Using the HashChain to Improve the Security of the Hadoop

Ying MEI[a]
School of Computer, Communication University of China, Beijing 10024, China
[a]178445547@qq.com

Keywords: Hadoop, Security, Hash Chain, Identity Authentication.

Abstract. For the problem of enforcing mandatory access control onthe data nodes in Hadoop, there exists some limitations by analyzing its own the token-based authentication scheme. Therefore, we use the hash chain password method to improve the original authenticationscheme. By analyzing the security, the novel method is effective againstthe impersonation and replay attacksbyincreasing the confidentiality of the token compared with the original one. At the same time, it is more efficient inthe authentication process for its simplification.

Introduction

Apache Hadoop is an open-source software framework [1,2], which is known for implementing the MapReduce model tosupport the data-intensive distributed applications. Hadoop Distributed File System (HDFS) is a distributed, extensible, and lightweight file system, which is widely adopted as the cloud storage system [3]. Therefore, a high security authentication system is needed to restrict the access control to confidential data.

TheKerberos authentication scheme came with the release of the Hadoop 1.0.0, which achieve the authentication of nodes in the cluster. It isavailableto use only when the nodes pass the authentication,which ensure the security and credibility of the cluster. However, Kerberos does not solve a package for all security issues [4]. Many researchers also have put forward the improvement schemes constantly.For instance, Zhou et al.proposed a data security access scheme based on attribute-group. In the scheme, the data ownersdo not participate in the specific operation of the property and user rights [5],and Somuet al. proposed a novel authentication model using one time pad algorithm that removes communication of passwordsbetween the serversto enhance the security in Hadoopenvironment [6], andYang et al. achieved a tripleencryption scheme for Hadoop-based cloud data to enhance the data security [7].

In this article, the hash chain password method[8]is used to improve the token-based authentication in Hadoop to respond the security authentication problem of the communication between the client in Hadoop and HDFS services, which is effective toimplement the authentication between users and HDFS.The schemehas more efficiency, simpler processes, higher security, and nicer compatibility than the original one.

Hash Chain Model and Application

The hash chain is a basic password method to prevent from being stolen. Lamport first proposed the method [8], and the hash chain of length N can be constructed from applying the one-way hash function h (.) to an initial seed value recursively (see Equations 1).

$$h^N(s) = h\left(h\left(h \ldots h(s)\right)\right) (\text{N times}) \quad (1)$$

If just the $h^N(s)$ is known and the seed values not, the adversary cannot know the $h^{N-1}(s)$. However, it is easy to verify the $h^{N-1}(s)$ through the $h^N(s)$. The characteristic of hashchain is derived from the nature of one-way hashfunction [9]. The model can be used to design cryptographic authentication schemes in insecure environments.Even if the adversary can read the system data and even eavesdrop on the communication between users and the system, it also can ensure the password security.

The security password authentication between the userand the systemcan be achieved by the hash chain perfectly. Here's as follows:

The user generatesa password sequence (e.g., N=1000), let the i th password is $x_i = h^{1000-i}(x)$.Thus, the sequence of passwords is:

$$h^{999}(x), \ldots, h\left(h\left(h(x)\right)\right), h\left(h(x)\right), h(x), x \quad (2)$$

The sequence of passwordsneeded by the system to authenticate the above passwords is:

$$h^{1000}(x), \ldots, h\left(h\left(h(x)\right)\right), h\left(h(x)\right), h(x) \quad (3)$$

According to the above definitions,$y_i = x_{i-1}$ each user password is the next password value that the system needs to authenticate. Therefore, the system only needs to initializethe value $y_1 = h^{1000}(x)$, and thensubsequentlyremember only the last password sent by the user. In this way, the system saves a secret hash value, and the adversary can't recover the original secret even if the system has been attacked. In the communication between the user and the system, the delivered password is also the secret hash, the hash value of the next password. Based on one-way hash function, theadversary cannot calculate the next password value even if this password is eavesdropped on.

HadoopClients and HDFS Authentication and Improvement

Token Scheme in Hadoop

The communication between Hadoop client and HDFS consists of two processes: (1) the RPC connection from client to NameNode; (2) the block transport from client to DataNode. The RPCconnection can pass the Kerberos or the delegation token authentication. Thedelegation token is the shared secret between the client and the NameNode,which can be used for the subsequent access authenticationwithout usingthe Kerberos key server. To obtain a delegation token, the client must use a Kerberos authentication connection.The Block transport uses block access token authentication, and each block access token is specific to a concrete block generated by the NameNode. Although you can achieve the target authentication by only usingKerberos on the NameNode RPC, the delegation token has some critical advantages [10, 11].

However, the file access permissions are set on the NameNode. The DataNode needs to obtain the information whether the client is authorized to access these blocks from the NameNode. This can achieve via a block access token. When a client visits the NameNode to access a file, the NameNode will check the file permissions, and generatea block access token for each block based on the permissions, and return the block access token as well as the position of each piece to the client. When the client visits the DataNode to access data block, the block access token is passed to the DataNodeto authenticate. Only the NameNode can issue a block access token to verify [12] by the DataNode.

