International Journal on
Advanced Science
Engineering
Information Technology

Vol.9 (2019) No. 3
ISSN: 2088-5334

Static Knowledge-Based Authentication Mechanism for Hadoop Distributed Platform using Kerberos

Thoyazan Sultan Algaradi[#], B.Rama[#]

[#]*Department of Computer Science, Kakatiya University, Warangal, India*
Email: yaz.sul77@gmail.com; rama.abbidi@gmail.com

Abstract— **With the quickened phenomenal expansion of data, storing massive data has become important and increasingly growing day by day. Thus, big data came to express this large data and handling it properly under three important characteristics such as volume, veracity, and Variety. One practical of big data problems is user and services authentication. Kerberos v5 protocol provided a new solution to such this problem in the Hadoop-distributed platform (HDP). In this paper, we suggest a credible scheme by adding one more level of protection and authentication security to the Kerberos v5 protocol by using a static knowledge-based authentication (SKBA). Where in the login and verification phase by using Kerberos protocol, the KDC will replay with a question to the user-side to check the actual presence of user which the user already answered this question in his registration phase. Our credible scheme is useful in case of capturing messages that enable an eavesdropper to get the ticket that allows getting access to the HDFS as well as to avoid the common attacks with less computation, communication and storage cost. The proposed scheme works seriously and strictly to ensure the registration by delivery of user information over an insecure network in a safe manner and store this information in the KDC-database to be used later for getting access with HDFS.**

Keywords— **authentication protocol; kerberos; hadoop distributed file system; static based knowledge; big data.**

I. Introduction

Big data is a utilized term to express data of enormous volume and complexity [1]; these data grow very fast thus difficult to be efficiently processed [2]. Modern software is needed such Apache Hadoop which uses map reduce for handling such massive data volume and make it able to be processed and analyzed by parallelizing the processing and using distributed hardware [3], which considered as a big-data processing framework of open source software used by several of massive online media such as Yahoo, Facebook, and Twitter. Because of the insensitive data used previously, Hadoop was not had that much of importance [4]–[6]. What makes it imperfect, limited access solution to the cluster cloud and other systems with traditional methods, which designed to process structured data such as data warehouse and management systems of a relational database are unable to handle large-scale data of unstructured type using an effective way [7]. Thus, HDFS is used to deal with such data type of sensitive and personal information, but its problems still existed until that time, especially in its authentication where it is possible for any user to penetrate and spy another user or cluster services [8].

Moreover, the security of Big Data is an important issue [9, 10], which must focus on precisely to avoid its caused problems which may result by the access of unauthorized part, such illegal-modification, impersonated-publication and impersonated-retrieval [11]. To inhibit these, then proceed to build and design an adaptable authentication scheme has become indispensable, which enable to verify users and prevent impersonation as long as such other threaten security attacks.

In the field of authentication of big data, many proposals have been formulated and which in turn led to results that may be somewhat satisfactory. Ruidongli at al [12]declared some requirements that have to be met where integrity for big data, secure registration, secure data retrieval, efficient and flexible authorization, and distributed design are some. For that, they proposed DAAS (Distributed Authentication and Authorization Scheme) to achieve the above requirements and in its turn solve problems of AAA and CP-ABE based scheme systems. Their proposed scheme has some additional properties, thus met the proposed requirement such as integrity for big data, trustworthy registration for entities, publisher, user identity verification, security key establishment, flexible authorization, and automatic attribute updates, what make it as secure as IBS and CCP-ABE. Wang et al. [13] proposed a scheme, where the cloud center user gets access to decide who to share with the data. They consider starting to share data only with a receiver who have certain attributes, in this case, providers of data along with the receivers have to verify the authenticity of each other to be sure that data and the identity will not be leaked out.

Moreover, they proposed a verification mechanism to verify the authenticity of users' attributes. This total system called pre-authentication approach to proxy re-encryption where providers can verify the receivers' authenticity where the security under several attacks are ensured such as selective condition CCA, selective identity CCA and collusion attack, and make sure that the re-encryption keys, cipher text, and the attributes of users are anonymous. Shen et al. [14] proposed a protocol, which resorts to the tree-based signature to significantly improve the security of attribute authorization. And they extend the proposed authentication protocol to support multiple levels in the hierarchical attribute authorization structure to satisfy the big data requirements, where they show that their protocol can resist the forger attack and replay attack. In the framework of other, Ouda [2] investigates the development of a new authentication system that related to 'something you do' to find out unique patterns of the users' dynamic behaviors their new framework is for user authentication that leverages big data analytics.

Whereas organizations can collect identifiable information, Ibrahim and Ouda [15] presented an investigated for the implementation option for the new authentication approach [2]. IDA is an implementation of the second component of the big data-based user authentication framework. IDA investigation development stands on two main activities, first to create a user profile to study human dynamics and behavior, then analyze the practices and actions of the user and then classify user behavior accordingly. Second is to generate a questionnaire to authenticate users. Such that the real-time analysis of user's profiles helps generate a random set of challenging questions what made the authentication on demand feature is obtained.

Thus, their investigation approaches helped create a highly distributed authentication model, minimizing the storage of secrets, and lesser secret management overhead. By talking about the environment distributed, Abdullah et al. [16] published a proposal, where they presented the drawbacks of the normal Kerberos with identifying most of the authentication requirements that can enhance the big data security. Then introduced their improved proposal which based on the rising technology of block chain as a suggestion of utilizing the advantages of block chain technology could be leveraged to harden security systems, including authentication and authorization of big data.

