International Journal of Advances in Engineering, 2015, 1(2), 66 - 71
ISSN: 2394-9260 (printed version); ISSN: 2394-9279 (online version)
url:http://www.venuspublications.com/ijae.html

SHORT COMMUNICATION

An Enhanced Data Anonymization Technique to Preserve Privacy in Big Data

Deepalakshmi V, Mayuranathan M and Balasubramani S
Department of Computer Science and Engineering
Valliammai Engineering College, India
deepavijay.kpm@gmail.com

Received 23 January 2015 / Accepted 17 February 2015

Abstract : For Big data processing framework which rely on cluster computers with a high performance computing platform some parallel programming tools like map-reduce has been used on a large number of computing node. To preserve privacy among the data being shared an effective anonymization technique is used. Anonymizing the data is important if we are to reconcile the conflicting demands arising from the desire to release the data for study and the desire to protect the privacy of individuals represented in the data. This paper focuses on describing a system that anonymizes the data by partitioning the attributes and applying appropriate map-reduce framework on the Hadoop Distributed File System. It also focuses on providing the authorized data and also preserving the identity of the authorized user. Hence a high level of privacy on the data as well as the identity of the author will be achieved.
Keywords: Big data, anonymization

I. INTRODUCTION

Big data is the ocean of information we swim in every day that can be described as a massive volume of both structured and unstructured data that is difficult to process using traditional databases and software techniques [19]. Every day we create quintillion bytes of data through social media sites, purchase transaction records, weather reports, cell phone GPS signals, etc. Instead of relying on expensive, proprietary hardware and different systems to store and process data, Hadoop enables distributed parallel processing of huge amounts of data across inexpensive, industry-standard servers that both store and process the data, and can scale without limits. Hadoop can handle all types of data from disparate systems: structured, unstructured, log files, pictures, audio files, communications records, email. Data on individuals and entities are being collected widely. These data can contain information that explicitly identifies the individual (e.g., social security number). Data can also contain other kinds of personal information (e.g., date of birth, zip code, gender) that are potentially identifying when linked with other available data sets. Data are often shared for business or legal reasons. This paper addresses the important issue of preserving the anonymity of the individuals or entities during the data dissemination process. The primary focus of this work is to perform anonymization on the data being shared on the public pools that ensures enough privacy on the bigdata. The major focus of this system lies on the cost effectives so that all the people could easily access the data being shared on the public cloud and they can share their own data with the hadoop environment with the full confidence that the sensitive data those they think that are not to be revealed directly to the public can be preserved from others by generalising those data to a certain level. The generalization of data is implemented by specializing or detailing the level of information in a top-down manner until a minimum privacy requirement is violated.

II. RELATED WORK AND PROBLEM ANALYSIS

a) Top down specialization: The generalization of data is using a specialization or detailed level of information in a top-down manner until a minimum privacy requirement is violated. The privacy goal is given by the *anonymity* on a combination of attributes called a *virtual identifier*; the description on a virtual identifier is required to be shared by some minimum number of records in the table. A generalization taxonomy tree is specified for each categorical attribute in a virtual identifier. [1] Map Reduce Top down Specialization (MRTDS) generalizes the table by *specializing* it iteratively starting from the most general state. At each step, a general (i.e. parent) value is specialized into a specific (i.e. child) value for a categorical attribute, or an interval is split into two sub-intervals for a continuous attribute. This process is repeated until further specialization leads to a violation of the anonymity requirement. The scale of data in many cloud applications increases tremendously, in accordance with the recent trends in Big Data. The centralized top down specialization approach exploits the taxonomy indexed partition data structure to improve the scalability and efficiency by indexing anonymous data records and retaining statistical information. But in this approach there is an assumption that all data proposed should fit in memory for the centralized approaches. The amount of metadata retained to maintain the statistical information and linkage information is larger.

