Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:12:2021
DOI 10.17605/OSF.IO/GPQBE

EFFICIENT SCHEME FOR PRIVACY PRESERVING REAL TIME BIG DATA MINING

ILACHANDRAKAR and VISHWANATH R HULIPALLED

Abstract
With the evolution of Big data, data owners require the assistance of a third party (e.g.,cloud) to store, analyse the data and obtain information at a lower cost. However, maintaining privacy is a challenge in such scenarios. It may reveal sensitive information. The existing research discusses different techniques to implement privacy in original data using anonymization, randomization, and suppression techniques. But those techniques are not scalable, suffers from information loss, does not support real time data and hence not suitable for privacy preserving big data mining. In this research, a novel approach of two level privacy is proposed using pseudonymization and homomorphic encryption in spark framework. Several simulations are carried out on the collected dataset. Through the results obtained, we observed that execution time is reduced by 50%, privacy is enhanced by 10%. This scheme is suitable for both privacy preserving Big Data publishing and mining.

Index Terms— Big data, Privacy Preserving, Real Time, Homomorphic encryption

1 INTRODUCTION

Heterogeneous Big data sets requires better mechanisms to handle and analyse it in real-time. No doubt, Big data technologies like Hadoop developed by Apache gives efficient solutions to analyse Big data [1] [2]. High-end resources are required to store huge data and it incurs huge cost for enterprises to procure these resources and technologies. Cloud computing technology can help data owners to store their data at lesser cost. But since, cloud is a third party, privacy is a big challenge for Big data in cloud. Big data privacy is a major issue as it may disclose sets of sensitive data with regard to hospitals, banks, identity of an individual etc. For example, bank account data can disclose the financial condition of a person, health data can disclose the type of disease a person is suffering from, which is considered as a sensitive data for any individual [3][4].

Data owners take the help of third party to analyze the data and get results by paying lesser cost. Big data analysis is more beneficial if it is published. But, if published to third party, privacy cannot be assured and it remains to be a big challenge in today's scenario [7] [8]. So, to preserve privacy of sensitive Big data, we need techniques which can be called as privacy preserving Big data publishing. Further, third party like Cloud can be used by data owners for data mining, pattern mining, web mining etc. The sensitive mining results should not be disclosed to third party. This can be achieved by using privacy preserving Big data mining. Hadoop is a big innovation for analyzing Big data in which map reduce technique is used to divide the data into different parts and

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:12:2021
DOI 10.17605/OSF.IO/GPQBE

execute in parallel using different mappers. Spark is a Hadoop tool which is an improvement over traditional map reduce, because it is a lot times faster and is able to analyze real time data [12]. In [24], researchers have discussed greedy algorithms for anonymization.

This research aims to ensure data privacy and user privacy. While maintaining the user's privacy, the scheme prevents user from getting any information other than a single physical bit of data. The evaluation of this is based on combining pseudonymization with homomorphic encryption. Specifically, the contributions of this research are as follows.

1. Two level privacy using pseudonymization and holomorphic encryption
2. Homomorphic encryption implementation in Spark cluster.

2 LITERATURE SURVEY

Homomorphic encryption is significant in solving data security problems. In recent times, the requirements of privacy for the digital data and algorithms to process them have increased exponentially. Here, we discuss some related works on privacy preserving techniques for Big data applications.

In [1] [2] [3], researchers opine that calculating quasi-identifiers manually leads to disclosure of explicit identifiers and hence inaccurate results. So, they proposed entropy based detection of quasi-identifiers. The attributes with high entropy is found to be more sensitive. Further, the quasi-identifiers are anonymized along with the explicit identifiers to achieve privacy before publishing data. Similar research carried out in [4] [5] has scalability issues. Research discussed in [13] implemented k-anonymity and l-diversity privacy in Hadoop framework and achieves scalability using Big data technology. But, it still suffers from disclosure attacks and it can be better used for privacy preserving data publishing rather than privacy preserving data mining.

