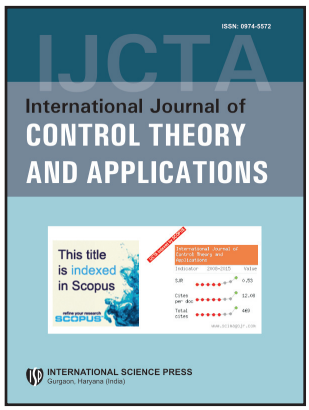

International Journal of Control Theory and Applications

ISSN : 0974-5572

Volume 9 • Number 44 • 2016

Hadoop Used Medical Analytics and Privacy Preservation

Balaji K. Bodkhe[a] and Sanjay P. Sood[b]

[a]*DPTU, Jalandhar, Punjab, India. Email: balajibodkheptu@gmail.com*
[b]*CDAC, Mohali. Email: spsood@gmail.com*

Abstract: A lot of progress is being made in Medical Science to push for a better human healthcare system, high-level research and experiments are being conducted all over the globe for the same cause. Information Technology has been one of the most powerful force to help human healthcare system bloom. It has provided a wide range of technologies which have been used directly as well as indirectly by the Medicial Science industry. According to recent surveys, the data is being generated at a rate of more than 100 Terabytes a day, which is humongous. With the widespread of computing services all over the world and increase in technological dependency, medical data serves as a huge contributor to Big Data. As the data generation in the healthcare field is increasing, privacy preservation of this data presents itself as a challenge, hence, the need to protect the identity of a person is becoming more critical. This paper is a survey for implementing privacy preservation of Big Data using Hadoop Analytic Tools. This is a survey regarding anonymization techniques used in data processing phase. It also talks about *k*-anonymity and *l*-diversity and their respective attacks. We use HBase to store the data in the Hadoop framework. Hive is a querying language that is used for data aggregation and data visualisation. Pig is a scripting language that is used to process unstructured data such as images and videos in the database. The paper proposes to use the Hadoop ecosystem along with its tools, Apache Pig and Hive to store the healthcare data and use it for access and analytics.

Keyword: Big Data, Hadoop, Analytics, MapReduce, HealthCare. Generalization, K-anonymity.

1. INTRODUCTION

It is known that data is of two type, the one which resides in OLTP (Online Transactional Processing) and OLAP (Online Analytical Processing). This data which is stored in OLAP consists of data whose size is more than a TB, which is structured, unstructured and from multiple sources. This data overall converges to something also called as Big Data.

According to Interactive Data Corporation (IDC) health insights study, the worldwide health care data is growing tremendously. From a mere 500 petabytes in 2012, it has been predicted to reach a humongous Figure of 25,000 petabytes by the year of 2020. To process such large amount of data using traditional RDBMS methods, it would require hours or sometimes even days to access this data and then analyze or perform operations on it. This data includes the Electronic Health Record (EHR) and medical data of the human population of this

world. As the size of this data is extremely high it is also called as Big Data. This medical Big Data is extremely structured, unstructured, semi-structured whilst it is also volatile and of multiple formats such as text, images, audio and videos. Big Data can be characterized to some extent using the 3 main 'V's, i.e. Volume, Velocity and Variety. Volume indicating the size, Velocity describing the speed at which this data is being generated and Variety putting light on the various sources of Big data.

Privacy Preservation has become one of the leading aspects in Big Data Analytics. It plays an important role in the field of Healthcare due to the immense development in medical facilities all over the world. There is a need for validation and accurate analysis of the enormous volume of data.

In the recent years, many anonymization techniques have been developed in order to preserve the privacy of healthcare data. These techniques also have been developed in order to retain the data utility. Generalization[1] is one of the more popular techniques for *k*-anonymity and Bucketization[1] for *l*-diversity. Since, these techniques have some setbacks such as-considerate amount of information loss for high dimensional data, difficulty of supporting marginal publication. In Generalization, each attribute is generalized separately hence correlations between different attributes are lost. The drawback of Bucketization is that it does not prevent membership disclosure[2].