b) Two phase top down specialization: Two phase top down approach is to conduct the computation required in TDS in a highly scalable and efficient fashion. [6] The two phases are based on the two levels of parallelization provisioned by MapReduce on cloud. Basically, MapReduce on cloud has two levels of parallelization, i.e., job level and task level. Job level parallelization means that multiple MapReduce jobs can be executed simultaneously to make full use of cloud infrastructure resources. Combined with cloud, MapReduce becomes more powerful and elastic as cloud can offer infrastructure resources on demand, for example, Amazon Elastic MapReduce service [5]. Task level parallelization refers to that multiple mapper/reducer tasks in a MapReduce job are executed simultaneously over data splits. It achieves high scalability by parallelizing multiple jobs on data partitions in the first phase, but the resultant anonymization levels are not identical. To obtain finally consistent anonymous data sets, the second phase is necessary to integrate the intermediate results and further anonymize entire data sets. In the first phase, an original data set D is partitioned into smaller ones. Then a subroutine is run over each of the partitioned data sets in parallel to make full use of the job level parallelization of MapReduce. The subroutine is a MapReduce version of centralized TDS (MRTDS) which concretely conducts the computation required in TPTDS. Two Phase MapReduce Top Down Specialization (TPMRTDS) anonymizes data partitions to generate intermediate anonymization levels. An intermediate anonymization level means that further specialization can be performed without violating k-anonymity. MRTDS only leverages the task level parallelization of MapReduce. In the second phase, all intermediate anonymization levels are merged into one. The basic idea of TPTDS is to gain high scalability by making a tradeoff between scalability and data utility. The slight decrease of data utility can lead to high scalability.

*2.3 Generalized Ring Signatures:*The ring signature specifies a set of possible signers instead of revealing the actual identity of the message signer. The verifier can verify that the signature is generated by one of the ring members still he cannot identify which member produced this signature. This can achieve unconditional signer ambiguity and is secure against adaptive chosen-message attacks in the random oracle model. There are certain [7,17]Threshold ring signature enables any group of t entities spontaneously conscripting arbitrary n-t entities to generate a publicly verifiable t-out-of-n threshold signature on behalf of the whole group of the n entities, while the actual signers remain anonymous. [18] A highly efficient ID-based ring signature from pairings that requires only one pairing operation is employed. It has the least complexity among its counterparts.

Table.1 Comparison with Existing Mechanisms

TECHNIQUES	MRTDS	TPMRTDS	IDRS	PROPOSED
Data Privacy	Yes	Yes	No	Yes
Identity Privacy	No	No	Yes	Yes
Public Auditing	No	No	Yes	Yes

III. DATA ANONYMIZATION TECHNIQUES

a) Encryption: Data encryption anonymizes data by replacing selected sensitive data with encrypted data. Data encryption is easy to put in place and provides good anonymization since it is almost impossible to revert to the original data without knowledge of the encryption key and encryption algorithm. With respect to producing undisclosed data, encryption is a fast and efficient way to proceed.

b) Substitution: With substitution technique the data anonymization techniques generate anonymized data that are irreversible without keys and the generated anonymized data maintain relational data integrity. Instance of this tech: replacing all values of sensitive column with a standard character or standard value, nulling out or blanking out/ removing out the sensitive. *Substitution* consists of replacing the contents of a database column with data from a predefined list of factious but similar data types so it cannot be traced to the original subject. *Shuffling* is similar to substitution, except the anonymized data is derived from the column itself. Both methods have their pros and cons, depending on the size of the database in use. For example, in the substitution process, the integrity of the information remains intact (unlike the information resulting from the encryption process). But substitution can pose a challenge if the records consist of a million usernames that require substitution. An effective substitution requires a list that is equal to or longer than the amount of data that requires substitution. Substitution is very effective in terms of preserving the look and feel of the existing data. The downside is that a largish store of substitutable information must be available for each column to be substituted. For example, to sanitize surnames by substitution, a list of random last names must be available. Then to sanitize telephone numbers, a list of phone numbers must be available. Substitution data can sometimes be very hard to find in large quantities; however any data masking software should contain datasets of commonly required items.

c) Shuffling: Shuffling is similar to substitution except that the substitution data is derived from the column itself. Essentially the data in a column is randomly moved between rows until there is no longer any reasonable correlation with the remaining information in the row. There is a certain danger in the shuffling technique i.e. the original data is still present and sometimes meaningful questions can still be asked of it. Another consideration is the algorithm used to shuffle the data is if the shuffling method can be determined, then the data can be easily "un-shuffled". For example, if the shuffle algorithm simply ran down the table swapping the column data in between every group of two rows it would not take much work from an interested party to revert things to their un-shuffled state. Shuffling is rarely effective when used on small amounts of data. For example, if there are only 5 rows in a table it probably will not be too difficult to figure out which of the shuffled data really belongs to which row. On the other hand, if a column of numeric data is shuffled, the sum and average of the column still work out to the same amount. This can sometimes be useful. Shuffle rules are best used on large tables and leave the look and feel of the data intact. They are fast, but great care must be taken to use a sophisticated algorithm to randomise the shuffling of the rows.