The Improved Authentication Scheme

The token scheme does improve the efficiency of authentication in Hadoop, and reduce the dependency on KDC. However, fundamentally, the token is the shared secret by both sidesand only two sides know. The loss of the secret by either side will lead to the collapse of the authentication. So the secret is better not to transmit on an insecure network environment, and at least to reduce the frequency of transmission on its network. Neither the delegation token nor the data access token shouldnot transfer its shared secret directly. Thus, we put forward an improved token scheme. The schemeuses hash chain method to realize the authentication between the users, the NameNode andDataNode (the data flow of authentication is shown in Fig. 1), and ensure that the secret do not transmit on the network repeatedly, which improve the efficiency and securityof the authentication. Specifically, the authenticationprocess is as follows:

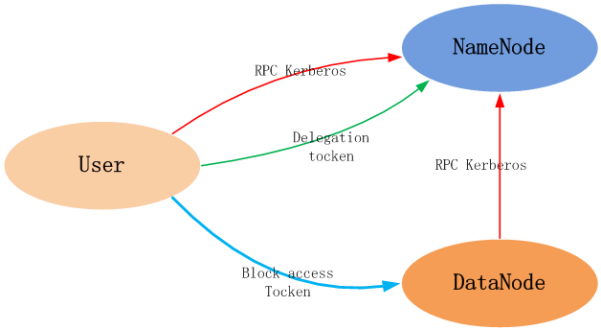

Fig.1.The data flow of authentication

(1) After the user completes the initial authentication through the Kerberos certificationand the NameNode, the delegation token is obtained that can be used to authenticate the subsequent user's work to the NameNode. The user randomly generates a secret x, assuming that the secret is used for subsequent 1,000 authentications, and then applies for a new token throughthe Kerberos certification to the NameNode. The user save the secret x, and send the secret hashvalue $y_1 = h^{1000}(x)$(see section 1, Equations3) to the NameNode as a shared secret of both for subsequent validationof the user'sdelegation token. TheNameNodekeeps the user ID and y_1 in the database. The user delegation token is generated by its own hashchain $h^{999}(x), \ldots, h\left(h(h(x))\right), h(h(x)), h(x), x$. Each time the token is authenticated to theNameNode, a value is extracted from the hash chain to send the NameNode. Because each authentication uses a different token, there is no token update and tokens transmit repeatedly on the Network.

The structure of the delegation token is as follows:

Delegation Token = {ownerID,x_i}

The user delegation token is a hash chain generated based on the secret x that shared with the NameNode, and the next value is extracted from the hash chain and passed to the NameNode when each access request.

(2) When the user applies for data access to the NameNode, the access request is generated to obtain the data access token. The request contains user ID, delegation token, and access files and operations. After the NameNode receives the user requests, the delegation tokenis first extracted andcarried outa hash, and then compared with the user correspondingvaluey that extracted fromthe database. If it matches, then it passes the authentication. Once each certification is completed, the old y value is replaced with a secret hash from the user. Then the NameNode checks the permissions of application. If satisfied, it generates the data access token, the corresponding data block, and the permissions of operationsto send to the user together.

(3) After the user has obtained the data access token, the related data operation can be carried out. The data access token is a secret shared by the NameNode and all DataNode, which is used to pass the user access rights granted by the NameNode to the DataNode. That is, each DataNodekeeps this secret. The adversary can attack any node to get the secret to generate a valid token. Therefore, the data access token is better not transferable, and only its owner can use it without worry about the steal of the token. In order to achieve this goal, we adopt the method of hash chain. When using Kerberos authentication to register on the NameNode, the NameNodecheck its validity. After passing the validation, the NameNode randomly choose a secret z and generate the hash chain sequence (e.g., the Equations 2 of section 1, assumes that the length of sequence is 1000), and the $h^{1000}(z)$issent to the DataNode, whichcan be used to validate the subsequent data access token. The structure of the data access token is as follows:

TokenID = {ownerID, blockID, accessModes}

Block Access Token = {TokenID, TokenAuthenticator}

Where the ownerID representsthe user identity of requesting data access, the blockID represents the block number needed to access, and the accessModes represents theauthorized operation

privileges that can be a combination of the following set {READ, WRITE, COPY, REPLACE}, and the TokenAuthenticator is a hash value that NameNode extract from the hash chain, which shared with the DataNode to verify the authenticity of the token. After receiving the data access token, the DataNodecarries out a hash operationfor TokenAuthenticator, and compare itwith the value extracted from the database, which is a shared secretwith the NameNode. If it matches, then it passes the authentication, anduses the TokenAuthenticator to replace the original value in the database in orderto verify the next data access token and allows users to perform the corresponding data operations.

