Jain *et al. J Big Data* (2019) 6:30
https://doi.org/10.1186/s40537-019-0193-4

Journal of Big Data

RESEARCH Open Access

# Enhanced Secured Map Reduce layer for Big Data privacy and security

Priyank Jain*, Manasi Gyanchandani and Nilay Khare

*Correspondence:
Priyankjain1984@gmail.com
Department of Computer Science and Engineering, MANIT, Bhopal, MP, India

**Abstract**

The publication and dissemination of raw data are crucial elements in commercial, academic, and medical applications. With an increasing number of open platforms, such as social networks and mobile devices from which data may be collected, the volume of such data has also increased over time move toward becoming as Big Data. The traditional model of Big Data does not specify any level for capturing the sensitivity of data both structured and unstructured. It additionally needs to incorporate the notion of privacy and security where the risk of exposing personal information is probabilistically minimized. This paper introduced security and privacy layer between HDFS and MR Layer (Map Reduce) known as new proposed Secured Map Reduce (SMR) Layer and this model is known as SMR model. The core benefit of this work is to promote data sharing for knowledge mining. This model creates a privacy and security guarantee, resolve scalability issues of privacy and maintain the privacy-utility tradeoff for data miners. In this SMR model, running time and information loss have a remarkable improvement over the existing approaches and CPU and memory usage are also optimized.

**Keywords:** Big Data, Security and privacy, Privacy preserving, HDFS, HADOOP, HIVE, Secured Map Reduce (SMR) layer

## Introduction

Data security, according to the common definition is confidentiality, integrity, and availability of data. It is the act of guaranteeing that the information is safe from unauthorized access, ensures that the information is reliable and accurate which is accessible whenever it is required. An information security design incorporates features, for example, gathering the required data, protecting it, and obliterating any data that is never again required [1]. Privacy, on the other hand, is the appropriate use of information. In other words, merchants and companies should use the data provided to them only for the intended purpose. For example, if an individual purchase a product from XYZ Company and provides them with their personal information like address, card number, etc. then this company cannot sell that information to the third party. Companies need to enact a data security policy for the sole purpose of ensuring data privacy of their consumer personal information. Moreover, companies must ensure data privacy because the information is an asset to the company. However, no data security policy can overcome the

willingness to sell or soliciting of the information of consumer data that was entrusted to an organization [1, 2].

### The challenges of privacy and security in Big Data

The term Big Data [3–8] refers to the massive digital information which is collected by different companies and government organization. Everyday quintillion bytes of data have been created i.e. 90% of the data in the world today has been created in the last 2 years alone. Security and privacy issues [9–15] are magnified by velocity, volume, and variety of Big Data, such as large-scale cloud infrastructures, diversity of data sources and formats, streaming nature of data acquisition and high volume inter-cloud migration. The use of large-scale cloud infrastructure, with a diversity of software platforms, spread across large networks of computers, also increase the attack on the system. The main security and privacy challenges in Big Data are the following: [16–18].

a. Secure computations in distributed program frameworks.
b. Security best practices for non-relational data sources.
c. Secure data sources and transition logs.
d. Endpoint input validation/filtering.
e. Real-time security/compliance monitoring.
f. Versatile and composable privacy-data mining and analytics.
g. Cryptographically enforced access control and secure communication.
h. Granular access control.
i. Granular audits.
j. Data provenance.

### The need of light weighted encryption

Heavy and light weighted encryption algorithms are used for secured communication over the Internet. Light weighted encryption algorithms are preferred over heavyweight encryption algorithms in low power designs and devices mainly because of their reduced resource requirements. A light weighted encryption technique takes less time for encryption and provides better security than existing heavyweight algorithms such as AES, RSA, PGP, TEA, and RC6 [19]. The proposed solution employs multi-level light weighted encryption along with key encryption and thereby decreases the possibility of various threats by attackers. In recent years, a large amount of person-specific data were collected by both government and private entities [20]. Laws and regulations require that some collected data must be made public (Example: Census and Healthcare Data).

### Contribution

The major contribution of the paper is toward proposing a novel approach called Secured Map Reduced Layer (SMR) for Big Data. This model is based on Input–Output Privacy and Data Security. Other key aspects of this approach are as follows:

a. Achieving high data utility with the same level of privacy compared with the existing approach.
b. Proposed SMR Layer reduces information loss.
c. SMR model is based on lightweight encryption so execution time is optimized.
d. This algorithm resolves the scalability issues related to privacy.

The rest of the paper is organized as follows. "Related work" section discusses privacy and security concerns in related work. "Problem discussion" section discusses the problems in related work. "The proposed Secure Map Reduce (SMR) model" section covers the proposed Secure Map Reduce (SMR) model. "Dataset used" section discusses understanding of Twitter Dataset. "Results and discussion" section covers the encryption and decryption side implementation, Performance measure, results and discussion. Finally, "Conclusions and future work" section concludes the paper and suggests future work.

## Related work

The various profound Big Data Privacy and Security methods analyzed or proposed by the researchers categorized into three types i.e. Input Privacy, Output Privacy, and Data Security.

