ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 9, September 2015

Data Security Accessing for Hadoop based on Masked CP-ABE

Suparna Gaur

Student, Dept. of CSE, UIET, Kurukshetra University, Kurukshetra, Haryana, India

ABSTRACT: The hadoop platform of cloud computing stores and processes massive data sets (big data) arriving from various sources hence security of these data sets is the issue of most concern in today's time so as to protect them from malicious attacks and unauthorized access. In the existing CP-ABE scheme for hadoop system, the types of attributes and access control policy has not been specified. We have proposed a Masked CP-ABE scheme in which access control is provided in the form of masking and then matching is done between user key and file key mask to obtain a matchfactor which is then compared with a threshold value. This masking and matching grant the permissions to various users for decrypting and accessing hadoop dataset files on the basis of their levels obtained after comparison with the threshold value.

KEYWORDS: CP-ABE, Encryption, Hadoop, Masking, Security.

I. INTRODUCTION

Hadoop is an open source framework that is based on Java programming language.It allows the storing, managing and processing of large scale datasets under the distributed environment efficiently[1]. This is the project of Apache and managed by the same software company , that is the reason it is referred as "Apache Hadoop". Hadoop has the capability of scaling up from single server to lots of machines, each having its own storage and computation[2]. It can analyse both structured and unstructured data quickly and reliably.This framework is mainly used by Google, IBM,Yahoo and Facebook [1][2].

The two layers of hadoop are:

- **Computation Layer:** This layer uses Map Reduce as its framework for providing computational capabilities.
- **Distributed Storage Layer:** In this layer,Hadoop distributed file system (HDFS) provides storage .

Architecture of Hadoop

Hadoop follows a master-slave architecture for both the tasks, whether distributed storage or distributed processing. It has two main components. For distributed storage, HDFS i.e hadoop distributed file system and for distributed processing, MapReduce paradigm has been provided by Hadoop [6]. Other components are Yarn and Hadoop Common The architecture is shown in the figure below.

- **HDFS:** It has been designed to manage large files and run on clusters of commodity hardware. It consists of a master node called NameNode and one or more slave nodes called DataNode. Its primary goal is to allow the system to continue the operation and preserve the data reliably in the case of any failure [3]. Namenode performs operations on files like open, rename, close etc. Datanodes store blocks which contain splitted files and create, delete, replicate blocks when namenode instructs [6][7][8].
- **Map Reduce:** It has a master node called Job Tracker and some slave nodes called Task Tracker. Request submitted by client is called job and execution of that job is done as map reduce task. It has map function and reduce function and breaks the client's request into small tasks and process them parallely. It accepts the request from users, splits it into small tasks controls, monitors and distributes these tasks to task tracker nodes. Task tracker performs map-reduce tasks [6][7][8].

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 9, September 2015

- **Yarn:** Yet Another Resource Negotiator,This is one of the main features in second generation hadoop [4].Resources like CPU time, storage and memory are assigned to the applications running on hadoop by Yarn. Through this, multiple applications in hadoop can share a common resource management.[5]

- **Hadoop Common:** The set of utilities and libraries whose work is to provide support to rest all hadoop modules[1], [2].

