International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2019 IJSRCSEIT | Volume 5 | Issue 1 | ISSN : 2456-3307
DOI : https://doi.org/10.32628/CSEIT19516

Big Data Anonymization in Cloud using k-Anonymity Algorithm using Map Reduce Framework

Anushree Raj[1], Rio G L D'Souza[2]
[1]Department of M.Sc. Big Data Analytics, St Agnes Autonomous College, Mangalore, Karnataka, India
[2]Department of Computer Science and Engineering, St Joseph Engineering College, Mangalore, Karnataka, India

ABSTRACT

Anonymization techniques are enforced to provide privacy protection for the data published on cloud. These techniques include various algorithms to generalize or suppress the data. Top Down Specification in k anonymity is the best generalization algorithm for data anonymization. As the data increases on cloud, data analysis becomes very tedious. Map reduce framework can be adapted to process on these huge amount of Big Data. We implement generalized method using Map phase and Reduce Phase for data anonymization on cloud in two different phases of Top Down Specification.

Keywords: Anonymization, Big Data in cloud, k-Anonymity, Map Reduce, Privacy Preserving

I. INTRODUCTION

Security and privacy are a significant obstacle that is preventing the extensive adaptation of public cloud [1]. Cloud Computing is a revolutionary computing approach, which provides massive storage and computational capability [2]. It provides user to implement applications cost effectively without investment and infrastructure. It enables users to have flexible, on demand access to computing resources via Internet [3]. These advantages are the causes of security and privacy problems, which emerge because data owned by different users are stored in cloud servers and users are losing their own control on data.

Data anonymization is the process of hiding identity or sensitive data from original data. Various algorithms are in use for implementation [4], [5], [6], [7]. The most efficient algorithm for data anonymization is the Top Down Specialization algorithm [8]. Using data anonymization key pieces of confidential data are obscured in a way that maintains data privacy. Data can still be processed to gain useful information. Anonymization data can be stored in cloud and even processed without concern that other individuals may capture the data [9]. Later, the results can be collected and mapped to the original data in a secure area.

In this paper we propose an anonymization model using the Top Down Specification techniques on huge amount of data along with Map reduce framework. The Top Down Specification has two phases which includes the map phase which reduces the huge amount of data into small chunks of data and the reduce phase combines the intermediate results to give back the k anonymous data [10]. Anonymization takes place at both the phases. Generalization method is used on Data that usually contain sensitive information and this proves the importance of employing anonymity approaches.

Generalization is the most common method to make the data anonymous to provide the privacy. In this paper we discuss on data anonymization technique in section II and then show the implementation using Map Reduce in section III.

II. DATA ANONYMIZATION

Data anonymizaton is a deep study, broadly adopted for privacy preserving in data publishing. Anonymization techniques highly focus on data security in public cloud [11]. The main feature of anonymization techniques is hiding few key data to maintain data privacy. Data can further be processed to achieve necessary information by analysing it. The simple way of preserving privacy is by removing all those information, which will directly link to data of an individual. This process s referred to anonymization [12]. There are many approaches to gain data anoymization, such as k-Anonymity, l-Diversity and t-Closeness which helps to make the data more private and anonymous, to protect data access from any third party users.

k- Anonymity:
k-Anonymity is used to hide some key information of the users in a database of k users. Then the attackers can make sure there are only k users but cannot interpret the information of an individual user in particular [13], [14]. This method is used to anonymize the quasi-identifiers to provide privacy to the data. The main aim of k-Anonymity is to make each record indistinguishable from at least k-1 other records. k-Anonymity assigns properties in specific ways [15]. The k anonymity data attributes are of three types

Key Attribute: Attribute which can identify n individual directly.
Quasi - identifier: Attribute which can be linked with external information to identify an individual.
Sensitive Attribute: Attribute or data that an individual is sensitive about revealing.

There are two main ways through which k-Anonymity can be achieved:
Bottom Up Generalization: In Bottom-Up Generalization strategy, the data is initialized to its current state and generalizations are carried out over attribute values until k-Anonymity is not violated [16].

