The Hadoop Security in Big Data: A Technological Viewpoint and Analysis

Yusuf Perwej

International Journal of Scientific Research in Computer Science and Engineering
Vol.7, Issue.3, pp.1-14, June (2019)
DOI: https://doi.org/10.26438/ijsrcse/v7i3.114

Research Paper

E-ISSN: 2320-7639

The Hadoop Security in Big Data: A Technological Viewpoint and Analysis

Yusuf Perwej

Department of Information Technology, Al Baha University, Al Baha, Kingdom of Saudi Arabia (KSA)

*Corresponding Author: yusufperwej@gmail.com

Available online at: www.isroset.org

Received: 12/May/2019, Accepted: 12/Jun/2019, Online: 30/Jun/2019

***Abstract*—** At present, the world is going to become more digital. As every person using the internet today, a huge amount of data gets generated day-to-day. The data are very essential with respect to carrying out their day-to-day activities, and also helping the business management to achieve their aims and make the best decisions on the basis of the information extracted from them. The big data phenomenon is a direct consequence of the digitization and 'datafication' of nearly every activity in public, private, and commercial life. Despite that, big data originated new matter related not only to the volume or the diversity of the data, but also to data security. There is need to endow security of such data. The Apache Hadoop platform is used to handle, store, manage, and distribute big data across many server nodes. Here are different tools, which research on the top of Apache Hadoop stack to provide security for data. In order to obtain a full perspective of the problem, we decided to carry out examine with the objective of existing security methods for Apache Hadoop security in big data.

***Keywords*—** Big Data; Hadoop; Kerberos Protocol; Hadoop Metrics; Amazon Web Services; Security Protocols; Hadoop Security.

I. INTRODUCTION

The term Big data refers to a framework that permits the analysis and management of a huge amount of data than the traditional data processing technologies [1]. Information from data and the ability to combine data from various sources and different formats. Big data has also altered the way in which organizations store data, and have allowed them to develop a more via and in-depth understanding of their business, which implies a great benefit. [2] This tendency towards increasing the volume and detail of the data that is collected by companies will not transform in the near future, as the rise of social networks, multimedia, and the Internet of Things (IoT) [3] is producing an overwhelming flow of data. Besides, this data is mostly unstructured, signifying that conventional systems are not capable of analyzing it. The organizations are willing to extract more advantageous information from this high volume and diversify of data. In this context, that the Internet of Things (IoT), the growing network of day-to-day objects equipped with [4] sensors that can record, send, and receive data over the Internet without human intervention. Gartner Inc. estimates that the IoT currently includes 4.95 billion connected "things" a 35 percent increase from 2014 analysts predict that the number will hit 30 billion by 2020 [5] of course, all these devices will generate data.

The Hadoop is a framework developed by Apache that permits the distributed processing of hue data sets across clusters of computers using programming models [6]. It is designed to be scalable from a single server to thousands of them, each of which offers computation and local storage. It is at the center of a growing ecosystem of big data technologies that are firstly used to support advanced analytics initiatives, [7] contain predictive analytics, machine learning applications and data mining. The input for the Hadoop framework is the data that feed the big data system. Hadoop has its own distributed file system (HDFS) which stores the data in various servers with various functions, like as NameNode, which is used to store the metadata, or the DataNodes, which store the application data. The principal specialty of Hadoop is, however, that of being an open-source implementation of MapReduce [8]. MapReduce is a [9] programming model that is particularly focused on processing and generating huge data sets. As big data grow via streaming cloud technology [10], conventional security mechanisms tailored to securing small scale, stable data on firewalls and semi isolated networks are insufficient. There are some challenges for managing a huge data set in a secure manner [11] and incompetency tools, public and private database contains more threats and penetrability, volunteered and unexpected leakage of data, and insufficiency of public and private policy makes a hacker to collect their resources whenever needed. They protect the data in the presence of unbelievers, people is more arduous and when moving from the same kind data with the miscellaneous data certain tools [12] and technologies for massive data set are not often

developed with more security and policy certificates [13]. Each new unruly technology brings new issues with it. In the case of big data, these issues belong to not only the volume or the variety of data, but also to data quality, and data security [14]. This paper will focus on the subjects of Hadoop security in big data.

II. THE BASIC CONCEPTS OF SECURITY

At present technology driven world, computers have penetrated all walks of day-to-day life, and more of our personal and corporate data is available electronically than ever. Unhappily, the same technology that provides so many advantages can also be used for damaging [15] purposes. The information security has become an ongoing concern in all areas of an Information system. The Security is neither a product, nor a software, it is a discipline that needs to be taken into consideration in any organizational decision. Nowadays, independent hackers, who earlier worked mostly for own benefit, have organized into groups working for financial benefit, making the threat of corporate or personal data being thieve for illegitimate purposes much more serious and real. Malware spread our computers and redirects our browsers to specific advertising, web sites depending on our browsing context. The phishing emails entice us to log into websites that appear real, but are designed to steal our passwords. Viruses or direct attacks breach our networks to steal passwords and data. It is indeed true that there is no such thing as a perfectly secure system. But it is also correct that by increasing the security measures that protect your assets, you are making [16] your system a much more arduous target for intruders, which, in turn, decrease the chances of becoming a victim when the right security technologies are in place. Supposing we want to antagonize these attacks on our personal property or our corporate property, you have to understand utterly the threats as well as your own vulnerabilities. We are working toward devising a strategy to secure our data, be it personal or corporate. The security strategy establishes guidelines and procedures for securing against malicious threats and often involve encryptions for passwords and sentient data, policies for configuring antivirus software and configuring firewalls etc.

III. THE SKELETON OF SECURITY ENGINEERING

The security engineering is about designing and implementing systems that do not disclose private information and can reliably withstand malicious attacks, errors, or mishaps [17]. It mainly attention on the tools, processes, and methods needed to design and implement complete systems and adapt existing systems. The security engineering need expertise that spans such different disciplines as cryptography, computer security, computer networking, economics, applied psychology, and law. The security needs vary from one system to another [18]. Normally, we need a balanced combination of user authentication, integral transactions, fault tolerance, encryption, authorization, policy definition, auditing, and isolation. A many systems fail because their designer's attention on the erroneous things. The securing a system thus depends on many types of processes [19]. We need to determine your security necessity and then how to implement them. Also, we have to remember that safe systems have a very vital component in addition to their technical components. In figure 1 shown the security engineering depend on the following five factors to be considered while conceptualizing a system.

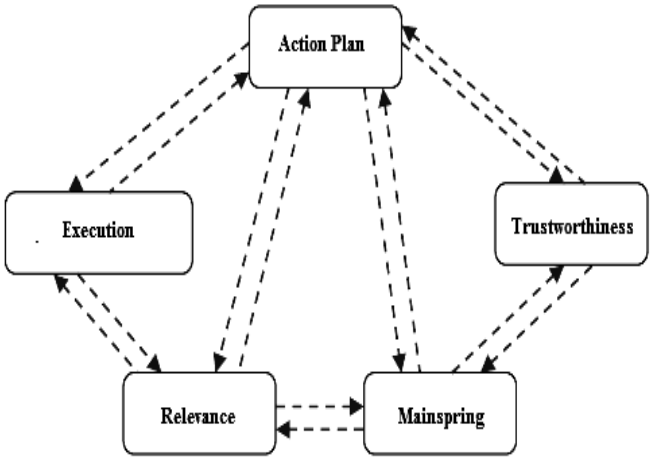

Figure 1. The Security Engineering Depend on the Following Five Factors

A. Action Plan

Our action plan revolves around our aim. A specific aim is a good starting point to define authentication, authorization, integral transactions, fault tolerance, encryption, and isolation of our system. We also need to consider and account for possible error conditions or malicious attack scenarios.

B. Execution

Execution of our plan involves procuring the necessary hardware and software components, designing and developing a system that fulfill all, our aim, defining access controls, and thoroughly testing our system to match our plan.

C. Trustworthiness

Trustworthiness is the amount of faith we have for each of our system components and our system as a whole. Trustworthiness is measured against lack of success as well as malfunction.