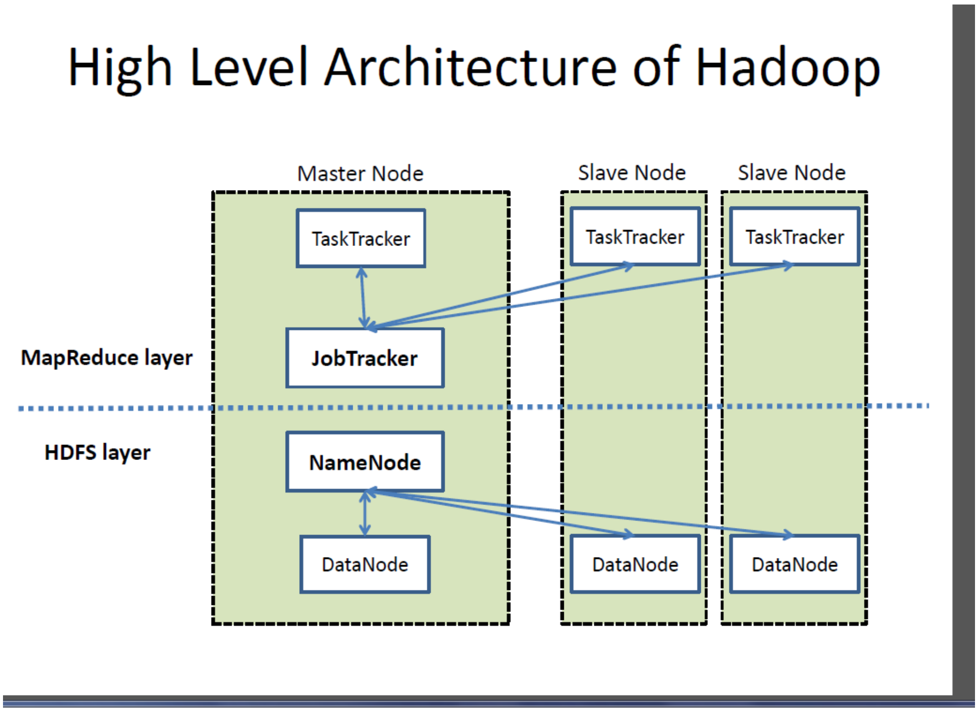


Figure 1: Architecture of Hadoop system [9].

Hadoop Security Challenges

- Data which is at rest on hdfs is not encrypted by hadoop at present. Organizations which are very strict for the encryption of their data residing hdfs use third party tools for ensuring data security.
- Hadoop’s authentication for security totally depends on kerberos. Organizations which some other methods or protocols for security, need to set up their personal authentication system in the enterprise.
- Most of the organisations use access control policies based on XACML and attribute based access control.This is because authorisation performed by hadoop is not sufficient.
- Authentication in hadoop requires kerberos, delegation token, block token etc and SSL, encryption of transfer of hdfs data etc for network encryption. There all types of encryption methods are required to be installed seperately increasing complexity[10].

CP-ABE

The algorithm of CP-ABE scheme consists of four steps:

- Setup: This algorithm takes as input security parameters that are implicit. It provides a master key (MK) and public key (PK) as output.
- Encrypt: The input to this algorithm is PK, M-message and A-access structure which is defined over a universe of attributes. Message M will be encrypted and converted to a ciphertext (CT).

- KeyGen: The input to this algorithm is MK and a set of attributes S that defines this key. The output of this algorithm is SK, which is a private key.
- Decrypt: The input to this algorithm is PK, CT, SK. The ciphertext will be decrypted to the original message M only when the attribute set S satisfies the access structure A in the ciphertext [11].

DES

Data Encryption Standard is a method of encrypting electronic data. It is based on symmetric key encryption as the sender and receiver both use same key for encryption and decryption so it is must for both of them to know the private key used in the process. It has been developed at IBM in 1975.