In [6], map reduce framework along with top down specialization technique helps to achieve anonymization. In that work, the large data set is divided into smaller sets and anonymized using different mappers. Later, the results are combined through reducer and next level anonymization is implemented. To achieve it, the data is first clustered using k-means clustering to place similar data items in same partition. It reduces data distortion and achieves some scalability, but it is unable to tackle large data processing problems. Also, the work is prone to several attacks and cannot be used for privacy preserving outsourced mining. In [10], researchers proposed a hybrid approach of top down specialization and up generalization to achieve anonymization on the given datasets. That approach is implemented in map reduce platform.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:12:2021
DOI 10.17605/OSF.IO/GPQBE

For privacy preservation, the researchers in [11] employed machine-learning algorithms. The training data is segmented for analysis in the map reduce framework. Local mappers analyze data in parallel and generate results for the related data set. The reducer then combines the results and applies privacy logic to avoid complexity. Although that study overcomes the challenge of Big Data analytics, it lacks privacy because it employs readily hacked cryptographic approaches. Local sensitive hashing technique is proposed in [7]. That technique is implemented in map reduce framework to handle privacy in Big data. The data is divided and local sensitive hashing technique is implemented on each data set and then results are combined. The privacy technique is parallelized in [15] using map reduce framework. Firstly, huge data set is divided into many small sized clusters. Then, anonymization is used. Later in reducer, the mapper results of anonymized parts of data are combined. The information loss is found to be less thereby enhancing data utility.

In [8], local recoding problem in anonymization of Big data is solved and scalability increased using map reduce framework. The proximity aware clustering is used which assures privacy. A novel technique called FAST is proposed in [12], that uses high speed technique to anonymize Big data streams. In it, researchers have used multithreading technique for parallel computation. The stream is divided into parts and assigned to threads. Each thread processes one set of data. The work gives better performance compared to few works and lesser information loss. Map reduce k-anonymity is introduced in [13]. Researcher have taken Indian election data set from one of its state and applied generalization and suppression in quasi-identifier attributes. The improved k-anonymity implemented in map reduce framework achieved better privacy and performance as compared to k-anonymity. Big data privacy scheme based on optical geometric transformations is discussed in [29]. It achieves better privacy but not suitable for big data sets. In [9], micro aggregation method is used to hide original data, where the explicit identifiers and quasi identifiers values are aggregated to hide original values. Researchers in [6] have discussed various privacy preserving techniques used for classification and their merits and demerits. Summary and performance of existing works is given in Table 1 and Table 2 respectively.

TABLE 1
SUMMARY OF THE WORKS ON PRIVACY IN BIG DATA

[Ref.]	Technique used	Advantage	Limitations
[1]	Two-phase entropy	The quasi-identifiers are anonymized along with explicit identifiers to achieve privacy before publishing data	The work does not include multiple attributes while anonymizing dataset based on quasi-identifier to reduce the

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:12:2021
DOI 10.17605/OSF.IO/GPQBE

			overall complexity of the anonymization process.
[3]	MapReduce based approach	A highly scalable median-finding algorithm combining the idea of the median of medians and histogram technique is proposed and the recursion granularity is controlled to achieve cost-effectiveness	Ensuring privacy preservation of largescale data sets needs to be done.
[8]	Proximity-aware local-recoding anonymization with mapreduce	A scalable two-phase clustering approach consisting of a t-ancestors clustering (similar to k-means) algorithm and a proximity-aware agglomerative clustering algorithm is proposed	Not suitable for preserving privacy in cloud
[25]	Privacy-preserving big data publishing	Addresses privacy problem in big data, evaluates big data components from privacy perspective, privacy risks and protection methods in big data publishing	Complexity with regard to privacy not addressed.
[26]	An Efficient Hybrid Clustering Preserving Differential Privacy	Proposes a novel, effective hybrid k-means clustering preserving differential privacy in Spark, namely Differential Privacy Hybrid k-means (DPHKMS).	Needs to ensure the level of privacy protection further.
[27]	Middle layer for preserving privacy	Based on lightweight encryption, which uses randomization and perturbation methods for maintaining security and integrity.	Focus needed on the privacy and security of Big Data, which is generated in real-time.
[29]	Efficient privacy preservation of big data	Proposed an efficient and scalable non-reversible perturbation algorithm for privacy preservation of big data via optimal geometric	The efficiency and scalability of the algorithm for big datasets not discussed.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:12:2021
DOI 10.17605/OSF.IO/GPQBE

		transformations.	
Proposed	Homomorphic encryption and pseudonymization.	The proposed work aims to ensure data privacy and user privacy.	Needs to test the system for providing privacy in federated cloud system

