# Big Data: Technical Review of Apache Hadoop Security

Muhammad Bukhari Bin Burhanuddin
School of Computer Sciences
Universiti Sains Malaysia
USM, 11800 Georgetown, Penang, Malaysia
P-COM0071/19

***Abstract*—Due to the exponential growth of big data, there is an increasing number of people and organizations that are adapting Hadoop. Hadoop is an open source distributed system for data storage and parallel computations but it is not without fault as it has plenty of security issues. In this paper, Hadoop security vulnerabilities are discussed as well as its consequences for lack of security. Available security solutions provided from Kerberos, Apache Knox, Apache Ranger, Apache Sentry and Project Rhino are then elaborated from the point of view of four imperative security factors which are authentication, authorization, audit and data protection.**

***Index Terms*—hadoop, big data, security, solution**

## I. Introduction

### A. Apache Hadoop

A huge amount of structured and unstructured data is produced on a daily basis in today's digitized world. This occurrence of ever-increasing rates of a diverse set of information is called big data. Big data offers new opportunities of knowledge especially in providing businesses a new form of value in capturing their customers' needs and wants but the data is deemed too large to be stored, processed and analyzed using traditional data-processing application software. In order to tackle this issue, the Apache Software Foundation develops Apache Hadoop which uses simple programming models allowing distributed processing of large data sets across clusters of computers.

Hadoop is a framework that utilizes commodity hardwares to build a powerful system that is able to process petabytes of data rapidly and efficiently [1]. It is based on the concept of distributed computing making it easy for horizontal and vertical scaling of machines with each of them offering local computation and storage [1]. The core components of Hadoop ecosystem are Hadoop Distributed File System (HDFS), MapReduce and Yet Another Resource Negotiator (YARN) [2].

HDFS is written in Java and it sits on top of the native file system for an operating system. The role of HDFS in the Hadoop framework is to distribute and provide redundant storage for large amounts of data using low-cost hardwares [3]. The files are stored locally as blocks where the sizes are much larger than what is implemented in other file systems (64MB block size by default) and they are replicated thrice in DataNodes for redundancy [3]. The DataNodes are managed by NameNode where it stores metadata providing addresses of data blocks allowing all clients applications read/write data to the DataNodes [3].

The large data sets that are stored in a parallel and distributed way among the DataNodes are processed using MapReduce [3]. MapReduce consists of two stages which are Map and Reduce. Map tasks are responsible to transform the input into intermediate data and then pass the key-value pairs to the Reducer for sorting as well as merging the data to write into an output file [3]. The MapReduce framework employed two Hadoop daemons (or services) which are JobTracker and TaskTracker. JobTracker is responsible to administer all jobs submitted for a Hadoop cluster and it runs on the master node [3]. TaskTrackers run on the slave nodes and they are responsible to execute tasks given by the JobTracker [3].

YARN is described as a management framework for Hadoop resources by implementing three important daemons which are ResourceManager (RM), ApplicationMaster (AM) and NodeManager [2]. YARN strives to make sure data processing in Hadoop runs smoothly.

### B. Hadoop Security

The security measures of the Hadoop framework are not the focus in the early development as the attention is more towards making the technology work and dealing with the complexities ingrained in distributed systems such as fault tolerance. Since Hadoop is iterated from the Google File System, it was meant to process a massive amount of web data in the public domain [4] [5]. As such, the security stance that was settled during that time is the entire clusters of machines and user access are part of a network that is trustworthy and secured [3].

Due to the original Hadoop design, authentication method or security gateway is not implemented. Although the later build of Hadoop default installation has a basic authentication level enforced such as using username and password, it is deemed as insufficient especially for HDFS as it is quite easy to impersonate another user and a large number of DataNodes being potential entry points for attacks and exploitation [3] [6].

In addition, Hadoop does not provide granularity in defining user roles across its components making all users having the same access level with the exception of only one superuser in the admin role [3]. There is also no consistent way of

encrypting data both in motion and at rest. Security vulnerability of data in motion describes unencrypted data by default being transmitted through the network during internode communication. While for data at rest, Hadoop does not encrypt data that is stored on disk by default which makes the data that is stored easily exposed to hackers. Other security vulnerabilities and threats in Hadoop are described below [3] [7].