Moreover, Hadoop has used the delegation tokens [8] by proposing a scheme that allows the clients of HDFS to authenticate through the block-access-token (BAT). This authentication token provided by data-node, where it used to protect sensitive data in HDFS to contra some of the attacks such replay attacks as well as impersonation attacks, the proposed HDFS authentication scheme hire elliptic-curve-cryptography (ECC) to create delegation tokens which are destined to works on master-slave architecture. Which work with smaller key size, reducing storage and transmission requirements [16] it supported the protection of the exchanged information between name-node and data-node in term of authentication. This scheme protects sensitive data stored in HDFS. However, have a weakness observed when users share public and private keys in both TLS and SSH implementation within a cloud environment [17].

Moreover, Rahul and GireeshKumar [18] proposed a new authentication framework for clients, by using cryptographic functions such as public key cryptography, private key cryptography, hashing functions, and random number generator, which helped to enhanced Hadoop security by adding a unique key to be used along with all clients. It is a secure protocol despite its drawback of the type of communication that all go through the layer by using the encrypted symmetric key. Hadoop used symmetric essential operation of Kerberos [19] too as an authentication solution to both users and services in the Hadoop-distributed platform (HDP), by using automatic fashion of implementation to MIT Kerberos for authentication that creates users and gives them to Ambari with admin privileges where Ambari hold all the services, clients and Keytabs.

In this paper, in the proposed work, a new proposal has been devised using the Kerberos v5 protocol with one more level of protection. It uses static knowledge-based authentication (SKBA) by a trusted phase for registration, login, and verification. It is intended to create a compelling and strict method allows users to access the HDFS server in a secure and protected manner in a perfect way which it will avoid famous common threats on the communication channel. The proposed tightly controlled schemes are for recording users and their required data in a safe way to later use in the login and verification process as an additional level that supports Kerberos in the failure case as well as to avoid most of the common attacks with less computation, communication and storage cost.

For more clarity, the proposed scheme strengthens the protection of authentication information exchanged between users and HDFS. The proposed design uses communication on a Kerberos v5 protocol using static knowledge-based authentication (SKBA) as an additional level. This is after the user registered his required information in the KDC database. Means it consist of two main phases first is for user registration where it will be done between users and KDC of their realm. Which allow users to be registered by a strong authentication way providing resistance to attacks of both passive and active. This phase ensures users to deliver the requested data such a password and the answer to the question that the server is putting in a safe manner on a confidential database in the KDC server side. Second is the login and verification phases, which are proceed by Kerberos v5 where the KDC will ask the user who wants access to answer his secret question that he already answered during his registration phase, to verify his identity in the event that someone could penetrate the protocol to impersonate his personality, since in this case will not be useful. As Even if the invader can get the ticket that enables him to access the HDFS server, will not be capable of replaying with the correct answer in the pre-limited time, so the HDFS server will reject him directly. Fig.1. illustrate the general structure of the adopted scenario, which consists of several points, shown as follows:

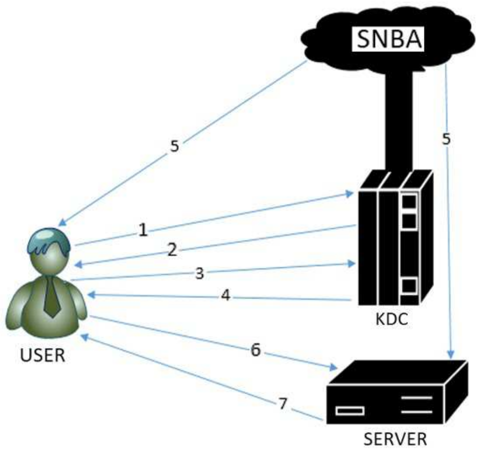

Fig.1. The general structure of the proposed approach

1- User request authentication to KDC
2- TGT + Session Key
3- Request ticket + Authenticator
4- Ticket _ Session Key
5- Retrieve Question and Answer as a static knowledge-based authentication (SNBA) then send a question to the user and Answer to the server.
6- Provide answer + Request serves + authenticator.
7- Compare Answers if valid send server authentication to the user.

Moreover, some principals related to the current study are provided as follow:

A. Kerberos v5:

Kerberos is an ideal protocol of authentication for trusted hosts on untrusted channel [20]. It uses to ensure secure communication by verifying users identities on an unprotected network to ensure the privacy of data by allowing the authentication between users and services. Meaning, each of users and services managed to identify each other [21], which is summarized as not requiring from the user to be registered in each server, and user's username and password do not need to be known in their storage. Users can enter to the server through two-step login; first, they get authentication at authentication-server (AS), and second, gets a real ticket for a specific server [22]. The client uses this ticket to log onto an application server. As shown in the above fig.2.

B. Knowledge-Based Authentication [23]:

It is one of the commonly used methods of user authentication, where the user asked to answer at least one 'secret' question. It used to use in multifactor authentication (MFA) as well as self-service password retrieval. The question should be appropriate for a large segment of inhabitancy similarly in case of the answer which it has to be an only one correct answer for each question and difficult to guess or discovered. There are two cases of the question posed, where it can be static or dynamic, wherein in the static scheme case, the user is the one who will choose questions that he wants to answer from the list of the predetermined list stored by the host and then will provide his correct answer. Questions should be realistic, clear and easy to save for the user such as "when did you complete your master degree?", "what is your favorite food?"