TABLE 2
PERFORMANCE METRICS COMPARISON ON PRIVACY IN BIG DATA

[Ref.]	Privacy	Information loss	Time
[1]	H	H	M
[3]	M	L	H
[8]	H	L	H
[25]	M	L	M
[26]	H	L	L
[27]	M	H	H
[29]	H	L	H
Proposed	H	H	H

3 PRELIMINARIES

Following are the preliminaries for the scheme proposed in section 4.

3.1 Pseudonymization

Pseudonymization processes personal data in a way that no longer attributes to specific meaning without using extra information. Random codes are used to hide the original identities. In other words, pseudonymization is an information management and de-identification technique by which recognizable data fields inside an information record are replaced by at least one pseudonyms. Pseudonymization does not remove all recognizing data from the information, yet reduces the link ability of a dataset with the identity of an individual.

3.2 Homomorphic Encryption

Homomorphic Encryption (HE) is a technique for encryption that permits any information to remain scrambled while it is being prepared and controlled. It permits clients or a third party (for example, a Cloud service provider) to apply functions on scrambled information without needing to reveal the values of the data. HE resembles different types of open encryption in that it utilizes an open key to encode information and permits just the person with private key decode the information. What separates it from different types of encryption is that it utilizes a mathematical framework to permit

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:12:2021
DOI 10.17605/OSF.IO/GPQBE

elements to perform a variety of computations (or procedures) on the scrambled information. The analysis process on encrypted data is shown in Figure 1. Data owners, who want to store their sensitive data, encrypts the data first and then send to cloud. When user needs some information from stored data, he generates query. Cloud performs the computation on encrypted data, which generates encrypted result. Cloud sends these encrypted results back to user.

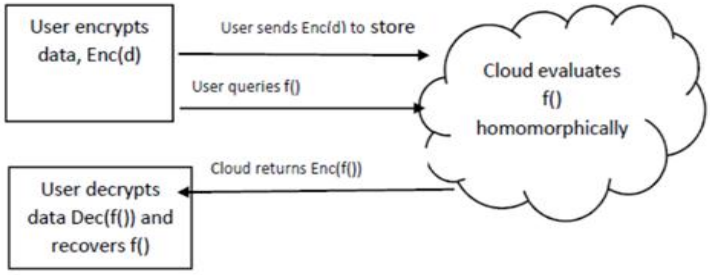

Figure 1: Analysis on homomorphic encrypted data

3.3 Apache Spark

Apache Spark core is an execution engine which provides in-memory computing and can also reference data from an external storage. Spark SQL has schema RDD that handles both structured and unstructured data. The data structure in Spark is Resilient Distributed Data sets (RDDs). RDD is a collection of objects, which is distributed among different nodes in Spark cluster. In Spark, state of memory is stored as objects and these objects can be shared among nodes for different jobs. These objects cannot be changed over time. RDDs can contain any type of Python, Java, or Scala objects, including user-defined classes.

RDD is the read-only partitioned collection of records and can be written only once. RDD stores the state of memory as an object across the jobs and the object is sharable between those jobs. Data sharing in memory is 10 to 100 times faster.

4 PROPOSED REAL TIME BIG DATA PRIVACY SCHEME

We consider a dataset (relevant to Big data application) and apply two-level privacy scheme on the original data to perform pseudonymization and homomorphic encryption. The proposed architecture is shown in Figure 2, and the procedure given in Algorithm 1. The Apache Spark stream data processing is aimed at processing data in batches for faster and parallel execution. Pseudonymization replaces most distinguishing inside an information record by at least one pseudonym. The work is implemented on Spark framework with an aim to improve the performance of the privacy technique.