- Malicious user and rogue process could intercept internode communications due to the channels being unencrypted and unsecured by default.
- Cluster might be accessed by unauthorized clients impersonating as an authorized user.
- Absence in tracking of user activities and processes for risk assessment.

Consequence for a lack of security or not implementing security measures in Hadoop architecture is dire as many companies such as Facebook, Amazon and Netflix begun adopting the technology to handle their huge amount of data everyday particularly users' personal information. Since the law of data privacy is enacted in almost all countries, companies could be sued for negligence for data leakage of user's personal information and this potentially will damage the reputation of a company especially if it is technological one. For example, Netflix was sued over leakage and mishandling of user's personal information [8].

Companies also may experience revenue loss when the operations are forced to shut down to solve the security issues. When classified documents such as the blueprint of a new technology are stolen or leaked, an entity may lose the competitive edge over their competitors and hamper the business from expanding. In the user's perspective, data breach is very damaging to them as they could become the victim of identity theft and their details of monetary transactions using credit cards getting exploited. Hence, the security of an architecture especially for one that handles a massive amount of data is very important and the cost of implementing security measures in Hadoop far outweighs the option of not implementing one.

### C. Important Security Aspects

Data needs to be hidden behind multiple layers of defenses for a secure distributed system and Apache Hadoop is no exception. Aspects such as authentication, authorization, auditing and data protection ought to be considered and designed as solutions to the security vulnerabilities described in the previous subsection. These factors are defined below with respect to the Hadoop framework.

- Authentication: Function or process that verify whether a person is an authentic user of Hadoop or not. This mechanism allows a Hadoop cluster to be secured when a client connects to it.
- Authorization: Function that specify the resources that a user could access in Hadoop after being authenticated.
- Auditing: The process where the activities in the Hadoop ecosystem of any authenticated and authorized user are recorded and reported.
- Data Protection: The uses of mechanism such as encryption to protect sensitive data in Hadoop from being accessed by unauthorized applications and entities.

## II. Available Security Solutions

Subsections below elaborate the tools and/or solutions that are available to manage security in Apache Hadoop. The available solutions that were reviewed in this paper are Kerberos, Apache Knox, Apache Ranger, Apache Sentry and Project Rhino. The important security aspects described previously are addressed and implemented by these security solutions. These security solutions are chosen to be reviewed in this paper because their modules are widely adapted in today's Hadoop framework and they are also made easily available from popular Hadoop distribution vendors.

### A. Kerberos

In response to Hadoop authentication vulnerability [9], a central authentication method to Hadoop services is introduced through Kerberos. Kerberos (Cerberus) was a giant, three-headed hound that guards the Underworld's gate based on the Greek Mythology. In cyber security, Kerberos is a network authentication protocol with three important entities which are client, server and Key Distribution Center (third-party authentication service) developed for Project Athena by Massachusetts Institute of Technology (MIT) [10]. KDC has three components and their roles are explained below.

- Database: Stores credentials of users and service/server known as principals.
- Authentication Server (AS): Authenticate the user by issuing a Ticket Granting Ticket (TGT) and only valid for a specific time.
- Ticket Granting Server (TGS): Application server of KDC and provides a service ticket to a client in order to access any service on a Hadoop cluster.

All the current major Hadoop distributions come with Kerberos installed and it is commonly used to safeguard Hadoop clusters by providing secure user authentication. Fig. 1 is referred to assist in understanding the methods and features of Kerberos. Assuming that a user wanted to get information of a file from HDFS on a Kerberos Hadoop cluster, the steps are as below.

1) The user enters login ID and password. A request is sent to the AS by the client on the user's behalf.
2) AS communicate with the database to decrypt the request using the user's ID and password. AS responds back to the client with TGT on a successful authentication.
3) The user executes a command for example *bin/hdfs dfs -stat /filename* and the Hadoop client sends the encrypted TGT to TGS requesting a service ticket for the NameNode service.

