Sensitivity-based Anonymization of Big Data

Mohammed Al-Zobbi, Seyed Shahrestani, Chun Ruan
School of Computing, Engineering & Mathematics.
Western Sydney University, NSW, Australia

***Abstract*—Data Analytics is widely used as a means of extracting useful information from available data. It is only natural that it is increasingly adapted for processing big data. The rapidly growing demand for big data analytics has several undesirable side-effects. Perhaps, the most significant of those relates to increased risks for data disclosure and privacy violations. Data anonymization can provide promising solutions for minimizing such risks. In this paper, we discuss some of the specific requirements of the anonymization process when dealing with big data. We show that in general, information loss is the result of avoidable generalization of similar or equivalent data. Using these analyses, we propose a novel framework for data anonymization, which expands the *k-anonymity* properties and concepts and takes the data class values and the sensitivity of data into account. As such, the proposed process can utilize a bottom-up approach, in contrast to most other anonymization methods. The top-down approaches usually generalize all records, the equivalent and the non-equivalent ones. Ours is more methodical, as it avoids the generalization of the equivalent records. With the inclusion of sensitivity levels, we demonstrate that our framework can reduce the iteration steps and the time required to finalize the anonymization, and therefore enhance the overall efficiency of the process**

Keywords*— *Access Control; Anonymization; Big Data; k-anonymity; MapReduce; Sensitivity.

I. INTRODUCTION

It is hard to define big data precisely. The term big data has appeared in the literature since the world faced technical difficulties in storing, retrieving and analyzing the massive volumes of data. Big data is predominantly associated with data storage and data analytics processes [1]. Hardware developments, particularly memory and CPU technologies, have not been able to meet the processes that go with the massive growths in data volumes [2]. These necessitate the development of novel approaches and revamping of the methods that process big data [3]. Data analytics is among the prominent approaches in such operations. The significance of data analytics increases in direct proportions to the growing volume of data [4].

Big data analytics, however, may result in greater security risks and user privacy invasions. To overcome these problems, data anonymization has been suggested and implemented [5]. The majority of anonymization approaches have been proposed for conventional data, and may not be suitable for big data processes. However, at least conceptually, the more scalable approaches can be adapted for use in big data environments. Two primary methods are used for conventional data privacy protection. These are perturbation techniques and methods for *k-anonymity* [6]. Many variations of these two approaches have been devised and reported. Some of their adaptations have also been developed for privacy preservation while dealing with big data. These have been mainly achieved by amending the conventional data methods, to make them fit the big data characteristics. However, these modifications do not fully satisfy the requirements arising in dealing with big data [7]. On the other hand, MapReduce is a parallel distributed process that has gained popularity in managing big data. Big data privacy preservation approaches that mimic the MapReduce structure are shown to produce promising results [8].

Many users from various organizations may inquire access to big data analytics for different reasons. With a large number of users, it is reasonable to expect differing access privileges, for instance, due to the roles of the various users [9]. Current big data anonymization methods, however, do not provide for multiple levels of access privilege. Nevertheless, compared to conventional data, big data by nature is more prone to external attacks. So, it is beneficial to have anonymization methods that are capable of providing hierarchical access control levels, similar to those provided by Role-Based Access Control (RBAC), without significantly affecting the big data analytics performance.

A widely cited anonymization method, utilized generalization and suppression of data to yield *k-anonymity* properties for the data [10]. The method is based on anonymization of quasi-identifiers (Q-ID), which can be used in the discovery of a group of attributes that in turn may identify other tuples in the database [11]. The method tends to hide the sensitive values by ensuring the equivalency between records by at least k times [10]. One of the main reasons behind the large information loss in *k-anonymity* method was the single dimensional operation. The *k-anonymity* adheres one group for all data, which considerably reduces the gained information. This concern was resolved by proposing the top-down specialization method. The TDS is capable of endorsing the multi-dimensional operation. Eventually, the TDS method was proposed based on LKC method in a multi-dimensional operation [12]. Hence TDS is known by multi-dimensional TDS (MDTDS) [13] and [14]. All *k-anonymity* methods implement the grouping process as a major task of anonymization. Data is usually grouped into equivalent or similar records, known as compressions. MDTDS intensively compresses data to the top-most, by generalizing Q-ID attributes to the top-most values. This technique increases the information loss rate.

DOI 10.1109/LCNW.2016.25

Another big data privacy method mutates between top-down specializations (TDS) and bottom-up generalization (BUG) in a hybrid fashion [15, 16]. The method calculates the value of K, where K is defined as workload balancing point if it satisfies the condition that the amount of computation of anonymization required by MRTDS is equal to that by MRBUG. The K value is calculated separately for each group of data set, so TDS operation is triggered when anonymity of k > K, while BUG operation is triggered when anonymity of k > K. However, the iteration technique of this method is quite similar to MDTDS. Also, finding the value K for each group of record consumes even more time.

Any big data anonymization process is supposed to operate in a parallel distributed paradigm, which splits the large tasks into smaller subtasks and scopes. To achieve this, utilization of approaches similar to those implemented by the MapReduce slave servers may prove to be fruitful. In this fashion, each node spends very little time on each process, before transferring the rest of the job to another server. Task splitting is not implemented in MDTDS. Furthermore, MDTDS splits the large size of data into small chunks. This technique negatively affects the information gain, resulting in increased information loss.