Algorithm 1

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:12:2021
DOI 10.17605/OSF.IO/GPQBE

Procedure for the Proposed system
Inputs: Huge data stream
Output: encrypted data;
Description:
1. Initially sample dataset is taken from the owner.
2. This dataset contain some unwanted and incomplete data.
3. Using pre-processing, the unwanted/ / incomplete datacan be removed. This step helps to speed up the processof structuring
the dataset on which
pseudonymization canbe applied.
4. The data set is divided into several parts of different sizefor conversion into a stream for real time processing.
5. The explicit identifier attributes are converted into pseudocodes.
6. Encryption is implemented on pseudonymized data onSpark platform.
7. Data owner takes the cloud services to store this encrypteddata. Since it is homomorphic encryption, data can beencrypted
leading to privacy preservation.
8. Spark technique of Hadoop is used to parallelize thehomomorphic encryption for real-time data. It returns the encrypted data.

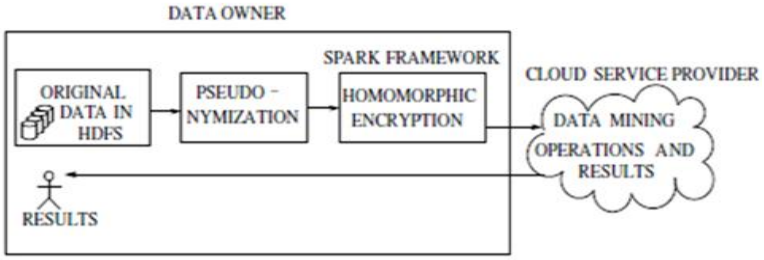

Fig. 2 : Proposed Model

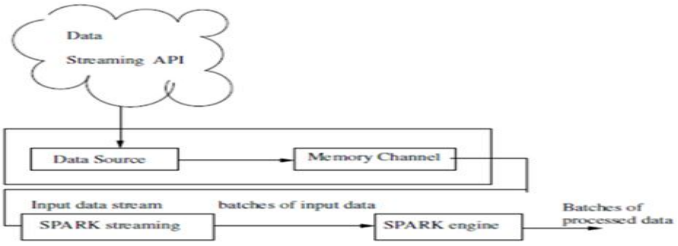

Fig. 3: Apache Spark Stream Data Processing

The data set used in proposed work is online retail data of several transactions of gift items in UK online ecommerce website for two years duration available at Kaggle [29]. This data has fields such as Invoice Number, Stock Code, Description, Quantity, Invoice

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:12:2021
DOI 10.17605/OSF.IO/GPQBE

Date, Unit Price, Customer Id and Country. It has 54900 tuples. Data is pre-processed to remove missing values and outliers. In the first step of the work, pseudonymization is applied to the pre-processed data. Algorithm 2 is used for pseudonymization. In the pseudonymized data set, the sensitive explicit identifier values are replaced by pseudo-strings for that value.

Consider the sample dataset of online retail in Table 3. Let us assume that the attribute 'Description' is the sensitive attribute and explicit identifier which discloses the identity of the particular product. There are total of 4862 unique products in the data set. We generate the dictionary of three letter random words for each of those products. This dictionary is created using python random string generation. To pseudonymize the product name, we generate the same number of random strings as products. So, each product name is pseudonymized to a different string. These strings are three letter random words. The random pseudo-numbers generated for the considered dataset are: ['YWS', 'RDF', 'ILA', 'UTM', 'FHP', SHU, TUY]. The pseudonymized data is given in Table IV. The original data is now seen to be secured as its identity is hidden.

TABLE 3 SAMPLE DATASET

InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
536365	85123A	WHITE HANGING HEAR	6	12/1/2010 8:26	2.55	17850	United Kingdom
536365	71053	WHITE METAL LANTERN	6	12/1/2010 8:26	3.39	17850	United Kingdom
536365	84406B	CREAM CUPID HEARTS	8	12/1/2010 8:26	2.75	17850	United Kingdom
536365	84029G	KNITTED UNION FLAG	6	12/1/2010 8:26	3.39	17850	United Kingdom
536365	84029E	RED WOOLLY HOTTIE W	6	12/1/2010 8:26	3.39	17850	United Kingdom
536365	22752	SET 7 BABUSHKA NEST	2	12/1/2010 8:26	7.65	17850	United Kingdom