D. Relevance

Relevance decides the ability of a system to antagonize the latest threats. For it to remain reasonable, especially for a security system, it is also extremely important to update it periodically to maintain its ability to antagonize new threats as they arise.

E. Mainspring

Mainspring relates to the drive or dedication that the people accountable for handling and maintaining our system have for doing their job appropriately, and also refers to the temptation for the attackers to try to defeat our plan.

IV. THE SECURITY PROTOCOLS

A security system be made up of components such as users, companies, and servers, which communicate using a number of channels including phones, satellite links, and networks, while also using physical devices such as laptops, portable USB drives [20]. A security protocol is basically a communication protocol an agreed sequence of actions performed by two or more communicating entities in order to accomplish some mutually desirable goal that makes use of cryptographic techniques, allowing the communicating entities to attain a security objective. The security protocols are the rules governing these communications and are designed to efficaciously antagonize malicious attacks [21]. The protocols are often evaluated by considering the possibility of occurrence of the threat they are designed to antagonize, and their effectiveness in neutralize that threat [22].

A. The Needham Schroeder Symmetric Key Protocol

The Needham Schroeder Symmetric Key Protocol, based on a symmetric encryption algorithm. It forms the basis for the Kerberos protocol [23]. This protocol objective to establish a session key between two parties on a network, generally to protect further communication. A user needs to access a file from a secure file system. As a first stage, the user appeals a session key to the authenticating server by conferring her temporarily and the name of the secure file system to which she needs access. The server endue a session key, encrypted using the key shared between the server and the user. The session key also contains the user's temporary, just to confirm it's not a replay. In the end, the server endue the user a copy of the session key encrypted using the key shared between the server and the secure file system. The user forwards the key to the secure file system, which can decrypt it using the key shared with the server, thus authenticating the session key. The secure file system sends the user a temporary encrypted using the session key to show that it has the key. The user performs a straightforward operation of the temporary, re-encrypts it, and sends it back, verifying that she is still alive and that she holds the key. Thus, secure communication is established between the user and the secure file system. The issue with this protocol is that the secure file system has to assume the key that it receives from the authenticating server is fresh. This may not be correct. Besides, if a hacker gets hold of the user's key, he could use it to set up session keys with many other principals. It's not possible for a user to revoke a session key in case she explore imitation or improper use via usage logs. Eventually, the Needham Schroeder protocol is vulnerable to replay attack, because it's not possible to determine if the session key is latest or current.

B. The Kerberos Protocol

The Kerberos is a network authentication protocol and it is designed to endue robust authentication for client and server applications by using secret key cryptography. That works on the basis of tickets to permit nodes communicating over a non safe network to vindicate their identity to one another in a safe manner. The protocol was named after the character Kerberos (or Cerberus) from Greek mythology, the ferocious three-headed guard dog of Hades [24]. Its designers aimed it firstly as a client server model and it provides fraternal authentication both the user and the server verify each other's identity. If a user needs to access a secure file system that uses Kerberos. Firstly, the user logs on to the authentication server using a password. The client software on the user's PC bring in a ticket to this server that is encrypted under the user's password and that contains a session key. Presuppose the user is authenticated, he now uses the session key to get access to secure file system that's controlled by the ticket granting server. Afterwards, the user requests access to the secure file system from the ticket-granting server. If the access is allowed, a ticket is created containing an appropriate key and provided to the user. The user also gets a copy of the key encrypted under the session key [25]. The user now confirms the ticket by sending a timestamp to the secure file system, which confirms it's alive by sending back the timestamp incremented by 1. After that, the user can interact with the secure file system. This protocol messages are protected against eavesdropping and replay attacks. The Kerberos makes on symmetric key cryptography and need a sure of third parties, and optionally may use public key cryptography during certain stages of authentication. Kerberos is extensively used and is incorporated into the Windows Active Directory server as its authentication procedure. In practice, Kerberos is the most extensively used security protocol, Kerberos uses UDP port 88 by default.

V. THE HADOOP

The Hadoop is an open-source software framework for storing data and running applications on clusters of

commodity hardware [6]. It endue enormous amount of storage for any kind of data, enormous processing power and the ability to handle virtually limitless concurrent tasks or jobs. These are mostly useful for achieving greater computational power at small cost. Hadoop does not rely on hardware to endue fault-tolerance and high availability, rather Hadoop library itself has been designed to detect and handle lack of success at the application layer [26]. Servers can be added or delete from the cluster dynamically and Hadoop continues to operate without any obstacle.

A. The Hadoop Stack

Hadoop core modules and primary components are mentioned to as the Hadoop Stack. At the same time, the Hadoop core modules provide the basic working functionality for a Hadoop cluster [6]. In figure 2 shown the Hadoop common module confers the shared libraries, and HDFS offers the distributed storage and functionality of a fault-tolerant file system. YARN or MapReduce confers the distributed data processing functionality. So, in the absence of all the bells and whistles, that's a functional Hadoop cluster [1]. You can configure a node to be the NameNode and add a couple of DataNodes for an elementary, functioning Hadoop cluster. Here's a concise introduction to each of the core modules.

1) Hadoop Common: These are the common libraries or usefulness that support functioning of other Hadoop modules. Since the other modules use these libraries heavily, this is the backbone of Hadoop and is completely essential for its working.

2) Hadoop Distributed File System (HDFS): The HDFS is at the heart of a Hadoop cluster. It is a distributed file system that is fault tolerant, comfortably scalable, and provides high throughput using local processing and local data storage at the data nodes [1]. HDFS holds a very huge amount of data and provides comfortable access. To store such large data, the files are stored across multiple machines. These files are stored in an inessential fashion to rescue the system from possible data losses in case of lack of success. HDFS also makes applications available to parallel processing.

3) Hadoop YARN: The YARN is a structure for job scheduling and cluster resource management. It uses a global resource manager process to efficaciously [27] manage data processing resources in a Hadoop cluster in conjunction with Node Manager on each data node. The resource manager also has a pluggable scheduler that can schedule jobs and works with the application master process on DataNodes. It uses MapReduce as a distributed data processing algorithm by default, but can also use any other distributed processing application as needed.

4) Hadoop MapReduce: A YARN based system for parallel processing of huge data sets. MapReduce is the algorithm that takes processing of data. All the data nodes can process maps and reduce local, autonomously and in parallel, to provide the highest throughput that's needed for very huge datasets [8]. The MapReduce algorithm contains two vital tasks, namely Map and Reduce. Firstly, the Map takes a set of data and converts it into another set of data, where individual elements are split into tuples for instance key & value pairs. Secondly, reduce task, which takes the output from a map as an input and integrate those data tuples into a smaller set of tuples.

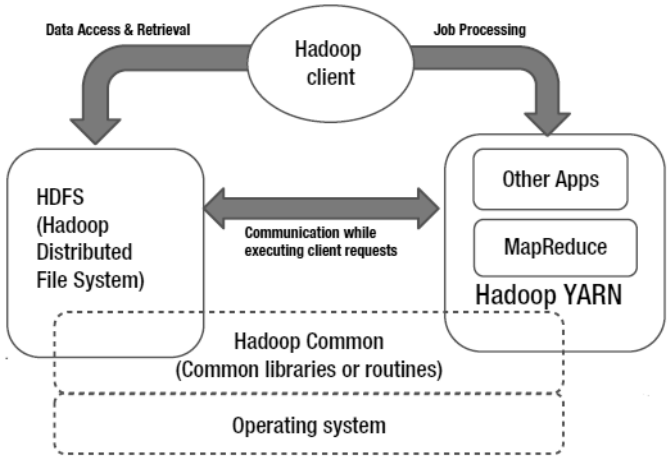

Figure 2. The Hadoop Core Modules

VI. THE HADOOP SECURITY

We live in a very insecure world, for instance starting with the key to your home's front door to those all necessary virtual keys, your passwords, everything need to be secured and well. In the world of big data where humungous amounts of data [28] are processed, transformed, and stored, it's all the more necessary to secure your data [29]. Despite the fact that, Hadoop does have inherent security concerns due to its distributed architecture, the circumstances described is extremely unlikely to occur on a Hadoop installation that's managed securely [30]. A Hadoop installation that has clearly defined user roles and multiple levels of authentication and encryption for confidential data will not let any unauthorized access go through. Hadoop was meant to process huge amounts of web data in the public domain, and hence security was not the focus of development. That's why it lacked a security model and only provided basic authentication for HDFS, which was not very convenient, since it was extremely simple to imitate another user. Another problem is that Hadoop was not designed and

developed as [31] a cohesive system with predefined modules, but was rather developed as a collage of modules that either correspond to different open source projects or a set of extensions developed by different vendors to supplement functionally deficient within the [32] Hadoop ecosystem. Presently, the standard community supported way of securing a Hadoop cluster is to use Kerberos security and its major [33] components now fully support Kerberos authentication.