Quasi-Identifiers (QI) may be two or more attributes that can be combined to recover the patients identity; attributes such as age, sex and race.

Sensitive attributes (SA) are the attributes that need to be preserved in the dataset, for example the disease, doctors, salary.

In distributed computing, extremely large datasets are processed and stored using an open source, Java based programming framework called Hadoop.

Hadoop includes a distributed file system called Hadoop Distributed File System(HDFS). Large data sets running on clusters of commodity hardware can be stored in HDFS. It has a block size of 64MB, block size can be increased to 128 or 256MB depending upon the requirement of the system. HDFS replicates each block to multiple machines in a cluster, typically 3 but can also be user-specified.

HDFS includes a NameNode, secondary NameNode, and multiple DataNodes.

NameNode: It is used for storing file system metadata, mapping of files to the blocks and provides a global picture of the filesystem.

Secondary NameNode: It performs internal NameNode transaction log check-pointing. It is a helper to the NameNode but it is not a replacement for the primary NameNode. In order to overcome failure of primary NameNode, Hadoop implemented a secondary NameNode whose main function is to edit the log file and store a copy of Fslmage file.

DataNode: It stores data in the Hadoop file system in the form of blocks. It connects to the NameNode on startup and responds to the requests from the NameNode to perform file system operations. It has local access to one or more disks in a server on which it is permitted to store data

Hadoop includes a database called HBase, which stores and searches data from a large data table. It is a column-oriented database management system which automatically shares the data table across multiple nodes so that MapReduce jobs can run locally.

Hadoop MapReduce is a framework to write applications that process large amounts of data in parallel on large clusters of commodity hardware. It contains two important functions which are Map and Reduce. It splits

the input data into independent blocks which are then processed by the Map function in a parallel way. The outputs of Map function are then sorted and given as input to the Reduce function. The framework of MapReduce consists of single master which is the *JobTracker* and slaves which are *TaskTrackers*(one per cluster). The master schedules the jobs and assigns them to the specific slaves. It also monitors the jobs and re-executes them in case of failure.

2. LITERATURE SURVEY

In [1] the author talks about Generalization for Privacy preserving of data publishing. Along with this it also throws light upon various drawbacks of Generalization. It develops ANGEL that is a new anonymization technique which is as effective as Generalization but can retain significantly more useful correlations in the database called Microdata.

Generalization is regarded as a point to rectangle transformation in the space formed by QIattributes. Various Generalizations may provide drastically different Privacy protection and therefore Generalization needs to be guided by anonymization principles such as *k*-anonymity, *l*-diversity and *l*-closeness.

In [2] the author talks about Generalization and Bucketization along with its drawbacks. The paper presents a novel technique called Slicing which is used for Privacy preserving data publishing. It also talks about various advantages of Slicing when compared with Bucketization and Generalization. Data utility is better preserved in Slicing than in Generalization. It also preserves more attribute correlation than Bucketization. It handles high dimensional data well without clear separation of SA and QI.

Slicing divides the data both horizontally and vertically. Vertical partitioning is facilitated by grouping attributes into columns based on correlations among each attribute. Every column contains a subset of attributes which are highly correlated. Tuples are grouped into buckets in horizontal partitions. Within each bucket, the values of each column are randomly permutated to break the linking between different columns. In Slicing the association across different columns is broken but association within each column is preserved. Due to this, dimensionality of the data is considerably reduced and utility is preserved better than Generalization and Bucketization.

In this technique, the output table also satisfies *l*-diversity. The associations between attributes which are uncorrelated are broken.

In [3] the author talks about updates made to confidential and anonymous databases where a database owner A needs to determine whether a database when inserted with a tuple owned by a person B is still secure. It talks about two protocols which contain Generalization based and Suppression based *k*-anonymous confidential databases.

In [4] the author talks about focusing on Data Publication in a dynamic dataset. It talks about *k*-anonymity and its vulnerabilities such as homogeneity attacks and background knowledge attacks. They propose a stronger model which is *l*-diversity which needs every QI group to have at least one well represented SA. They are throwing light upon m-invariance that limits the risks of privacy disclosure in re-publication. It considers insertions, updates and deletions in the microdata.