TABLE 4 PSEUDONYMIZED DATA

InvoiceNo	StockCode	Descriptic	Quantity	InvoiceDa	UnitPrice	Customer	Country
536365	85123A	YWS	6	40513.35	2.55	17850	United Kingdom
536365	71053	RDF	6	40513.35	3.39	17850	United Kingdom
536365	84406B	ILA	8	40513.35	2.75	17850	United Kingdom
536365	84029G	UTM	6	40513.35	3.39	17850	United Kingdom
536365	84029E	FHP	6	40513.35	3.39	17850	United Kingdom
536365	22752	SHU	2	40513.35	7.65	17850	United Kingdom

Partial homomorphic encryption is applied in pseudonymized data, as it is possible to add and multiply encrypted values. In this work, multiplicative homomorphic encryption

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:12:2021
DOI 10.17605/OSF.IO/GPQBE

is used. The steps in multiplicative homomorphic encryption algorithm is explained in Algorithm 3.

Algorithm 3: Multiplicative homomorphic encryption algorithm
Input: Pseudonymized Data
Output: Encrypted Data
Step 1: Key Generation

- Generate two random large prime numbers ‘a’ and ‘b'.
- Calculate modulus $n = a \times b$
- Calculate PHI, $\emptyset(n) = (a-1)(b-1)$
- Choose an integer e such that 1<e<∅(n) and e is coprime to ∅(n)
- Compute d so that $de = 1(mod\ \emptyset(n))$

Public key is(n,e) and private key is (n,d).
*Step 2: Encryptio*n:

- If m is the message then cipher text can be calculated as $c = m^e\ mod\ n$
- Two ciphers are generated as $c1 = m1^e\ mod\ n$ and $c2 = m2^e\ mod\ n$.
- Calculate multiplication of ciphers as$C = (c1 \times c2)^e\ mod\ n$.

Step 3: Decryption:

- Message can be decrypted as $m = c^d\ mod\ n$
- Multiplicative message can be decrypted as $m = (c1 \times c2)^d\ mod\ n$

Spark streaming is added as an add-on to the core Spark API for the processing of live data streams with the features of increased scalability, high throughput and fault tolerance. Spark streaming divides live stream data into chunks called as Dstreams which is a sequence of RDDs, which are subsequently processed to form a final stream. The Dstreams are ingested to Worker nodes of spark cluster for parallel processing. Spark supports in memory processing so it is faster than Hadoop - 100 times for data in RAM and upto 10 times for data in storage. The process of RDD creation is shown in figure 4. Steps in implementation of Spark streaming are:

- Create a Streaming context with two execution threads and interval of 2 seconds. This Dstream is connected to a TCP port.
- Apply transformation and output operations to DStreams to define the streaming computations.
- Use streamingContext.start() to begin receiving and processing data.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:12:2021
DOI 10.17605/OSF.IO/GPQBE

- Wait for the processing to be stopped using streamingContext.awaitTermination() or stop it manually using streamingContext.stop().

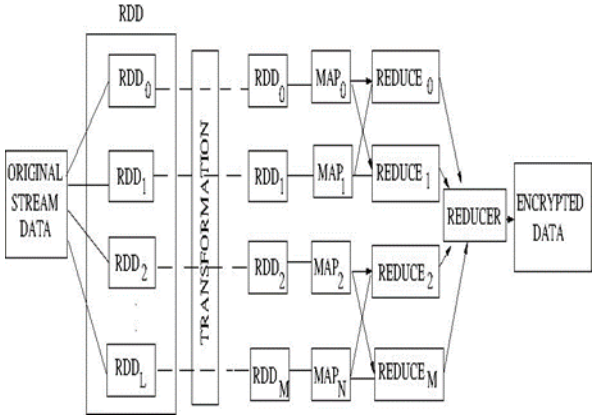

Fig. 4: Sequence of RDDs

5 RESULTS AND DISCUSSIONS

We have used Databricks [29] as spark cluster. DataBricks is an open and unified data analytics platform for data engineering, data science, machine learning, and analytics. This is developed by Apache. It is created by the original creators of Apache Spark, Delta lake, MLflow, and Koalas. Simulation is carried out in 15.8 GB memory with 2 GB RAM. Following are the parameters assessed:

- Running time: The running time is referred to as the time for the encryption and decryption process.
- Privacy: Privacy refers to the protection of information on individual data.
- Information loss: It is the amount of loss in data after encryption/decryption process.