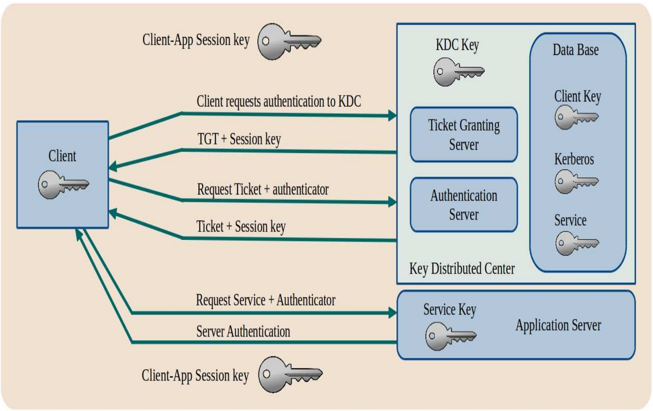

Fig.2.Kerberos key distributed center and its component

II. Material And Method

The proposed scheme is shown in this part, which mimics the authentication between the user and the server in the environment of the Hadoop-distributed platform. Kerberos protocol used to define the rules of users, servers, and applications on communication to guard Hadoop distributed platform against any possible failure in term of user authentication before accessing HDFS. Kerberos v5 with using a static knowledge-based authentication (SKBA) had recycled as provided an additional level to protect the user from impersonation attack or any real impersonate to the user's personality by checking the presence of the true user. where the KDC will replay with question that related to the user in the database and he already answered due to its registration phase, this phase are built in a serious and strict way to ensure the registration and delivery of user information over an insecure network in a safe manner, and store it in the KDC database.

The proposed scheme of conception consists of three entities: 1] HDFS client that belongs to the realm of the KDC: the person, or entity who want to get ability to access to HDFS, 2] KDC, the key distribution center, it is the third party contains the database in which user information will be stored. 3] HDFS server, the place the user aims to access, it known as the Hadoop distributed file system. There are a basic four storing identification parameters assigned to each user identified by ID in their registration phase, this parameter utilized to authenticate users through their login and verification phase, where they are as the user identification, user's password, question and user's answer, respectively. The password-footprint along with a random salt in the proposed scheme will be applied to generate keys encryption/decryption to ensure the user identification and messages confidentiality exchanged during communication.

A. Terms and Definitions

- g: a primitive root modulo P (often called a generator).
- B: a random private key.
- S: salt
- ID: user identity.
- Pw: the user's password

- Q: question
- ANS: answer to the question
- A, B: corresponding public keys.
- H( ): one-way hash function.
- K: session key.
- T1: the current time of the input device when writing the answer
- T2: a time when the server receives the answer.
- T3: the difference between T1 and T2 and it have been within a predefined threshold

Notes:

- The values P, g are well-known values agreed to beforehand.
- b is a random private key on the KDC-side.

B. The conception of the Proposed Approach

1) Registration phase: In this phase, the user will be capable of generating an individual username and password, which will also enable him to answer one of the secret questions that he will choose and then submit this information and store it safely in the KDC database. Thus, four authentication parameters will be generated based on mathematical operations formulated for authentication. Each user must have a valid password, and unique-ID, then, he must choose an appropriate question, as well as provided his answer. The following fig.3. Shows the dialogue of this phase:

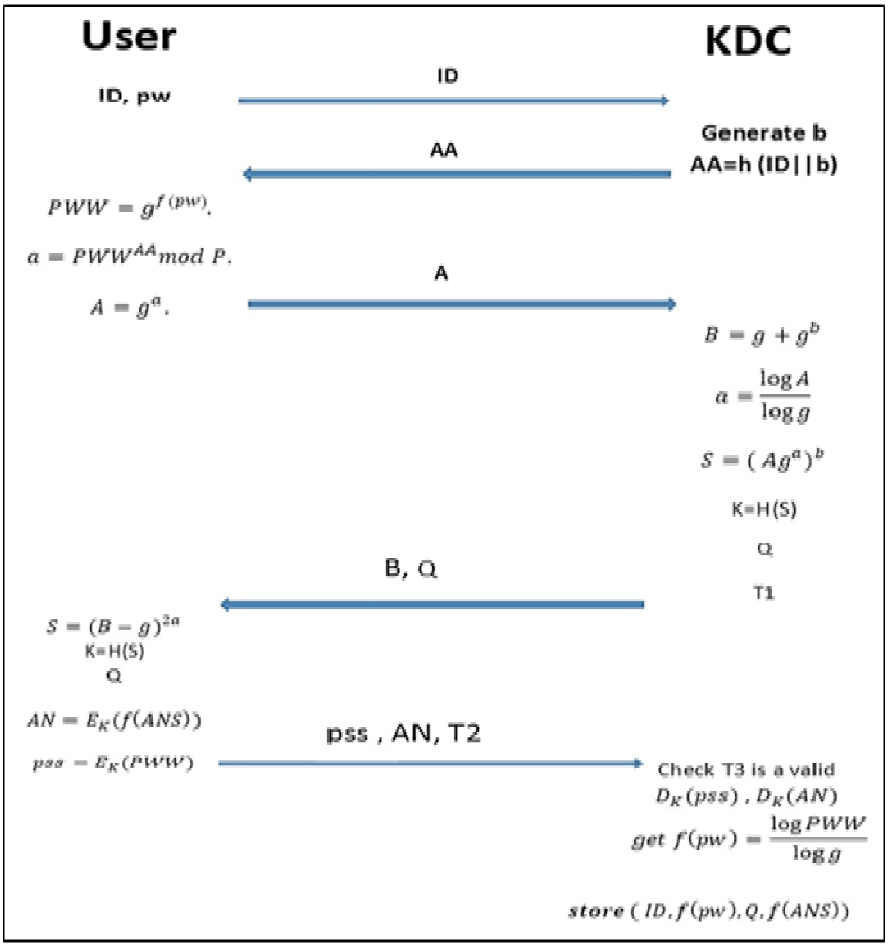

Fig.3. Registration phase of the proposed approach

Algorithm:

Step1: user side - Enter user-ID and pw.

Step2: Send user ID to KDC-side.

Step3: KDC side - Generate b as a random private key, and calculate AA=h(ID||b).

Step4: send AA to user-side.

Step5: user side - Calculate $PWW = g^{f(pw)}$, Calculate $a = PWW^{AA} mod\ P$, and calculate $A = g^{a}$.