In [5] the author talks about enhanced slicing models due to the drawback of Slicing i.e. when more number of similar attribute values are present along with their sensitive values in different tuples, the output maybe the original tuple despite performing random permutation. Also, the utility of the data set may be lost due to the generation of the fake tuples. Enhanced Slicing models such as suppression slicing which is done by suppressing any one of the attribute value in the tuple. Only a very few values are suppressed and random permutation is

used to maintain privacy. It also throws light upon the Mondrain Slicing in which random permutation is done not within a single bucket but within all buckets.

In [6] the author talks about handling privacy preservation using a new technique called Overlapping Slicing which handles the data attributes based on the idea of Fuzzy Clustering. In this paper multiple data tables are generated which satisfy *l*-diversity. It also processes high dimensional data effectively. In the first step Fuzzy Clusters are formed for the target attribute until the minimum value of the objective function is derived. Then Slicing is performed on the data.

[7] author talks about the attacks on data such as record linkage and attribute linkage. Now privacy preservation is achieved through generalizing the QI by setting range values and performing record elimination. This method attempts privacy preservation at static microdata which only contains numeric QI.

3. TECHNOLOGIES AND TOOLS

A. Hadoop

Hadoop is a framework which has been overseen by Apache software foundation. It is a framework used for storing and processing large data sets but not recommended for working with small data sets. HDFS and Map Reduce are two key components of Hadoop. HDFS to store the data and map reduce to process the data. Hadoop supports the concept of distributed architecture. Standalone, pseudo-distributed and fully distributed are the 3 modes of Hadoop configuration [8], [9]. The Hadoop cluster consists of master and slave nodes. The core services of Hadoop are Name Node, Data Node, Resource Manager, Application Master, Node Manager, Secondary Name Node. Generally, in fully distributed mode the Name node, Secondary name node and Application Master are identified as Master services whereas Data node and Node Manager are classified as slave services. [8], [9].