### Input privacy

In input privacy, one is primarily concerned with publishing anonymized data with models such as k-anonymity and l-diversity. Work-based on Input Privacy is discussed in this section. Mohammadian [21] work describes an algorithm which is Fast Anonymization of Big Data Streams (FAST). It results in the efficiency of anonymizing Big Data and also decreases the degree of information loss and cost metric. Evfimievski [22] work refers to randomized based privacy approach for a small amount of data. Client-side contains personal information, when this information is sent to the server side, then only statistically important properties are sent. The clients can guard the privacy of their data by unsettling the data with a randomization algorithm and further giving in the randomized version. It describes certain ways and consequences in randomization for numerical and categorical data and argues the concern of measuring privacy. The exploration in utilizing randomization for preserving privacy, which gives an impression of being a piece of some more profound measurable way to deal with security and privacy. This then provides a connection to the groundbreaking work which is done by Claude Shannon. His work is on the secrecy of a system which enables us to look into privacy under a different angle than the conventional cryptographic approach. Randomization does not count on complexity hypotheses from algebra or number theory and does not need costly cryptographic operations or sophisticated protocols. It is likely that future studies will combine statistical approach to privacy with cryptography and secure multiparty computation, for their mutual benefit. Tripathy [23] work presents a procedure which can be used to achieve anonymization using k-anonymity and l-diversity on social networking site data. This algorithm is altered reasonably from their corresponding algorithm for microdata and also relies upon some modified algorithms developed for anonymization against neighborhood attack. The algorithm still needs little advancement in order to decrease the complexity, so it can be applied to large social networks. The p-sensitivity

issue as specified by Machanavajjhala is yet to be dealt with so far even in the relational database case. Only distinct l diversity has been measured and utilized up to this point. Jain et al. [24] work on improved k-anonymity algorithm applied to a Big candidate election data set acquired from the Madhya Pradesh (MP, India) State Election Commission. Somayajulu [25] work refers to perturb attribute associations in a controlled way, by shifting the data values of specific columns by rotating fields. Zakerzadeh et al. [26] propose the well-established multidimensional k-anonymization Mondrian algorithm for MapReduce framework. MapReduce based anonymization (MRA) [27] is proposed in which instead of generating a single global file for all the nodes, chunks of the file are generated and distributed among all the nodes. In the mapping step, each node appends a unique file id to each part of the file for identification purpose. Hence, in the next iteration, each node needs to access only those files which were updated by its corresponding reducer. This way load of maintaining a global file is removed. Multiple iterations and file management are two major drawbacks of this technique. As the number of iteration increases, the performance of the system reduces and file management using MapReduce is also a difficult task.

#### Output privacy

In output privacy, one is generally interested in problems such as association rule hiding and query auditing, where the output of different data mining algorithms is either perturbed or "audited" in order to preserve privacy. Work-based on Output Privacy is discussed in this section. Roy [28] work refers to a map-reduce based system that provides strong privacy, security and provides assurances for distributed computations on sensitive data known as Airavat model. This model comprises of a novel integration of mandatory access control (MAC) and differential privacy (DP). Here, data providers control the security plan for their sensitive data, including a mathematical assurance on potential privacy violations. Airavat considered the first model which integrates MAC with DP and enables many privacy-preserving Map-Reduce calculations without the need for an audit of untrusted code. Derbeko [29] work on map reduce when a Map-Reduce computation is implemented in public or hybrid clouds, privacy, security, and output of Map-Reduce are essentially considered. In public and hybrid cloud environment, implementation of the Map-Reduce paradigm requires privacy, integrity, and correctness of the outputs as well as verification of mapper's reducers. Mehmood [2] work provides a complete outline of the privacy preservation techniques in Big Data and presents the challenges of existing mechanisms. They explained the infrastructure of Big Data and the privacy, maintaining techniques in each phase of the Big Data lifecycle. They also explained fundamental difficulties in deploying homomorphic encryption in the framework of Big Data analytics. This is to keep the computational complexity as low as possible. Chaudhari and Tiwari [30] work on heuristic-based association rule hiding using oracle real application clusters by introducing the concept of the impact factor of the transaction on the rule. Yadav and Ojha [31] work on data hiding in a generic grid that could be of pixels or bits.

### Data security

Security is the "confidentiality, integrity and availability" of data. Security offers the ability to be confident that decisions are respected. Work-based on Data Security is discussed in this section. According to the Terzi [32] work, network traffic should be expressed in code with suitable standards, access to devices should be checked, employees should be authorized to access systems, analysis should be done on anonymized data sending and receiving should be made for the secure channel to prevent data drip, and network should be observed for threats. Kacha [33] work describes principal issues related to data security that is raised by cloud environment are classified into three categories: (1) data security issues raised by single cloud characteristics, (2) data security issues raised by data lifecycle in cloud computing, (3) data security issues associated with data security attributes (confidentiality, integrity, and availability). Ilavarasi [34] states the concern on the security while distributing microdata about population. The emerging area called privacy-preserving data mining (PPDM) concentrates on any person-specific privacy without negotiating data mining results. The enhancing growth of PPDM algorithms enhances the concept of investigating the privacy inferences and the crosscutting issues between privacy versus the utility of the data.

## Problem discussion

1. Existing approaches as discussed in related work in "Related work" section are categorized into three different category's i.e. Input privacy, output privacy, and data security. Each of these three categories has a specific purpose. The proposed SMR Layer is a combination of all three categories i.e. Input–Output Privacy and Data Security. It provides not only input privacy on raw data (Twitter Dataset) but also applies query auditing in Output Privacy. This layer also maintains confidentiality, integrity, and availability as a part of data security.
2. Input and Output privacy are based on partial encryption having high information loss. Data security is based on data encryption which is applied on an entire dataset which is time-consuming. Our proposed model is based on lightweight encryption. It does not only provide full encryption in optimal time but also maintains it optimizes information loss. Due to the use of lightweight encryption, it utilizes the effect of large crowd i.e. Big Data. In increasing data size, execution time difference does not proportionally increase. So it resolves scalability issues of Privacy.