Step6: send A to KDC-side.

Step7: KDC side - calculate $B = g + g^{b}$, Get $a = \frac{\log A}{\log g}$, calculate $S = (Ag^{a})^{b}$, Get K=H(S), Q and select T1

Step8: send B,Q to user-side.

Step9: user side - Calculate $S = (B - g)^{2a}$, Get K=H(S), Q , $AN = E_K(f(ANS))$, and $pss = E_K(PWW)$

Step10: Send PSS, ANS, T2 to KDC-side

Step11: KDC side - Check T3 is a valid time where T3= T2-T1, $D_K(AN)$ and $get\ f(pw) = \frac{\log PWW}{\log g}$.

Step12: KDC side - $store\ (ID, f(pw), Q, f(ANS))$.

2) *Login and verification phase:* In this phase, users must prove their self by user-ID to the KDC server, as the KDC based on a trusted third party [24]. For this reason, the user-side sends user-ID to the KDC server without sending the password, KDC server must checks this user-ID in its database if it exists it will start with its normal steps. For more secure authentication of KDC, it will retrieve the user's question from the database and submit it to that used to verify his or her credibility if he who asking the access or another person impersonates him. While the user has sent the correct answer after he showed his ticket, he will be capable of accessing, unless the access will be rejected directly from the HDFS server itself. The following fig.4. Shows the dialogue of this phase:

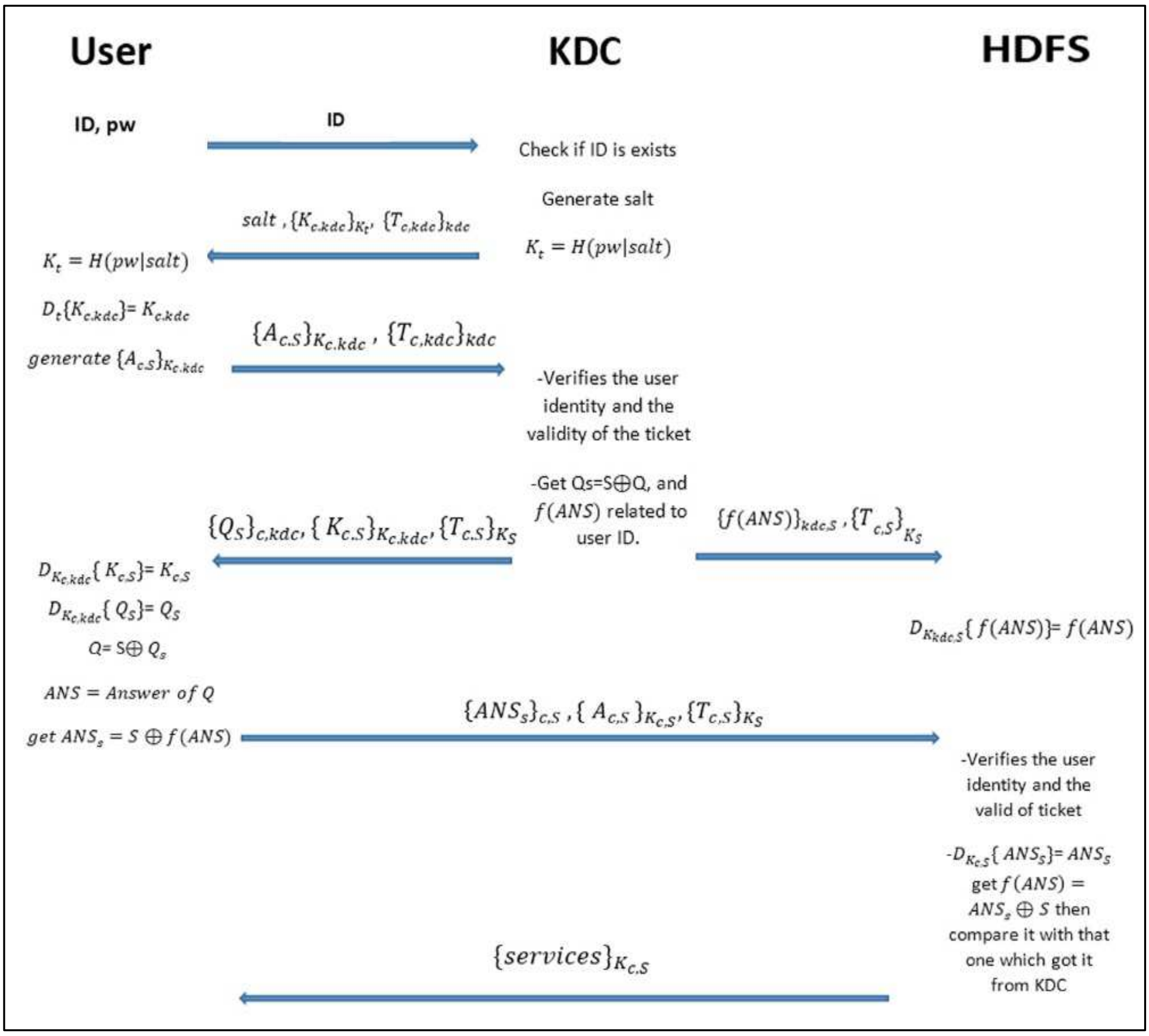
User
KDC
HDFS
ID, pw
ID
Check if ID is exists
Generate salt
$K_t = H(pw|salt)$
salt , $\{K_{c.kdc}\}_{K_t}$, $\{T_{c,kdc}\}_{kdc}$
$K_t = H(pw|salt)$
$D_t\{K_{c.kdc}\}= K_{c.kdc}$
$\{A_{c.S}\}_{K_{c.kdc}}$, $\{T_{c,kdc}\}_{kdc}$
generate $\{A_{c.S}\}_{K_{c.kdc}}$
-Verifies the user identity and the validity of the ticket
-Get Qs=S⊕Q, and $f(ANS)$ related to user ID.
$\{Q_S\}_{c,kdc}$, $\{K_{c.S}\}_{K_{c.kdc}}$, $\{T_{c.S}\}_{K_S}$
$\{f(ANS)\}_{kdc,S}$, $\{T_{c,S}\}_{K_S}$
$D_{K_{c.kdc}}\{K_{c.S}\}= K_{c.S}$
$D_{K_{c.kdc}}\{Q_S\}= Q_S$
Q= S⊕ Q_s
$D_{K_{kdc,S}}\{f(ANS)\}= f(ANS)$
$ANS = Answer\ of\ Q$
$\{ANS_s\}_{c,S}$, $\{A_{c,S}\}_{K_{c,S}}$, $\{T_{c,S}\}_{K_S}$
get $ANS_s = S \oplus f(ANS)$
-Verifies the user identity and the valid of ticket
-$D_{K_{c,S}}\{ANS_S\}= ANS_S$ get $f(ANS)$ = $ANS_s \oplus S$ then compare it with that one which got it from KDC
$\{services\}_{K_{c.S}}$