Top Down Specialization: In Top-Down Specialization strategy, all the attribute values are initialized to the root value of the hierarchy tree [17]. The specialization is carried out iteratively over the attribute values, until the k-Anonymity is violated.
Top Down Specification

Let D be a data set to which the anonymization technique is applied. R denotes the records in the data set. Let A be the attributes, which can identify N individuals through A_i where $1 \leq i \leq N$, k is the anonymity parameter used in the anonymization levels. Anonymization levels are the single mapreduce job execution level from multiple mapreduce jobs. The domain values are represented by the taxonomy tree [18],[21]. Q determines the quasi identifier. A record can contain any N number of Quasi Identifiers, $Q_i \in R$, $1 \leq i \leq N$. the sensitive values are denoted by S. In Top Down Specification, a data set is anonymized through specialization operations. The domain values are replaced with all its sub domain values.

In Top Down Specialization, a hierarchy tree is used to represent the attribute values which is initiated by a root value. The specialization performed iteratively over the attribute values, till the k - anonymization is satisfied. The specialization is performed by substitution of the parent attribute value by its child value in taxonomy tree. For example, the patient record of a hospital is anonymized and published for

data analysis using Top Down Specialization which is a generalization algorithm then Table I and II represent the anonymization for the quasi identifiers. Here Q = {AGE, SEX}

Table 1. Original Data

NAME	AGE	SEX	EDUCATION	PROFESSION
SAM	34	M	MASTER	BUSINESS
FUNG	45	M	DOCTRATE	DOCTOR
SONA	53	F	MASTER	PRINCIPAL
RAFI	25	M	BACHELORS	HR
ZAMA	37	F	MASTERS	MANAGER
ALICE	28	F	BATCHELORS	BUSINESS
BOB	52	M	DOCTRATE	PROFESSOR

The specialization is performed by replacing the parent attribute value by its child value in Taxonomy Tree

TAXONAMY TREES AND QUASI IDENTIFIERS

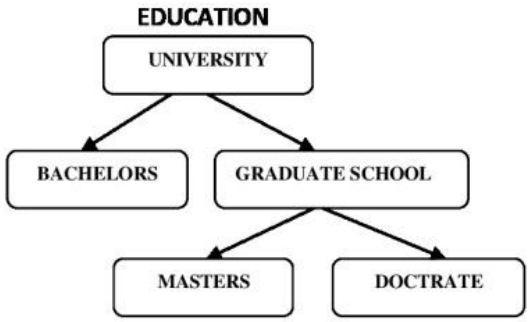

Figure 1. Taxonomy tree for Education

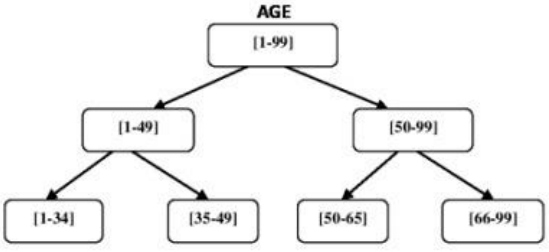

Figure 2. Taxonomy tree for Age

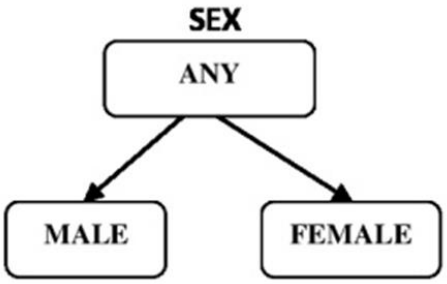

Figure 3. Taxonomy tree for Sex

Table 2. Anonymized data

NAME	AGE	SEX	EDUCATION	PROFESSION
*	[1-34]	ANY	GRADUATE SCHOOL	BUSINESS
*	[35-49]	ANY	GRADUATE SCHOOL	DOCTOR
*	[50-63]	ANY	GRADUATE SCHOOL	PRINCIPAL
*	[1-34]	ANY	BACHELORS	HR
*	[35-49]	ANY	GRADUATE SCHOOL	MANAGER
*	[1-34]	ANY	BACHELORS	BUSINESS
*	[50-63]	ANY	GRADUATE SCHOOL	PROFESSOR

