A Protection Layer over MapReduce Framework for Big Data Privacy

Hidayath Ali Baig
Department of Information Technology
University of Technology and Applied Sciences
Sur, Sultanate of Oman
Email: hidayath.sur [AT] cas.edu.om

Abstract— **In many organizations, big data analytics has become a trend in gathering valuable data insights. The framework MapReduce, which is generally used for this purpose, has been accepted by most organizations for its exceptional characteristics. However, because of the availability of significant processing resources, dispersed privacy-sensitive details can be collected quickly, increasing the widespread privacy concerns. This article reviews some of the existing research articles on the MapReduce framework's privacy issues and proposes an additional layer of privacy protection over the adopted framework. The data is split into bits and processed in the clouds, and two other steps are taken. Hadoop splits the file into bits of a smaller scale. The task tracker then allocates these bits to several mappers. First, the data is split up into key-value pairs, and the intermediate data sets are generated. The efficiency of the suggested approach may then be effectively interpreted. Overall, the proposed method provides improved scalability. The following figures compare execution time with relation to file size and the number of partitions. As privacy protection technique is used, the loss of data content can be appropriately handled. It has been demonstrated that MRPL outperforms current methods in terms of CPU optimization, memory usage, and reduced information loss. Research reveals that the suggested strategy creates significant advantages for Big Data by enhancing privacy and protection. MRPL can considerably solve the privacy issues in Big Data.**

Keywords- Data Analytics; HADOOP; HDFS; Map Reduce Protection Layer (MRPL); Privacy-preserving.

I. Introduction

There is a sharp rise in public sector data, business organizations, companies, scientific research, academics, and individuals. The vast volume of complex and heterogeneous information from anywhere, any time, and any device is undeniably considered an age of big data [1]. There are many big data platforms available to store and process this information. These platforms and systems are suitable for big data analytics [2]. For many purposes, Big Data analytics is used more commonly. Indeed, these modern ways of implementing analytics will offer organizations innovative changes.

Cloud computing and Big Data, two new disruptive technologies, greatly impact the IT academia and industry cultures [30] [31]. Cloud computing enables vast computing and storage space, allowing users to deploy applications without additional infrastructure, a financial commitment. Data sets have become more valuable as a result of cloud computing. It has grown to such a size and complexity that it is a significant challenge; traditional data processing methods face difficulties in dealing with these data are being analyzed through a system. The word "Big Data" encompasses a broad range of amounts of digital data generated by businesses and government agencies [26] [27] [28] [28] [29] [30]. Every day, a quadrillion byte of data is generated, which means that 90 percent of the data on the planet today was generated in the last two years.

A. Big Data Privacy

The main concern with the availability of such a large amount of data and analytical methods is the confidentiality of information. In the current period of innovative developments, the organizations' data need to be safeguarded [3]. Further research is required on privacy issues in big data. Privacy laws have been put in place to protect the people, but these privacy policies are outdated [4]. Protecting the privacy and maximizing the total value of data should no longer be at odds. We are only scratching the surface of the societal benefits of what big data can bring. Data is everywhere, but it is not being used proficiently despite billions of bytes of data because people are skeptical about information privacy. A general approach to data privacy can lead to liability issues.

It is necessary to put technologically strengthened controls in the data to determine how the data can be used, where, and to what extent. It is essential to maximize the value of data without compromising confidentiality to be used. The benefits of injecting confidentiality into security controls allow sensitive and restricted personal information to be shared in a controlled manner without disrupting individual privacy and minimizing risk.

B. MapReduce Preliminary

MapReduce is a programming model and an accompanying implementation to analyze and generate massive datasets appropriate for a broad spectrum of real-world jobs. A significant handful of firms and organizations have used the MapReduce system to process large datasets [33]. Unlike conventional MapReduce frameworks, cloud-based

MapReduce frameworks are more versatile, scalable, and cost-effective. The Amazon Elastic MapReduce (Amazon EMR) service is a prime illustration. Users specify the map and reduce functions, and the intrinsic runtime system will automatically mirror the computation for large computer clusters, manages software errors, and schedules machine-to-machine communication [5].