A. Authentication and Authorization

The first and most essential consideration for security is authentication. A user needs to be authenticated before he is permitted to access the Hadoop cluster. Because Hadoop doesn't do any security, authentication, Kerberos is often used by Hadoop to provide authentication. When implementing security, your next step is authorized. In particular, how can you implement fine grained authorization and roles in Hadoop. There is no concept of a table and that makes it harder to authorize a user for partial access [34] to the stored data. Authorization is a much dissimilar beast than authentication. Authorization tells us what any given user can or cannot do within a Hadoop cluster, after the user has been triumphingly authenticated. In HDFS this is the first instance of governed by file permissions. Infrequently the necessary permissions for data don't match the existing group structure for an organization. New versions of HDFS support ACL (Access Control List) functionality, and this will be very useful in such circumstances [6]. In ACL we can specify read & write permissions for specific users or groups as needed.

At present, Hadoop is configurable in either secure or non-secure mode. The main dissimilarity is that secure mode needs authentication for every user and service. Kerberos is the basis for authentication in Hadoop secure mode [35]. The data is encrypted as part of the authentication process. Several organizations perform authentication in the Hadoop environment by using their Active Directory or LDAP solutions. This approach formerly wasn't consistent with the Hadoop environment and is a good representation of how Hadoop is maturing and evolving.

B. Trouble Free Administrate HDFS

The securely administering HDFS presents a number of challenges, due to the design of HDFS and the way it is structured [6]. Monitoring can help with security by forewarn we are unauthorized access to any Hadoop cluster resources [1]. We can design countermeasures for malicious attacks based on the severity of these warning. The Hadoop provides metrics for this monitoring, they are unwieldy to use. The standard Hadoop distributions by Hortonworks and Cloudera provide their own monitoring modules. Audit logs supplement the security by recording all access that flows via to the Hadoop cluster [36]. We can decide the level of logging, and advanced log management provided by modules like Log4j confers a lot of control and flexibility in the logging process. The Log4j API is at the heart of Hadoop logging. The Log4j module confers extensive logging capabilities and contains many logging levels that we can use to limit the outputting of messages by category as well as limiting the messages by their class.

C. Encryption

Being a distributed system, Hadoop has data spread across a huge number of hosts and stored locally. There is a huge amount of data communication between these hosts, hence data is vulnerable in transit as well as when at rest and stored on local storage. Hadoop started as a data store for collecting web usage data as well as other forms of non confidential huge volume data [37]. That's why Hadoop doesn't have any built-in provision for encrypting data. At present, the circumstances are changing and Hadoop is increasingly being used to store confidential warehoused data in the corporate world. This has created a need for the data to be encrypted in transit and at rest. Presently, there are a number of substitutes available to help you encrypt our data. Internode communication in Hadoop uses protocols such as RPC, TCP/IP, and HTTP. RPC communication can be encrypted using a straightforward Hadoop configuration option and is used for communication between NameNode, JobTracker, DataNodes, and Hadoop clients [38]. That leaves the genuine write & read of file data between clients and DataNodes (TCP/IP) and HTTP communication unencrypted [39]. It is possible to encrypt TCP/IP or HTTP communication, but that needs the use of Kerberos or SASL frameworks.

There are a number of preferences for implementing encryption at rest with Hadoop, but they are offered by various vendors and rely on their distributions to implement encryption. Most notable are the Intel Project Rhino (Apache Software Foundation and open source) and AWS (Amazon Web Services) offerings, which confer [40] encryption for data stored on disk. AWS encrypts data stored within HDFS [6] and also supports encrypted data manipulation by other components such as HBase or Hive. This encryption can be transparent to users or can prompt them for passwords before allowing access to confidential data, can be applied on a file-by-file basis, and can work in combination with external key management applications. This encryption can use symmetric as well as asymmetric keys. To use this encryption, confidential files must be encrypted using a symmetric or asymmetric key before they are stored in HDFS [36]. When an encrypted file is stored within HDFS, it remains encrypted. It is decrypted as needed for processing and re-encrypted before it is moved back into storage. The outcome of the analysis is also encrypted, including intermediate findings.

VII. THE KERBEROS AUTHENTICATION IN HADOOP

Authentication is the primary level of security for any system. It is all about validating the identity of a user or a process. It means verifying a username and password. In a secure system, the users and the processes need to identify themselves [25]. Then the system needs to validate the identity. The system must assure that you are the one who you claim to be. The authentication doesn't end there. Once our identity is validated, it must flow further down to the system. Our identity must propagate in the system along with your all possible action and to all possible resources that you access on the network. This kind of authentication is not only needed for users, but it is also compulsory for [34] every process or service. The non appearance of an authentication, a process or a user can pose itself to be a trusted identity and gain access to the data. Most of the systems implement this capability.

The Hadoop works with a group of computers and every computer executes a separate operating system. The operating system authentication works within the boundary of an operating system. However, Hadoop works across those boundaries [41]. So, preferably, Hadoop should have a network-based authentication system. However unluckily, Hadoop doesn't have a built-in capability to authenticate users and propagate their identity. So, the community has following options. Firstly, develop an authentication capability into Hadoop [42]. Secondly, integrate with some other system that is purposefully designed to confer the authentication capability over a networked environment. They decided to go with the second option. So, Hadoop uses kerberos for authentication and identity propagation [43]. We ask a question here. Why Kerberos? Why not something else like SSL certificates because, Kerberos performs better than SSL, and managing users in Kerberos is much more [44] simple shown in figure 3. To eliminate a user, we just delete it from Kerberos whereas revoking an SSL certificate is a complicated thing.

- **AS_REQ** is the preliminary user authentication request. This message is directed to the KDC component known as an authentication server (AS).
- **AS_REP** is the reply of the authentication server to the earlier request. Principally it contains the TGT (encrypted using the TGS secret key) and the session key (encrypted using the secret key of the requesting user).
- **TGS_REQ** is the request from the client to the Ticket Granting Server (TGS) for a service ticket. This packet contains the TGT acquired from the earlier message and an authenticator generated by the client and encrypted with the session key.
- **TGS_REP** is the reply of the Ticket Granting Server to the earlier request. Located inside is the requested service ticket (encrypted with the secret key of the service) and a service session key generated by TGS [44] and encrypted using the earlier session key generated by the AS.
- **AP_REQ** is the request that the client sends to an application server to obtain a service. The components are the service ticket acquired from TGS with the earlier reply and an authenticator again originate from the client, but this time encrypted using the service session key.
- **AP_REP** is the reply that the application server gives to the client to prove it really is the server the client is expecting [45]. This packet is not always requested. The client requests the server for it only when mutual authentication is compulsory.

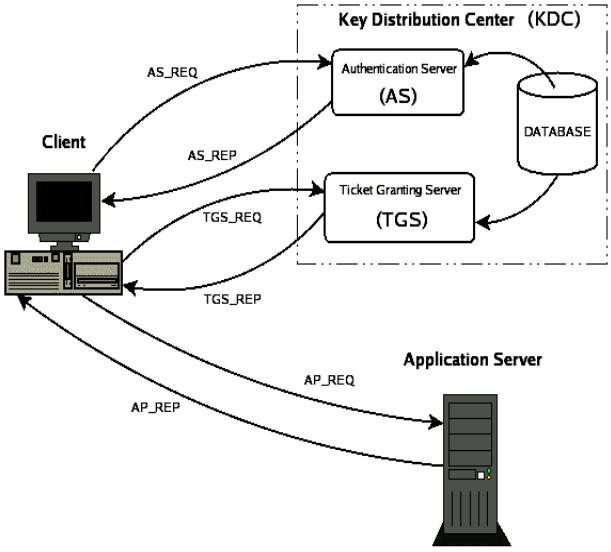

Figure 3. The Kerberos Architecture

1) Realm: The term realm designates an authentication, administrative domain. Its aim is to establish the boundaries within which an authentication server has the authority to authenticate a user, host or service. this does not mean that the authentication between a user and a service that they must belong to the same realm, if the two objects are part of dissimilar realms and there is a trust relationship between them, then the authentication [45] can take place. The name of a realm is case sensitive, i.e. There is a difference between upper and lower case letters, but usually realms always appear in upper case letters.