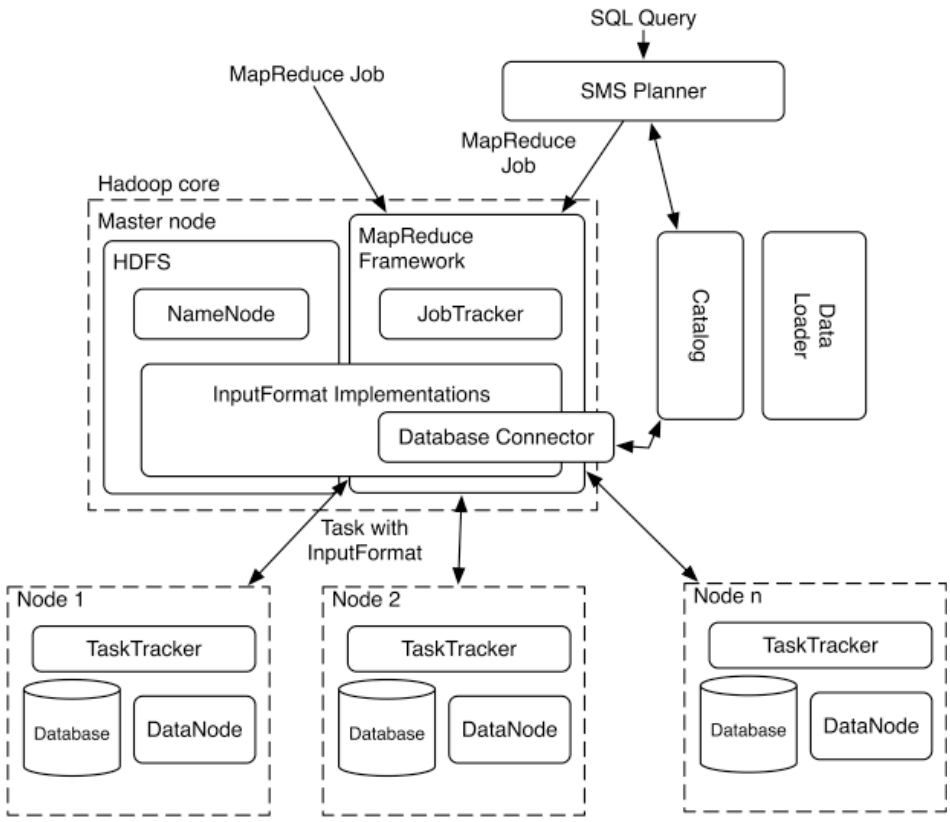

Figure 1: Hadoop Architecture

Map Reduce Layer: Map-reduce is a data processing part in Hadoop. Map Reduce working very closely with HDFS really knows about where the data is placed. This fact makes Map Reduce very efficient. Map Reduce is actually aware of fact that your input data is replicated and the same data is available on multiple nodes. In Map Reduce two types of trackers are used namely Job Tracker and Task Tracker [8], [9].

Job Tracker: JobTracker process runs on a separate node and not usually on a DataNode. JobTracker is an essential Daemon for MapReduce execution in MRv1. It is replaced by Resource Manager/ApplicationMaster in MRv2. JobTracker receives the requests for MapReduce execution from the client. JobTracker talks to the NameNode to determine the location of the data. JobTracker finds the best TaskTracker nodes to execute tasks based on the data locality (proximity of the data) and the available slots to execute a task on a given node. JobTracker monitors the individual TaskTrackers and the submits back the overall status of the job back to the client. JobTracker process is critical to the Hadoop cluster in terms of MapReduce execution. When the JobTracker is down, HDFS will still be functional but the MapReduce execution can not be started and the existing MapReduce jobs will be halted. [8], [9].

Task Tracker: TaskTracker runs on DataNode. Mostly on all DataNodes. TaskTracker is replaced by Node Manager in MRv2. Mapper and Reducer tasks are executed on DataNodes administered by TaskTrackers. TaskTrackers will be assigned Mapper and Reducer tasks to execute by JobTracker. TaskTracker will be in constant communication with the JobTracker signaling the progress of the task in execution. TaskTracker failure is not considered fatal. When a TaskTracker becomes unresponsive, JobTracker will assign the task executed by the TaskTracker to another node. Task Tracker is present in the Master node as well as in Slave node. The main job of a Task Tracker is to accept and execute the tasks as directed by the Job Tracker. There are set of slots configured to each Task Tracker. A number of tasks that a Task Tracker can accept are indicated by a set of slots [8], [9].

Name Node: NameNode is the centerpiece of HDFS. NameNode is also known as the Master. NameNode only stores the metadata of HDFS – the directory tree of all files in the file system, and tracks the files across the cluster. NameNode does not store the actual data or the dataset. The data itself is actually stored in the DataNodes. NameNode knows the list of the blocks and its location for any given file in HDFS. With this information, NameNode knows how to construct the file from blocks. NameNode is so critical to HDFS such that when the NameNode is down, HDFS/Hadoop cluster is inaccessible. NameNode is a single point of failure in Hadoop cluster.

Data Node: DataNode is responsible for storing the actual data in HDFS. DataNode is also known as the Slave NameNode and DataNode is in constant communication. When a DataNode starts up it announce itself to the NameNode along with the list of blocks it is responsible for. When a DataNode is down, it does not affect the availability of data or the cluster. NameNode will arrange for replication for the blocks managed by the DataNode that is not available. DataNode is usually configured with a lot of hard disk space. Because the actual data is stored in the DataNode. HDFS uses replication for the reliability of the data. Typically files are divided into blocks and multiple replicas of the block are stored in HDFS. These replicas are placed on data node and a client that is reading a block usually, reads it from the local node [9].

HDFS: HDFS is a custom file system for storing data sets of large size on a cluster of commodity hardware and with streaming access patterns. HDFS is highly scalable, reliable and manageable file system. Parallel reading and processing of data are supported by HDFS. Also, HDFS is fault tolerant and easy to manage. Automatically manages addition/removal of nodes [8], [9].