A MapReduce role comprises two basic operations, Map and Reduce, described over a key-value pair data structure (key, value). In particular, the Map function can be written as Map: (k1, v1) (k2, v2). The map component uses a pair of inputs (k1, v1) and returns another indirect key-value pair (k2, v2). Reduce function will accumulate these intermediate pairs. Reduce: (k2, list(v2)) (k3, v3) is a formal representation of the Reduce function, which takes intermediary k2 and all of its related variables list(v2) as input and outputs another pair of values (k3, v3). The outcomes that MapReduce users typically attempt to get are the (k3, v3) list. Data users define both the Map and Reduce functions in terms of their particular applications.

MapReduce applications have core frameworks, such as data replication and data sorting, to effectively and efficiently make such a basic programming model function. Furthermore, distributed file systems, such as Hadoop shared file system [34], are critical for making the MapReduce architecture highly scalable and fault-tolerant. The basic MapReduce has currently been extensively revised into several to manage data in various situations. Incoop [35], for example, is designed to support exponential MapReduce computation, while Haloop [36] and Twister [37] are built to assist iterative MapReduce computation.

C. *Execution Overview*

Map invocations are spread through several machines by randomly splitting the input data into a collection of M splits. Different devices will parallel process the input units. The Reduce is the function of partitioning the intermediate keyspace into R bits. The user determines the number of parts (R) and the partitioning function.

When completed, output from MapReduce is available in R output files (one per miniaturization, with user-specified file names). Typically, users do not need to merge these output R files into one file; they frequently transfer these files as inputs to some other MapReduce functions or use them from a different program to manage several file partitioned inputs.

II. Related Work

Much analysis has been undertaken to protect data privacy, and research communities have made substantial strides. In this section, a summary of the recent MapReduce privacy analysis is included.

Researchers and industry propose many privacy protection mechanisms for big data platforms. Baig et al. provided a comprehensive analysis of Big data's existing privacy protection schemes [6]. Since Hadoop version 1.0, the Kerberos authentication mechanism has been implemented into the Hadoop MapReduce system [7]. In essence, access control does not preserve anonymity since data users can presume privacy-sensitive details by accessing unencrypted data. Roy et al. addressed data security in the MapReduce architecture and suggested the Airavat system that allows for data access with compulsory sparse representation [8]. Mandatory access control is allowed when privacy violations exceed a level so that privacy integrity is preserved and data usefulness is maintained. However, this approach's results are associated with inevitable noise, which is insufficient for such applications requiring quiet data sets, like data mining and clinical experiment analysis.

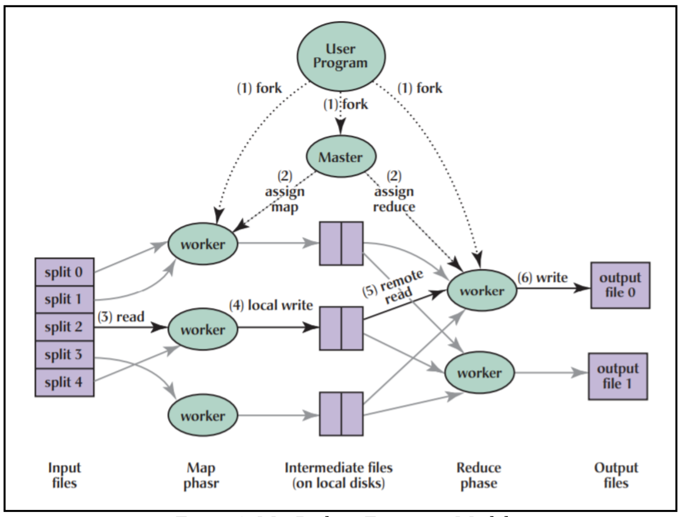

Figure 1: MapReduce Execution Model

Encryption is generally used as an easy workaround for preventing unauthorized users with data privacy in the cloud. Specific activities, such as queries or searches for encrypted cloud-based datasets, have been extensively studied [9]–[11]. Puttaswamy et al. suggest that instead of encrypting all data sets, a series of Silverline tools be developed that can separate all logically encrypted data from any cloud applications [12]. On the other hand, Zhang and colleagues suggested a privacy leakage solution based on the upper limit restriction to secure the privacy of multiple databases by encrypting just part of the datasets throughout the cloud [13]. In particular, various mechanisms have been suggested for the MapReduce platform to solve computation and storage protection issues within the system. A MapReduce, privacy protection scheme, named PRISM that runs parallel word searches for encrypted datasets is proposed by Blass et al. [14].

