages by client’s public key and tests the effectiveness of timestamp to prevent replay attacks. Then NameNode uses its private key KnPr to decrypt Tn, TGS’ public key KtgsPu to verify the signing information of Tn and checks the valid identification of ticket lifetime in Tn.
There’ s a need to compare IDc decrypted from Ac,tgs and Tn respectively. If the above processes are successfully validated, NameNode regards the client as the credible client that passes KDC

authentication. Finally, NameNode encrypts Random by Kc,n and sends back to client. Client decrypts the message by Kc,n and compares it with the random preserved itself to verify the identity of the server.

These are the whole processes of identity authentication.

B. Client request datanode

This part is same as Client Request NameNode.

VI. ANALYSIS OF IMPROVED KERBEROS PROTOCOL WITH RSA

A. Analysis of safety

It solves potential safety hazard of the original authentication mechanism by introducing public key encryption and data signature mechanism into Kerberos protocol. The security analysis is as follows:

(1) The improved Kerberos protocol based on public key encryption system uses distributed keys management strategy. Users keep the private keys themselves and can get the public keys from cluster. Therefore, KDC neednot to save the secret information like passwords intensively. Due to this change, it greatly improves the security of KDC. Even if the KDC is invaded, the attacker can’t get the user's private key and impersonate the user to obtain service.

(2) **Public key encryption uses longer key compared with the symmetric key encryption.** Theoretically, if the key is more than 1024 bit, it will be safety. **The improved protocol can defense dictionary attack effectively.**

(3) **Public key encryption and private key signature are used in the processes** of sending and receiving authentication messages of the user, KDC server, NameNode and DataNode. On account of the privacy of private key, the identity of the sender can be verified.

(4) **The requirements of time synchronization are reduced after the improvement of the authentication protocol.** The data encrypted by private key can only be decrypted by private key. In addition, public key encryption system is difficult to broken. As a result, timestamp is used to judge the validity of ticket and prevent replay attacks as auxiliary in improved authentication protocol.

(5) **The data when send over any communication channel will be encrypted using RSA.** If in any case, the intruder able to succeed in getting the control over the client and server then also not able to access any type of data over the communication channel.

B. Analysis of efficiency

Symmetric key encryption needs shorter time than public key encryption for the same data.

However, KDC bottleneck caused by identity authentication needs to be considered in large clusters. Specific efficiency analysis is as follows:

(1) **The improved Kerberos protocol removes the redundancy information** such as IP address transmitted between client and KDC and retains IDc and Random merely. Less data makes encryption and decryption more efficient in authentication process.

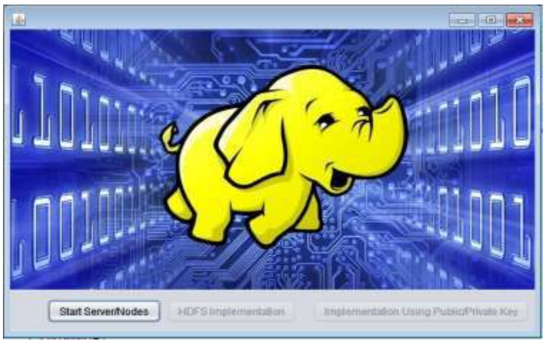
(2) The improved Kerberos protocol does not abandon the symmetric encryption mechanism absolutely, but makes an improvement. Firstly, it makes KDC liberated from generating symmetric key that the clients are responsible for doing it. So even if a large number of clients need to request for KDC, it won’t occupy KDC’s limited computing capability to generate symmetric key. Secondly, the client sends symmetric key information encrypted by public key encryption to KDC so that the information can only be decrypted by KDC. KDC uses the symmetric key to encrypt Ack information and client decrypts Ack by its own symmetric key.

(3) The use of RSA algorithm helps efficiently in a way that it will provide the encryption over the network so that there is little or no unauthorized access.

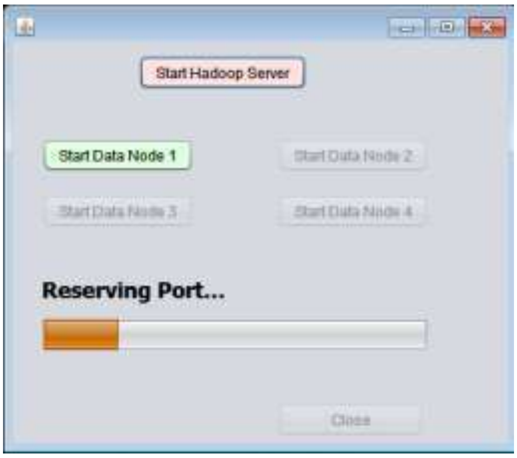
In a word, the design achieved the balance between security and efficiency

VII. OUTPUT SCREEN OF THE PROPOSED MODEL

a. This is the first screen of the project.

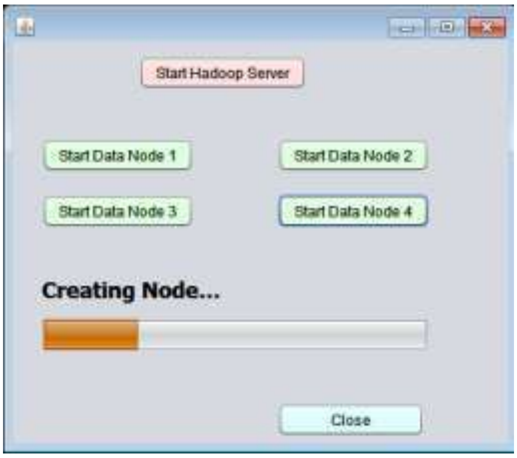


b. By clicking the button **Start Hadoop Server,** the Hadoop server initiates.