The homomorphic encryption algorithm is run on different size of pseudonymized data in spark cluster. Figure 5 shows evaluation of time for encryption in minutes among various algorithms on different size of data. Research works such as [25] and [27] use Hadoop for parallel processing on data stored in hard disk. Those works are faster than traditional single system computation but not very fast when data size increases with very high speed. The proposed system is seen to perform encryption/decryption faster owing to application of spark in memory processing and is also scalable for the same reason. The execution time is reduced by 50% than existing techniques.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:12:2021
DOI 10.17605/OSF.IO/GPQBE

Between HDFS and the map reduce layer, a secure layer was implemented in [27]. Perturbation and Randomization techniques were used in this research to implement privacy in original data on map reduce framework. Better performance is achieved in [27] as compared to work in [25] as lightweight algorithms are used but since perturbation is used, data is modified and which caused decrease in data utility. The work in [27] has only one level of privacy. This technique was able to provide privacy while publishing the big data but cannot be used for privacy in big data mining because of perturbed data. In the proposed work, we have used two level privacy as pseudonymization and homomorphic encryption, so it achieves 85% privacy. Figure 6 shows the comparison of privacy for existing techniques and proposed scheme.

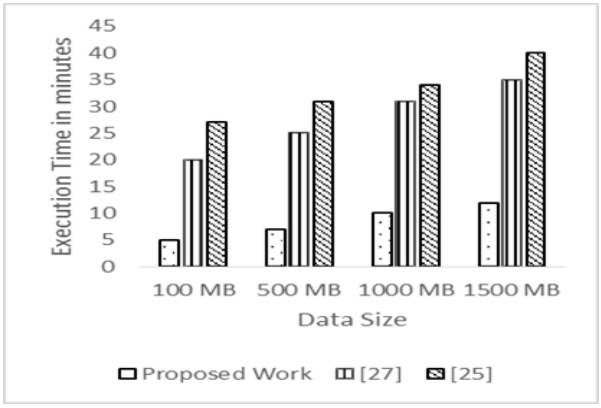

Fig. 5: Data size vs. Execution Time

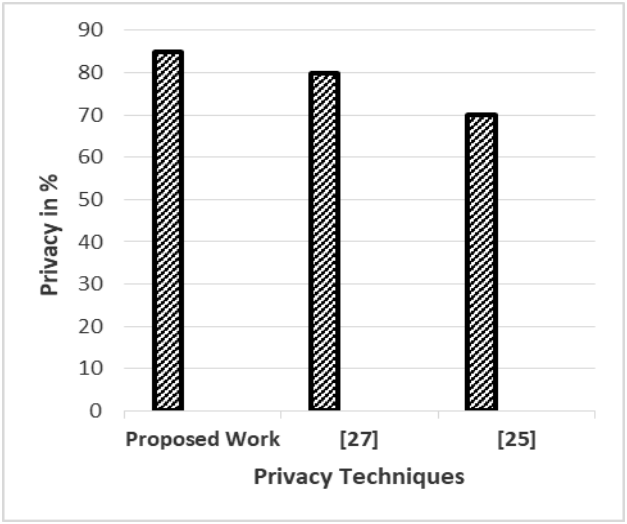

Fig. 5: Data size vs. Execution Time

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:12:2021
DOI 10.17605/OSF.IO/GPQBE