II. RELATED WORK

John Bethencourt *et al.* [11] stated that in most of the distributed system environment, only users with some particular credentials or attributes have the right to access the data. At present, these kind of policies could only be incorporated by storing the data in a server that is trusted and mediating the access control. But if this server is compromised anyhow, then the confidentiality of the whole data which is stored in this server is completely at the risk. This paper has presented a system that protects the encrypted data by applying complex access control policies over it and this access control policy is called as CP-ABE scheme. Not only this method protects the data from collusion attacks but also ensures to maintain the confidentiality of the encrypted data if in case the server is compromised. Existing ABE schemes define access policies and cipher texts based on the attributes of users but in the proposed scheme, credentials of the users are based on attributes and the one who encrypts the data defines the access policy to decrypt the data. Proposed method is quite closer to traditional access control methods. **Yanli Ren** *et al.* [12] stated that in the existing CP-ABE (CipherText Policy Attribute Based Encryption) scheme, encryptor assigns an access policy over the ciphertext and also describes who can decrypt the message into plaintext in the encryption algorithm. Mostly, ciphertext’s size is not constant in CP-ABE schemes rather the size is linearly dependent on the number of attributes used in defining the access policy for that ciphertext. Selective secure without random oracles is the one and only CP-ABE scheme whose size is constant. This paper has constructed a CP-ABE scheme whose size is constant and that provides complete security that too without random oracles. Threshold decryption policies based on IBE (Identity based encryption) scheme has been admitted by the proposed scheme. **Chao YANG** *et al.* [13] stated that the security model of hadoop and HDFS is weak as interaction among data nodes is not yet encrypted and data stored in HDFS is in clear text. The triple encryption scheme for ensuring hadoop security is as follows, encryption of HDFS files using DEA (Data Encryption Algorithm), encryption of data key using RSA and encryption of RSA private key of user using IDEA (International Data Encryption Algorithm). Hybrid encryption is used to encrypt the files residing in HDFS. To encrypt files and get data key, DES algorithm is used by hybrid encryption and then to encrypt data key, RSA algorithm is used. To decrypt the data key, private key is kept by the user. So, this triple encryption scheme was founded to be feasible in providing confidentiality to HDFS data but it was lacking somewhat in performance. **Zhiqian Xu** *et al.* [14] stated that cloud stores a lot of data and protection of that data is one of the significant challenges in cloud scenario. In order to protect the data in untrusted and unreliable environments, ABE (Attribute Based Encryption) scheme is found to be useful mechanism for providing security. Through ABE, not only fine grained access control can be applied over encrypted data stored in cloud but also provides data owners with a mechanism to define an access policy over the encrypted data. In this paper, generic practical deployment framework has been proposed that could be applied over ABE scheme so as to incorporate some practical features like key escrow avoidance, user revocation and key refreshing etc, which are currently not incorporated in existing ABE schemes. Proposed scheme is efficient , scalable and flexible as it can be applied to any ABE scheme with no modification. Moreover, when any user is revoked, this framework doesn’t require key or ciphertext to be generated and encrypted again respectively. **Huixiang Zhou** *et al.* [15] They described CP-ABE scheme for access control because the traditional schemes like PKI and IBE requires all relevant information of user to be sent to resource provider hence damaging user’s privacy, also more bandwidth and processing overhead is required . CP-ABE (Ciphertext policy attribute based encryption) is better as it identifies user using a collection of properties or multiple attributes. Though CP-ABE provides good security and reliability, yet its efficiency is still need to be improved. **Masoumeh RezaeiJam** *et al.* [16] stated that international IT universe is currently focussed upon the core technology of cloud computing that is security and privacy of data and services and trust based computing. These issues are biggest challenges in today’s scenario. Cloud computing and big data’s open source framework that is hadoop is the latest one and is widely used in business environment. But the main challenge in its development is the poor security mechanism. It is used for distributed processing and distributed storage and it is resilient that is it continue to operate even in case of any node failure and it makes 3 duplicate copies of the data on various nodes so as to recover if any node fails. It is used by companies like Amazon, Yahoo, IBM and facebook etc. Hadoop in its default state is not protected by any encryption mechanism, no security model, no authentication, data stored over the network is not encrypted and it pre-assumes network as already trusted. For addressing these issues present security mechanisms are as follows, Kerberos protocol for authentication firewalls for perimeter level security, HDFS permissions for authorization. Security leak in hadoop after these mechanisms is like submitting and deleting of job with the privileges of valid users, accessing and modification of data by hackers, impersonation by malicious users etc. These problems is avoided in this paper by implementing the following, Apache sentry for providing access control, triple encryption of data by RSA, DES, IDEA algorithms, designing a secure file system based on fully homomorphic encryption. At the file system level, security of hadoop is claimed by the paper but lacking in granular support for securing access to data. **Mounira Msahli** *et al.* [17] proposed a new access control paradigm using the graph transformation called Profile Centric Modelling (PCM). This model used obligation, condition and authorization and it could be generalized by using more access specifications like role etc. to define the elementary profile which was the basic element of the PCM. The model used graph formalism and its implementation using matrix. The profile was defined as the combination of all possible authorization, obligation, condition, role, etc. Profile centric modelling was an optimum paradigm to define access control policy in complex distributed and elastic system like cloud computing. The proposed scheme was verified and implemented over Hadoop distributed file system in the context of Safe Box as a service. The proposed framework was suitable for the Cloud constraints like distributed environment, complexity and decentralization, web based applications and also provided elasticity by using it in Safe Box as a service.

III. PROBLEM FORMULATED

John Bethencourt [11] and Huixiang zhou [15] proposed a CP-ABE scheme in order to secure the data stored on HDFS by applying access control even if the server storing the data is untrusted and also to prevent the data from collusion attacks too. This scheme uses multiple attributes to identify any user rather than using only identity information.

The base researches neither specify the types of attributes to compare and the method to compare user attributes with file attributes nor even define the access policy to be applied over the encrypted data so as to define which user is given what permission to access the data.