## The proposed Secure Map Reduce (SMR) model

The enterprise organizations are facing deployment and management challenges with Big Data. Hadoop's core specifications are still being developed by the Apache community and, thus far, do not adequately address enterprise requirements for more robust privacy and security, policy enforcement, and regulatory compliance. While Hadoop may have its challenges, its approach, which allows for the distributed processing of large data sets across clusters of computers, represents the future of enterprise computing. In

order to fill the privacy and security gaps that exist in all open source Hadoop distributions, a solid pathway for securing distributed computing environments in the enterprise is provided by proposed Secured Map Reduce Model. The traditional model of Big Data does not specify any level for capturing the sensitivity of both structured and unstructured data. It additionally needs to consolidate the thought of protection and security where the danger of uncovering individual data is probably limited. Given the high volume of enormous information, and the combination of structured and unstructured data requires some set of new models for Big Data so as to increase privacy and security. These algorithms build on current privacy-preserving data techniques, thus comes up with a new model which incorporates a new layer of privacy on the map reduces phase of Big Data architecture. This new layer thus implements the security algorithms on the data individually as the data come across the map-reduce phase. The security algorithm should be light weighted encryption techniques so that the overhead of new algorithms does not affect the main functionality of the Big Data. The data thus can be protected and secured, when it is processed through this new proposed Secured Map Reduce (SMR) layer of Big Data. It starts from a collection of data from weblogs, Social Data, Streaming Data, and then the collected data is sent to HDFS (Hadoop Distributed File System) [35–39]. This proposed model introduces a privacy layer between HDFS and MR Layer (Map Reduce) known as Secured Map Reduce (SMR) Layer as shown in Fig. 1. To increase the security and privacy of the data, perturbation and randomized techniques were used.

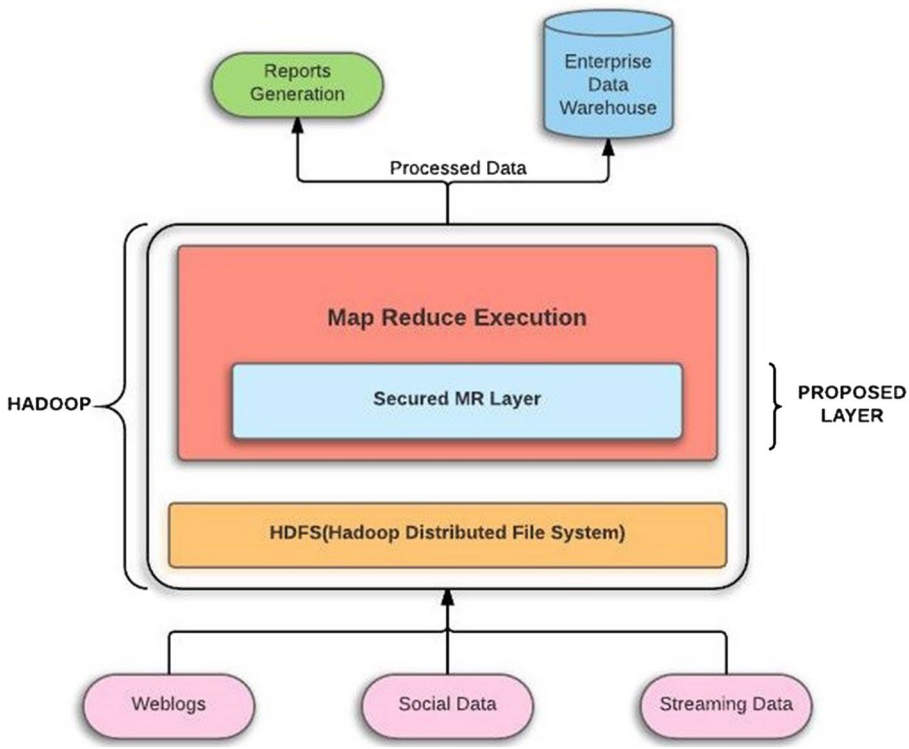


**Fig. 1** New proposed model of Big Data with secured MR layer

SMR layer is based on a proposed lightweight encryption process, which satisfies the big data security and privacy in optimal timing requirement [40]. In lightweight encryption process original data are passed to HDFS (Hadoop Distributed File System) then the data from HDFS will be passed to Map Reduce Layer. The original data may be in the form of weblogs, streaming data, social data. The encryption process takes place in map reduce layer. Once the data entered the map reduce layer the encryption starts. Encryption is the step where encode or encrypt the given data. There are two levels involved in this encryption. At the first level converting text data to number, to do this, first, consider the text and dividing each word text into tokens. It will take key-value pairs (KVP) model, by considering each unique word and counting the number of times the word is repeated in the given data. Where key is each unique word and number of times the word is repeated is value. This process not only provides lightweight encryption but also provide high privacy by giving data. The second level performs the randomization process [41–45] in converted number data, which enhance the privacy level. Vertical partitioning of the HybrEx model [46] shown in Fig. 2 has been implemented where the data is first processed in the private cloud at the time of encryption and then the data is processed in public cloud at the time of decryption.

Decryption is the step where decrypt the encrypted data. It is the reverse process of encryption. In this process processed data (key-value pairs) is passed to HDFS then the output of this is passed into the map-reduce layer. Where decryption takes place in map reduce layer. This phase is called a reconstruction phase. There are two levels involved in this decryption process. First is reverse randomization, this level uses randomization to decrypt the encrypted message to some extent. Second is