The Security Analysis

In the improved scheme we propose, the method of the hash chain is adopted. According to the definition, each user password is that the confirmer needs to use toauthenticatethe next password value. Thus, the confirmer just needs to remember orderly the last password that the user sends. By the application of hash chain, The HDFS authentication of the user in Hadoophas the following security features:

(1) Anti-counterfeiting

This method can be used to against the tampering withor eavesdropping on the authentication password during the communication process safely. It is supposed that an adversary knows the previous 987 passwords sequence $(h^{999}(x), \dots, h^{13}(x))$, he can findthe nextpasswordh$^{12}(x)$. Then $y' = h^{13}(x)$ is given, the $h^{14}(x), \dots, h^{999}(x)$ can be calculated, thus you can calculate the $x' = h^{12}(x)$to makey$' = h(x')$.It conflicts the one-way hash function, and so the assumption fails. And thehash sequence of password local generate by the user in advance without the network interactive communication, so the intruder cannot through the tampering with or eavesdropping on the communication between the two sides for the secret and further impersonate as a legitimate user.

(2) Prevent replay attacks

In our hash chain approach, it may lose synchronization between the system and the user because of a system crash. For instance, the user sends$x_i$to the system, while the authentication code of the system still kept in y_{i-3}. That is, the user may send two data access tokens during the system crash, which happens to be record by a malicious user. When the system restores to its origin, the malicious user can take his record of token to impersonate as its corresponding user. We can solve this problem by the jump forward ways, and assumes that the system restore to its origin $y_{374} = h^{626}(x)$, and send the password to the user will not be more than two during the collapse, and the system may require the user to provide x_{376} as a password to validate. Because only the user can provide x_{376}, so it can effectively prevent the intruder from utilizingthe failure loopholes that the systemrestore to for the system collapse and impersonating as a legitimate user bythe eavesdroppedkey. It also preventsthe malicious users from implementing replay attacks effectively.

(3) Confidentiality

The hash chain method is used in the process of authentication, whether thedelegation token or data access token will not transfer twice on the network. According to the one-way hash function, the adversary cannot infer the subsequent password from the previous passwords. Thebasic process of authentication is similar to the one time pad methods, which greatly improve the security of the password. The method simplifiesthe authentication process, appliesa password toeach task individually, without the needfor updating token regularly, effectively reducesthe traffics of communication, and improvesefficiency of the authentication.

Summary

The communication between Hadoop client and HDFS needs a trusted client to authenticate, in order to ensure that the legitimate users perform related operations andprevent the malicious users to impersonate as other users to invade the HDFS cluster. At the same time, the DataNode also authenticate the access rights of the users to realize the access control of data in HDFS. We propose

a solution to use the hash chain password method to improve the delegation token and the data access token scheme, whicheffectively improve the efficiency and security of the authentication process. Using public key scheme to the identify authentication is also a possible method. Althoughit is expensive to calculate, it is still a direction worth studying.

References

[1] Apache Hadoop! Available: http://hadoop. apache. org/

[2] Dean J, Ghemawat S. MapReduce: simplified data processing on large clusters[J]. Communications of the ACM, 2008, 51(1): 107-113.

[3] Jin S, Yang S, Zhu X, et al. Design of a trusted file system based on hadoop[C]//International Conference on Trustworthy Computing and Services. Springer Berlin Heidelberg, 2012: 673-680.

[4] Jam M R, Khanli L M, Javan M S, et al. A survey on security of Hadoop[C]//Computer and Knowledge Engineering (ICCKE), 2014 4th International eConference on. IEEE, 2014: 716-721.

[5] Zhou H, Wen Q. Data security accessing for hdfs based on attribute-group in cloud computing[C]//International Conference on Logistics Engineering, Management and Computer Science (LEMCS 2014). Atlantis Press, 2014.

[6] Somu N, Gangaa A, Sriram V S S. Authentication service in hadoop using one time pad[J]. Indian Journal of Science and Technology, 2014, 7(4): 56-62.

[7] Yang C, Lin W, Liu M. A novel triple encryption scheme for hadoop-based cloud data security[C]//Emerging Intelligent Data and Web Technologies (EIDWT), 2013 Fourth International Conference on. IEEE, 2013: 437-442.

[8] Lamport L. Password authentication with insecure communication[J]. Communications of the ACM, 1981, 24(11): 770-772.

[9] Rogaway P, Shrimpton T. Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance[C]//International Workshop on Fast Software Encryption. Springer Berlin Heidelberg, 2004: 371-388.

[10] Sharma P P, Navdeti C P. Securing big data hadoop: a review of security issues, threats and solution[J]. Int. J. Comput. Sci. Inf. Technol, 2014, 5.

[11] O'Malley O, Zhang K, Radia S, et al. Hadoop security design[J]. Yahoo, Inc., Tech. Rep, 2009.

[12] Jeong Y S, Kim Y T. A token-based authentication security scheme for Hadoop distributed file system using elliptic curve cryptography[J]. Journal of Computer Virology and Hacking Techniques, 2015, 11(3): 137-142.