2) Principal: A principal is the name used to refer to the entries in the authentication server database. A principal is correlated with every user, host or service of a given realm. A principal in Kerberos five is a type.

3) Ticket: A ticket is something a client offer to an application server to demonstrate the authenticity of its identity. Tickets are issued by the authentication server and are encrypted using the secret key of the service they are intended for. Because this key is a secret shared only between the authentication server and the server providing the service, not even the client, which requested the ticket can know it or transformation its contents. Each ticket has an expiration (generally 10 hours). This is necessary since the authentication server no longer has any control over [46] an already issued ticket. Even though the realm administrator can impede the issuing of new tickets for a certain user at any time, it cannot impede users from using the tickets they already possess. This is the reason for limiting the lifetime of the tickets in order to limit any exploitation over time.

4) Encryption: The Kerberos often needs to encrypt and decrypt the messages such as tickets and authenticators passing between the different participants in the authentication. It is essential to note that Kerberos uses only symmetrical key encryption. The objective of the Kerberos protocol is to impede the user's password from being stored in [47] its unencrypted form, even in the authentication server database. Each encryption algorithm uses its own key length, if the user is not to be forced to use a dissimilar password of a fixed size for each encryption method supported, the encryption keys cannot be the passwords. For these reasons the string2key function has been introduced, which alter an unencrypted password into an encryption key appropriate for the type of encryption to be used. This function is called each time a user alter password or enters it for authentication. The string2key is called a hash function, meaning that it is static, given that an encryption key cannot determine the password which generated it and the renowned hashing algorithms are crc32 and md5.

5) Key Distribution Center: The authentication server in a Kerberos environment, based on its ticket distribution function for access to the services, is called key distribution center or more briefly kdc [48]. It lives entirely on a single physical server, it can be logically considered divided into three parts first database, second authentication server (as) and third ticket granting server (tgs). The database is the container for entries correlated with users and services. The authentication server is the part of the kdc, which replies to the primary authentication request from the client, when the user, not yet authenticated, must enter the password. The ticket granting server is the kdc component which distributes service tickets to clients with a valid tgt, promise to the authenticity of the identity for obtaining the requested resource on the application servers. The tgs can be considered as an application server, which endue the issuing of service tickets as a service.

6) Session Key: The users and services share a clandestine with the kdc. For users, this clandestine is the key derived from their password, while for services, it is their secret key. These keys are called long term, since they do not alter when the work session alter. However, it is important that the user also shares a clandestine with the service, at least for the time in which a client has a work session open on a server, this key, generated by the kdc when a ticket is issued, is called the session key. The copy intended for the service is enveloped by the kdc in the ticket, while the copy intended for the user is encapsulated in an encrypted packet with the user long term key. The session key plays a primary role in demonstrating the authenticity of the user.

7) Authenticator: Furthermore, if the user principal is present in a ticket and only the application server can extract and possibly manage such information, this is not enough to promise the authenticity of the client. A renegade could capture the ticket when it is sent by a legitimate client to the application server, and at a propitious time, send it to illegitimately obtain the service. On the contrary, including the IP addresses of the machine from where it is possible to use it is not very advantageous, it is known that in an open and insecure network addresses are easily disproved [49]. To solve the issue, one has to exploit the fact that the client and server, at least during a session have the session key in common that only they know [50]. The following policy is applied along with the request containing the ticket, the client adds another packet, where the user principal and time stamp are included and encrypts it with the session key, the server which must proposal the service, upon receiving this request, unpacks the first ticket, extracts the session key and, if the user in fact who he says, the server is able to un- encrypt the authenticator extracting the timestamp. If the latter differs from the server time by less than 2 minutes, then the authentication is triumphant.

8) Replay Cache: The possibility exists for a renegade to at the same time steal both the ticket and the authenticator and use them during the 2 minutes the authenticator is valid. This is very arduous, but not impossible. To solve this issue with kerberos 5, replay cache has been introduced. In the application servers, there exists the capacity to remember authenticators which have arrived within the last 2 minutes, and to denial them if they are replicas. With this the issue is resolved as long as the renegade is not smart enough to copy the ticket and authenticator and make them arrive at the application server before the legitimate request arrives. This really would be a deception, since the authentic user would

be rejected while the renegade would have access to the service.

9) Credential Cache: The client not ever keeps the user's password, nor does it memorized the confidential key obtained by applying string2key, they are used to decrypt the replies from kdc and straightaway discarded. However, on the other hand, to implement the single sign-on characteristic, where the user is asked to enter the password just once per work session, it is essential to memorize the tickets and related session key. The place where this data is stored is called the "credential cache". Where this cache needs to be located does not depend on the protocol, but differ from one implementation to another. Often for portability purposes, they are located in the filesystem. In other implementations, in order to increase security in the event of vulnerable clients, the credential cache is placed in an area of the memory accessible only to kernels and not mutable on the disk.

A. How Kerberos Authentication Work in Hadoop

Let's suppose we want to list a directory from HDFS on a Kerberos enabled Hadoop cluster [51].

Step 1. First activity, we must be authenticated by Kerberos. On a Linux machine, we can do it by executing the kinit tool. The kinit program will ask you for the password. Then, it will send an authentication request to the Kerberos authentication server.

Step 2. On a successful authentication, the authentication server will respond back with a ticket granting server.

Step 3. The kinit will store the ticket granting server in credentials cache. So, now we have ticket granting server that means, we have got our authentication, and we are ready to execute a Hadoop command.

Step 4. Let's say we run following command. hadoop fs --ls / So, we are using Hadoop command. That's a Hadoop client. Right?

Step 5. At the moment, the Hadoop client will use ticket granting server [25] and reach out to ticket granting server. The client approaches the ticket granting server to ask for a service ticket for the Name Node service.

Step 6. The ticket granting server will grant as a service ticket, and the client will cache the service ticket.

Step 7. At the moment, we have a ticket to communicate with the Name Node. So, the Hadoop RPC will use the service ticket to reach out to Name Node.

Step 8. They will again swap the tickets. Ticket proves our identity and Name node's Ticket determines the identification of the Name Node [34]. Both are sure that they are talking to an authenticated entity. We call this a mutual authentication.

Step 9. The next part is authorization. If we have permissions to list the root directory, the Name Node will return the outcome to us. That's all about Kerberos Authentication in Hadoop.

VIII. THE MONITORING SYSTEM IN HADOOP

The monitoring a distributed system is eternally challenging, not only are multiple processes interacting with users and each other, but you must monitor the system without influence the performance of those processes in any way [52]. A system like Hadoop presents an even greater challenge, therefore the monitoring software has to monitor individual hosts and then consolidate that data in the context of the whole system. It also needs to consider the roles of different components in the situation of the whole system [53]. Afterward, the monitoring system needs to have a capacity of abbreviate monitoring thresholds by role as well. The monitoring system for distributed systems needs to have access to details of processes executing at any time. This is essential for generating alerts or performing any deterrent action. Hadoop distributed architecture is a marked reform in efficiency over conventional client and server processing, a distributed processing model can make better a unsophisticated monitoring system as well. If a localized monitoring process stores and captures monitoring data for each node in a Hadoop cluster. Each of these localized processes can then transmit data to other nodes in the cluster and also receive copies of data from other nodes in the cluster [54]. A polling process can poll monitoring data for the entire cluster from any of the nodes within the cluster at any predefined frequency shown in figure 4. The data can be written to a repository and stored for further processing or displayed by web based front-end or a graphical.