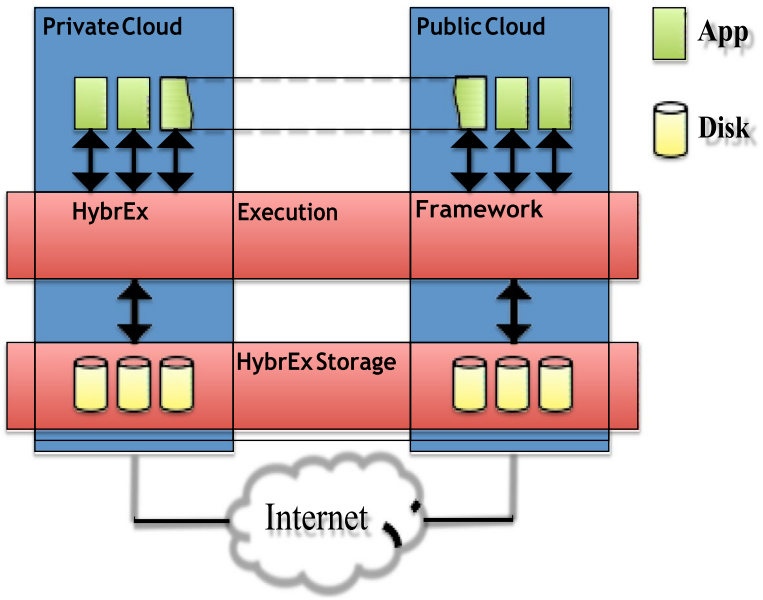


**Fig. 2** The architecture of the HybrEx Model

convert number to text data, using the concept of Key-Value pairs (KVP) model after reverse randomization. Every word (token) is key and a number of times it is repeated is value. But as the order is not mentioned here to maintain the order will retrieve the data from the file where wrote the order. By this, it will successfully complete the number to text data conversion.

**Algorithm 1 Secure Map Reduce (SMR) Encryption Algorithm**

**Input:** File data F.

**Output:** Encrypted file SMR, an Encrypted file of frequencies EC.

**Mapper Phase:**

1. Partition (F).
2. for each line $L_i$
   - A. read();
   - B. tokenize();
   - C. for each word $w_j$
     - i. Convert $w_j$ to number $n_j$ ; // here we covert the word into the number and store this into a hash map and the SMR file.
     - ii. rand($n_j$ ); // here the number is replaced by the random

3. A. Write(rand($n_j$ ), SMR); // this original-random number pair is written into the SMR encrypted file for the process of reverse randomization.

   B. Write(mapper id, SMR);

**Reducer Phase:**

1. Combine(mapper result);
2. Count($w_j$ );
3. Encrypt(Count($w_j$ ));

#### Description of SMR encryption algorithm

In Algorithm 1 Partition(F), the HDFS partition the file data (F) into n number of blocks each of size 128 MB and distribute them to m nodes where mapper and reducer working on. Now by read() function, the mapper read the part of the data file line by line, then tokenize the string into separate words using tokenize(). At that point, each word will be changed over to various one-way privacy purpose, then in two-way privacy purpose, it further changed to another arbitrary number through the procedure of randomization for which utilize the function Rand(). The write() function would lead to this random number being written to a common file which is called the SMR encrypted file, where the pairs of a random number corresponding to the original number are written and also the order of the original numbered data is written. This file is then used at the time of reverse randomization. Thus, the original numbered data are retrieved and also the order of the sentence is maintained. Here each mapper maintains the count of each number. Now the results of all the mappers passed to the reducer. The file which is generated on the mapper side is the encrypted SMR file which maintains the (noisy number) − (original number) pairs and the entire order of the original numbered file along with the mapper ids. It is to be considered that the mapper task also writes the mapper id at the end of each sentence to maintain the regular order of sentences. In the reducer phase, each reducer combines the result of mappers and maintain the count of words in the entire file. Then the frequency of each word is also encrypted. So the word is encrypted as well as its frequency is also encrypted.

#### Application of query on the encrypted data

Analyzing the Twitter dataset could help one's improvement in many spheres like marketing companies for increases the popularity of their products, with many surveys for seeing who is most influential currently, for finding out the latest trends and patterns which in turn would help to enhance profit and business. Applying for any of the above queries the component used in this paper is the hive. After the data is transferred from HDFS to MapReduce, where it is encrypted, the component hive [47] comes into play and answers these queries. In this way, the identity of the person is preserved along with the queries being answered. Storing the data into HDFS [48] and then processing the data through a secured map-reduce phase where encryption process is performed, all this is done in a private cloud and then the encrypted data is given to the public cloud where only that person can get the data who have the decryption key. The decryption key is nothing but the SMR encrypted file which contains (noisy number) -(original number) pairs and the order of the entire original numbered file and also the mapper ids along with the hash map. Now if someone tries to query the data using hive, the person will get an encrypted answer and hence in this way this work provided two-way security to the data. Only that person having this decryption key can get the information uses of SMR layer to secured data.

**Algorithm 2 Secure Map Reduce (SMR) Decryption Algorithm**

**Input:** Encrypted file of words SMR, Encrypted file of frequencies EC.
**Output:** Decrypted file of words D, a Decrypted file of frequencies DC.

**Mapper Phase:**

1. Partition(SMR);.
2. for each line $L_i$
   A. read(mapper id);
   B. add mapper id to hash map(H);
   C. tokenize();
   D. S = reverse randomization(number); // reverse randomization is done with the help of the SMR encrypted file.
   E. add string S to hash map(H);
3. Decrypt(C);

**Reducer Phase:**

1. read(hash map(H));
2. generate(D);
3. generate(DC);

#### Description of SMR decryption algorithm

In Algorithm 2, initially, the server receives the encrypted file from the client through a network connection. This encrypted file, SMR, act as an input to the server cluster. Now HDFS at this side first partition the file data into l blocks, then distribute them to several nodes again (Partition (SMR)). The partition of a file is again read line by line. It reads the mapper id first and creates a hashmap based on these mapper ids. The mapper read one line and again tokenize it and then decrypt the number into the corresponding word by the process of reverse randomization. And further, add the whole decrypted string to the hashmap under the matching mapper id. Likewise, all the mappers add their decrypted strings to the same hash map. And now this hashmap is passed onto the reducer side. Also, the file containing the word and its frequency also be decrypted and pass onto the reducer end. The reducer will again perform two tasks simultaneously. It first read the hash map line by line and generate a decrypted file which contains the whole data in order (D) and it also combines the results of mapper to generate an output file of words and their frequency (DC).