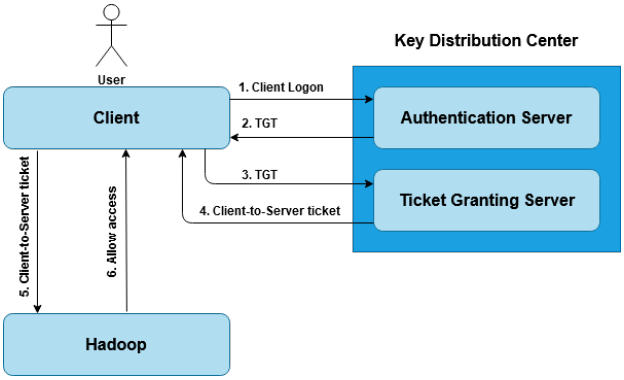


Fig. 1: Basic architecture of Kerberos

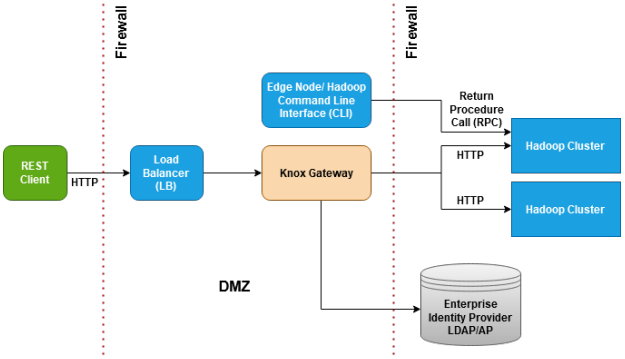


Fig. 2: Apache Knox architecture

4) TGS grants the client with a service ticket and it will be used to reach out to the NameNode.
5) The client sends a request to the NameNode with the encrypted service ticket for authentication.
6) Then the NameNode allows access to its service based on the executed command for a certain period of time specified in the service ticket.

The last step of Kerberos authentication is of course dependent on the authorization level of the user. For example, if the user does not have the permission to list the files in the root directory from the NameNode then the request is denied. Kerberos is highly effective in protecting a network and system from unauthenticated user although it is also easily exploited through password guessing attack [11] [12].

### *B. Apache Knox*

Apache Knox is introduced by Hortonworks and it is a parameter security gateway interacting with representational state transfer application programming interfaces (REST APIs). The gateway is based on the concept of stateless reverse proxy framework and set to run on a secure socket layer (SSL). Knox also acts as a single access point for authentication to a Hadoop cluster by providing enterprise integration such as single sign-on (SSO), Active Directory (AD), Lightweight Directory Access Protocol (LDAP) and other authentication systems [13]. Other features provided by Knox are described as below [14] [15].

- Network topology hiding allowing a layer of abstraction where Hadoop services can be accessed via standardized Knox gateway URL. For example:
  - http://oozie-host:11000/oozie
  - http://hbase-host:60080
  - https://knox-host:8443/gateway/default/oozie
  - https://knox-host:8443/gateway/default/hbase
- Kerberos encapsulation enabling it to take advantage of features provided by Knox such as integration of enterprise identity management solutions. Thus, eliminating further identity configuration on the client side and simplifies the authentication method.
- Service level auditing capturing events across the Knox gateway.

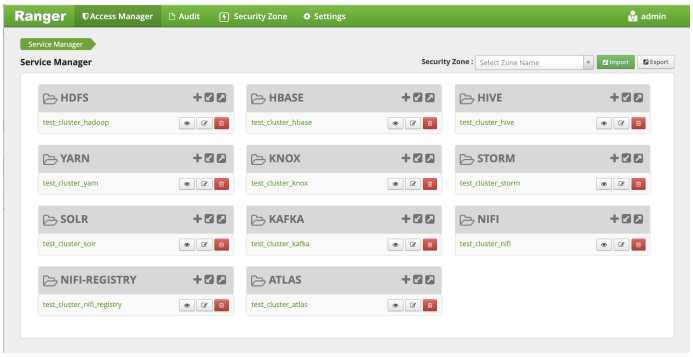


Fig. 3: Apache Ranger web application GUI

The basic architecture of Knox is provided in Fig. 2. Knox is often integrated together with Apache Ranger for authorization which will be explained later.