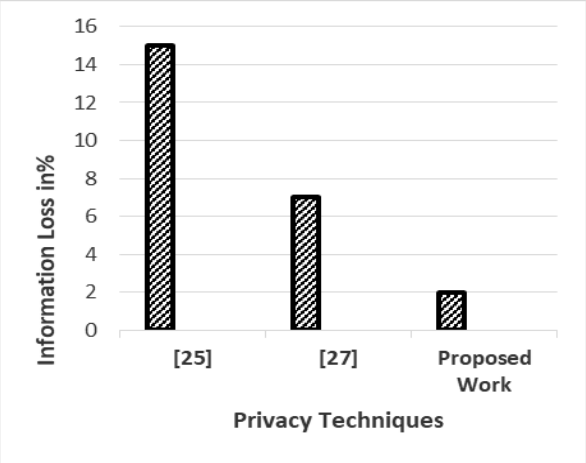

Fig. 7: Information Loss vs. Privacy Technique

Our work can be used to provide privacy while executing data mining algorithms also along with privacy preserving data publishing because of allowed computation on encrypted data using homomorphic encryption. Proposed work uses spark for homomorphic encryption implementation. The in-memory processing of Spark is used for faster access. So,it supports real time privacy preserving of streaming big data without storing. This is very beneficial, as maximum data in modern times is generated in real time. The other techniques discussed in [25] and [27] do not support privacy on real time data. Table V shows the comparison on application of different techniques.

TABLE 5
COMPARISON ON APPLICATIONS

Technique	**Privacy Preserving Big data publishing**	**Privacy Preserving Big data mining**	**Preserving Privacy on real time data**
[25]	Yes	No	No
[27]	Yes	No	No
Proposed Work	Yes	Yes	Yes

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:12:2021
DOI 10.17605/OSF.IO/GPQBE

5 CONCLUSION

The proposed work aims to ensure data privacy and user privacy using homomorphic encryption and pseudonymization, respectively. Owing to usage of two level privacy, the proposed scheme is robust than other techniques in terms of privacy. Further, the usage of Spark for running homomorphic encryption leads to higher scalability and support for preserving privacy in real time data. On using the proposed scheme for privacy preserving big data mining in cloud for third party, privacy of original data is ensured. Demonstrating the feasibility of the proposed scheme in heterogeneous settings such as federated cloud computing environment will be the focus of future research.

REFERENCES

[1] A. Ranjan and P. Ranjan, "Two-phase entropy based approach to Big data anonymization", Proceedings of the International Conference on Computing, Communication and Automation (ICCCA), 2016, pp. 76-81.
[2] D. S. Terzi, R. Terzi, and S. Sagiroglu, "A survey on security and privacy issues in Big data", Proceedings of the International Conference for Internet Technology and Secured Transactions (ICITST), 2015, pp. 202-207.
[3] X. Zhang, C. Yang, S. Nepal, C. Liu, W. Dou, and J. Chen, "A MapReduce based approach of scalable multidimensional anonymization for Big data privacy preservation on cloud", Proceedings of the International Conference on cloud and Green Computing (CGC), 2013, pp. 105-112.
[4] W. Li and H. Li, "LRDM: Local record-driving mechanism for Big data privacy preservation in social networks", Journal of Data Science in Cyberspace (DSC), 2016, pp. 556-560.
[5] B. C. Fung, K. Wang, A. W. C. Fu, and S. Y. Philip, "Introduction to privacy-preserving data publishing: Concepts and techniques", CRC Press", 2010.
[6] DediGunawan, “Classification of Privacy Preserving Data Mining Algorithms: A Review”, JurnalElektronikadan Telekomunikasi, Vol 2, No. 20, 2020.
[7] X. Zhang, C. Leckie, W. Dou, J. Chen, R. Kotagiri, and Z. Salcic, "Scalable local-recoding anonymization using locality sensitive hashing for Big data privacy preservation", Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, 2016, pp. 1793-1802.
[8] X. Zhang, W. Dou, J. Pei, S. Nepal, C. Yang, and C. Liu, "Proximity-aware local-recoding anonymization with mapreduce for scalable Big data privacy preservation in cloud", IEEE Transactions on Computers, Vol. 64, pp. 2293-2307, 2018.
[9] Jae-Seong Lee, Seung-Pyo Jun, “Privacy-preserving data mining for open government data from heterogeneous sources”, Government Information Quarterly,Volume 38, Issue 1, 2021, 101544, https://doi.org/10.1016/j.giq.2020.101544
[10] X. Zhang, C. Liu, S. Nepal, C. Yang, W. Dou, and J. Chen, "A hybrid approach for scalable sub-tree anonymization over Big data using MapReduce on cloud", Journal of Computer and System Sciences, Vol. 80, pp. 1008-1020, 2014.
[11] K. Xu, H. Yue, L. Guo, Y. Guo, and Y. Fang, "Privacy-preserving machine learning algorithms for Big data systems", Proceedings of the IEEE 35th International Conference on Distributed Computing Systems (ICDCS), 2015, pp. 318-327.
[12] E. Mohammadian, M. Noferesti, and R. Jalili, "FAST: Fast anonymization of Big data streams", Proceedings of the 2014 International Conference on Big Data Science and Computing, pp. 23-34, 2014.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:54 Issue:12:2021
DOI 10.17605/OSF.IO/GPQBE