## Dataset used

The way that researchers and other people who want to get large publicly available Twitter datasets [49] are through their Application Programming Interface (API). A large dataset of Twitter is available on their API from where a researcher can download the data. There are two unique types of Twitter API: RESTful and Streaming. The RESTful API is helpful for getting things like arrangements of supporters and the individuals who take over a specific client and is the thing that most Twitter customers are working off of. This work concentrates on the Streaming API. The Streaming API works by making a demand for a particular kind of information sifted by watchword, client, geographic range, or an irregular specimen and afterward keeping the association opens the length of there are no blunders in the association. For this purpose, using the tweepy bundle to get to the Streaming API.

### Collecting data

The initial step is to get a copy of tweepy [49] (either by looking at the store or simply downloading it) and then introducing it. The following steps to do are, firstly it has to make an occasion of a tweepy Stream Listener, which will deal with the approaching information (Begin another document for every 20,000 tweets, labeled with a prefix and a timestamp. This record is called' slistener.py') secondly, it needs the script that does the gathering itself. This record is called'streaming.py', which can accumulate clients, watchwords, or particular areas characterized by bouncing boxes. The API documentation has more information on this. For the present, some well-known keywords like Delhi and India, etc. (keywords are case-insensitive) are being used in the proposed model.

## Results and discussion

The platform used for the deployment is HP Z840 workstation. It consists of 64-bit dual core processors and 8 GB of RAM. The proposed SMR layer codes written in Java and executed in Hadoop multi-node environment. The multi-node environment created by using 5 number of workstations. Each Workstation has 40 Cores. In our experiments 40 Cores are used for Name Node and 160 cores are used for data nodes for implementing SMR.

When the input file is provided to the master node, vertical partitioning of the HybrEx model [46] has been implemented where the data processed in the private cloud at the time of Encryption. Hadoop mechanism partition the whole file into smaller parts. Then these parts are distributed to many mapper tasks by the job tracker. Inside each mapper, every word of a sentence gets encrypted with the specified logic of randomization and is written to a file along with the mapper Id of the corresponding mapper task of the slave node. Now the reducer task will aggregate the results of all mapper tasks and finally generate the encrypted file as output, which is to be transmitted to the public cloud. Anyone accessing this data from the public cloud will always get an encrypted answer when the person queries the encrypted data. Only that individual who is having the decryption key will be able to get the original data and the desired outputs to the queries. Now, what happens when the person tries to decrypt the data with the decryption key? Now, when the encrypted file is provided to the master node, the Hadoop mechanism partitions the

whole file into small parts. Then these parts are distributed to many mapper tasks by the job tracker. Inside each mapper, every word of a sentence gets decrypted with the specified logic of randomization and a hash map is created according to the mapper Id and all the sentences belonging to the one mapper are placed in one place. Now the reducer task will finally write the whole hash map in order to a file and generates the decrypted file as output which the same as the original file is.

The following performance measures are used to measure the performance of proposed SMR model.

Running time: The running time is measured in terms of the wall-clock time (milliseconds). So the performance of the proposed method can be effectively comprehended. This will provide overall better scalability.

CPU utilization: By using the proposed methodology which is based on parallel and distributed architecture tends to increase CPU utilization. CPU performance increases by efficiently using other resources such as memory space, input–output devices etc.

Memory usage: Memory usage not only depends on the amount of data to be processed but also on a data structure that has been used in the algorithm. As it is obvious that a large amount of data take more memory space, but in the proposed method, there is an effective use of memory space while processing huge amount of data sets. The appropriate data structure is used in the proposed method in order to occupy less space.

Information loss: When privacy preserving technique is applied, one should take care of degradation in data quality. Data quality (also called functionality loss metric) is a widely used metric that captures the entire amount of loss information due to encryption. For an encrypted dataset with n tuples and m attribute, the information loss I [50] is computed as follows in Eq. 2.

$$I = \sum_{i=0}^{n} \sum_{j=0}^{m} \frac{\left|upper_{ij} - lower_{ij}\right|}{m \cdot n\left|max_j - min_j\right|} \tag{2}$$

In the above equation, $lower_{ij}$ and $upper_{ij}$ represent lower and upper bound of attribute j in tuple i after generalization respectively, $min_j$ and $max_j$ represent the minimum and maximum values respectively taken by attribute j over all records.

In Fig. 3 graph shows that the parallel execution of Hadoop SMR layer tasks markedly reduces the time when it comes to increasing the number of cores 40–160 as the nodes will simultaneously perform with the same speed of processing the data. The execution time reduction is not proportional to the increase of the number of cores/nodes due to input–output operation and CPU time required for the shuffle process in MapReduce. And also, there is number of nodes takes to combine their result because it is not necessary that jobs completed by all nodes at the same time. SMR layer approach, when increase the data size time difference get minimized. Due to the use of lightweight encryption, it utilizes the effect of large crowd i.e. Big Data. In increasing data size, execution time difference does not proportionally increase. So it resolves scalability issues of Privacy.

Table 1 shows that the existing anonymization algorithms Mondrian [26] and MRA [27] take much more time when compared with the proposed SMR layer in implementing the security algorithms. The running time of SMR lies between MRA and

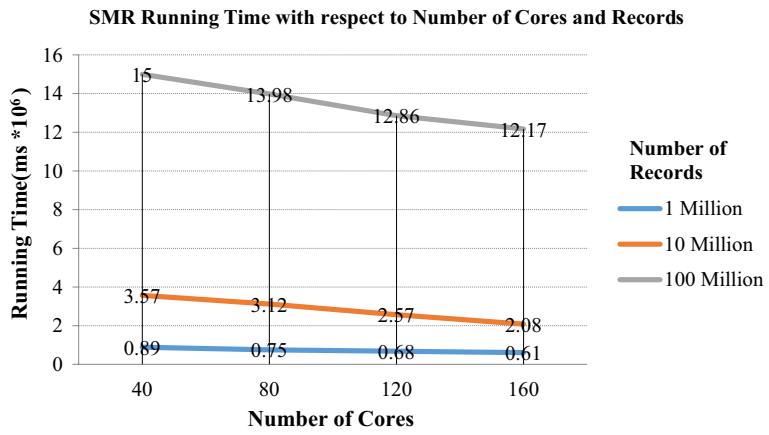


**Fig. 3** Time taken with respect to Number of Cores and Records

**Table 1 Running time comparison of mondrian, MRA, and SMR on 1 M, 10 M, and 100 M data sets**

| Data size | Time in milliseconds taken with respect to data size (ms*10$^{6)}$ | | |
|---|---|---|---|
| Data volume (number of records) | Mondrian [26] | Map-reduce anonymization (MRA) algorithm [27] | Proposed SMR model |
| 1 million | 0.53 | 0.86 | 0.62 |
| 10 million | – | 5 | 2.09 |
| 100 million | – | 56 | 12.19 |