The above table describes the anonymization using generalization algorithms like TDS for the selected Quasi Identifiers with the taxonomy tree hierarchy.
Algorithm 1 TDS

1) Identify every value in *HT* to the top most value of the hierarchy tree.
2) Let *ListAttribute*$_i$ initially include the top most value.
3) While some *ca*(categorical attribute) in *ListAttribute*$_i$ is valid and useful, do step 4, 5, 6
4) Find the Best specialization from *ListAttribute*$_i$
5) Perform Best on *HT* and update *ListAttribute*$_i$

6) Update InfoGain(*ca*) and AnonymityLossValue(*ca*), and validity for *ca* ∈ $ListAttribute_i$
7) End while
8) return Generalized *HT* and $ListAttribute_i$

Initially $ListAttribute_i$ contains only the topmost value $ListAttribute_i$, in which case it returns the masked table together with $ListAttribute_i$. In order to maintain the data utility we have to find the best specialization, otherwise the data utility will decrease considerably and can leave data useless for mining purpose.

The metrics used to find the best specialization are, the Information Gain (*InfoGain*) and Anonymity loss value (*AnonymityLossValue*). For each specialization SP:P child (P) is calculated.
The Information Gain (*InfoGain*) and Anonymity loss value (*AnonymityLossValue*) values are calculated to generalize the record R ϵ D using the SC(R).

$$\mathrm{SC}(R) = \begin{cases} \dfrac{InfoGain(R)}{AnonymityLossValue(R)} & \text{if } AnonymityLossValue(R) \neq 0 \\ InfoGain(R) & \text{else} \end{cases} \quad \ldots\ldots\ldots\ldots\ldots\ldots (1)$$

1) Information gain,
IG(R) is only positive values

$$\mathrm{IG(R)} = I(Rc) - \sum_{c} \left(\frac{|\mathrm{Rc}|}{|\mathrm{Rr}|}\right) \mathrm{I(Rc)} \quad \ldots\ldots\ldots\ldots\ldots\ldots (2)$$

I(Rx) is the entropy measure of Rx where Rx is the number of records. (IRx, Sl)is the records with sensitive value and entropy is computed by,

$$I(Rx) = -\sum_{c} \left(\frac{|\mathrm{Rca, S}|}{|\mathrm{Rr}|}\right) \mathrm{X} \log 2 \left(\frac{|\mathrm{Rca, S}|}{|\mathrm{Rr}|}\right) \quad \ldots\ldots\ldots\ldots\ldots\ldots (3)$$

2) Anonymity Loss Value (*AnonymityLossValue*)
Here the anonymity before specialization is denoted by BeforSpecfication(Q) and after specialization is denoted by AfterSpecification(Q). BeforSpecfication(Q) - AfterSpecification(Q) is the loss of anonymity.
AnonymityLossValue(v) = $AVERAGE_{loss}$*(BeforSpecfication(Q) - AfterSpecification(Q))*

III. MAP REDUCE FRAMEWORK

Map Reduce is a programming model, which computes large volume of data sets with parallel and distributed algorithm on cluster. A Map Reduce program is composed of two user defined functions [22]. A Map function which performs sorting of data and Reduce function that performs summary operation. Map Reduce is a framework for processing parallelizable problems across large datasets using large number of computers.