### *C. Apache Ranger*

Apache Ranger is a security solution introduced by Hortonworks with the focus of centralized security framework to manage fine grained authorization mechanism and auditing over Hadoop cluster resources. A user request is assumed to be already authenticated when it comes to Ranger for authorization to the Hadoop cluster resources.

On the front-end, Ranger provides a centralized security approach with its web application as shown in Fig. 3 which can be maintained easily in managing access of users/groups to Hadoop cluster resources and those accesses are tracked through an integrated audit location. Ranger also provides encryption of data at rest for HDFS through its Ranger Key Management System (KMS) [16]. The KMS is an extension of Hadoop KMS developed by the Apache community providing scalable cryptographic key management service [17].

Referring to its architecture in Fig. 4, on the back end Ranger's main components can be dissected into two which are elaborated below.

- Ranger Administration Portal: The portal is the central location or interface for security administration where it mainly manages Ranger Audit Server and Ranger Policy Server. Ranger Policy Server is in charge of maintaining

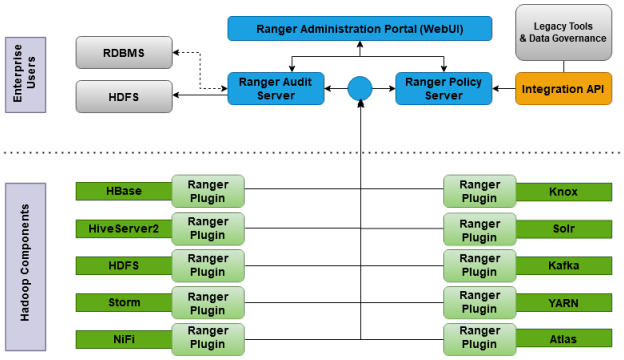


Fig. 4: Apache Ranger architecture

access policies to Hadoop cluster resources and then storing those policies in the database. Ranger Audit Server collects and sends audit data collected from the plugins to be stored in the database or HDFS.

- Ranger Plugin: Each Hadoop component is embedded with a lightweight Java program called Ranger Plugin. Other than collecting and sending audit data to the Ranger Audit Server, the plugins also evaluate an access request to the component by pulling the security policies from Ranger Policy Server.

Let's assume a situation where a user from a Finance Department is using Hive and requesting access to make changes to the Sales table. Hive will use its embedded Ranger Plugin to validate the user's access request by retrieving his/her authorization level from the Ranger Policy Server. Once valid, then the user can make the changes on the table and these events will be logged by the Ranger Audit Server.

### *D. Apache Sentry*

Apache Sentry is offered by Cloudera and in terms of functionality, it is the same as Apache Ranger where both of them offer granular, role-based authorization modules and security policies across the Hadoop clusters [18]. The architecture provided in Fig. 5 which is adapted from its project page shows the similarities between Sentry and Ranger [18].

However, it is noticeable that Sentry is integrated with fewer Hadoop components where it is missing HBase, Knox, YARN, Storm, NiFi, Atlas and Kafka compared to Ranger. Sentry also provides its own auditing although its functionality is not as expanded as Ranger where it is only limited to logging the changes of its role-based authorization metadata in terms of create role, drop role, add role to group, delete role from group, grant privilege and revoke privilege [19] [20]. Unlike Apache Ranger that supports user friendly web GUI, Apache Sentry is command line interface (CLI) based.

### *E. Project Rhino*

Project Rhino is an open source contribution pioneered by Intel with an effort to address the security concerns of Hadoop. Not much is known currently about the initiative as it was introduced in 2013 and the project is still in development [21].

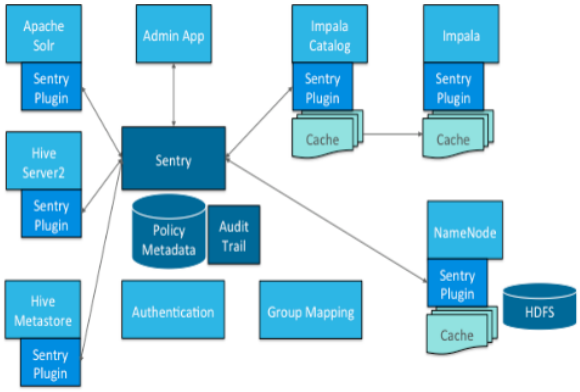


Fig. 5: Apache Sentry architecture

The resources regarding the project are also scarce and mostly the information could be gathered from the documentation in its GitHub page [22]. However, a researcher opine that Rhino will potentially become a standard part of other Hadoop distributions given its excellent partnership with other vendors [23]. The key features of Rhino adapted from [22] are as below.