However, many cloud systems need the MapReduce platform to perform data mining, and data set analysis tasks. The HybrEx paradigm is proposed as a method for sensitive and secure data to be maintained in a secure cloud by Ko et al. [15]. Likewise, Zhang et al. presented Sedic, a system for distributing MapReduce commands in terms of protection labels on that data they are operating on, and then assigns computing to a decentralized cloud that does not include private data [16]. The sensitivity of the data must also be pre-acquired in the two systems described above. Wei et al. suggested a system for safeguarding the integrity of resources for the MapReduce Framework called the SecureMR [17]. Jain et al. proposed a Secure MapReduce Layer for privacy protection that uses encryption and randomization techniques for data anonymization [4]. However, on large datasets, randomization cannot be applied because of the time complexity and lack of data utility [18]. We concentrate primarily on privacy concerns in the proposed layer, which aims to generate anonymized databases in compliance with the data owner's privacy criteria for the MapReduce tasks.

To summarize the above work, the majority of privacy protection methods implemented in the MapReduce system are ccess management [12], encryption [9], auditing [19], and differential privacy [8]. These structures are well-known foundations of privacy protection that remain subject to concerns in the context of big data and cloud computing [20]. Typically, databases submitted to the cloud are used for more than just storage; they can also be used for online cloud functions, making the datasets dynamic. Because the current systems operate only on unencrypted datasets, the effective retrieval of data would be challenging if we encrypt these datasets. In the recent past, significant progress has been made in homomorphic encryption, which technically enables computations to be made on coded data sets. The usage of this approach is costly due to its inefficiency [21].

III. PROPOSED PROTECTION FRAMEWORK.

This section proposes a new approach to protect privacy by designing a layer of protection on the MapReduce framework deployed on the Hadoop ecosystem.

The presented system includes a new privacy layer in the Big Data architecture over the Map-Reduce framework. As a result, this new layer applies data anonymization algorithms independently as data crosses the map-reduce process. A privacy layer known as Map-Reduce Protection Layer (MRPL) is proposed to be implemented between Map Reduce and Hadoop Distributed File System (HDFS) and as shown in Fig. 2.