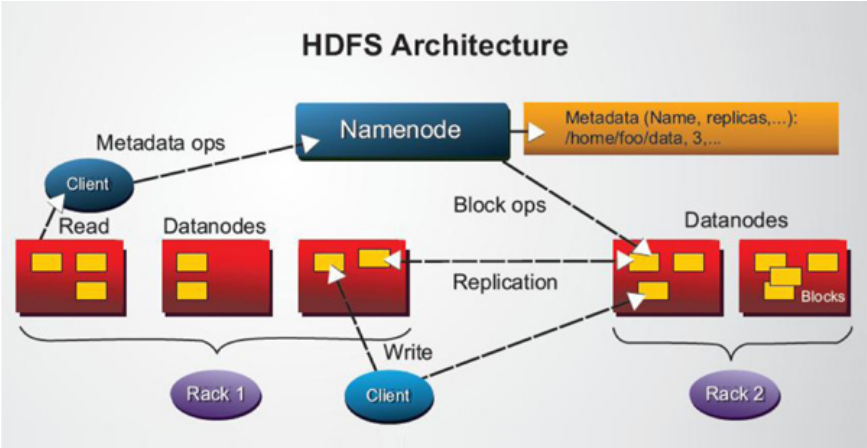

Figure 2: Hdfs Architecture

B. Apache Pig

Pig is a high-level platform for creating MapReduce programs used with Hadoop. The language for this platform is called Pig Latin.Pig Latin abstracts the programming from the Java MapReduce idiom into a notation which makes MapReduce programming high level, similar to that of SQL for RDBMS systems. The important property of Pig programs is that they are amenable to a high amount of parallelization, which in turns allows them to handle very large and big data sets. At the present time, Pig's architectural layer consists of a compiler that creates a sequence of Map-Reduce programs, for which large-scale parallel implementations exists. Pig uses a language which is known as Latin Pig [10].

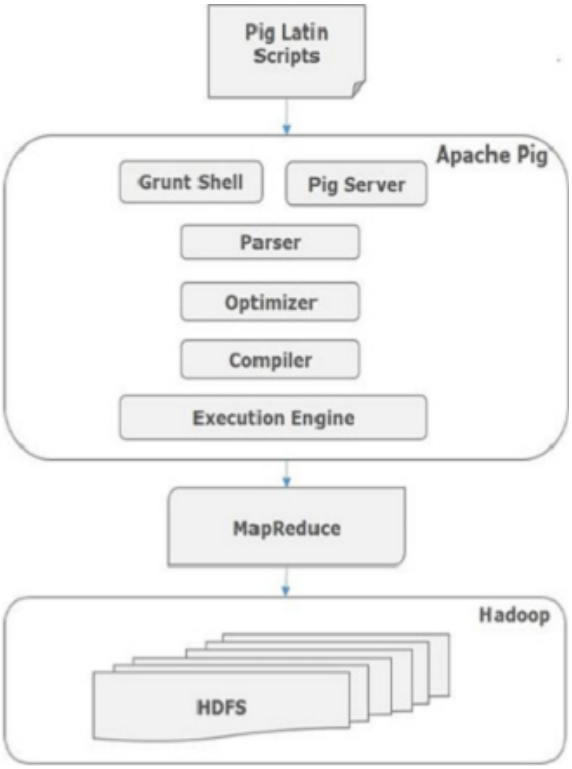

Figure 3: Apache Pig Architecture

C. Apache Hive

The Apache Hive data warehouse software facilitates querying and managing large datasets residing in distributed storage. Hive provides a mechanism to project structure onto this data and query the data using a SQL-like language called HiveQL.

D. Apache Spark

With the Hadoop framework. Also, them traditional SQL queries should be implemented in MapReduce to execute SQL applications as well as queries over a distributed data. Hive also provides the required SQL abstraction to converge SQL-like Queries into Java API [16].

Apache Spark offers the programmers with an API centered on a data structure known as the Resilient Distributed Dataset (RDD), that is maintained in a fault-tolerant manner. It was aimed to be developed in a response to provide limitations in a MapReduce cluster computing domain, which forces a linear data flow structure on distributed programs [16].