d)Generalization: Generalization is one of the commonly used anonymization approach that replaces quasi-identifier values with values that are less-specific but semantically consistent. Then, all quasi-identifier values in a group would be generalized to the entire group extent in the QID space. [1] If at least two transactions in a group have distinct values in a certain column then all information about that item in the current group is lost. The QID used in this process includes all possible items in the log. Due to the high-dimensionality of the quasi-identifier, with the number of possible items in the order of thousands, it is likely that any generalization method would incur extremely high information loss, rendering the data useless [8]. In order for generalization to be effective, records in the same bucket must be close to each other so that generalizing the records would not lose too much information. However, in high-dimensional data, most data points have similar distances with each other. This is an inherent problem of generalization that prevents effective analysis of attribute correlations. In order to perform data analysis or data mining tasks on the generalized table [9], the data analyst has to make the uniform distribution supposition that each value in a generalized interval is equally possible, as no additional distribution assumption can be justified. This significantly decreases the data utility of the generalized data.

Table. 2 Classification of attributes

Attribute type	Property	Example	Action required
Key	Can identify an individual directly	Name, social society number	Remove or obscure
Quasi identifier	Can be linked with external information to identify an individual	Zip code, gender, birthday	Suppress or generalize
Sensitive	Data that an individual is sensitive about revealing	Income, type of illness	Needs to be delinked from individual

e)Perturbation: Data perturbation represents one common approach in privacy preserving data mining. It is built on a longer history in the areas of statistical disclosure control and statistical databases where the original dataset is perturbed and the result is released for data analysis. Typically, a "privacy/ accuracy" trade-off is faced. On one hand, perturbation must not allow the original data records to be adequately recovered. On the other hand, it must allow "patterns" in the original data to be mined. Data perturbation includes a wide variety of techniques including: additive, multiplicative [2], matrix multiplicative, k-anonymization [3,10], micro-aggregation [4,11], categorical data perturbation [12,13], data swapping [15], resampling, data shuffling [14]. Perturbation methods are mainly used with a compromise on data utility, as the data are altered and or not reversible. But the privacy is provided to an extent except closeness attack.

IV. PROPOSED SYSTEM

A scalable two-phase top-down specialization approach to anonymize large-scale data sets using the MapReduce framework has been used. In both phases the approach deliberately designs a group of innovative MapReduce jobs to concretely accomplish the specialization computation in a highly scalable way. The proposed system also focuses on providing the authorized data and also preserving the identity of the authorized user, preserving a highly level privacy on the data as well as the identity of the author. This system not only focuses on the privacy of the data but also the privacy of the author who owns the data is also preserved using the ring signature. This project is aimed at sharing a data on the public cloud by preserving the sensitive data and also preserving the identity of the author who owns the data. To preserve privacy among the data being shared an effective anonymization technique is used. Data anonymization enables the transfer of information across a boundary, such as between two departments within an agency or between two agencies, while reducing the risk of unintended disclosure, and in certain environments in a

manner that enables evaluation and analytics post-anonymization. The proposed system anonymizes the data by partitioning the attributes and applying appropriate map-reduce framework on the Hadoop Distributed File System.

Algorithm: Data Partition map and reduce
MapReduce can take advantage of locality of data, processing data on or near the storage assets to decrease transmission of data. **"Map" step:** The master node takes the input, divides it into smaller sub-problems, and distributes them to worker nodes. A worker node may do this again in turn, leading to a multi-level tree structure. The worker node processes the smaller problem, and passes the answer back to its master node. **"Reduce" step:** The master node then collects the answers to all the sub-problems and combines them in some way to form the output – the answer to the problem it was originally trying to solve.
Step 1: Input data set D, anonymity parameters k, k^I and the number of partitions p.
Step 2: Partition D into $D_i, 1 \leq i \geq p$.
Step 3: Execute MRTDS(D_i, K^I, AL^0) $\rightarrow AL'_i$, $1 \leq i \leq p$ i parallel as multiple MapReduce jobs.
Step 4: Merge all intermediate anonymization levels into one, Merge ($AL'_1, AL'_2, ..., AL'_p$) $\rightarrow AL^I$.
Step 5: Execute MRTDS (D, k, AL^I) $\rightarrow AL^*$ to achieve k-anonymity.
Step 6: Specialize D according to AL^*, Output D^*.
MapReduce is as a 5-step parallel and distributed computation:
Prepare the Map() input – the "MapReduce system" designates Map processors, assigns the K1 input key value each processor would work on, and provides that processor with all the input data associated with that key value.
Run the user-provided Map () code – Map () is run exactly once for each K1 key value, generating output organized by key values K2.
"Shuffle" the Map output to the Reduce processors – the MapReduce system designates Reduce processors, assigns the K2 key value each processor would work on, and provides that processor with all the Map-generated data associated with that key value.
Run the user-provided Reduce () code – Reduce () is run exactly once for each K2 key value produced by the Map step.
Produce the final output – the MapReduce system collects all the Reduce output, and sorts it by K2 to produce the final outcome.