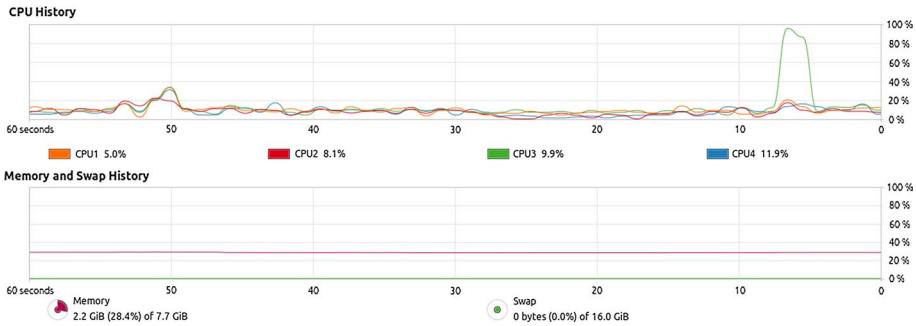


**Fig. 4** CPU and Memories uses in SMR in multimode environment

Mondrian, this gap becomes less significant for larger data set (> 10 M). Since Mondrian cannot run on multiple machines in parallel, it always holds the issue of scalability and cannot be applied over large data sets. Also, as the size of the data increases the SMR algorithm seems to take optimized time than the existing MRA algorithm.

SMR takes less CPU use in the multi-node environment, here having four HP Z840 workstations 64-bit dual-core processors, the range of CPU uses between 5% and 11.9% shown in Fig. 4. Memories uses during the running of SMR is 28.4%, i.e. 2.2 GB of 7.7 GB and swap is 0%. The result shows the SMR layer approach is suitable for Big Data.

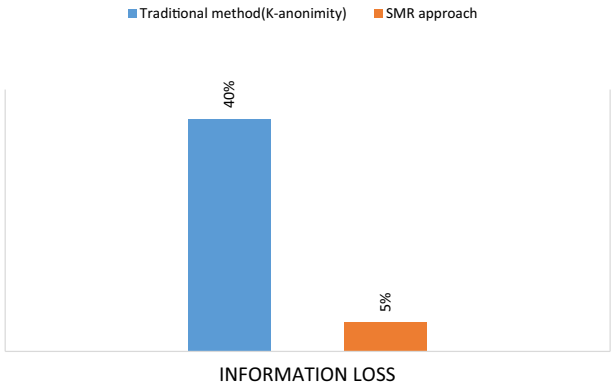


**Fig. 5** Information loss comparison of MRA [27] and SMR

**Table 2 Information loss of SMR using twitter dataset**

| Number of execution | Information loss (%) |
|---|---|
| 1 | 4.85 |
| 2 | 3.55 |
| 3 | 6.45 |
| 4 | 4.9 |
| 5 | 5.1 |

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 appear in the release. The information loss in a traditional method such as MRA [27] is around 40% and this new proposed approach SMR layer is reduced to only 5% using 100 million dataset as shown in Fig. 5. Therefore, this new SMR layer is an improvement over the existing methods and also maintain the privacy-utility tradeoff for data miners.

The mathematical formula used in the SMR layer to calculate the information loss in Eq. 3 of the randomized data in the reverse process is defined as:

$$\text{Information loss} = \frac{\left(\text{Original value} - \text{values after radomization}\right)^2}{\left(\text{Original value} + \text{values after radomization}\right)} \tag{3}$$

Experiments have been done several times on the same data, which gives variation in Information Loss in Table 2. On average an information loss (in each experiment) approximate to 5%.

Overall Information Loss $= (4.95 + 3.55 + 6.45 + 4.9 + 5.1)/5 = 4.97 \approx 5$

## Conclusions and future work

This paper emphasizes on privacy and security of Big Data. SMR model proposes a methodology to protect Big Data information. It similarly needs to ensure privacy and security where the risk of revealing individual information is probabilistically

constrained. SMR model is based on lightweight encryption, which uses randomization and perturbation methods for maintaining security and integrity. Existing anonymization methods take much more time compared to proposed SMR model for implementation of security algorithms. The experimental result shows that this approach is an advantage for Big Data, which provides better privacy and security. When increasing the data size, the running time difference gets remarkable minimized as compared to existing approaches, So SMR layer resolves scalability issues of privacy. Analysis results demonstrate that CPU utilization, Memory usage, and Information loss are optimized in proposed SMR layers. SMR layer also maintains the privacy-utility tradeoff for data miners. Future work will focus on the privacy and security of Big Data, which is generated in real-time.

**Abbreviations**
SMR layer: Secured Map Reduce layer; MR: map reduce; KVP: key-value pairs; HDFS: Hadoop Distributed File System; API: Application Programming Interface.

**Authors' contributions**
PJ performed the primary work and analysis of this manuscript. MG worked with PJ to develop the article framework and focus, and MG also drafted the manuscript. NK introduced this topic to PJ. MG and NK revised the manuscript for important intellectual content and have given final approval of the version to be published. All authors read and approved the final manuscript.