E. Map Reduce

MapReduce by Hadoop is nothing but a software framework which is used for building applications that process data of huge amounts (multi-terabyte) on large clusters which work in parallel, in a reliable, fault-tolerant manner [15], [16].

MapReduce splits the input data-set into chunks which are independent of each other, which are then processed by the map function task in parallel. The result of the map function is sorted by the framework, which is then given as an input to the next stage which is the reduce function task. The important activities of scheduling tasks, monitoring the tasks and re-executing the failed tasks is taken care by the framework as well. [15], [16].

This MapReduce framework consists of a single master JobTracker and one slave TaskTracker. The master JobTracker is responsible for scheduling the MapReduce jobs' component tasks on the slave nodes, monitor them and re-execute the failed tasks. The slaves are responsible for executing the tasks as directed by the master. The Hadoop framework is implemented in Java but it is not compulsory for the MapReduce applications to be written in Java as well [15], [16], [15].

The MapReduce framework uses <key, value> pairs, the framework displays the input to the job as a set of <key, value> pairs and gives a set of <key, value> pairs as the output.

Input and Output types of a MapReduce job:

(input) <key1, value1> -> map -> <key2, value2> -> combine -> <key2, value2> -> reduce -> <key3, value3> (output) [9].

Mapper Function: The Mapper function maps the given input key/value pairs into a set of intermediate key/value pairs.Maps are the tasks that convert input data into intermediate data. The transformed intermediate data does not need to be of the same type as of the input data. Zero or many output pairs may be produced for a given input pair [9].

4. DEFINITIONS

***k*-anonymity:** *k*-anonymity is a property of anonymized data, i.e, data in which there is removal of identifiable information from data sets in order to retain the privacy of the database. K-anonymity refers to person-specific and field-structured data which when released with scientific guarantee, makes sure that the person/persons are

not re-identified from the released data, yet the data is practically useful. "A release of data is said to have the *k*-anonymity property if the information for each person contained in the release cannot be distinguished from at least *k*-1 individuals whose information also appears in the release".

Following are the methods of *k*-anonymization:

Suppression: In suppression, all or some values of an attribute are replaced by an '*'. Values such as Name or Religion can be suppressed.

Generalization: In generalization, a broader category replacement is done for the attributes. For example, the value 18 of the attribute 'Age' can be replaced by Age <= 25.

Attacks on *k*-amonymity:

Homogeneity Attack: In case all sensitive values within a set of *k* records are identical, even though the data has under gone *k*-anonymization, the sensitive value can be exactly found out.

Background knowledge attack: In[M], it is shown that there is an association between 1 or more QI attributes within the SA to reduce the set of possible values for the SA.

l-diversity: It is a group based anonymization technique that is used for privacy preservation of data. Privacy is preserved by reducing granularity of representation of data. This reduction is like a trade off which results in loss of effectiveness of mining algorithms or data management in order to gain privacy. This is an extension of *k*-anonymity, where there is loss of information due to generalization and suppression. It must be taken care that the SA must be diverse within each QI equivalence class. That means, each equivalence class has at least *l* well represented Sensitive values.

In [M], 3 distinct methods are used to implement *l* diversity.

Distinct *l*-diveristy

Entropy *l*-diversity

Recursive $(c - l)$ diversity

Attacks on *l*-diversity:

Skewness: The attribute values may be semantically similar which causes difficulty in creating good *l*-diverse representations in the data. L-diversity maybe insufficient to prevent attribute disclosure.

t-closeness: " An equivalence class is said to have *t*-closeness *f* distance between the distribution of SA in this class and distribution of attribute in the entire table is not more than a threshold value *y*. If all equivalence classes have *t*-closeness, then the table as *t*-closeness." *t*-closeness is a further improvement of *l*-diversity group based anonymization. It takes into account, the distribution of data values for a particular attribute.

5. PRIVACY IN DATA GENERATION PHASE

In the Data Generation phase, the privacy of user's data is at risk as the data can be collected by a third party without the consent of the user. However, we can minimize the risk of privacy violation by using two methods namely Access Restriction and Falsifying Data.