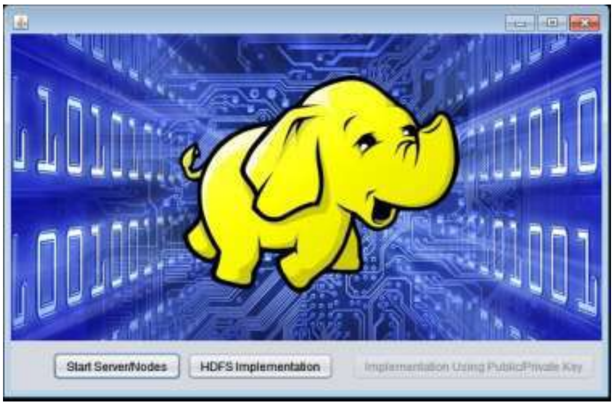
Reserving Port...

c. On clicking the **Start Data Node 1,2,3….** The data nodes gets initiated. Here, four data nodes have been created.

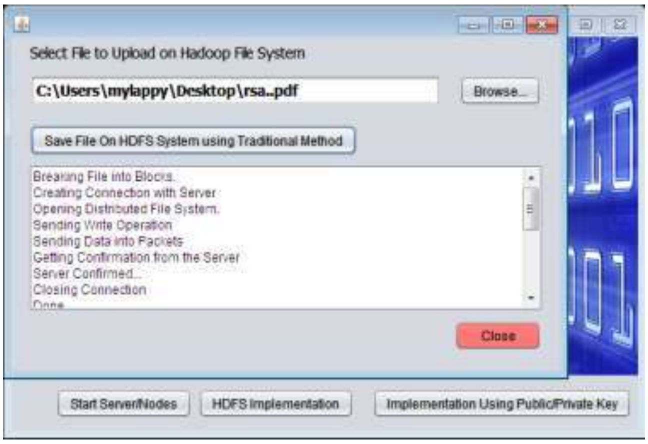


Creating Node...

d. Once the server nodes and data nodes gets initiated, the implementation of HDFS started from here.



e. After clicking on **HDFS Implementation,** it wants to browse any file or document to decrypt. And **save file on HDFS using Traditional method.**

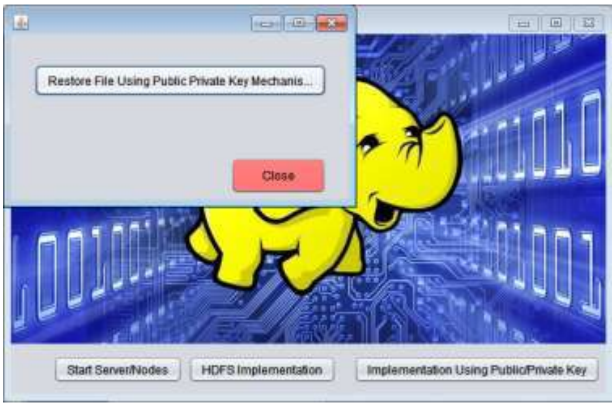

Browse...
Save File On HDFS System using Traditional Method
Close
Start Server/Nodes
HDFS Implementation
Implementation Using Public/Private Key

Select File to Upload on Hadoop File System

C:\Users\mylappy\Desktop\rsa..pdf

```
Breaking File into Blocks.
Creating Connection with Server
Opening Distributed File System.
Sending Write Operation
Sending Data into Packets
Getting Confirmation from the Server
Server Confirmed...
Closing Connection
Done
```

f. Once the browsed file has been saved. Now, restore file using public-private key mechanism. File has been saved where the user wants in its system.


Restore File Using Public Private Key Mechanis...
Close
Start Server/Nodes
HDFS Implementation
Implementation Using Public/Private Key

CONCLUSION AND FUTURE SCOPE

The authentication mechanism of Hadoop varied from nothing to Kerberos protocol. According to the features of Hadoop cluster, thesis carries on research about identity authentication based on Kerberos Protocol.

Firstly, it analyzes the security mechanism of Kerberos protocol and points out the problems such as time synchronization, KDC security, dictionary attacks and denial mechanism of symmetric key system of the original Kerberos protocol in Hadoop cluster.

Secondly, aiming at the related question, it proposes an improved strategy of Kerberos protocol based on public key encryption system with secure communication channel with the help of RSA.

Finally, it verifies the feasibility of the improved protocol by specific analysis. Of course, authentication is just one of the security problems of Hadoop cluster. Hadoop cluster faced a series of security problems including access control, security of data storage, *etc*. These problems need research and settlement in the future.

ACKNOWLEDGMENT

This research paper is supported by MIET, Meerut (UP). I would like to thank my guide Dr. Nidhi Tyagi (Associate Professor- MIET) who provided insights and expertise that greatly guided the research. I would also like to show my gratitude to all the authors of below references for sharing their pearls of wisdom through their papers.

REFERENCES

[1] ZHOU Xianwei, LI Shuai, and LIN Fuhong," Big Data security and privacy: Review",2014:135-145.

[2] Bindiya M.K & Ravi Kumar G.k, "Securing Big Data over network using MD5 algorithm technique", IJCS, Volume 123-No.15, August 2015. Kalyani Shirudkar & Dilip Motwani, "Big-Data security", IJARCSSE, Volume 5, Issue 3, March 2015.

[3] Lohr s, "The age of big data[J]". New York Times, 2012, 11.

[4] Steven M. Bellovin,Michael Merritt, " Limitations of the Kerberos Authentication System", AT&T Bell Laboratories.

[5] Tom Davenport, IIA Director of Research & faculty leader. http://www.sas.com/en_ca/news/sascom/2014q3/big-data-davenport.html

[6] McCune J C. Data, data everywhere [J]. Management Review, 1998, 87(10): 10-12.