IV. PROPOSED METHODOLOGY

To address the above mentioned issues M-CP-ABE (Masked CP-ABE) scheme has been proposed in the paper which further extends the base algorithm CP-ABE by specifying the types of attributes and specified an access policy in the form of masking to be applied over the encrypted data to provide the access permission to various

users to decrypt and access dataset files of hadoop on the basis of their matchfactor obtained. The pseudocode is mentioned below:

Pseudocode

Input: File key, User key, File Mask.

Output: Access Permissions granted.

```
Step1: Calculate the length of file key, initialize temp and match variable to 0, repeat steps3 & step4 upto the key length calculated.
Step2: If file key contains 1 at any position then check the value of user key and file mask at the same position and increment the temp variable by 1. If user key belongs to or equal to file mask then increment match variable by 1.
Step3:  Calculate matchfactor = match/temp. Compare this matchfactor with a threshold value say 0.6.
Step4: If matchfactor<=0.6 then access is denied for that user.
Step5: If matchfactor lies between 0.6 & 0.8 then only read access is granted to the user.
Step6: If matchfactor>=0.8 then user is given the key to decrypt the file and read/write access is granted.
Step7: Return.
```

Complexity: Complexity of matchfactor is O (n^2).

Generation of File Key and Key Mask

Suppose we have a file named ABE1.txt and the attributes on which the file key is based are (id, usertype, location, age, gender). We generate a random binary string in which one's shows relevant attributes and zero's shows ignorable attributes like, 10110 i.e id, location and age must be present, rest are ignorable. Key mask specifies the possible values that relevant attributes can take. Like, id= {100 to 110}, location= {chd, kkr, ggn, mrt}, age= {30 to 40}. So, 10110 & (105)| (3)| (kkr)| (35)| (0) is valid file key and key mask respectively.

Encryption Time: The time taken by the DES algorithm to encrypt the hadoop splitted dataset files into ciphertext using a key. It is calculated in milliseconds.

Encryption Time= System current time before encryption – System current time after encryption.

Decryption Time: The time taken by the DES algorithm to decrypt the hadoop splitted dataset files back into the plain text using the same key. It is calculated in milliseconds.

Decryption Time= System current time before decryption – System current time after decryption.

Framework of Proposed Method

Hadoop sample dataset is splitted into some files and they are encrypted using DES. Each file has been given a unique key and a key mask. Extract the file key and key mask for that file and also extract the user key of the user who wants to access the file. Match the user key and key mask to obtain the matchfactor. This matchfactor is used to decide what permission is granted to the user. Only the users having matchfactor above the threshold value are given the key to decrypt and access the file. Framework is shown in the figure below:

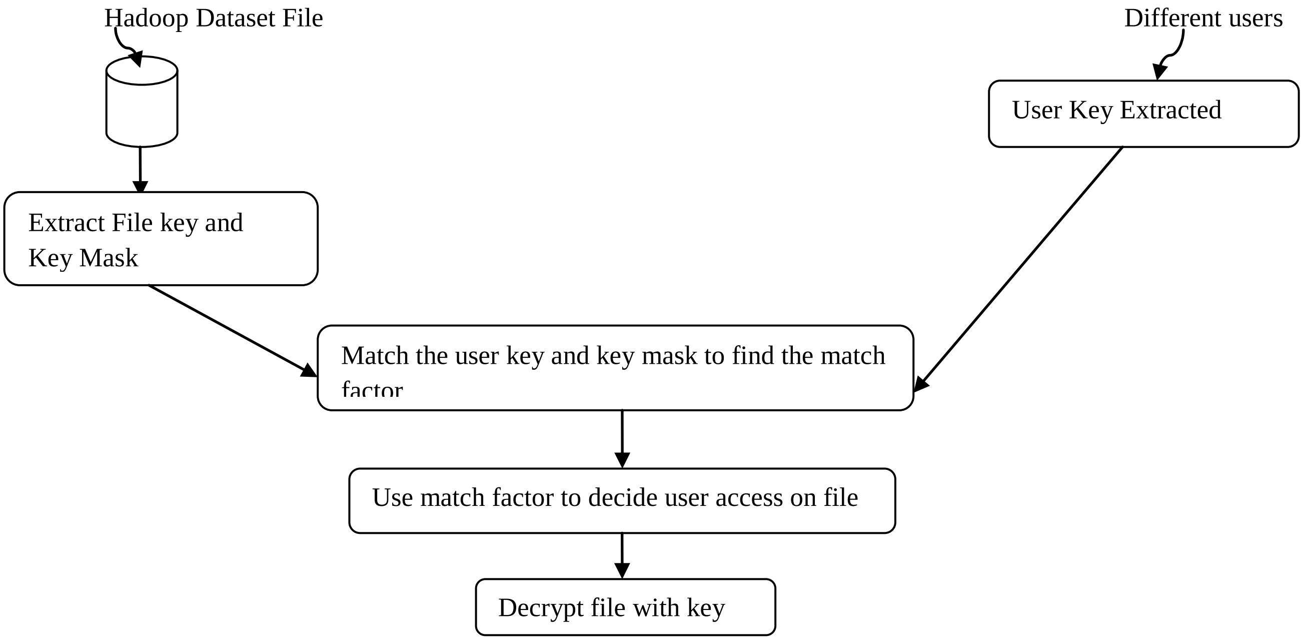

Hadoop Dataset File
Different users
User Key Extracted
Extract File key and Key Mask
Match the user key and key mask to find the match factor
Use match factor to decide user access on file
Decrypt file with key

Figure 2: Architecture of M-CP-ABE

DOI: 10.15680/IJIRCCE.2015. 0309115

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 9, September 2015

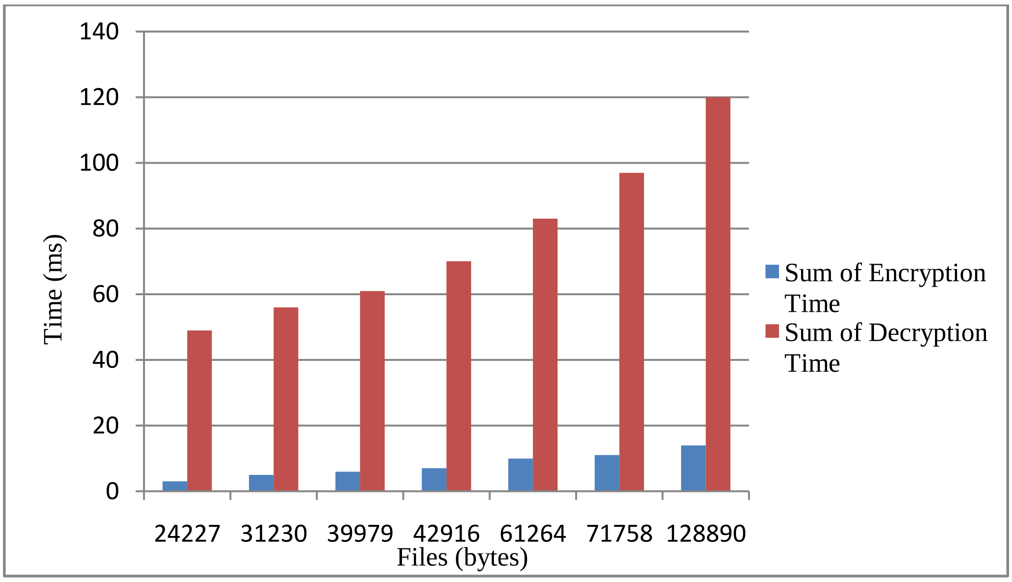
V. EXPERIMENTAL RESULTS

Results for performance analysis are shown in tabular format as well as in the graphical output too. These results are obtained by implementing the algorithm on ubuntu 12.04 in the language java version 7 with eclipse kepler IDE and hadoop 2.5.2 version installed on the operating system. Tha hadoop sample dataset is splitted into some files and the encryption and decryption time for each of the file using DES algorithm is shown in the table below.

S.No.	File Size	Encryption Time	Decryption Time
1	24227	3	49
2	31230	5	56
3	39979	6	61
4	42916	7	70
5	61264	10	83
6	71758	11	97
7	128890	14	120

Table 1: Encryption and Decryption time (ms).

This graph shown below is plotted between various files of hadoop sample dataset and encryption and decryption time for each of the splitted files according to their size. Graph shows that encryption and decryption time of files increases as the size of files increases and also that more time is spent in decryption of the files rather than encryption. The hadoop dataset used for implementing the algorithm is Google ngram viewer (Web Data)[18].



Graph 1: Encryption and Decryption time for each file.