In the Access Restriction method, if the data provided by the owner is passive (data generated during user's online activity which can be collected by a third party which the user might not be aware of) then certain measures such as encryption tools, anti-tracking extensions and advertisement/script blockers should be used to ensure privacy of the user's data. Though a user's data cannot be completely protected from unauthorised parties,

it is advisable to use certain privacy preserving access restriction tools. Anti-malware and anti-virus software can also be used for this purpose.

Certain sensitive data cannot be protected by using access restriction algorithms, but however they can be protected using the method of Falsifying Data. By using this method, sensitive data can be distorted using certain tools to prevent revelation of true information to the third parties that try to gain access to this data. Two techniques namely SocketPuppet and MaskMe is used for Falsifying data.

1. SocketPuppet is a tool used to hide the online identity of a user. The tool does this by creating a false identity for a user and hence a user's true activities are concealed. Thus, if multiple such SocketPuppets are used, then multiple false identities can be generated. Relating different SocketPuppets to one user is not possible by the third party as it does not have enough knowledge. The user's data cannot be discovered easily and thus the data is secured.
2. MaskMe is a tool used to mask user's identity. Aliases are created for personal information. These masks can be used by the user whenever information is needed.

6. PRIVACY IN DATA STORAGE PHASE

- *Identity Based Encryption (IBE):* With the help of Identity based systems one can generate public keys from any known identities (ASCII String). IBE uses a novel approach to encryption key management problem. Data can be protected without the need for certificates by using public keys which can be any arbitrary string. A key server provides protection which controls on the go generation of private keys which correspond to public identities and the key servers root key. Authentication and authorization from private key is separated by the key server, permissions for generation of keys is controlled dynamically on basis on granular policy. IBE simplifies operation and scaling due to its stateless nature.
- *Key-Policy Attribute Based Encryption (KP-ABE):* This is a type of encryption in which the private key and the ciphertext of the user are dependent upon attributes (Example. Blood group, Locality, Gender, etc.). Decryption of ciphertext is possible when the user key matches the attributes. An important aspect of ABE is collision resistance. ABE reduces the number of keys used for encryption in log files, as it is possible for encryption of the log file with attributes that match recipients' attributes.
- *Homomorphic Encryption:* In this, computations can be performed on encrypted text (ciphertext) which generates a result which when decrypted is same as the operations that are performed on the plaintext. This allows chaining of different services without exposing the data contained in each of these services. Example- A chain of different services with respect to healthcare could be (1) Disease susceptibility (2) Insurance claim (3) Patient history without exposing un-encrypted data to each of these services. Homomorphic encryption ensures the confidentiality of processed data, this property can be used to create secure systems.
- *Storage Path Encryption:* This is a scheme for secure storage of Big data on clouds. The big data is separated into multiple sequenced parts after which each part is stored on various storage media owned by cloud storage providers. To access this data, the different parts are collected from different data centers and then restored to original form and presented to data owner. In this, the data stored is classified into two types- public data and confidential data. Public data does not have extra security requirements (free access of data), confidential data is secure and not accessible to irrelevant users.

- *Usage of Hybrid Clouds:* Hybrid cloud is a combination of public and private cloud. It takes into consideration the advantages of both models. Features such as scalability, processing power, etc. are taken from public clouds; also security and research opportunity of storage and processing of big data from private clouds. This approach has a drawback which is when we adopt hybrid cloud directly, all sensitive data is stored on the private could which requires a lot of storage. To avoid this, users prefer storage to be done on public cloud. A scheme has been presented for reduction of communication overheads between public cloud and private cloud.

7. PRIVACY IN DATA PROCESSING PHASE

In the Data Processing phase, before publishing the data, the privacy of the data should be maintained. This can be done using Anonymization techniques:

1. *Generalization:* In this method, the values of certain specific QI attributes are replaced with less specific values. This is done so as to not provide a detailed description of the data which will be useful in preserving the privacy of the data. Various generalization techniques such as: subtree generalization, full domain generalization, multidimensional generalization, cell generalization etc are used.
2. *Suppression:* In this method, the values of certain specific QI attributes are not disclosed. This is achieved by replacing those values with a special symbol ('*'). This maintains the privacy of user's data.
3. *Anatomization:* This works by de-associating the relationship between QI and SA. Two separate tables are formed for QI and SA. Both the tables contain a single common attribute known as GroupID (GID).