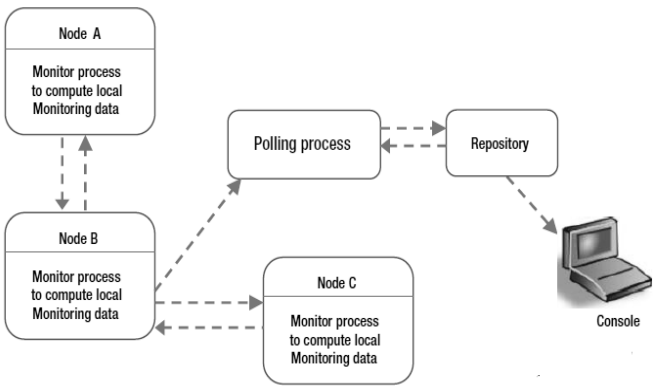

Figure 4. Hadoop Monitoring System

In this architecture, even adding 2000 hosts for monitoring would not contrarily affect performance. No extra load burdens any of the existing nodes, or the polling process, in view of the fact that the polling process can still poll from any of the nodes and doesn't have to make multiple passes. The cluster nodes transmit data to a common channel that is received by all other nodes. So, rise the number of nodes does not influence polling process or system performance in any way, making the architecture highly scalable.

A. Hadoop Metrics

The Hadoop metrics are straightforwardly information about what's happening within your system, like as memory usage, number of open connections, or remaining capacity on a node. We can configure every Hadoop daemon to collect metrics at a regular period and then output the data using a plug-in [55]. The collected data can contain information about Hadoop daemons, events, and measurements. The output plug-in you use determines the metrics destination. Depending on the information they contain metrics are classified into four type jvm, dfs, rpc, and mapred. The metrics for jvm contain basic statistics for JVM such as memory usage or thread counts, etc. This type is applicable for all Hadoop daemons. The dfs (distributed file system) type is applicable to NameNode and DataNode. Some of the metrics for this type output information such as capacity or number of files, number of failed disk volumes, remaining capacity on that particular worker node. The JobTracker and TaskTracker use the mapred context for their counters. These metrics contain pre job counter data, job counters, and post job counters. The rpc type is used for remote procedure call metrics like as the average time taken to process an rpc, number of open connections, and the like, and is applicable to all Hadoop daemons.

B. Security in Hadoop Metrics

The Hadoop Metrics can provide useful security information, including the following.

1) Responsibility for NameNode: It's necessary to monitor the activity on NameNode, as it can confer a lot of information that can warn you of security problem. Being the "brain" of a Hadoop cluster, NameNode is the hub of all the file creation activity. If the number of newly created files transformation drastically or the number of files whose permissions are transformation increases drastically, the metrics can trigger warning, so we can investigate.

2) Responsibility for a DataNode: For a DataNode, if the number of reads or writes by a local client rise unexpectedly, you definitely need to examine. Also, if the number of blocks added or removed modification by a huge percentage, then metrics can trigger alerts to warn [56].

3) Responsibility for RPC-Related Processing: For the NameNode, we need to monitor closely the RPC metrics, like as the number of processed RPC requests, number of failed RPC authentication calls [57], or number of failed RPC authorization calls. We can differentiate the daily numbers with weekly averages and generate warning if the numbers differ by a threshold percentage.

4) Responsibility for Unexpected Alteration in System Resources: It is advantageous to monitor for unexpected alteration in any of the major system resources, like as available memory, CPU, or storage. Hadoop provides metrics for monitoring these resources, and you can either define a specific percentage or monitor for a percent deviation from weekly or monthly averages. The ensuing method is more precise, as some of the clusters may never hit the target warn percentage even with a malicious attack. If you have defined a warning threshold of 75% or 95%, then you will never get a warning. On the other hand, if you have defined we warning threshold for 50%, then we will definitely get a warning.

C. Security Monitoring in Hadoop

The optimal security monitoring system for your Hadoop cluster is a system that matches our environment and necessity. In some circumstance, making sure that only authorized users have access may be most important, while in another circumstance, you may need to monitor the system resources and raise an immediate alert if an unexpected alteration in their usage occurs. Some cluster administrators merely want to monitor failed authentication demand. The Hadoop security monitoring, Ganglia and Nagios, meet this challenge by providing pliability and varied means of monitoring the system resources, connections, and any other part of your Hadoop cluster that's technically possible to monitor. Both are open source tools [58] with dissimilar strengths that complement each other nicely. Ganglia is very good at gathering metrics, tracking them over time, and aggregating the outcome, while Nagios mainly focuses more on providing an alerting mechanism. Both these tools have agents running on all hosts in a cluster and gather information via a polling process that can poll any of the hosts to get the essential information.

IX. THE ENCRYPTION IN HADOOP

The majority of the Hadoop distributions now have Kerberos installed and implemented and include a simple option to implement authorization as well as encrypted in transit, however the options are limited for at rest encryption for

Hadoop, especially with file level granularity. Encryption is the last line of defense when a hacker gets complete access to our data [59]. It is a satisfying feeling to know that our data is still going to be safe, since it can't be decrypted and used without the key that scrambled it. The most recent versions of Hadoop supports encryption [60]. We can create an encrypted region and the data that we transfer to these regions will be encrypted automatically and the data retrieved from this region will be decrypted automatically. This is also known as REST data encryption. There are two fundamental keys based encryptions first symmetric and second asymmetric. The symmetric algorithms use the same key for encryption as well as decryption. Two users share a secret key that they both use to encrypt and send information to the other as well as decrypt information from the other [61]. Therefore a separate key is needed for each pair of users who plan to use it, key distribution is a main issue in using symmetric encryption. In this context mathematically, m users who need to communicate in pairs need $m \times (m-1)/2$ keys. So, the number of keys increases almost exponentially with the number of users. Two famous algorithms that use symmetric key are DES and AES. Asymmetric or public key systems don't have the problem of key distribution and an exponential number of keys. A public key can be distributed through an e-mail message or be copied to a shared directory. A message encrypted using it can be decrypted using the commensurate private key, which only the authorized user possesses. The famous encryption algorithm RSA [62] uses public key. Public key encryption, however, is generally ten thousand times slower than symmetric key encryption because the modular exponentiation that public key encryption uses involves multiplication and division, which is slower than the bit operations for instance addition, exclusive OR, substitution, shifting, that symmetric algorithms use. For this cause, symmetric encryption is used more commonly, while public key encryption is reserved for specialized applications where speed is not a constraint. The symmetric and asymmetric encryptions, and DES, AES, and RSA in specifically, are used as building blocks to perform like computing tasks as signing documents, detecting a modification, and exchanging confidential data.

A. Hadoop Encryption Using Intel

The Hadoop was designed without much security in mind, hence a spiteful user can bypass the NameNode and access a data node directly and if we know the block location of data, that data can be retrieved or alter. Therewith, data that is being sent from a DataNode to a client can be comfortably sniffed using generic packet sniffing technology.

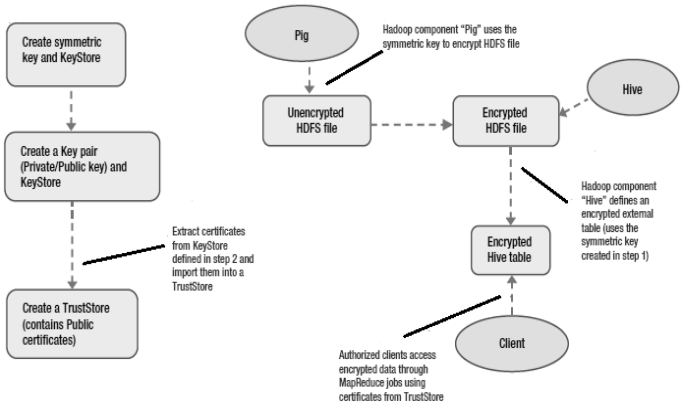

Figure 5. Hadoop Encryption Using Intel

Intel, however, promise the Hadoop world that its [63] intentions were only to contribute to the Hadoop ecosystem and assist out with Hadoop security concerns. Intel claimed its Hadoop distribution [64] worked in perfect harmony with specific Intel chips (used as the CPU) to perform encryption and decryption about 10 to 20 times swifter than current substitute. Intel distribution used codecs to implement encryption and offered file level encryption [65] that could be used with HBase or Hive. It used symmetric as well as asymmetric keys in conjunction with Java KeyStores. The client's need was encryption at rest for confidential financial data stored within HDFS and accessed using Hive. So, I had to make sure that the data file, which was pushed from SQL server as a text file, was encrypted while it was stored within HDFS and also was accessible normally via Hive, to authorized users only shown in figure 5.