[13] P. Jain, M. Gyanchandani, and N. Khare, "Improved k-Anonymity privacy-preserving algorithm using Madhya Pradesh state election commission Big data", Studies in Computational Intelligence, Vol. 771. Springer, Singapore, pp. 1-10, 2019.
[14] A. Mehmood, I. Natgunanathan, Y. Xiang, G. Hua, and S. Guo, "Protection of Big data privacy", IEEE access, Vol. 4, pp. 1821-1834, 2016.
[15] N. P. KS and T. Pratheek, "Providing anonymity using top down specialization on Big Data using Hadoop framework", Annual IEEE India Conference (INDICON), 2015, pp. 1-6.
[16] H. K. Patil and R. Seshadri, "Big data security and privacy issues in healthcare", IEEE International Congress on Big Data (BigData Congress), 2014, pp. 762-765.
[17] N. Victor, D. Lopez, and J. H. Abawajy, "Privacy models for Big data: a survey", International Journal of Big Data Intelligence, Vol. 3, pp. 61-75, 2016.
[18] I. Olaronke and O. Oluwaseun, "Big data in healthcare: Prospects, challenges and resolutions", Future Technologies Conference (FTC), 2016, pp. 1152-1157.
[19] M. Tanwar, R. Duggal, and S. K. Khatri, "Unravelling unstructured data: A wealth of information in Big data", Proceedings of the 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions), 2015, pp. 1-6.
[20] L. Hbibi and H. Barka, "Big Data: Framework and issues", Proceedings of the International Conference on Electrical and Information Technologies (ICEIT), 2016, pp. 485-490.
[21] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, and M. McCauley, "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing", Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation, 2012, pp. 2-2.
[22] Apache Spark, Available at http://spark.apache.org/, Accessed on Sep. 9, 2020.
[23] S. Kavitha, S. Yamini, and R. Vadhana, "An evaluation on Big data generalization using k-Anonymity algorithm on cloud", Proceedings of the IEEE 9th International Conference on Intelligent Systems and Control (ISCO), 2015, pp. 1-5.
[24] K. LeFevre, DJ DeWitt, and R. R. Mondrian, "Multidimensional k-anonymity", Proceedings of 22nd International Conference on Data Engineering, Washington, DC, USA: IEEE Computer Society, April 2006, pp. 1-11.
[25] H. Zakerzadeh, C.C. Aggarwal, and K. Barker, "Privacy-preserving big data publishing", Proceedings of 27th International Conference on Scientific and Statistical Database Management, New York, 2015, pp. 11-26.
[26] Z.Q. Gao and L.J. Zhang, "DPHKMS: An Efficient Hybrid Clustering Preserving Differential Privacy in Spark", Proceedings of the International Conference on Emerging Internetworking, 2017, pp. 367-377.
[27] P. Jain, M. Gyanchandani, and N. Khare, "Enhanced Secured Map Reduce layer for Big Data privacy and security", Journal of Big Data, Springer, Vol. 6, no. 30, 2019, pp. 1-17.
[28] Available at https://www.kaggle.com/ mathchi/ online-retail-ii-data-set, "Online Retail sales dataset", Accessed on September 29, 2020.
[29] M.A.P. Chamikara, P. Bertok, D. Liu, S. Camtepe, I. Khalil, “Efficient privacy preservation of big data for accurate data mining”, Information Sciences, Volume 527, 2020, pp. 420-443.