**Authors' information**
Priyank Jain: Mr. Priyank Jain is working as a Ph.D. Research Scholar. He is having more than 8 years' Experience as an Assistant professor & in the research field. Mr. Priyank Jain has experience From Indian Institute of Management, Ahmedabad, India (IIM A) in the research field. His Ph.D. is in the Big Data Privacy area. His Educational Qualification is M.Tech & BE in Information Technology. Mr. Priyank Jain areas of specialization are Big Data, Big Data Privacy & Security, data mining, Privacy-Preserving, & Information Retrieval. Mr. Priyank Jain has publications in various International Conference, International Journal & National Conference. He is a member of HIMSS.

Dr. Manasi Gyanchandani: Dr. Manasi Gyanchandani working as Assistant Professor in MANIT Bhopal. She is having more than 20 years' experience, Her Educational Qualification is a Ph.D. in Computer Science & Engineering. Dr. Manasi Gyanchandani area of Specialization in Big Data, Big Data Privacy & Security, data mining, Privacy-Preserving, Artificial Intelligence, Expert System, Neural Networks, Intrusion Detection & Information Retrieval. Dr. Manasi Gyanchandani, publications in 08 International Conference, 15 International Journal & 08 National Conference. She is a Life member of ISTE.

Dr. Nilay Khare: Dr. Nilay Khare is working as Associate Professor in MANIT Bhopal. He is having more than 21 years' experience, His Educational Qualification is a Ph.D. in Computer Science & Engineering. Dr. Nilay Khare area of Specialization in Big Data, Big Data privacy & security, Wireless Networks, Theoretical computer science. Dr. Nilay Khare, publications in 54 National and International conference He is a Life member of ISTE.

**Acknowledgements**
We acknowledge to Madhya Pradesh Council of Science and Technology, Bhopal, India for providing us funds to carry out this research work.

**Competing interests**
The authors declare that they have no competing interests.

**Availability of data and materials**
The data used in this article are publicly online available at Tweepy dataset and the link for the same is mentioned in the references section.

**Consent for publication**
Not applicable.

**Ethics approval and consent to participate**
Not applicable.

**Funding**
We are presenting this article without any kind of funding.

## Publisher's Note

Received: 29 November 2017 Accepted: 9 March 2019
Published online: 28 March 2019

**References**

1. Jain P, Gyanchandani M, Khare N. Big data privacy: a technological perspective and review. J Big Data. 2016;3:25. ISSN 2196-1115.
2. Mehmood A, Natgunanathan I, Xiang Y, Hua G, Guo S. Protection of Big Data Privacy. IEEE Access. 2016;4:1821–34. https://doi.org/10.1109/access.2016.2558446.
3. Sagiroglu S, Sinanc D. Big Data: a review. J Big Data. 2013;1:20–4.
4. Chavan V, Phursule RN. Survey paper on big data. Int J Comput Sci Inf Technol. 2014;5(6):7932–9.
5. Groves P, Kayyali B, Knott D, Kuiken SV. The big data revolution in healthcare. New York: McKinsey & Company; 2013.
6. Lin J. MapReduce is good enough? The control project. IEEE Comput. 2013;32.
7. Patel AB, Birla M, Nair U. Addressing Big Data Problem Using Hadoop and Map Reduce,"Nirma University International Conference On Engineering in Proc., 2012.
8. Cevher V, Becker S, Schmidt M. Convex optimization for Big Data: scalable, randomized, and parallel algorithms for Big Data analytics. In: IEEE Signal Processing Magazine. 2014; 31(5), p. 32–43.
9. Kuo M-H, Sahama T, Kushniruk AW, Borycki EM, Grunwell DK. Health Big Data analytics: current perspectives, challenges, and potential solutions. Int J Big Data Intell. 2014;1(1/2):114–26.
10. Fung BCM, Wang K, Chen R, Yu PS. Privacy-preserving data publishing: a survey of recent developments. ACM Comput Surveys. 2010;42:4.
11. Machanavajjhala A, Gehrke J, Kifer D. L-diversity: privacy beyond k-anonymity," 22nd International Conference on Data Engineering (ICDE'06), Atlanta, GA, USA, 2006, p. 24.
12. R. Nix, M. Kantarcioglu, and K. J. Han, Approximate privacy preserving data mining on vertically partitioned data, in Data and Applications Security and Privacy XXVI, Springer, 2012, p. 129-144.
13. Jain P, Pathak N, Tapashetti P, Umesh AS. Privacy-preserving processing of data decision tree based on sample selection and Singular Value Decomposition, 2013 In: 9th International Conference on Information Assurance and Security (IAS), Gammarth, 2013, p. 91–5.
14. Jain P, Gyanchandani M, Khare N. Privacy and security concerns in healthcare big data: an innovative prescriptive. J Inform Assur Secur. 2017;12(1):18–30.
15. Yin C, Zhang S, Xi J, Wang J. An improved anonymity model for Big Data security based on clustering algorithm" Combined Special Issues on Security and privacy in social networks (NSS2015) and 18th IEEE International Conference on Computational Science and Engineering (CSE2015). Volume 29, Issue 7 10, 2017.
16. Big Data Top challenge 2016. Online. https://downloads.cloudsecurityalliance.org/initiatives/bdwg/BigDataTopTenv1.pdf. Accessed 15 Jan 2018.
17. Big Data Submits Online. https://theinnovationenterprise.com/summits/big-data-innovation-mumbai/eventactivities=5546. Accessed 17 Feb 2018.
18. The intersection of privacy and security data privacy day event 2012. https://concurringopinions.com/archives/2012/01/the-intersection-of-privacy-and-security-data-privacy-day-event-at-gw-law-school.html. Accessed 16 Feb 2018.
19. Savas O, Deng J. Big data analytics in cybersecurity. CRC Press, Taylor Francis Group, 2017.
20. Priyank Jain, Manasi Gyanchandani and Nilay Khare,"Data Privacy for Big Data Publishing Using Newly Enhanced PASS Data Mining Mechanism", Data mining book chapter, Intech open Publisher,2018. DOI: http://dx.doi.org/10.5772/intechopen.77033.
21. Mohammadian E, Noferesti M, Jalili R. FAST: Fast Anonymization of Big Data Streams. In: Proc. of the 2014 International Conference on Big Data Science and Computing, p. 23, 2014.
22. Evfimievski S. Randomization techniques for privacy preserving association rule mining. In: SIGKDD Explorations. 2002; 4(2).
23. K. Tripathy, Anirban Mitra, An Algorithm to achieve k-anonymity and l-diversity anonymization in Social Networks, In Proc. of Fourth International Conference on Computational Aspects of Social Networks (CA-SoN), Sao Carlos, 2012.
24. Jain P., Gyanchandani M., Khare N, Improved k-Anonymity Privacy-Preserving Algorithm Using Madhya Pradesh State Election Commission Big Data, Integrated Intelligent Computing, Communication, and Security. Studies in Computational Intelligence, vol 771. Springer, Singapore p. 1–10, 2019.
25. Kadampur MA. A data perturbation method by field rotation and binning by averages strategy for privacy preservation. In: Fyfe C, Kim D, Lee SY, Yin H, editors. Intelligent data engineering and automated learning—IDEAL, vol. 5326., Lecture Notes in Computer ScienceBerlin: Springer; 2008.
26. LeFevre K, DeWitt DJ, Ramakrishnan R. Mondrian multidimensional k-anonymity'. Proc. 22nd Int. Conf. Data Engineering, Ser. ICDE'06, Washington, DC, USA: IEEE Computer Society, April 2006, p. 1–11.
27. Zakerzadeh, H., Aggarwal, C.C., Barker, K.: 'Privacy-preserving big data publishing'. Proc. 27th Int. Conf. Scientific and Statistical Database Management, Ser. SSDBM '15, New York: ACM; 2015, p. 26:1–26:11.
28. Roy I, Ramadan HE, Setty STV, Kilzer A, Shmatikov V, Witchel E. Airavat: Security and privacy for MapReduce, In: Castro M, eds. In: Proc. of the 7th Usenix Symp. on Networked Systems Design and Implementation. San Jose: USENIX Association; 2010.
29. Derbeko P, et al. Security and privacy aspects in MapReduce on clouds: a survey. Comput Sci Rev. 2016;20:1.
30. Pathak K, Chaudhari NS, Tiwari A. Privacy preserving association rule mining by introducing concept of impact factor. In: 2012 7th IEEE Conference on Industrial Electronics and Applications (ICIEA), Singapore, 2012, p. 1458–61. https://doi.org/10.1109/iciea.2012.6360953.
31. Yadav GS, Ojha A. Multimed Tools Appl. 2018;77:16319. https://doi.org/10.1007/s11042-017-5200-1.