B. Hadoop Encryption Using Amazon Web Services

The Amazon EMR is a managed and handle cluster platform, that over-simplify running big data structure, like as Hadoop on AWS to process and analyze massive amounts of data [66]. This structure and related open source projects, like as Apache Hive and Apache Pig, we can also process data for analytics intention and business intelligence workloads. Besides, we can use Amazon EMR [67] to transform and move huge amounts of data into and out of other AWS data stores and databases. AWS provides many configurations or models for encryption usage. The first model, model A, lets you control the encryption method as well as KMI (key management infrastructure). It proposal the highest flexibility and control, but you do all the work. The model B lets you control the encryption method while AWS stores the keys. The most rigid preference, model C, gives you no control over KMI or encryption method, while it is the easiest to implement because AWS does it all for us. To implement model C, we necessity to use an AWS service that supports server side encryption. The EMR model provides server side encryption of our data handle and manages the encryption method as well as keys guileless for us.

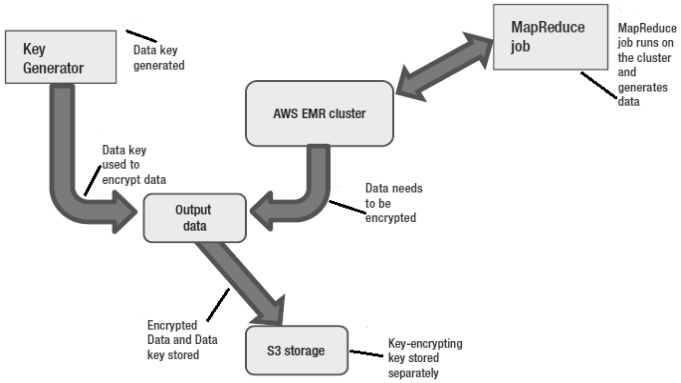

Figure 6. Hadoop Encryption Using Amazon Web Services

In figure 6 illustrate the envelope encryption procedure AWS is used for server side encryption. The primary steps are as follows.

Step 1. The AWS service originates a data key when you need that your data be encrypted.

Step 2. AWS uses the data key for encrypting our data.

Step 3. The encrypted data key and the encrypted data are stored using S3 storage.

Step 4. AWS uses the key encrypting key (unique to S3 in this case) to encrypt the data key and store, it independently from the data key and encrypted data.

For data retrieval and decryption, this process is reversed. Firstly, the encrypted data key is decrypted using the key encrypting key, and then it is used to decrypt our data. The S3 storage service supports server side encryption shown in figure 6. The Amazon S3 server side encryption uses 256-bit AES symmetric keys for data keys as well as master keys.

X. THE SECURITY CHALLENGES

Big data systems can store huge amounts of data can handle and manage that data across many systems and provide some facility for data queries, data consistency, and systems management, etc. The Big data is nothing new to large organizations, however, it's also becoming well-liked [68] among smaller and medium sized firms due to cost reduction and provided easy to handle and manage data. The big data explosion has given rise to a host of information technology tools and capabilities that enable organizations to capture, handle, manage and analyze huge sets of structured and unstructured data for actionable insights and [69] competitive advantage. However, this new technology comes the challenge of keeping sensitive information private and secure. Due to the sensitive nature of all of this data and the losses that can be done should it fall into the erroneous hands, it is compulsory that it be protected from unauthorized access [70]. Also, these security technologies are ineffective to manage dynamic data and can control static data only. To that end, we are exploring some common security challenges in this section.

A. Secure Transaction Logs and Data

The data stored in a storage medium, like as transaction logs and other sentient information, may have differing levels, but that's not enough. For example, the transfer of data between these levels gives the IT manager insight over the data which is being moved. Data size being endlessly increased, the scalability and availability makes auto-tiering essential for big data storage management. Yet, new challenges are being posed to big data storage as the auto-tiering method doesn't keep track of data storage location.

B. Monitor, Detect and Resolve Problem

Even the optimal security models will be found wanting without the capability to detect non-compliance problem and suspected or genuine security breaches and swiftly resolve them. The company needs to make sure that best practice monitoring, and detection processes are in place.

C. Validation and Filtration of End Point Inputs

The end point devices are the primary factors for maintaining big data. The processing, storage and other necessary tasks are performed with the help of input data, which is provided by end points. Consequently, an organization should make sure to use an authentic and legitimate end point device.

D. When Generate Information for Big Data

The company have to ensure that they have the right balance between utility of the data and privacy. Before the data is stored it should be adequately anonymised, alienate any unique identifier for a user. This in itself can be a security challenge as alienate unique identifiers might not be enough to guarantee that the data will remain anonymous. The anonymized data could be could be cross referenced with other obtainable data following de-anonymization method.

E. Securing and Safe Data in Real Time

Due to huge amounts of data generation, most organizations are unable to maintain regular checks. However, it is most advantageous to perform security checks and observation in actual time or almost in actual time.

F. Granular Auditing

Analyzing various kinds of logs could be beneficial and this information could be helpful in recognizing any kind of

cyber attack or malicious activity. Hereupon, regular auditing can be advantageous.

G. Adequate Access Control Mechanisms will be key in Securing the Data

The access control has conventionally been provided by operating systems or applications limited access to the information, which commonly uncover all the information if the system or application is hacked. A superior procedure is to protect the information using encryption that only permits decryption if the entity trying to access the information is authorized by an access control policy.

H. Missing Security Audits

In the big data security audits help companies gain consciousness of their security gaps. In spite of the fact that, it is advised to perform them on a regular basis, this recommendation is rarely met in reality. Working with big data has adequate challenges and concerns as it is, and an audit would only add to the list. In addition, the lack of time, resources, qualified personnel or lucidity in business side security needs makes such audits even more unfeasible.

I. Investigation the Cloud Providers

If we are storing a big data in the cloud, you must make sure that your provider has competent protection mechanisms in place. Do see that the provider does regular basis security audits and agree to penalties in the circumstance, when enough security standards are not met.

J. Vulnerability to Imitation Data Generation

Before proceeding to all the operational security challenges of big data, we should describe the concerns of an imitation data generation. To intentionally undermine the quality of your big data analysis, cyber criminals can counterfeit data and pour it into your data lake. For example, if your manufacturing company uses sensor data to detect malfunctioning production processes, cyber criminals can penetrate your system and make your sensors show imitation outcome, say, erroneous temperatures. This way, we can stupid mistake to notice alarming trends and miss the opportunity to solve issues before serious damage is caused. Above mentioned type of challenges can be solved via applying fraud detection mechanism.

XI. CONCLUSIONS

Over the last few years, data has become one of the most essential assets for companies in almost every field. The term "Big Data" refers to the huge amounts of digital information companies and governments collect about human beings and our environment. Every single person has his own account on various social sites, shopping sites, commercial sites. The billion kilobytes of data get generated day-to-day. This data may essential for the user. We need to store this data for future use. The Apache Hadoop is used to store and analyze the data. The companies of all sizes and in virtually every industry are contending to handle and manage exploding amounts of data. But as both business and IT executives know all too well, handle and managing big data involves far more than just dealing with the storage and retrieval challenges it needs addressing a diversity of security issues as well. The purposes of this paper are to highlight the main Hadoop security, technological viewpoint and analysis that may affect big data. Furthermore, big data can be advantageous as a base for the development of the future technologies that will transform the world as we see it, like the cloud computing, Internet of Things (IoT), or on-demand services, and Blockchain, that is the reason why big data is, after all, the future.