Fig.4. Login and verification phase of the proposed approach

Algorithm:

Step1: user side - Inter ID and password (pw).

Step2: Send ID to KDC.

Step3: KDC side - Check if ID exists, Generate salt, and get $K_t = H(f(pw)|salt)$.

Step4: $send\ salt$, $\{K_{c.kdc}\}_{K_t}$, $\{T_{c,kdc}\}_{kdc}$ To user.

Step5: user side - Get $K_t = H(f(pw)|salt)$, $D_t\{K_{c,kdc}\}$=$K_{c,kdc}$, and Generate $\{A_{c,S}\}_{K_{c,kdc}}$

Fig.4. Login and verification phase of the proposed approach

Step6: Send $\{A_{c,S}\}_{K_{c,kdc}}$, $\{T_{c,kdc}\}_{kdc}$ to KDC.

Step7: KDC side - Verifies the user identity and the validity of the ticket then get Qs=S ⊕ Q and $f(ANS)$ related to the user ID.

Step8: Send $\{Q_S\}_{c,kdc}$, $\{K_{c,S}\}_{K_{c,kdc}}$, $\{T_{c,S}\}_{K_S}$ from KDC-side to the user-side.

Step9: Send $\{f(ANS)\}_{K_{kdc,S}}$, $\{A_{c,S}\}_{K_{c,S}}$, $\{T_{c,S}\}_{K_S}$ from KDC-side to the server

Step10: server-side – get $f(ANS) = D_{kdc,S}\{f(ANS)\}$

Step11: user side – $D_{K_{c,kdc}}\{K_{c,S}\}$ = $K_{c,S}$, $D_{c,kdc}\{Q_S\} = Q_S$, Q= S ⊕ Q_s , ANS= Answer of Q, and get $ANS_s = S \oplus f(ANS)$

step12: $send$ $\{ANS_S\}_{c,S}$, $\{A_{c,S}\}_{K_{c,S}}$, $\{T_{c,S}\}_{K_S}$ from User-side to the server.

Step13: in Server side - Verifies the user identity and the valid of the ticket, then get$ANS_S = D_{c,S}\{ANS_S\}$, get $f(ANS) = S \oplus ANS_S$ then compare f(ANS) with that one which got it from KDC in step10.

Step14: if the comparison is valid, send $\{services\}_{K_{c,S}}$ from server-side to the user side.

III. Result And Discussion

A. Security Analyses

In this section, we will present the security analysis of the proposed approach by discussing some of the common attacks we have addressed through a powerful and strict way to enable users access to HDFS in a more robust secure manner over the untrusted channel. And explain the explanation of how to avoid them, these attacks mentioned and described below.

1) Guessing attack: The proposed registration phase provide zero-knowledge security by using the unify key K=H(S), which is the hash result calculated in both side based on some other mathematical formulae as we see in eq1 in the user-side and eq2 in the KDC-side

$$S = (B - g)^{2a} \tag{1}$$

$$S = (Ag^{a})^{b} \tag{2}$$

Moreover, the password never be shared during the transmission, it is a cryptographic function "pss" of 'PWW' that we got it by the predefined g power the virtualization function of the original password f(pw) as shown in eq3 and eq4. Finally, the result will be sent

$$PWW = g^{f(pw)} \tag{3}$$

$$pss = E_k(PWW) \tag{4}$$

Moreover, the scheme will help to avoid the attacks of password guessing even after the key detected because the password will be in formulae of eq3, and only the KDC server can solve this formula to know the value of "g" which never sent through the channel. The same with the proposed login and verification proposed phase where the password never needs to share over the network, where we used salt to get the first key $E_t$to reciprocation between the user and HDFC server as shown in eq5.

$$K_t = Hpw|salt) \tag{5}$$

Where the password already exists in both side what makes guess the original password is almost impossible. The same happens with an answer too during translation, where first the virtualization function will be calculated then encrypt them and send in the form of eq6.

$$AN = E_k(f(ANS)) \tag{6}$$

2) Replay attack: A replay attack occurs when a third party captures a command in transmission and replays it at a later time. In the proposed registration phase, finding discrete logarithmic is a complicated problem where even in case the eavesdropper obtains the inverse logarithm of such a high exponential number to get 'a,' especially when the value g is a predefined value in both side, eq7, and eq9 show that.

$$a = PWW^{AA} \, mod \, P \tag{7}$$

$$A = g^{a} \tag{8}$$

As well as, it is also required from the user-side to replay during a limited predefined time which makes T3 is valid to be able to continue, where T3 as in eq9.

$$T3 = T2 - T1 \tag{9}$$

For the login and verification phase, we have used the advantages of Kerberos protocol to avoid this attack, adding another level of protection, which even if the intruder could get the ticket that enables him to access the HDFS server, he will be not able to provide the server by the correct answer, thus the server will reject him directly .