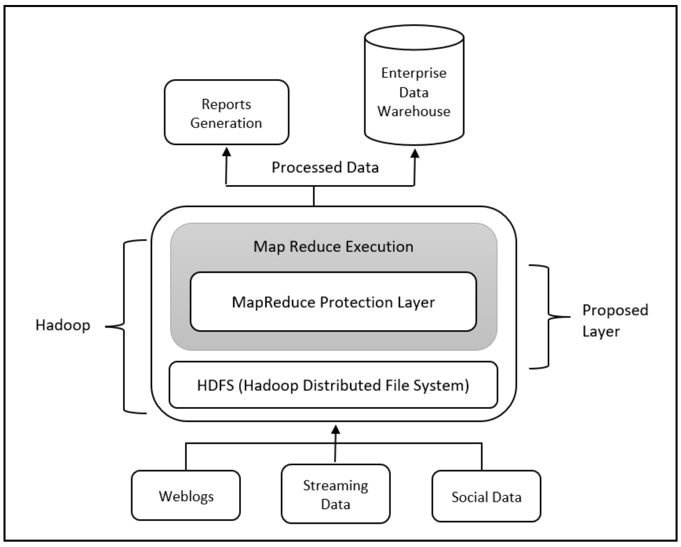

Figure. 2 MapReduce Protection Layer

It begins by collecting the data from different sources and submits the collected data to HDFS. With the proposed MRPL, the data can be securely shared by external data users and uploaded to public clouds.

```
Input: Data set $D$, anonymization level $AL$ and
  $k$-anonymity parameter $k$.
Output: Anonymization level $AL'$.
1: Initialize the values of search metric IGPL, i.e., for each
  specialization $spec \in \cup_{j=1}^{m} Cut_j$. The IGPL value of $spec$ is
  computed by job IGPL Initialization.
2: while $\exists\, spec \in \cup_{j=1}^{m} Cut_j$ is valid
  2.1: Find the best specialization from $AL_i$, $spec_{Best}$.
  2.2: Update $AL_i$ to $AL_{i+1}$.
  2.3: Update information gain of the new specializations
       in $AL_{i+1}$, and privacy loss for each specialization via
       job IGPL Update.
  end while
$AL' \leftarrow AL$.
```

ALGORITHM: TOP-DOWN SPECIALIZATION [22]

In the proposed layer design to achieve anonymization, we adopt the two-phase top-down specialization (TDS) technique proposed by Zhang et al. [22]. In Top-Down Specialization, the properties of the entities are initialized to their root value at the start. The k-anonymization is incremental to protect personal identity. Specialization is done in the taxonomy tree on the value of a child attribute.

IV. PROPOSED METHODOLOGY

The data is split into bits and processed in the clouds in this method, and two other steps are taken. Hadoop splits the file into bits of a smaller scale. The task tracker then allocates these bits to several mappers. First, the data is split up into key-value pairs, and the intermediate data sets are generated. Therefore, intermediate data sets are merged and specialized using the model suggested by Zhang et al. The following algorithm illustrates the process.

Algorithm:

```
Input: Data file DF
Output: Anonymized File
Map Stage:
Partition the dataset DF (DF1, DF2, DF3 ... DFn)
Input: Partitioned Data
Execute Map Function
Produce anonymized Intermediate result using TDS
Reduce Stage:
Merge the intermediate data sets ((DS1, DS2, DS3, . . . . , DSn,) into DS
Execute the TDS to achieve K-Anonymization
Specialize DS according to Anonymity level
Output: Anonymized data set by multiple MapReduce functions.
```

The MRPL is responsible for initial anonymizing datasets according to confidentiality criteria and handling private datasets. In the MapReduce task, data users may define their program logic and execute specific jobs on anonymized datasets. MRPL utilizes MapReduce tasks as the analytical engine. It is reasonable and appropriate to use the MapReduce system to anonymize and handle these datasets, as they appear to be significant in volume in cloud computing and big data.

A. Data Splitting Phase:

The data partition would be executed on the cloud. A vertical division HybrEx Model suggested by Ko et al. is applied with anonymized data when an input file is sent to the master node [23]. The big data sets are reduced to smaller groups of data. The random number for the data set is generated. Partitioning is the process that identifies the correct reducer instances for the intermediate keys and values. The reducer determines all the outputs that the output values go to. Each mapper instance must use the same destination partition regardless of any different mapper instances that generate the same key. If a pair of keys are similarly matched, then they are reduced.

- Phase I (Map Stage)

After individual data sets are created, the anonymization algorithm can be applied. Anonymization is a way to make a data set untraceable. Next, For the minor data sets, the algorithm measures estimation. For specialization, intermediate outcomes are used. An anonymization algorithm (k-anonymity) is used to generate non-readable and irreversible data.

- Phase II (Reduce Stage)

Results from several small data sets are combined in this stage. The TDS algorithm [22] is designed to organize a small intermediate data set into a large output set. Together the information is to be kept in the cloud. The result of the merger is used to anonymize further, called specialization.

B. Deployment Environment

A workstation with core i7 10th generation 64-bit processors and 12 GB of RAM is used to implement the proposed methodology. The MRPL is implemented in Java and run in a virtualized Hadoop multi-node environment.

V. Results and Analysis

Vertical partitioning of the HybrEx model (Gudditti & Krishna, 2021) [24] has been introduced when the input file is sent to the master node. The data is processed in the private cloud at the time of anonymization. Hadoop's mechanism divides the entire file into smaller portions. The work tracker then distributes these parts to several mapper tasks. Every word of a sentence is anonymized with the stated randomization logic within each mapper and written to a file along with the mapper Id of the slave node's corresponding mapper mission. The reducer task will now combine the results of all mapper tasks and create an anonymized file as an output, which will be sent to the public cloud. When anyone requests Confidential data from the public cloud, they will always receive an anonymized response.