DOI: 10.15680/IJIRCCE.2015. 0309115

VI. CONCLUSION AND FUTURE SCOPE

This paper has focused upon the security leaks associated with the hadoop system and proposed a method to overcome them in order to improve the security. The base researches don't specify the types of attributes and the access control policy in applying CP-ABE on hadoop system. In order to resolve these issues, Masked CP-ABE scheme has been proposed in the paper that specified the types of attributes and provided the access policy in the

form masking concept which further decides what permission to access the file is granted to a particular user on the basis of his matchfactor obtained. It is graphically demonstrated that encryption and decryption time of dataset files used in the Masked CP-ABE scheme is directly proportional to the file size. The proposed concept addressed the mentioned issues with more flexibility and provided enhancement in the security parameter of the hadoop system.

The concept of Masked CP-ABE described in the dissertation is not restricted to hadoop environment only, rather it can be used on all previously defined applications for ABE, because this method provides more flexibility than previous ABE specifications.

REFERENCES

[1]. Wikipedia "Hadoop definition and Hadoop Common", http://en.wikipedia.org/wiki/Apache_Hadoop. Accessed on 03-November-2014.

[2]. Apache "Hadoop definition and Hadoop Common", http://hadoop.apache.org/.Accessed on 03-November- 2014.

[3]. Wikipedia "History of Hadoop and HDFS", http://en.wikipedia.org/wiki/Apache_Hadoop. Accessed on 08-December-2014.

[4]. Search Data Management "Hadoop Yarn",http://searchdatamanagement.techtarget.com/definition/Apache-Hadoop-YARN-Yet-Another-Resource-Negotiator. Accessed on 04-November-2014.

[5]. IBM "Hadoop Yarn and Key features of Hadoop", http://www-01.ibm.com/software/data/infosphere/hadoop/. Accessed on 05-November- 2014.

[6]. Wikispaces "Hadoop Architecture", http://hadooptutorial.wikispaces.com/Hadoop+architecture.Accessed on 03-February-2015.

[7]. Aosa Book "Hadoop Architecture", http://www.aosabook.org/en/hdfs.html.Accessed on 03-February-2015.

[8]. Apache "Hadoop Architecture", http://hadoop.apache.org/docs/r1.2.1/hdfs_design.html.Accessed on 03-February-2015.

[9]. Google "Architecture of cloud computing and Architecture of Hadoop System", https://www.google.co.in/. Accessed on 26-November-2014 and Accessed on 03-February-2015.

[10]. Apache "Hadoop Security Challenges",https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/SecureMode.html . Accessed on 15-july-2015.

[11]. John Bethencourt, Amit Sahai, Brent Waters, "Ciphertext-Policy Attribute-Based Encryption", IEEE Symposium on Security and Privacy, pp: 321-334, May 2007.

[12]. Yanli Ren, Shuozhong Wang, Xinpeng Zhang, Zhenxing Qian, "Fully Secure Ciphertext-Policy Attribute-Based Encryption with Constant Size Ciphertext", Third International Conference on Multimedia Information Networking and Security, IEEE, pp: 380-384, November 2011.

[13]. Chao YANG, Weiwei LIN*, Mingqi LIU, "A Novel Triple Encryption Scheme for Hadoop-based Cloud Data Security", Fourth International Conference on Emerging Intelligent Data and Web Technologies,IEEE, pp: 437-442, September 2013.

[14]. Zhiqian Xu, Keith M. Martin, "A Practical Deployment Framework for Use of Attribute Based Encryption in Data Protection", IEEE, pp: 1593-1598, 2013.

[15]. Huixiang Zhou, Qiaoyan Wen, "A New Solution of Data Security Accessing for Hadoop Based on CP-ABE", IEEE, pp: 525-528, 2014.

[16]. Masoumeh RezaeiJam, Leili Mohammad Khanli, Mohammad Kazem Akbari, Morteza Sargolzaei Javan, "A Survey on Security of Hadoop", IEEE, pp: 716-721, 2014.

[17]. Mounira Msahli Xiuzhen Chen Ahmed Serhrouchni, "Towards a fine-grained access control for Cloud", 11th International Conference on e-Business Engineering, IEEE, pp: 286-291, 2014.

[18]. Googleapis"Google ngram viewer", http://storage.googleapis.com/books/ngrams/books/datasetsv2.htmlAccessed 15-July-2015.

BIOGRAPHY

Suparna Gaur is a research scholar pursuing M.Tech (2013-2015) in Software Engineering, Computer Science & Engineering Department, University Institute of Engineering and Technology, Kurukshetra University, Kurukshetra, Haryana, India. Her interest areas are Cloud Computing, Big Data, Hadoop.