- Authentication: Reinforce the current authentication mechanism such as strengthening Kerberos token-based authentication framework and SSO.
- Authorization: Extend and centralized the authorization mechanism. HBase 0.98 has been successfully developed and made available. HBase from Project Rhino enhances the security by introducing a more granular cell-level authentication and providing transparent encryption for HBase tables stored in Hadoop.
- Audit: Standardize audit logging framework and log formats for easy audit trail analysis.
- Data Protection: Protect data in motion and at rest by crypto codec framework and crypto coded implementation which provide block-level encryption.

## III. Comparative Analysis

It is important to note that Project Rhino is included in this comparative discussion between available security solutions although its modules are still in development. This is because the initiative has the support of multiple Hadoop distribution vendors such as Cloudera and IBM and it is deemed appropriate to highlight its potential in bringing more values to the landscape of Hadoop security. The functionalities of the available security solutions are summarized and compared in TABLE I in respect to the important security factors. TABLE II compares and summarizes the advantages and limitations of the Hadoop security solutions.

### *A. Authorization*

In Hadoop, Kerberos is the de facto of authentication where it is able to integrate with other security solutions such as Apache Ranger and Apache Sentry. Kerberos is able to provide

TABLE I: FUNCTIONALITY COMPARISON BETWEEN DIFFERENT SECURITY SOLUTIONS

| | **Kerberos** | **Apache Knox** | **Apache Ranger** | **Apache Sentry** | **Project Rhino** |
|---|---|---|---|---|---|
| **Authentication** | Secret-key cryptography, encrypted delegation tokens. | Single access point, integration with enterprise solutions. | Integration with other authentication methods. | Integration with Kerberos. | Token based & SSO. |
| **Authorization** | Not supported. | Service level authorization. | Centralized & fine-grained robust authorization. | Centralized & fine-grained role-based authorization. | Per cell authorization framework. |
| **Audit** | Not supported. | Log4j framework. | Centralized auditing. | Centralized auditing. | Unified and standardized logging framework. |
| **Data Protection** | Not supported. | Not supported. | Data at rest encryption. | Not supported. | Encryption support across the whole Hadoop ecosystem. |

TABLE II: ADVANTAGES & LIMITATIONS BETWEEN DIFFERENT SECURITY SOLUTIONS

| | Kerberos | Apache Knox | Apache Ranger | Apache Sentry | Project Rhino |
|---|---|---|---|---|---|
| Advantage | Fast authentication, open and accepted standard. | Security enhancement on top of other authentication methods such as Kerberos. | Centralized administration of authorization and audit, user friendly web GUI, authorization methods variety, expandable auditing, data at rest encryption. | Centralized administration of authorization and audit. | The goal of robust Hadoop security framework. |
| Limitation | Only provide authentication function, single point of failure, prone to password attack. | Fewer functions compared to Ranger and Sentry, does not provide any data protection mechanism. | Does not provide data in motion encryption. | Fewer functions compared to Ranger, does not provide any data protection mechanism. | Still in development phase. |

fast authentication with its ticket/token based system via a dedicated point of contact which is KDC although this could also potentially overload the server which will eventually makes it a single point of failure. It is also prone to password attacks such as password dictionary attack and brute-force attack which is why the authors in [24] opine that it should be integrated with other tools or techniques to strengthen its security.

Apache Knox is able to encapsulate "Kerberized" Hadoop clusters providing an extra layer of gateway security and provides SSL protocol on services that do not have it. Both Knox and Ranger are able to integrate with enterprise's identity management authorization such as LDAP and AD making it easy for clients to adapt to the security solutions. Project Rhino is aiming to enhance the authentication method provided by Kerberos using token based authentication framework and SSO [22].