The proposed MRLP model's efficiency is measured using

Running time: Running time estimated in time of the clock (milliseconds). The efficiency of the suggested approach may then be effectively interpreted. Overall, the proposed method provides improved scalability. Figure 3 (a-f) compares execution time with relation to file size and the number of partitions.

Loss of information: As privacy protection technique is used, the loss of data content can be appropriately handled. A widely used metric known as usability failure metric is used for collecting all the details lost due to anonymization to preserve the data quality.

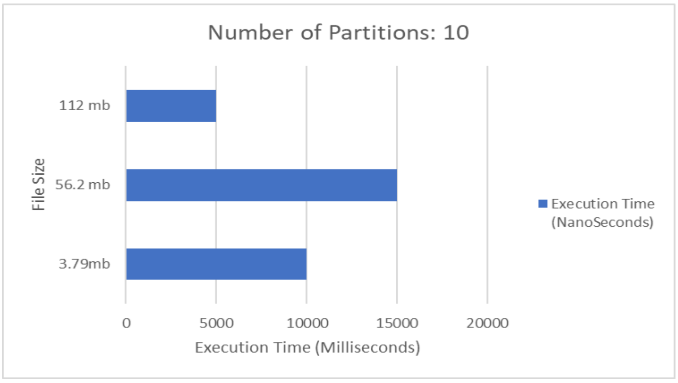

Figure 3a: Comparison of Execution time for a file with ten partitions.

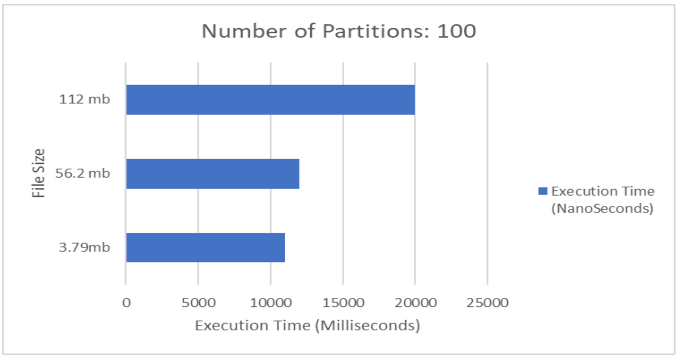

Figure 3b: Comparison of Execution time for a file with 100 partitions.

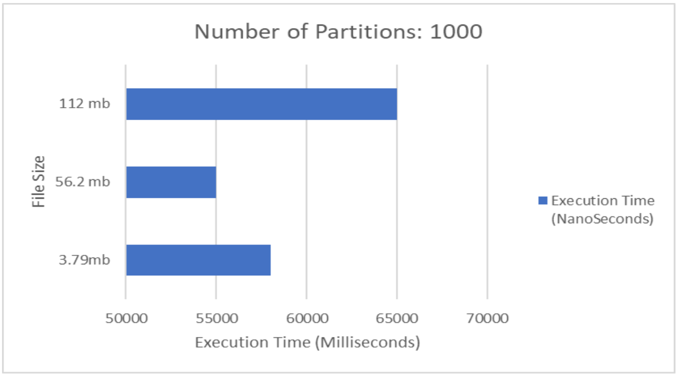

Figure 3c: Comparison of Execution time for a file with 1000 partitions.