3) Brute-force and dictionary attacks: the attacker tries to attain the password or any personal information, which may enable the intruder to get user's information on the channel to be stored in the database of KDC server. In the proposed registration phase, we send an exponential value over the channel to get the common key A and B, where they are shown as eq10, and eq11.

$$A = g^{a} \tag{10}$$

$$B = g + g^{b} \tag{11}$$

Thus, brute-force and dictionary attacks will be avoided and will be very costly and time waster. The same in the proposed login and registration phase where the effect of salt in eq12 used to disrupt the password and help to create a most reliable login and verification phases, therefore, listening to requests pass between the user and the KDC or by brute force to guess the original password is almost impossible.

$$K_t = Hpw|salt) \tag{12}$$

Also, dictionary attacks will be avoided too. As the proposed phase used Kerberos, and it established to hidden password. Thus the proposed principal minimizes the probability of finding the password that caused by disturbance by adding dynamic salt per session and used of virtualization function, therefore even if capturing several messages by intruders; they will not have the opportunity to get the password.

4) Man in the middle: In the man in the middle attack, an eavesdropper tries to impersonate either the client or server and thus change the message passed or attain valuable information. In the proposed registration phase, the eavesdropper in the KDC can intercept A and B only which shown in eq13 and eq14 as it sent over the channel. However, these values are not enough yet to calculate the session key "S," which calculate it as eq15, and eq16. For the both side respectively, because the value of "a" on the user-side is not known and to get it we have to calculate mathematical formulae shown in eq17, that include "AA" which need the value "b" to be calculated as shown in eq18, "b" generate randomly in the server-side. Moreover, "a, b" is set and well known to the user-side and server-side respectively and never shared through the channel.

$$A = g^{a} \tag{13}$$

$$B = g + g^{b} \tag{14}$$

$$S = (B - g)^{2a} \tag{15}$$

$$S = (Ag^{a})^{b} \tag{16}$$

$$a = PWW^{AA} \, mod \, P \tag{17}$$

$$AA = h(ID \,|\, b) \tag{18}$$

Moreover, Kerberos applied in login and verification phase use tickets that contain the authentication information of its holder and are encrypted using the key of the final recipient. Therefore, no one knows any knowledge about the ticket content nor modify it.

5) Insider attack: An insider attack is a malicious attack perpetrated on a network or computer system by a person with authorized system access [25]. Where in the proposed registration phase the value of "AA" is protected by secure one way hash function shown in eq18, and any modification to these value will be detected between the user and server, where it will result in a different value of the formula "S" in both side which Table 1 describes. Moreover, b is a random value in KDC-side.

TABLE I
THE VALUE OF THE FORMULA "S"

User	KDC
$S = (B - g)^{2a}$ $S = (g + g\text{^}b - g)^{2a}$ $S = (g^{b})^{2a}$	$S = (Ag^{a})^{b}$ $S = (g^{a}g^{a})^{b}$ $S = (g^{b})^{2a}$

For the proposed login and verification phase, the KDC will ask the user to answer his secret question that he already answered in the phase of registration to verify his identity if someone could penetrate the protocol to impersonate his personality. Since in this case, it will not be useful, because even if an intruder got the ticket that enables him to access the HDFS server, he would not be able to answer the question, so the HDFS server will reject him directly. Thus insider attack will be avoided.

6) User anonymity: Suppose that the attacker has intercepted user-side authentication messages such as "AA, A." Thus, the adversary may try to retrieve any static parameter from these massages. However, "AA, A" all are not have the same value in each session, but their value will be vary depending on the random generator value given on the KDC-side "b," as shown in eq18, eq19, eq20. , the random "b" never share over the channel. Thus, the proposed scheme of the proposed registration phase can overcome the security flow of user anonymity breach.

$$A = g^{a} \tag{19}$$

$$a = PWW^{AA} \, mod \, P \tag{20}$$

$$AA = h(ID \,|\, b) \tag{21}$$

Moreover, the effect of salt used to disrupt the password helps to create a most reliable login, and verification phases provided resistance to the user anonymity attack.

7) Impersonation attack: The proposed registration phase provide a secure and strong authentication protocol to supply resistances to attacks over a secure exchange on the channel it minimizes the guessing likelihood of parameters reciprocation between client and HDFS server. What makes the impersonation attack is a week to be active during its steps.

For the login and registration phase, also resistance to such attack by using a most trust Kerberos protocol by adding an additional level, for the purpose of user protection against impersonation attack, or real impersonate to the user's personality, by check the presence of the real user, where the KDC will retrieve the user's question from the database. Then submit it to the user in the encrypted form of eq21 to verify his credibility. Whether he who asking the access or another person impersonate him and wait to correct user answer to submit it back in the encrypted form of eq22. After the HDFS server checks the answer to be valid, then allow him to access, unless it will be rejected directly from the HDFS server itself.

$$Q_{S} = S \oplus Q \tag{22}$$

$$ANS_{S} = S \oplus f(ANS) \tag{23}$$

Table.2 shows the security features of the proposed scheme.

TABLE II
SECURITY FEATURES COMPARISON

Security analysis	Avoided
Password guessing attack	Yes
Replay attack	Yes
Man in the middle attack	Yes
brute-force	Yes
dictionary attacks	Yes
insider	Yes
impersonation	Yes
User anonymity	Yes

B. Performance Analysis

In this section, we compare the performance and security features of the proposed schemes of registration and login and verification phases with that of related schemes, which applied in the area of security of big data in Hadoop distributed file system. To prove the efficiency of the proposed scheme, we discuss the performance analysis by computing the cost of computation, communication, and storage overhead. Comparisons with some related schemes are also provided in this section. The identity ID, Password pw, salt, secret one-way hash function, visualization function, ticket, answer all recommended to be 128-bit long, while the random number, question, and the encrypted session key all 1024-bit long, other value as time stamp is of 32-bit long. The meaning of notations used in the comparison is given in table3.