8. CONCLUSION

With the amount of data increasing every day, the need to preserve this data is also rising. In this section, we discuss about different directions for privacy preservation in Big Data. Due to Big Data technology, there is an increase in demand to process healthcare related patients data for retrieving health care information and discovery in the medical stream using Big Data and its framework. In this paper, we clearly see the demand for Big Data and application using Hadoop and other frameworks in Healthcare stream. After knowing the Hadoop, Map Reduce, Spark, Pig, HBase, Hive framework and their components, it can be noted that there is an increasing demand of Big Data in the field of medical database. Doctors and Medical Professionals will be benefited using Big Data technology.

REFERENCES

[1] "ANGEL: Enhancing the Utility of Generalization for Privacy Preserving Publication" IEEE Transactions on Knowledge and Data Engineering, Yufei Tao, Hekang Chen, Xiaokui Xiao, Shuigeng Zhou, Vol 21 July 2009.

[2] "Slicing: A New Approach for Privacy Preserving Data Publishing" IEEE Transactions on Knowledge and Data Engineering, Tiancheng Li, Ninghui Li, Vol. 24, No. 3, March 2012.

[3] "Privacy-Preserving Updates to Anonymous and Confidential Databases", IEEE Transactions on Dependable and Secure Computing, Alberto Trombetta, Wei Jiang, Vol. 8, No. 4, July/August 2011.

[4] "Privacy Preserving Research for Re-publication Multiple Sensitive Attributes in Data", Xiaolin Zhang, Lifeng Zhang, IEEE 2011.

[5] "Enhanced Slicing Models For Preserving Privacy In Data Publication", International Conference on Current Trends in Engineering and Technology, ICCTET'13, S.Kiruthika, Dr.M.Mohamed Raseen.

[6] "A Data Anonymous Method based on Overlapping Slicing", Jing Yang, Ziyun Liu, yangyue, Jianpei Zhang, Proceedings of the 2014 IEEE 18th International Conference on Computer Supported Cooperative Work in Design.

[7] "Anonymization Technique through Record Elimination to Preserve Privacy of Published Data", Proceedings of the 2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering, February 21-22, R. Mahesh, T. Meyyappan.

[8] An Emerging 3-Tier Architecture model and frameworks for Big Data Analytics, Preetishree Patnaik 1, Pooja Batra Nagpal, Ulya Sabeel

[9] Big Data: A Review by Seref SAGIROGLU and Duygu SINANC Gazi University,IEEE 2014z

[10] C. Eaton, D. Deroos, T. Deutsch, G. Lapis and P.C. Zikopoulos, Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data, Mc Graw-Hil.

[11] Toward a Big Data Healthcare Analytics System: a Mathematical Modeling Perspective, Hamzeh Khazae, Carolyn McGregor, Mikael Eklund, Khalil El-Khatib,Anirudh Thommandram.

[12] Big data Analytics in Healthcare: A Survey Approach Dharavath Ramesh, Pranshu Suraj, and Lokendra Saini.

[13] Big Data Analytics in Healthcare- Its Benefits, Phases and Challenges Jasleen Kaur Bains Department of Computer Science and Applications, Punjab University, Chandigarh, India.

[14] A Survey On Big Data Analytics In Health Care Priyanka K Prof Nagarathna Kulennavar B.V.B C.E.T B.V.B.C.ET Hubli.

[15] Apache Hive for Apache Hadoop, https://hive.apache.org/

[16] Apache Spark for Apache Hadoop, http://spark.apache.org/.

[17] Aditya B. Patel, Manashvi Birla, Ushma Nair, "Addressing Big Data Problem Using Hadoop and Map Reduce", 2012, 6-8

[18] Hadoop Framework in depth analysis and tutorials, http://hadoop.apache.org/

[19] hadoopinrealworld.com