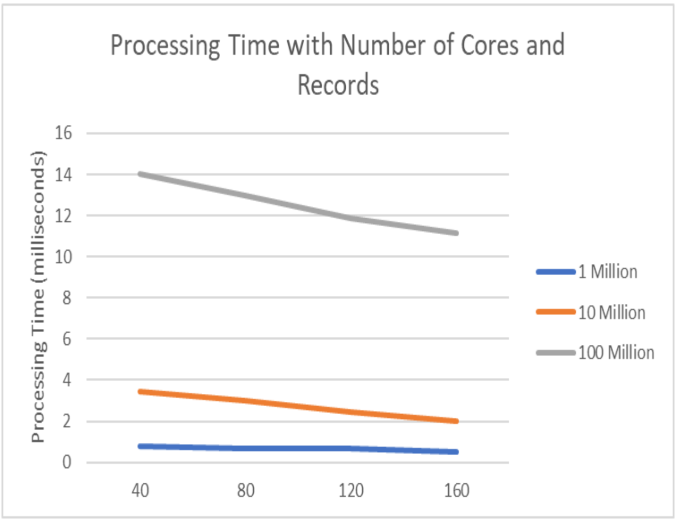

Figure 3d: Processing time with the number of Cores and Records

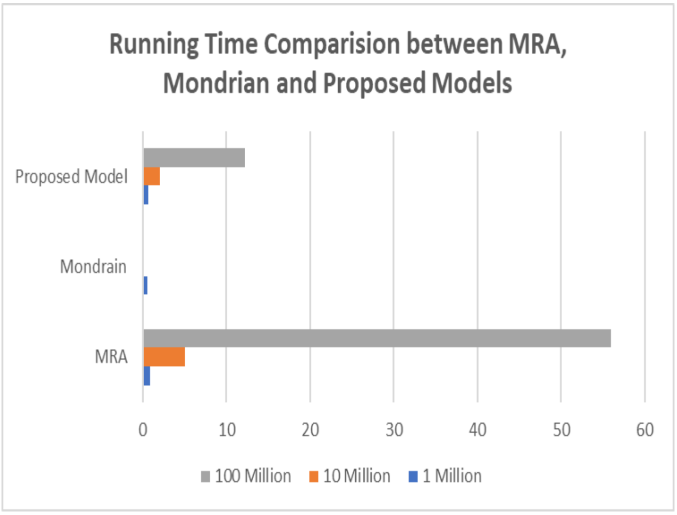

Figure 3e: Run-time Comparison between Proposed MRPL, Mondrian, and MRA.

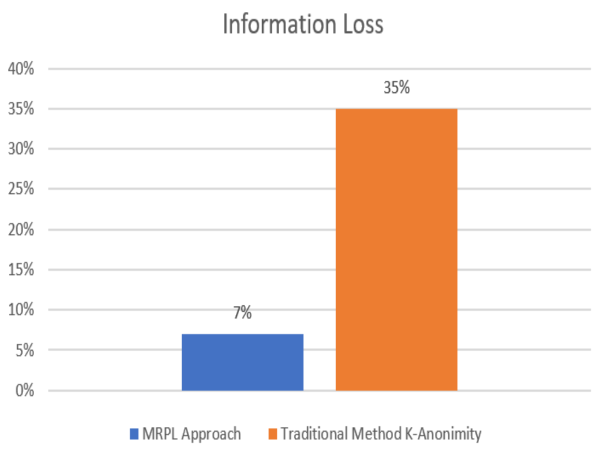

Figure 3f: Comparison of Information loss between MRPL and MRA approaches

TABLE I. COMPARISON OF RUNNING TIME BETWEEN MRA, MONDRIAN, AND PROPOSED MODELS

Data Size	**Time in milliseconds taken concerning data size (ms*10^6)**		
Data volume (number of records)	**Map-reduce Anonymization Algorithm**	**Mondrain**	**Proposed Model**
1 Million	0.86	0.53	0.62
2 Million	5	-	2.09
3 Million	56	-	12.19

VI. CONCLUSION

The emphasis of this research is on the Privacy of Big Data. Scalability of privacy-protection data publishing techniques is a still new field of data privacy. We also analyzed and discussed the shortcomings of some of the current privacy data publication approaches. We suggested a novel technique called MRPL that utilized MapReduce and compared it to current Mondrian and MRA techniques. The MRPL model is focused on data anonymization, which employs methods of top-down specialization to preserve anonymity. The methodological assessment of MRPL was conducted using generalized adult data sets. It has been demonstrated that MRPL outperforms current practices in terms of CPU optimization, memory usage, and reduced information loss. Research reveals that the suggested strategy creates significant advantages for Big Data by enhancing privacy and protection. As data size increases, the runtime gap is significantly reduced compared to current methods. Hence MRPL considerably solves the privacy issues in Big Data. Future research will focus on real-time or streaming Big Data privacy and security.