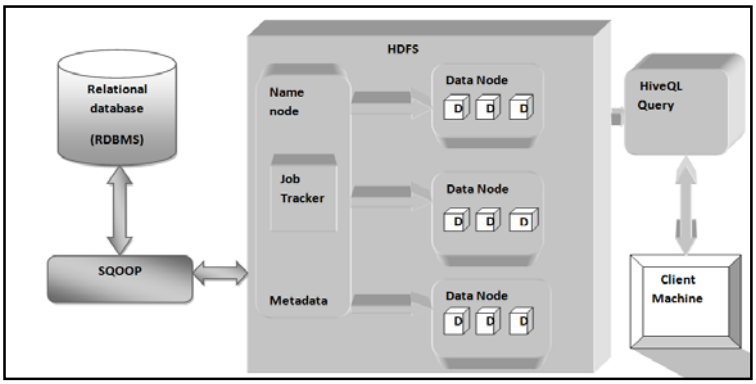

Figure.1 Architecture diagram

In the user interface level the client machine access the data on the hadoop distributed file system through the web browser. In the HDFS the tasks are performed using various nodes. Two important nodes which are to be considered in HDFS are name node and the data node. The name nodes are considered as the job tracker and the data nodes are considered as the task trackers. The JobTracker pushes work out to available TaskTracker nodes in the cluster, striving to keep the work as close to the data as possible. With a rack-aware file system, the JobTracker knows which node contains the data, and which other machines are nearby. If the work cannot be hosted on the actual node where the data resides, priority is given to nodes in the same rack. This reduces network traffic on the main backbone network. If a TaskTracker fails or times out, that part of the job is rescheduled. The TaskTracker on each node spawns off a separate Java Virtual Machine process to prevent the Task Tracker itself from failing if the running job crashes the JVM. The data in the RDBMS are imported to HDFS and exported from HDFS using the tool called sqoop, which is the command line interface application developed by apache hadoop.

a) MRTDS Driver: Usually, a single MapReduce job is inadequate to accomplish a complex task in many applications. Thus, a group of MapReduce jobs are orchestrated in a driver program to achieve such an objective. MRTDS consists of MRTDS Driver and two types of jobs, i.e., IGPL Initialization and IGPL Update. The driver arranges the execution of jobs. Step 1 initializes the values of information gain and privacy loss for all specializations, which can be done by the job IGPL Initialization.

b) IGPL Initialization Job: The main task of IGPL Initialization is to initialize information gain and privacy loss of all specializations in the initial anonymization level AL.[1] Information gain for a potential specialization in the corresponding Reduce function is computed. The first step is to accumulate the values for each input key. If a key is for computing information gain, then the corresponding statistical information is updated. A salient MapReduce feature that intermediate key-value pairs are sorted in the shuffle phase makes the computation of IG(spec) sequential with respect to the order of specializations arriving at the same reducer. Hence, the reducer just needs to keep statistical information for one specialization at a time, which makes the reduce algorithm highly scalable.

c)IGPL Update Job: The IGPL Update job dominates the scalability and efficiency of MRTDS, since it is executed iteratively, iterative MapReduce jobs have not been well supported by standard MapReduce framework like Hadoop. The IGPL Update job is quite similar to IGPL Initialization, except that it requires less computation and consumes less network bandwidth. Thus, the former is more efficient than the latter.

d) Ring Signature with one way accumulator: Ring signature, a type of digital signature is performed by any member of a group of users that each have keys. Therefore, a message signed with a ring signature is endorsed by someone in a particular group of people [18]. One of the security properties of a ring signature is it is computationally infeasible to determine *which* of the group members' keys was used to produce the signature. Ring signatures are similar to group signatures but differ in two key ways: first, there is no way to revoke the anonymity of an individual signature, and second, any group of users can be used as a group without additional setup. [17]A cryptographic accumulator which is a one way membership function is used. It answers a query as to whether a potential candidate is a member of a set without revealing the individual members of the set. One trivial example is how large composite numbers accumulate their prime factors, as it's currently impractical to factor the composite number, but relatively easy to find a product and therefore check if a specific prime is one of the factors. New members may be added or subtracted to the set of factors simply by multiplying or factoring out the number respectively.