TABLE III
THE NOTATION USED IN COMPUTATION COST COMPARISON

Symbols	Notation for
T_H	Hash function
T_E	Exponential operation
T_S	Symmetric key encryption/decryption

1) Computation cost: Table 4 shows the composition results of the computation cost of the proposed method over the compared methods for the registration phase, whereas table 5 show the result of the login and verification phase.

TABLE IV
THE COMPUTATION COST OF THE REGISTRATION PHASE

Our method	Jeong and Kim [10]	Rahul and GireeshKumar [18]
$2T_S + 3T_H + T_E$	$2T_H + T_E$	$4T_S + T_H$

TABLE V
THE COMPUTATION COST OF LOGIN AND VERIFICATION PHASE

Our method	Jeong and Kim [10]	Rahul and GireeshKumar [18]
$8T_S + 2T_H$	$5T_S + 6T_H + 5T$	$21T_S$

During the registration phase, the total computation cost of the user and the server is $2T_S + 3T_H + T_E$, which consider something less efficiency. Hence, the steps of the proposed scheme phase must be done accurately and with a complicated way to be something difficult to decipher or penetrate. It is sensitive data and the stage of communication will depend on it. Thus, it must be stored safely. Nevertheless, that will not affect the proposed work, because this phase will run only once for a single user only during his registration to the KDC server. In contrast, in the proposed login and verification phase, the total computation cost of the user and server is $8T_S + 2T_H$. Where the proposed scheme here is more efficient than Jeong and Kim [10] by the symbols used as well as efficient than Rahul and GireeshKumar [18] which need 21 crucial symmetric encryption.

2) Communication cost: To compute the cost of communication we mainly focus on the efficiency of login and verification phases since these phases are the main body of an authentication scheme and are executed much more frequently than the other phases. In the proposed scheme the communication overhead includes the capacity of the transmitted message involved in the authentication process, which is 5216 = (160×7 + 1024 ×4) bit. It is considered the best compared to some previous works. Table6 shows the cost of communication for each of [10] and [18] calculated in the same way and assuming that the variables in each of them have the same value.

TABLE VI
COMMUNICATION COST OF LOGIN AND VERIFICATION PHASE

Our method	Jeong and Kim [10]	Rahul and GireeshKumar [18]
5216	9280	6912

3) Storage overhead cost: In the proposed schemes, the parameters {ID, f(pw), Q, f(ANS)} are stored in the database of the KDC server. Thus the storage cost is 1504 = (3×160 + 1×1024)bit. Table7 provides the comparison results of the proposed method over the related scheme [10], [18]. It shows that the storage cost of the proposed method is the lowest and best.

TABLE VII
STORAGE COST OF LOGIN AND VERIFICATION PHASE

Our method	Jeong and Kim [10]	Rahul and GireeshKumar [18]
1504	6112	3392

C. Representation of the Result

Tables 6 and 7 above are graphically drawn in the Fig.5 and Fig.6 sequentially to show the difference of values of each work clearly.

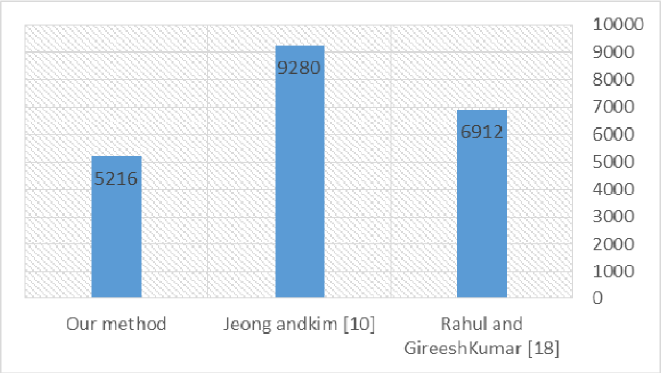

Fig. 5 communication cost of login and verification phase

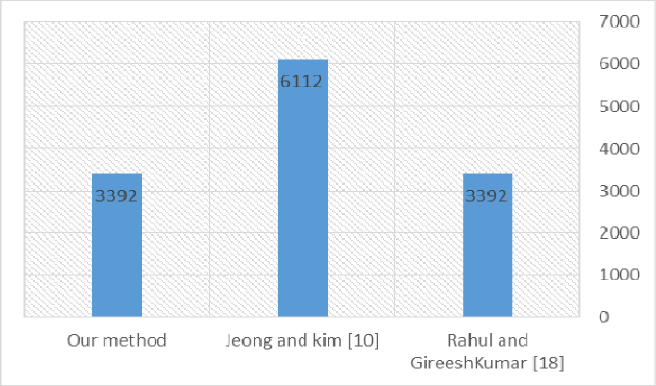

Fig. 6 Storage cost of login and verification phase

The results were gained based on the performance and security features analysis rule on the proposed schemes of registration and login and verification phases to get the

Computation cost, Communication cost and Storage cost by the same way used in Ding [26].

IV. Conclusions

We have suggested a new scheme using Kerberos with a static knowledge-based authentication to authenticate users in Hadoop distributed platform. It works in two phases, first a serious salt and strict way to ensure the registration phase, which allows the users to store their information by a robust authentication way providing resistance for many attacks. Second the login and verification phase that use registered users' information to get access to HDFS server that strives to provide a stronger and complex access mechanism.