REFERENCES

[1] O. O. Nathaniel and S. Muhammed, "APPLICATION OF BIG DATA ANALYTICS IN NIGERIA MOBILE TELECOMMUNICATION INDUSTRY," vol. 8, no. 1, p. 12, 2019.

[2] Z. Bi and D. Cochran, "Big data analytics with applications," J. Manag. Anal., vol. 1, no. 4, pp. 249–265, Oct. 2014, DOI: 10.1080/23270012.2014.992985.

[3] D. Y. K. Sharma, "Framework for Privacy-Preserving Classification in Data Mining," vol. 5, no. 9, p. 6, 2018.

[4] P. Jain, M. Gyanchandani, and N. Khare, "Enhanced Secured Map-Reduce layer for Big Data privacy and security," J. Big Data, vol. 6, no. 1, p. 30, Dec. 2019, DOI: 10.1186/s40537-019-0193-4.

[5] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," Commun. ACM, vol. 51, no. 1, pp. 107–113, Jan. 2008, DOI: 10.1145/1327452.1327492.

[6] H. A. Baig, Dr. Y. K. Sharma, and S. Z. Ali, "Privacy-Preserving in Big Data Analytics: State of the Art," SSRN Electron. J., 2020, DOI: 10.2139/ssrn.3713826.

[7] B. C. Neuman and T. Ts, "Kerberos: An Authentication Sewice for Computer Networks," p. 6, 1994.

[8] I. Roy, S. T. V. Setty, A. Kilzer, V. Shmatikov, and E. Witchel, "Airavat: Security and Privacy for MapReduce," Data Min., p. 51, 2010.

[9] N. Cao, C. Wang, M. Li, and K. Ren, "Privacy-Preserving Multi-Keyword Ranked Search over Encrypted Cloud Data," IEEE Trans. PARALLEL Distrib. Syst., vol. 25, no. 1, p. 12, 2014.

[10] V. Hu, T. Grance, D. Ferraiolo, and D. Kuhn, "An Access Control Scheme for Big Data Processing," presented at the 10th IEEE International Conference on Collaborative Computing: Networking, Applications, and Worksharing, Miami, United States, 2014, DOI: 10.4108/icst.collaboration.2014.257649.

[11] M. Li, S. Yu, N. Cao, and W. Lou, "Authorized Private Keyword Search over Encrypted Data in Cloud Computing," p. 10, 2011.

[12] K. P. N. Puttaswamy, C. Kruegel, and B. Y. Zhao, "Silverline: Toward Data Confidentiality in Storage-Intensive Cloud Applications," p. 13, 2011.

[13] X. Zhang, C. Liu, S. Nepal, S. Pandey, and J. Chen, "A Privacy Leakage Upper Bound Constraint-Based Approach for Cost-Effective Privacy Preserving of Intermediate Data Sets in Cloud," IEEE Trans. PARALLEL Distrib. Syst., vol. 24, no. 6, p. 11, 2013.

[14] E.-O. Blass, R. Di Pietro, R. Molva, and M. Önen, “PRISM – Privacy-Preserving Search in MapReduce,” in Privacy Enhancing Technologies, vol. 7384, S. Fischer-Hübner and M. Wright, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 180–200.

[15] S. Y. Ko, K. Jeon, and R. Morales, "The HybrEx Model for Confidentiality and Privacy in Cloud Computing," p. 5, 2011.

[16] K. Zhang, X. Zhou, Y. Chen, X. Wang, and Y. Ruan, "Sedic: privacy-aware data-intensive computing on hybrid clouds," p. 11, 2011.

[17] W. Wei, J. Du, T. Yu, and X. Gu, "SecureMR: A Service Integrity Assurance Framework for MapReduce," p. 10, 2009.

[18] P. Ram Mohan Rao, S. Murali Krishna, and A. P. Siva Kumar, "Privacy preservation techniques in big data analytics: a survey," J. Big Data, vol. 5, no. 1, p. 33, Dec. 2018, DOI: 10.1186/s40537-018-0141-8.