V. CONCLUSION AND FUTURE WORK

The very important issues, that is to be concentrated while accessing the data from the public domain is its originality. There are many possibilities in the freely available public cloud to encounter fake data. Vulnerability in publicly accessible software enables an attacker to puncture the cloud and expose data of other customers using the same service. So considering these issues we focused on ring signature which is similar to the digital signature that can be performed by any member of a group of users that each have keys. Therefore, a message signed with a ring signature is endorsed by someone in a particular group of people. One of the security properties of a ring signature is that it should be computationally infeasible to determine which of the group members' keys was used to produce the signature. Also the work is extended by allowing a third party authority to verify whether the data is authorized or not without revealing the identity of the user.

REFERENCES

[1] Yeye He, Jeffrey F. Naughton, “Anonymization of SetValued Data via TopDown, Local Generalization”, Proceedings of the VLDB Endowment Volume 2 Issue 1, August 2009
[2] J. J. Kim andW. E.Winkler, “Multiplicative noise for masking continuous data”, Technical Report Statistics 2003-01, Statistical Research Division, U.S. Bureau of the Census, Washington D.C., April 2003.
[3] L. Sweeney, “ k-anonymity: a model for protecting privacy”International Journal on Uncertainty, Fuzziness and Knowledge-based Systems,10(5):557–570, 2002.
[4] Charu C. Aggarwal and Philip S. Yu, “A condensation based approach to privacy preserving data mining” In Proceedings of the 9th International Conference on Extending Database Technology (EDBT’04), pages 183–199, Heraklion, Crete, Greece, March 2004.
[5] Amazon Web Services, “Amazon Elastic Mapreduce,” http://aws.amazon.com/elasticmapreduce/, 2013.
[6] Xuyun Zhang, Laurence T. Yang, Senior Member, IEEE, Chang Liu, and Jinjun Chen, “A Scalable Two-Phase Top-Down Specialization Approach for Data Anonymization Using MapReduce on Cloud”, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 25, NO. 2, FEBRUARY 2014.
[7] Jian Ren, Member, IEEE, and Lein Harn, “Generalized Ring Signatures”, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 5, NO. 3, JULY-SEPTEMBER 2008.
[8] K. Wang, P. Yu, and S. Chakraborty,” Bottom-up generalization: a data mining solution to privacy protection”, The Fourth IEEE International Conference on Data Mining 2004 (ICDM 2004), November 2004.
[9] R. Agrawal and S. Ramakrishnan, “Privacy preserving data mining”, In Proc. of the ACM SIGMOD Conference on Management of Data, pages 439–450, Dallas, Texas, May 2000.
[10] P. Samarati, “Protecting respondents identities in microdata release”, IEEE Transactions on Knowledge and Data Engineering, 13(6):1010–1027, November/December 2001.
[11] X.-B. Li and S. Sarkar, “A tree-based data perturbation approach for privacy-preserving data mining ”, IEEE Transactions on Knowledge and Data Engineering (TKDE), 18(9):1278–1283, 2006.
[12] Evfimevski, J.Gehrke, andR. Srikant , “Limiting privacy breaches in privacy preserving data mining”, In Proceedings of the ACMSIGMOD/PODS Conference, San Diego, CA, June 2003.
[13] V. S. Verykios, A. K. Elmagarmid, E. Bertino, Y. Saygin, and E. Dasseni, “Association rule hiding”, In IEEE Transactions on Knowledge and Data Engineering, volume 16, pages 434–447, 2004.

[14] K. Muralidhar and R. Sarathy. Data shuffling - a new masking approach for numerical data. Management Science, 52(5):658–670, May 2006.
[15] S. E. Fienberg and J. McIntyre, "Data swapping: Variations on a theme by dalenius and reissTechnical report", National Institute of Statistical Sciences, Research Triangle Park, NC, 2003.
[16] Ashish Kumar Kendhe, Himani Agrawal, "A Survey Report on Various Cryptanalysis Techniques", International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-3, Issue-2, May 2013.
[17] Quangang Zhao,"A New Type of Ring Signature Scheme Based on Group Signatures Idea", Journal of Convergence Information Technology (JCIT), Volume8, issue3.81, Number3, Feb 2013.
[18] Jiang Han, Xu QiuLiang ; Chen Guohua, "Efficient ID-based Threshold Ring Signature scheme", Embedded and Ubiquitous Computing, 2008. EUC '08. IEEE/IFIP International Conference on (Volume:2), Dec. 2008.
[19] Xindong Wu, Xingquan Zhu, Gong-Qing Wu, and Wei Ding, "Data Mining with Big Data", IEEE transactions on knowledge and data engineering, vol. 26, no. 1, January 2014.