REFERENCES

[1] Yusuf Perwej, "An Experiential Study of the Big Data," International Transaction of Electrical and Computer Engineers System (ITECES), USA, ISSN (Print): 2373-1273 ISSN (Online): 2373-1281, Vol. 4, No. 1, page 14-25, March 2017, DOI:10.12691/iteces-4-1-3

[2] V Mayer-Schonberger, K Cukier, Big data: a revolution that will transform how we live work and think, Boston:Houghton Mifflin Harcourt, 2013

[3] Yusuf Perwej, Mahmoud Ahmed AbouGhaly, Bedine Kerim and Hani Ali Mahmoud Harb,"An Extended Review on Internet of Things (IoT) and its Promising Applications", Communications on Applied Electronics (CAE), ISSN : 2394-4714, Foundation of Computer Science FCS, New York, USA, Volume 9, Number 26, Pages 8– 22, February 2019, DOI: 10.5120/cae2019652812

[4] Yusuf Perwej, Majzoob K. Omer, Osama E. Sheta, Hani Ali M. Harb, Mohmed S. Adrees, "The Future of Internet of Things (IoT) and Its Empowering Technology" , International Journal of Engineering Science and Computing (IJESC), ISSN: 2321- 3361, Volume 9, Issue No.3, Pages 20192– 20203, March 2019

[5] Gartner says 4.9 Billion Connected 'Things' Will Be in Use in 2015," Gartner Inc., 2014

[6] Nikhat Akhtar, Firoj Parwej, Dr. Yusuf Perwej, "A Perusal Of Big Data Classification And Hadoop Technology," International Transaction of Electrical and Computer Engineers System (ITECES), USA, ISSN (Print): 2373-1273 ISSN (Online): 2373-1281, Vol. 4, No. 1, page 26-38, May 2017, DOI: 10.12691/iteces-4-1-4

[7] Khadija Aziz, Dounia Zaidouni, Mostafa Bellafkih, "Real-time data analysis using Spark and Hadoop", 4th International Conference on Optimization and Applications (ICOA), IEEE, Mohammedia, Morocco , April 2018

[8] Yusuf Perwej, Md. Husamuddin, Fokrul Alom Mazarbhuiya ,"An Extensive Investigate the MapReduce Technology", International Journal of Computer Sciences and Engineering (IJCSE), E-ISSN : 2347-2693, Volume-5, Issue-10, Page no. 218-225, Oct-2017, DOI : 10.26438/ijcse/v5i10.218225

[9] Johnson Anumol, P.H. Havinash, Vince. Paul, Mr. Sankaranarayanan, "Big Data Processing Using Hadoop MapReduce Programming Model", International Journal of

Computer Science and Information Technologies, vol. 6, no. 1, pp. 127-132, 2015

[10] Tim Hegeman, Yong Guo, Mihai Capota, Bogdan Ghit, "Big Data in the Cloud: Enabling the Fourth Paradigm by Matching SMEs with Data Centers", 2nd ISO/IEC JTC 1 Study Group on Big Data, Amsterdam, 2014

[11] Youssef Gahi, Mouhcine Guennoun, Hussein T. Mouftah ," Big Data Analytics: Security and privacy challenges", IEEE Symposium on Computers and Communication (ISCC), Messina, Italy, June 2016

[12] Firoj Parwej, Nikhat Akhtar, Yusuf Perwej, "A Close-Up View About Spark in Big Data Jurisdiction", International Journal of Engineering Research and Application (IJERA), ISSN: 2248-9622, Vol. 8, Issue 1, (Part -I1), pp.26-41 January 2018, DOI : 10.9790/9622-0801022641

[13] Yusuf Perwej, "The Ambient Scrutinize of Scheduling Algorithms in Big Data Territory", International Journal of Advanced Research (IJAR), ISSN 2320-5407, Volume 6, Issue 3, PP 241-258, March 2018, DOI : 10.21474/IJAR01/6672

[14] A.A. Cardenas, P.K. Manadhata, S.P. Rajan, "Big Data Analytics for Security", IEEE Security & Privacy, vol. 11, no. 6, pp. 74-76, 2013

[15] Min Lei, Yixian Yang, Xinxin Niu, Yu Yang, Jie Hao," An overview of general theory of security", China Communications, Volume: 14 , Issue: 7,PP 1 – 10, IEEE, July 2017

[16] F. Greitzer, A. Moore, D. Cappelli, D. Andrews, L. Carroll, T. Hull, "Combating the insider cyber threat", IEEE Security Privacy, vol. 6, no. 1, pp. 61-64, Jan./Feb. 2008

[17] M. Clarkson and F. Schneider. Hyper properties. Journal of Computer Security, 18(6):1157-1210, 2010

[18] A. Datta, J. Franklin, D. Garg, L. Jia, and D. Kaynar. On adversary models and compositional security. Security Privacy, IEEE, 9(3):26-32, 2011

[19] Carsten Rudolph , Andreas Fuchs, "Redefining Security Engineering", 5th International Conference on New Technologies, Mobility and Security (NTMS), IEEE, Istanbul, Turkey, May 2012

[20] A. Yasinsac ; J. Childs ," Analyzing Internet security protocols ", Proceedings Sixth IEEE International Symposium on High Assurance Systems Engineering. Special Topic: Impact of Networking, Boco Raton, FL, USA, USA, Oct. 2001

[21] Yusuf Perwej, Firoj Parwej, Mumdouh Mirghani Mohamed Hassan, Nikhat Akhtar, "The Internet-of-Things (IoT) Security: A Technological Perspective and Review", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 5, Issue 1, Pages 462-482, February 2019, DOI: 10.32628/CSEIT195193

[22] Perrig Adrian, Szewczyk Robert, Wen Victor, Culler David, J.D. Tygar, M. Luk, G. Mezzour, A. Perrigo, V. GLigor, "SPINS: Security protocols for sensor networks", Seventh Annual ACM International Conference on Mobile Computing and Networks(MobiCom 2001) July 2001, 2007

[23] R.M. Needham and M. D. Schroeder," Using encryption for authentication in large networks of computers", Comm. ACM, Vol.21, No.12, pp. 993-999, 1978

[24] M. Sirbu, J. Chuang, "Distributed authentication in Kerberos using public key cryptography", IEEE Symposium On Network and Distributed System Security (NDSS'97), pp. 134-141, 1997

[25] A. Harbitter, D. Menasce, "Perofrmance of public-key-enabled Kerberos authentication in large networks" in Proceedings of 2001 IEEE Symposium on Security and Privacy, IEEE Computer Society Press, 2001

[26] Lin H., Seh S., Tzeng W., Lin B.P., "Toward Data Confidentiality via Integrating Hybrid Encryption Schemes and Hadoop Distributed FileSystem", 26th IEEE International Conference on Advanced Information Networking and Applications in 2012

[27] Yusuf Perwej, Bedine Kerim, Mohmed Sirelkhtem Adrees, Osama E. Sheta, "An Empirical Exploration of the Yarn in Big Data", International Journal of Applied Information Systems (IJAIS), ISSN : 2249-0868 , Foundation of Computer Science FCS, New York, USA Volume 12 , No.9, page 19-29 , December 2017, DOI: 10.5120/ijais2017451730

[28] Yusuf Perwej, Md. Husamuddin, Majzoob K.Omer, Bedine Kerim, "A Comprehend TheApache Flink In Big Data Environments", IOSR Journal of Computer Engineering (IOSR-JCE), e-ISSN: 2278-0661,p-ISSN: 2278-8727, USA, Volume 20, Issue 1, Ver. IV, PP 48-58, Feb. 2018, DOI : 10.9790/0661-2001044858

[29] Yusuf Perwej, Firoj Parwej, Mumdouh Mirghani Mohamed Hassan, Nikhat Akhtar, "The Internet-of-Things (IoT) Security: A Technological Perspective and Review", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 5, Issue 1, Pages 462-482, February 2019, DOI: 10.32628/CSEIT195193

[30] Huang Jing, LI Renfa, C. Tang Zhuo, "The Research of the Data Security for Cloud Disk Based on the Hadoop Framework", International Conference on Intelligent Control and Information Processing, June 9–11, 2013