### *B. Authentication*

Kerberos is a third party authentication centric security solution and as such, it does not address the other security aspects which make a system more secure. Ranger and Sentry are both front runners in providing comprehensive authorization methods in Hadoop. They offer centralized and fine-grained authorization where users could define control on the access to the Hadoop components and resources through a central administration tool. However, the authorization mechanism of Ranger is more robust than Sentry where the latter could only do role-based access control (RBAC) while Ranger supports more authorization methods such as attribute-based access control (ABAC) which is much more granular and dynamic.

For example, RBAC method could grant access to a resource for managers while ABAC could specify it by configuring that only managers in the finance department could access it. Meanwhile, Knox provides service level authorization which is also the same mechanism packaged with Hadoop framework and it is disabled by default [25]. It is also not centralized and tedious to configure unlike Ranger and Sentry. Project Rhino aspired to implement a centralized, fine-grained authorization mechanism to all Hadoop components and extend it further to add per-cell ("per-KeyValue") security [22].

### *C. Auditing*

Auditing in cyber security is important for risk assessment and keeping track of activities that happened in a system. Ranger provides a more powerful auditing method compared to Knox and Sentry. Knox utilizes log4j Java framework which is fitting considering Hadoop is written in Java. Its logging is more centred on the events happening in the Knox gateway. Comparatively, Ranger and Sentry are more centralized in their auditing with the implementations of plugins embedded to the Hadoop components and providing more detail in the authorization to their resources.

Ranger's auditing package is more expanded than Sentry where it manages auditing with its own dedicated audit server and tracked any activity that its policies have define for the system such as login sessions to each of the Hadoop component and the status of the Ranger Plugins [19]. While Sentry is limited to only logging the changes of it role-based authorization metadata [20]. Project Rhino criticizes that the current audit methods do not provide unified and consistent logging format. Thus, improving audit logging to make it centralized and standardized is one of its goals.

### D. Data Protection

For data protection, Ranger is the only security solution that address this factor with its Ranger KMS data at rest encryption for HDFS [17]. However, it does not provides encryption for data in motion. Project Rhino is seeking to enhance not only in HDFS but other upper layer applications such as MapReduce, Hive and Pig [22].

## IV. Discussion

Authentication, authorization, audit and data protection are important factors to be considered in protecting the Hadoop environment. In terms of completeness of covering all the important security aspects, Project Rhino seems to be a wise choice as it gives solutions to the authentication, authorizations, audit and data protection problems that Hadoop is facing. As mentioned before however, the initiative is still under development. Although certain milestones have been achieved it is nowhere close to its project goals of delivering robust Hadoop security yet.

As such, the next best alternative is Apache Ranger as it is the most comprehensive security framework providing centralized security administration via its user friendly web GUI. But it is not a standalone security solution as in the aspect of authentication, it does not offer its own mechanism and needs to be integrated and configured with other technology such as Kerberos. It also provides fine-grained authorization and auditing methods.

Conversely, having too many triggers on authorization and auditing may increase system overhead and degrade the performance of Hadoop services which is why the functions are disabled by default and users have to configure themselves [20] [26]. While Ranger is lacking in providing data in motion encryption, Hadoop at its core provides HDFS data encryption for both data in motion and data at rest [27]. Ranger KMS data at rest encryption extends and enhances the data protection security further but it is worrying that there is no mention of data protection in other Hadoop components.

It is great that various open source communities and Hadoop vendors are working together to improve Hadoop security but it is also easy to become confused and fatigued with a lot of different security solutions especially for new users. The security solutions are not designed and developed as a cohesive predefined module, but were rather developed to offer a unique value proposition marketed to the customers from the vendors such as IBM, Cloudera and MapR. More emphasis on data protection of the entire Hadoop ecosystem also should be recognized and not just on HDFS. As such, Hadoop is in need of a single security solution that acts as an umbrella offering users complete authentication, authorization, audit and data protection modules. Initiative such as Project Rhino by Intel is a good start in leading the effort of complete Hadoop security package.

## V. Conclusion

Hadoop security is an afterthought and it is not the focus during its early development. Nevertheless, significant effort has been made to address its security vulnerabilities with the offering of Kerberos, Apache Knox, Apache Ranger, Apache Sentry and Project Rhino. They are among distinguished security solutions for Hadoop built to accommodate to four important security aspects highlighted in this technical review paper which are authentication, authorization, audit and data protection.