A typical MR-MPI program makes at least three calls to the MapReduce -MPI library, to perform map(), collate(), and reduce() operations on a MapReduce object it creates. In a map() key-value pairs are generated by each processor. The key-value pairs produced are stored locally by each processor; a map() thus requires no inter-processor communication. Users call the library with a count of tasks to perform and a pointer to a user function; the MapReduce -MPI map() operation invokes the user function multiple times as a callback. Depending on which variant of map() is called, the user function may be passed a file name, a chunk of bytes from a large file, a task ID, or a v pair. Options for assigning map tasks to processors are specified by the user and include assigning consecutive chunks of tasks to processors, striding the tasks across processors, or using a master-slave model that is useful when tasks have widely varying workloads [19].

The basic data stored and operated on by any MapReduce framework are key-value pairs [20]. In the MapReduce -MPI library, individual keys or

values can be of any data type or length, or combinations of multiple types (one integer, a string of characters, two integers and a double, etc); they are simply treated as byte strings by the library. A key-value pair always has a key; its value may be NULL. Key-value pairs are stored within a MapReduce object. A user program may create one or more MapReduce objects to implement an algorithm [21]. Various MapReduce operations (map, reduce, etc) are invoked on an object and its key-value pairs; key-value pairs can also be passed between and combined across objects.

Conceptually the map and reduce functions supplied by the user have associated types.
map (k1,v1) → list(k2,v2)
reduce (k2,list(v2)) → list(v2)

Map Reduce approach involves the following four steps:

1. Map processors, assigns the K1 input Key value, and provides the processor with all the input data associated with that key value.
2. Run the user provided Map() exactly once for each key value, generating output organized by key values K2.
3. Reduce processors, assigns the K2 key value with all the Map generated data associated with that key value.
4. Run the user provided Reduce () exactly once for each K2 key value produced by the Map
5. Produce the final output –The Map Reduce system collects all the reduce output and sorts it by K2 to produce final outcome

IV. CONCLUSION

Privacy is very important to protect the sensitive data from the attacker. To provide data privacy the anonymization methods may be used. The data mining activities utilize huge amount of data gathered by enterprises and government agencies. Such data is subjected to, be abused by internal and external adversaries. Many anonymization techniques came into existence. They are k-Anonymity, l-Diversity, t-Closeness and m-Privacy. These privacy preserving data mining algorithms cannot work for big data analytics, as such data is computed using MapReduce programming paradigm in cloud computing environment. MapReduce is used to control the parallel processing power of cloud computing infrastructure. In this paper, we proposed algorithms which are used to parallelize k-Anonymity using MapReduce programming paradigm with Hadoop. First, we studied k-Anonymity model and its usefulness in anonymizing data. We implemented the Top Down Specialization, a generalization method on k-anonymity so as to anonymize the original data. We then implemented algorithms that could parallelize k-Anonymity using Map Reduce framework. The proposed algorithm is capable of anonymizing big data in order to support privacy preserving. In future, we can propose an algorithm using MapReduce programming towards efficiently anonymizing big data for preserving privacy.

V. REFERENCES

[1] L. Hsiao-Ying and W.G. Tzeng, “A Secure Erasure Code-Based Cloud Storage System with Secure Data Forwarding,” IEEE Trans. Parallel and Distributed Systems, vol. 23, no. 6, pp. 9951003, 2012.

[2] D. Zissis and D. Lekkas, "Addressing Cloud Computing Security Issues," Future Generation Computer Systems, vol. 28, no. 3,pp. 583-592,2011.

[3] RuilinLiu, Hui Wang, ”Privacy –Preserving Data Publishing “ IEEE ,2010.

[4] B.C.M. Fung, K. Wang, and P.S. Yu, "Anonymizing Classification Data for Privacy Preservation," IEEE Trans. Knowledge and

Data Eng., vol. 19, no. 5, pp. 711-725, May 2007.

[5] X. Xiao and Y. Tao, "Anatomy: Simple and Effective Privacy Preservation." Proc. 32nd Int'l Conf. Very Large Data Bases (VLDB'06), pp. 139-150,2006.

[6] K. LeFevre, D.J. DeWitt, and R. Ramakrishnan, "Incognito: Efficient Full-Domain K-Anonymity," Proc. ACM SIGMOD Infl Conf. Management of Data (SIGMOD '05), pp. 49-60, 2005.