[31] Chao YANG, Weiwei LIN, Mingqi LIU, "A Novel Triple Encryption Scheme for Hadoop-based Cloud Data Security", International Conference on Emerging Intelligent Data and Web Technologies, 2013

[32] Karim Abouelmehdi, Abderrahim Beni-Hssane, Hayat Khaloufi, Mostafa Saadi ," Big data emerging issues: Hadoop security and privacy ", 5th International Conference on Multimedia Computing and Systems (ICMCS), IEEE, Marrakech, Morocco, Oct. 2016

[33] J. Whitworth, S. Suthaharan, "Security problems and challenges in machine learning-based Hybrid Big Data processing network systems", ACM SIGMETRICS Performance Evaluation Review, vol. 41, no. 4, pp. 82-85, March 2014

[34] Kai Zheng , Weihua Jiang," A token authentication solution for hadoop based on kerberos pre-authentication", International Conference on Data Science and Advanced Analytics (DSAA), IEEE, Shanghai, China, Nov. 2014

[35] B. Saraladevi, N. Pazhaniraja, P. Victer Paul, M.S. SaleemBasha, P. Dhavachelvan, "Big Data and Hadoop-a Study in Security Perspective", Procedia Computer Science, vol. 50, pp. 596-601, 2015, ISSN 1877-0509

[36] J. Xie, S. Yin, et al. "Improving MapReduce performance through data placement in heterogeneous Hadoop clusters", In 2010 IEEE International Symposium on Parallel & Distributed

[37] GitHub, RJ97/Kuber: A Framework for Large Scale Encryption in Hadoop Environment, Mar. 2017

[38] Apache Hadoop 2.7.3 , Transparent Encryption in HDFS, Feb. 2017

[39] Simon Heron, "Advanced Encryption Standard CAES)", Network Security, vol. 2009, no. 12, pp. 8-12, December 2009

[40] P. Mehrotra, J. Djomehri, S. Heistand, R. Hood, H. Jin, A. Lazanoff, S. Saini, R. Biswas, "Performance Evaluation of Amazon EC2 for NASA HPC Applications", Proceedings of the 3rd Workshop on Scientific Cloud Computing, 2012

[41] Charles Schmitt, "Security and Privacy in the Era of Big Data" in RENCI (Renaissance Computing Institute), NCDS, White Paper, 2013

[42] Shuyu Li, Tao Zhang, Jerry Gao, Younghee Park, "A Sticky Policy Framework for Big Data Security", 2015 IEEE First International Conference on Big Data Computing Services and Application, pp. 71, 2015, ISBN 978-1-4799-8128-1/15

[43] J. Kohl, C. Neuman, "The Kerberos Network Authentication Service (V5)", Rfc, pp. 1510, September 1993
[44] S. M. Bellovin, M. Merritt, "Limitations of the kerberos authenication system", Computer Commun. Rev., vol. 20, no. 5, pp. 119-132, Oct. 1990
[45] J. T. Kohl, B. C. Neuman, T. Y. T'so, The evolution of the Kerberos authentication system. Distributed Open Systems, IEEE Computer Society Press, pp. 78-94, 1994
[46] C. Neuman, T. Yu, S. Hartman, K. Raeburn, "The Kerberos network authentication service (V5)", Network Working Group. Request for Comments: 4120, 2005
[47] F. Butler, I. Cervesato, A. D. Jaggard, A. Scedrov, "A formal analysis of some properties of Kerberos 5 using MSR", University of Pennsylvania Department of Computer & Information Science Philadelphia USA Technical Report MS-CIS-04-04, April 2004
[48] Qin Li, Fan Yang, Huibiao Zhu, Longfei Zhu, "Formal modeling and analyzing Kerberos protocol", IEEE World Congress on Computer Science and Information Engineering (CSIE) 2009
[49] William Stallings, "Cryptography and network security principles and practices" in , Pearson Prentice Hall, pp. 401-419, 2006
[50] A. Boldyreva, V. Kumar, "Provable-security analysis of authenticated encryption in Kerberos", IEEE Symposium on Security and Privacy (SP'07), May 2007
[51] S. Sakane, N. Okabey, K. Kamadaz, H. Esakix, "Applying Kerberos to the communication environment for information appliances", Symposium on Applications and the Internet Workshops (IEEE SAINT-w'03), 2003
[52] Joey Pinto , Pooja Jain , Tapan Kumar ," Hadoop cluster monitoring and fault analysis in real time ", International Conference on Recent Advances and Innovations in Engineering (ICRAIE), IEEE, Jaipur, India, Dec. 2016
[53] Kadirvel Selvi, Jeffrey Ho, José Ab Fortes, "Fault management in Map-Reduce through early detection of anomalous nodes", Proceedings of the 10th International Conference on Autonomic Computing (ICAC 13), pp. 235-245, 2013
[54] Hu Peng, Wei Dai, "Enhancing fault tolerance based on Hadoop cluster", International Journal of Database Theory and Application 7, no. 1, pp. 37-48, 2014
[55] Jianxi Yang , Chaoxiao Shen , Yaping Chi , Ping Xu , Wei Sun ," An extensible Hadoop framework for monitoring performance metrics and events of OpenStack cloud", IEEE 3rd International Conference on Big Data Analysis (ICBDA), Shanghai, China, March 2018
[56] Bao Rong, Chang Hsiu, Fen Tsai, Zih-Yao Lin, Chi-Ming Chen, "Access Security on Cloud Computing Implemented in Hadoop System", 2011 Fifth International Conference on Genetic and Evolutionary Computing IEEE, pp. 77-80, September 2011
[57] A. D. Birrell, D. J. Nelson, "Implementing Remote Procedure Calls", ACM Transactions on Computer Systems, vol. 2, no. 1, pp. 39-59, Feb. 1984
[58] J. Whitworth, S. Suthaharan, "Security problems and challenges in machine learning-based Hybrid Big Data processing network systems", ACM SIGMETRICS Performance Evaluation Review, vol. 41, no. 4, pp. 82-85, March 2014
[59] C. Yang, W. Lin, M. Liu, "A Novel Triple Encryption Scheme for Hadoop-Based Cloud Data Security", Emerging Intelligent Data and Web Technologies (EIDWT) 2013 Fourth International Conference, pp. 437-442, 2013
[60] J. Daemen, V. Rijmen, The design of Rijndael: AES-the advanced encryption standard, Springer Science & Business Media, 2002
[61] M. Hou, Q. Xu, "Secure and efficient two-party authenticated key agreement protocol from certificate less public key encryption scheme", INC IMS and IDC 2009. NCM'09. Fifth International Joint Conference on. IEEE, pp. 894-897, 2009
[62] Xin Zhou, Xiaofei Tang, "Research and Implementation of RSA Algorithm for Encryption and Decryption", the 6th International Forum on Strategic Technology, pp. 1118-1121, 2011
[63] https://pdfs.semanticscholar.org/a140/3588bbcb75452243bb8f3246dea5d49df4b1.pdf
[64] S. Gueron, "A Memory Encryption Engine Suitable for General Purpose Processors", Cryptology ePrint Archive report 2016/204, 2016
[65] Victor Costan, Srinivas Devadas, Intel sgx explained. Cryptology ePrint Archive Report 2016/086, 2016
[66] Robinson Glen, Narin Attila, Elleman Chris, "Amazon Web Services- Using AWS for Disaster Recovery", White Papers, October 2014
[67] Hamoud Alshammari , Jeongkyu Lee , Hassan Bajwa ," Evaluate H2Hadoop and Amazon EMR performances by processing MR jobs in text data sets", IEEE Long Island Systems, Applications and Technology Conference (LISAT), Farmingdale, NY, USA, April 2016
[68] C. Mora et al., "Top ten big data security and privacy challenges", Cloud Security Alliance, 2012
[69] A. Cuzzocrea, "Privacy and security of big data: Current challenges and future research perspectives", Proceedings of the First International Workshop on Privacy and Secuirty of Big Data PSBD '14, pp. 45-47, 2014
[70] M. Jensen, "Challenges of Privacy Protection in Big Data Analytics", Proceedings of the International Congress on Big Data, pp. 235-238, 2013