## Acknowledgment

I would like to express my gratitude and appreciation to Mr. G.C. Sodhy for his teaching and guidance.


## References


[1] T. White, *Hadoop: The definitive guide*. O'Reilly Media, Inc., 2012.
[2] D. Miner, "Hadoop: What you need to know," 2016.
[3] B. Lakhe, *Practical Hadoop Security*. Apress, 2014.
[4] M. Cafarella, B. Lorica, and D. Cutting, "The next 10 years of Apache Hadoop," 2016.
[5] S. Ghemawat, H. Gobioff, and S.-T. Leung, "The Google file system," in *Proceedings of the nineteenth ACM symposium on Operating systems principles*, pp. 29–43, 2003.
[6] R. R. Parmar, S. Roy, D. Bhattacharyya, S. K. Bandyopadhyay, and T.-H. Kim, "Large-scale encryption in the hadoop environment: Challenges and solutions," *IEEE Access*, vol. 5, pp. 7156–7163, 2017.
[7] D. Das, O. O'Malley, S. Radia, and K. Zhang, "Adding security to apache hadoop," *Hortonworks, IBM*, pp. 26–36, 2011.
[8] R. Singel, "Netflix spilled your brokeback mountain secret, lawsuit claims," Jan 2018.
[9] O. O'Malley, K. Zhang, S. Radia, R. Marti, and C. Harrell, "Hadoop security design," *Yahoo, Inc., Tech. Rep*, 2009.
[10] L. Zhu and B. Tung, "Public key cryptography for initial authentication in kerberos (pkinit)," tech. rep., RFC 4556, June, 2006.
[11] M. Walla, "Kerberos explained," *Windows 2000 Advantage*, 2000.
[12] M. A. Kâafar, L. Benazzouz, F. Kamoun, and D. Males, "A kerberos-based authentication architecture for wireless lans," in *International Conference on Research in Networking*, pp. 1344–1353, Springer, 2004.
[13] M. Priyadharshini, R. Baskaran, M. K. Srinivasan, and P. Rodrigues, "A framework for securing web services by formulating an collaborative security standard among prevailing ws-* security standards," in *International Conference on Advances in Computing and Communications*, pp. 269–283, Springer, 2011.
[14] T. Apache Software Foundation, "Apache Knox," 2019.
[15] M. R. Jam, L. M. Khanli, M. S. Javan, and M. K. Akbari, "A survey on security of hadoop," in *2014 4th International Conference on Computer and knowledge Engineering (ICCKE)*, pp. 716–721, IEEE, 2014.
[16] Cloudera, "Apache Ranger," 2020.
[17] M. Amirneni, "Migrate from Hadoop KMS to Ranger KMS," Oct. 2016.
[18] Y. Anne and S. Dapeng, "Sentry Tutorial - Apache Sentry - Apache Software Foundation," Mar. 2016.
[19] I. Hortonworks, "Managing Auditing in Ranger: Access," 2019.
[20] M. Colin, "Sentry Audit Log - Apache Sentry - Apache Software Foundation," Mar. 2016.
[21] J. Girish and D. Avik, "Securing Big Data for the Enterprise: Project Rhino and the Intel Distribution for Apache Hadoop* (IDH)," Sept. 2013.
[22] C. Haifeng, "Intel Hadoop Project Rhino," July 2015.
[23] H. Guy, "Does the World Really Need Another Hadoop Distribution?," Dec. 2013.
[24] B. C. Neuman and T. Ts'o, "Kerberos: An authentication service for computer networks," *IEEE Communications magazine*, vol. 32, no. 9, pp. 33–38, 1994.
[25] G. Barot, C. Mehta, and A. Patel, *Hadoop Backup and Recovery Solutions*. Packt Publishing Ltd, 2015.
[26] W. Panek, *MCSA Windows Server 2016 Complete Study Guide: Exam 70-740, Exam 70-741, Exam 70-742, and Exam 70-743*. John Wiley & Sons, 2018.
[27] T. Apache Software Foundation, "Transparent Encryption in HDFS," 2019.