[19] Z. Xiao and Y. Xiao, "Accountable MapReduce in cloud computing," in 2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Shanghai, China, Apr. 2011, pp. 1082–1087, DOI: 10.1109/INFCOMW.2011.5928788.

[20] Chaudhuri, S., "What next? A half-dozen data management research goals for big data and the cloud." Proc. 31st ACM SIGMOD-SIGACT-SIGAI Symp. Princ. Database Syst., p. (pp. 1-4)., May 2012.

[21] C. Gentry, "Fully homomorphic encryption using ideal lattices," in Proceedings of the 41st annual ACM symposium on Symposium on the theory of computing - STOC '09, Bethesda, MD, USA, 2009, p. 169, DOI: 10.1145/1536414.1536440.

[22] X. Zhang, L. T. Yang, C. Liu, and J. Chen, "A scalable two-phase top-down specialization approach for data anonymization using MapReduce on a cloud," IEEE Trans Parallel Distrib Syst, vol. 25, 2014, DOI: 10.1109/TPDS.2013.48.

[23] S. Y. Ko, K. Jeon, and R. Morales, "The HybrEx Model for Confidentiality and Privacy in Cloud Computing," p. 5, 2007.

[24] Gudditti, V., & Krishna, P. V. (2021). Light weight encryption model for map reduce layer to preserve security in the big data and cloud. Materials Today: Proceedings.

[25] Cevher, V., Becker, S. and Schmidt, M., 2014. Convex optimization for big data: Scalable, randomized, and parallel algorithms for big data analytics. IEEE Signal Processing Magazine, 31(5), pp.32-43.

[26] Patel, A.B., Birla, M. and Nair, U., 2012, December. Addressing big data problem using Hadoop and Map Reduce. In 2012 Nirma University International Conference on Engineering (NUiCONE) (pp. 1-5). IEEE.

[27] Lin, J., 2013. MapReduce is good enough? The control project. IEEE Comput, 32.

[28] Groves, P., Kayyali, B., Knott, D. and Kuiken, S.V., 2016. The'big data'revolution in healthcare: Accelerating value and innovation.

[29] Chavan, V. and Phursule, R.N., 2014. Survey paper on big data. Int. J. Comput. Sci. Inf. Technol, 5(6), pp.7932-7939.

[30] Sinanc, S.S., 2013. Big data: A review. In Proc. Int. Conf. CTS.

[31] Borkar, V., Carey, M.J. and Li, C., 2012, March. Inside" Big Data management" ogres, onions, or parfaits?. In Proceedings of the 15th international conference on extending database technology (pp. 3-14).

[32] Chaudhuri, S., 2012, May. What next? A half-dozen data management research goals for big data and the cloud. In Proceedings of the 31st ACM SIGMOD-SIGACT-SIGAI symposium on Principles of Database Systems (pp. 1-4).

[33] Dean, J. and Ghemawat, S., 2010. MapReduce: a flexible data processing tool. Communications of the ACM, 53(1), pp.72-77.

[34] Shvachko, K., Kuang, H., Radia, S. and Chansler, R., 2010, May. The hadoop distributed file system. In 2010 IEEE 26th symposium on mass storage systems and technologies (MSST) (pp. 1-10). Ieee.

[35] Bhatotia, P., Wieder, A., Rodrigues, R., Acar, U.A. and Pasquin, R., 2011, October. Incoop: MapReduce for incremental computations. In Proceedings of the 2nd ACM Symposium on Cloud Computing (pp. 1-14).

[36] Bu, Y., Howe, B., Balazinska, M. and Ernst, M.D., 2012. The HaLoop approach to large-scale iterative data analysis. The VLDB Journal, 21(2), pp.169-190.

[37] Ekanayake, J., Li, H., Zhang, B., Gunarathne, T., Bae, S.H., Qiu, J. and Fox, G., 2010, June. Twister: a runtime for iterative mapreduce. In Proceedings of the 19th ACM international symposium on high performance distributed computing (pp. 810-818).