32. Terzi, R. Terzi, and S. Sagiroglu. A survey on security and privacy issues in Big Data. In Proc. of ICITST 2015, London, UK, December 2015.
33. Kacha L, Zitouni A. An Overview on Data Security in Cloud Computing, CoMeSySo: cybernetics approaches in intelligent systems, Springer, 2017, p. 250–61.
34. Ilavarasi K, Sathiyabhama B. An evolutionary feature set decomposition based anonymization for classification workloads: privacy preserving data mining Journal of cluster computing. New York: Springer; 2017.
35. Acampora G, et al. Data analytics for pervasive health. In: Healthcare data analytics, ISSN: 533-576, 2015.
36. Kulkarni AP, Khandewal M. Survey on hadoop and introduction to YARN. Int J Emerg Technol Adv Eng. 2014;4(5):82–7.
37. Yu E, Deng S. Understanding software ecosystems: a strategic modeling approach, in Proceedings of the Workshop on Software Ecosystems 2011, IWSECO-2011. p. 6-16.
38. Shim K. MapReduce Algorithms for Big Data Analysis, DNIS, LNCS, 2013. p. 44–8.
39. Arora S, Goel DM. Survey Paper on scheduling in hadoop international journal of advanced research in computer science and software engineering. 2014; 4(5).
40. Jain P., Gyanchandani M., Khare N. "Big Data Security and Privacy: New Proposed Model of Big Data with Secured MR Layer", Advanced Computing and Systems for Security. Advances in Intelligent Systems and Computing, vol 883. Springer, Singapore 2019.
41. Sweeney L. K-anonymity: a model for protecting privacy. Int J Uncertain Fuzz. 2002;10(5):55770.
42. Zakerdah CC, Aggarwal KB. Privacy-preserving Big Data publishing. La Jolla: ACM; 2015.
43. Morey T, Forbath T, Schoop A. Customer data: designing for transparency and trust. Harvard Business Rev. 2015;93(5):96–105.
44. Friedman A, Wolff R, Schuster A. Providing k-anonymity in data mining. VLDB J. 2008;17(4):789–804.
45. Fung B, et al. Privacy-preserving data publishing: a survey of recent developments. ACM Comput Surveys (CSUR). 2010;1:42–4.
46. Ko SY, Jeon K, Morales R. The HybrEx model for confidentiality and privacy in cloud computing. In: 3rd USENIX workshop on hot topics in cloud computing, HotCloud'11, Portland; 2011.
47. Apache Hive. http://hive.apache.org. Accessed 18 Mar 2018.
48. ApacheHDFS. http://hadoop.apache.org/hdfs. Accessed 17 Mar 2018.
49. Tweepy dataset online. https://marcobonzanini.com/2015/03/02/mining-twitter-data-with-python-part-1/. Accessed 18 March 2018.
50. Ghinita G, Karras P, Kalnis P, Mamoulis N. Fast data anonymization with low information loss. In: Proc. Int'l Conf. very large data bases (VLDB), p. 758–69, 2007.