[7] K. LeFevre, D. l DeWitt, and R. Ramakrishnan, "Mondrian Multidimensional K-Anonymity," Proc. 22nd Int'! Conf. Data Eng. (ICDE '06), 2006.

[8] 1 Xu, W. Wang, 1 Pei, X. Wang, B. Shi, and A. W. Fu. , "Utility-based anonymization using local recoding ", In ACM SIGKDD, 2006.

[9] S.Yu, "Anonymizing Classification Data for Privacy Preservation". IEEE Transactions on Knowledge and Data Engineering ,vol 19 no 5 ,2007

[10] Benjamin C.M Fung, Ke Wang, Philip S.Yu "Top Down Specialization for Information and Privacy Preservation".

[11] Bayardo R and Agrawal R, Data privacy through optimal k-anonymization. In ICDE05: The 21st International Conference on Data Engineering, pages 217– 228, 2005.

[12] HIPAA 2012 "k-anonymity : A model for Protecting Privacy " International Journal Uncertain Fuzz .vol 10 ,no,5 pp 557-570 ,2002.

[13] Ciriani, S. De Capitani di Vimercati, S. Foresti, and P. Samarati, "k-Anonymity Data Mining: A Survey", Springer US, Advances in Information Security (2007)

[14] Latanya Sweeney, School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania, "Achieving k-anonymity privacy protection using generalization and suppression", International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, Volume 10 Issue 5, October 2002, Pages 571 - 588

[15] Meyerson A and Williams R, On the complexity of optimal k-anonymity. In PODS04: Proceedings of the twenty fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pages 223–228, 2004

[16] Ke Wang, Philip S. Yu, Sourav Chakraborty, "Bottom-Up Generalization: A Data Mining Solution to Privacy Protection", Fourth IEEE International Conference on Data Mining, 2004. ICDM '04 .. Pages 249 – 256

[17] Xuyun Z, Laurence T Yang ,"A scalable Two phase Top Down Specialization Approach for Data Anonymization using Map Reduce on cloud", IEEE Transaction on Parallel and Distributed Systems ,TPDSSI-2012.

[18] Zhang X, Yang LT, Liu C, Chen J. A scalable two-phase top-down specialization approach for data anonymization using MapReduce on cloud. IEEE Trans Parallel Distrib Syst. 2014;25(2):363–73

[19] Jefrey Dean and Sanjay Ghernawat "Map-Reduce: Simplified Data Processing on Large Clusters" Google,Inc.2004

[20] J. Dean and S. Ghemawat, "Mapreduce: Simplified Data Processing on Large Clusters," Comm. ACM, vol. 51, no. 1, pp. 107-113,2008

[21] Al-Zobbi M, Shahrestani S, Ruan C. Sensitivity-based anonymization of big data. In: Local computer networks work- shops (LCN workshops), 2016 IEEE 41st Conference on. IEEE; 2016. p. 58–64

[22] Al-Zobbi M, Shahrestani S, Ruan C. Implementing a framework for big data anonymity and analytics access control. In: Trustcom/BigDataSE/ICESS, 2017 IEEE. IEEE; 2017. p. 873–80.

[23] Al-Zobbi M, Shahrestani S, Ruan C. Multi-dimensional sensitivity-based anonymization method for big data. In: Elk-

hodr M, Shahrestani S, Hassan Q, editors. Networks of the future: architectures, technologies, and implementations. Boca Raton: Chapman and Hall/CRC Computer and Information Science Series, Taylor & Francis; 2017. p. 448.

Cite this article as :

Anushree Raj, Rio G L D'Souza, "Big Data Anonymization in Cloud using k-Anonymity Algorithm using Map Reduce Framework", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 5 Issue 1, pp. 50-56, January-February 2019.
Available at doi :
https://doi.org/10.32628/CSEIT19516
Journal URL : http://ijsrcseit.com/CSEIT19516