In this work, we have added a new protection level to leverage Kerberos, where made it more resistant to personality impersonation in case of capturing messages that allow an eavesdropper to get the ticket that will enable him accessing the HDFS server. We have shown that the scheme can handle password guessing, replay, brute force, dictionary, man-in-middle, user anonymity, and impersonation and insider attacks with less computation, communication, and storage cost.

References

[1] Chandra, Sudipta, Soumya Ray, and R. T. Goswami. "Big Data Security: Survey on Frameworks and Algorithms." In Advance Computing Conference (IACC), 2017 IEEE 7th International, pp. 48-54. IEEE, 2017.

[2] Ouda, Abdelkader. "A Framework for next generation user authentication." In Big Data and Smart City (ICBDSC), 2016 3rd MEC International Conference on, pp. 1-4. IEEE, 2016.

[3] Dean, Jeffrey, and Sanjay Ghemawat. "MapReduce: simplified data processing on large clusters." Communications of the ACM 51, no. 1 (2008): 107-113.

[4] Hong, Jinkeun. "Kerberos Authentication Deployment Policy of US in Big data Environment." Journal of Digital Convergence11, no. 11 (2013): 435-441

[5] Jeong, Yoon-Su, and Kun-Hee Han. "Service management scheme using security identification information adopt to big data environment." Journal of Digital Convergence 11, no. 12 (2013): 393-399.

[6] Lee, Seong-Hoon, and Dong-Woo Lee. "Current status of big data utilization." Journal of Digital Convergence 11, no. 2 (2013): 229-233.

[7] Vatamanu, Cristina, Dragoş Gavriluţ, and Răzvan-Mihai Benchea. "Building a practical and reliable classifier for malware detection." Journal of Computer Virology and Hacking Techniques 9, no. 4 (2013): 205-214.

[8] Jeong, Yoon-Su, and Yong-Tae Kim. "A token-based authentication security scheme for the Hadoop distributed file system using elliptic curve cryptography." Journal of Computer Virology and Hacking Techniques 11, no. 3 (2015): 137-142.

[9] Terzi, Duygu Sinanc, Ramazan Terzi, and Seref Sagiroglu. "A survey on security and privacy issues in big data." In Internet Technology and Secured Transactions (ICITST), 2015 10th International Conference for, pp. 202-207. IEEE, 2015.

[10] Wu, Dapeng, Boran Yang, and Ruyan Wang. "Scalable privacy-preserving big data aggregation mechanism." Digital Communications and Networks 2, no. 3 (2016): 122-129.

[11] Li, Ruidong, Hitoshi Asaeda, Jie Li, and Xiaoming Fu. "A Verifiable and Flexible Data Sharing mechanism for Information-Centric IoT." In Communications (ICC), 2017 IEEE International Conference on, pp. 1-7. IEEE, 2017.

[12] Li, Ruidong, Hitoshi Asaeda, Jie Li, and Xiaoming Fu. "A distributed authentication and authorization scheme for in-network big data sharing." Digital Communications and Networks 3, no. 4 (2017): 226-235.

[13] Wang, Kun, Jiahui Yu, Xiulong Liu, and Song Guo. "A pre-authentication approach to proxy re-encryption in the big data context." IEEE Transactions on Big Data (2017).

[14] Shen, J., Liu, D., Liu, Q., Sun, X. and Zhang, Y., 2017. Secure authentication in cloud big data with hierarchical attribute authorization structure. IEEE Transactions on Big Data, (1), pp.1-1.

[15] Ibrahim, Anas, and Abdelkader Ouda. "Innovative data authentication model." In Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2016 IEEE 7th Annual, pp. 1-7. IEEE, 2016.

[16] Abdullah, Nazri, Anne Hakansson, and Esmiralda Moradian. "Blockchain based approach to enhance big data authentication in a distributed environment." In Ubiquitous and Future Networks (ICUFN), 2017 Ninth International Conference on, pp. 887-892. IEEE, 2017.

[17] Bos, Joppe W., J. Alex Halderman, Nadia Heninger, Jonathan Moore, Michael Naehrig, and Eric Wustrow. "Elliptic curve cryptography in practice." In International Conference on Financial Cryptography and Data Security, pp. 157-175. Springer, Berlin, Heidelberg, 2014.

[18] P.K. Rahul and T. GireeshKumar "A Novel Authentication Framework for Hadoop" Advances in Intelligent Systems and Computing 324, Proceedings of ICAEES 2014, volume1, Springer.

[19] Grover, Chandni, and Manpreet Kaur Aulakh. "Big Data Authentication and Authorization in HDP (Hadoop Distributed platform) using Kerberos and Ranger." International conference on recent innovation in management and engineering, 24-June 2017.

[20] Kohl, John, and Clifford Neuman. The Kerberos network authentication service (V5). No. RFC 1510. 1993.

[21] Rathore, Romendrapal Singh, B. L. Pal, and Shiv Kumar. "Analysis and Improvement in Kerberos 5." (2015).

[22] Krishnamurthy, Anush. "Performance Impact of Encryption Algorithms on Kerberos Authentication Protocol." Ph.D. diss., Oklahoma State University, 2006.

[23] Knowledge-based authentication, Wikipedia website, https://en.wikipedia.org/wiki/Knowledge-based_authentication

[24] Tbatou, Zakariae, Ahmed Asimi, Younes Asimi, Yassine Sadqi, and Azidine Guezzaz. "A New Mutuel Kerberos Authentication Protocol for Distributed Systems." IJ Network Security 19, no. 6 (2017): 889-898.

[25] Insider attack, techopedia, https://www.techopedia.com/definition/26217/insider-attack.

[26] Ding, W. A. N. G. "Cryptanalysis and security enhancement of a remote user authentication scheme using smart cards." The Journal of China Universities of Posts and Telecommunications 19, no. 5 (2012): 104-114.