Sensitivity value is a real number that can be calculated from Q-ID probability and user ownership level. Sensitivity concept is related to hierarchical multi-level access control and anonymity process. It is essentially managing how much or to what extent the data attributes can be reorganized. The masking tool determines the level of anonymization in the reverse order. That is, the user with a higher access level will receive less reorganized data. The anonymity is provided through a number of iterations until the data reaches the desired *k-anonymity* conditions. As it will be shown, the sensitivity value can help in reducing the number of the required iterations, improving the overall anonymization process. The proposed anonymization approach benefits from utilization of sensitivity concepts.

In this work, we propose a novel method to addresses the mentioned concerns for MDTDS. It is referred to as Multi-Dimensional Sensitivity-Based Anonymity (MDSBA). It introduces a new technique of distributing the workload among the parallel MapReduce servers. The method is integrated with the Role-Based Access Control to provide hierarchical user access privileges. The proposed method, shortens the compression process through anonymizing groups from bottom up, eliminating the need for grouping of the equivalent records. Additionally, data is not split into small chunks. Rather, the data is distributed among many nodes, reducing its loss.

The rest of this paper is structured as follows. The next part discusses the adaptation of anonymization methods for use in big data. Section II describes the requirements for anonymization methods that deal with big data. Section III describes general requirements for any proposed anonymization method. Our proposed anonymization approach, MDSBA, is explained in Section IV. We then briefly discuss the RBAC integration with the MDSBA method. The last section gives our concluding remarks and the suggested future works.

II. ANONYMIZATION METHODS ADAPTED FOR BIG DATA

MapReduce transaction method is different from the classical transaction procedure in analytics process. MapReduce divides data process into two main tasks; reading data from multi-repositories and aggregating results in a reduce output. This imposes a new method of disposition in privacy related operations. The anonymization process can be amended to fit the reading, shuffling and reducing of data, as per MapReduce environment.

Some privacy preservation methods have been modified to fit the MapReduce framework and perform parallel data intensive computations on commodity computers [9]. Computation reads input data from a distributed file system, which splits the data into multiple chunks. Each chunk is assigned to a mapper which reads the data, performs some computation, and emits a list of key/value pairs. In the next phase, *Reduce* phases combine the values belonging to each distinct key according to some functions and write the result to an output file. The framework ensures fault-tolerant execution of mappers and reducers while scheduling them in parallel on any node in the system [8].

Since the MapReduce operations include; split, Map, shuffle and reduce, therefore, any practical security solution should take these transactions into consideration. Any tweaking in the available algorithms should consider the milestones of the scale-up efficiency and the data privacy [17]. A recently developed method in *k-anonymity* is Multi-Dimensional Top-Down Specialization method MDTDS. The method is separated into two-phase steps, known by Two-Phase TDS or TPTDS [18]. In perturbation, Airavat is the most popular method [19]. Besides, PINQ and GUPT [20].

TPTDS was proposed during the early release of Hadoop. Currently, MapReduce can be easily implemented by using Pig Latin, Hive, or SPARK , which makes the MapReduce job easier. This concern recalls for an indirect method that can provide better-performed operations. Previously, Hadoop scripts can be implemented by programming languages only; such as Java. Currently, Java can be replaced by Pig Latin queries or Hive. However, Java use can be reduced to the minimal, and on need only.

III. REQUIREMENTS OF ANONYMIZATION METHODS

Some specifications should be considered in developing anonymization methods. Developers need to distinguish the disparity between big data and conventional data. Most anonymization methods were developed for more traditional data, with a limited size of data and a computation cost. With big data, anonymization process should be able to reduce the computation costs, prevent high information loss and increase security. The larger size of data may increase the number of users who wish to access data. Because of the variance in the level of user access; there is a need for discriminating anonymization level.

Any big data anonymization developer should pay attention to the following specifications:

A. *Equivalency Increase*

The equivalency increase is a general specification that must be considered on proposing any *k-anonymity* method for big data. In *k-anonymity*, the percentage of equivalent records proportioned extrusive with the increasing number of records. The rising number of records can help the least common attributes to gain the equivalency. This is true for most attributes. Few attributes are excluded, as a reason for their solitary nature like; emails, usernames, phone or fax numbers, and primary keys.

Any Q-ID attribute can be presented by the probability of occurrences as; P(attr)=r/n, where any attribute attr can happen in r ways out of total number n. If we assume that each attribute allows r=1 of ways; then P (attr) =1/n.

The probability is defined by using the following assumptions. A group of records, N, contains some D attributes. The attributes D include m quasi-identifiers (Q-ID) [21] and one sensitive attribute. For each sensitive value, the probability factor of Q-ID records in the domain D, is described as:

$$\prod_{i=1}^{m} P[q\_id_i] = P[q\_id_1] \times P[q\_id_2] \times \dots P[q\_id_m] \quad (1)$$

Where P[Q_ID] is the probability for each Q-ID.

There are correlations of $n = \frac{1}{\prod_{i=1}^{m} P[q\_id_i]}$, where n denotes the maximum number of Q-IDs combinations in N. This means that any Q-ID record must be equivalent to one of the n combinations. If we assume that each value of the combinations appears only once; then we need at least n records to gain one-time occurrence. Also, we need k × n records to gain the *k-anonymity* for each combination value. Referring to *k-anonymity*, the equivalency q is defined as the total number of equivalent records q ≥ k for each occurrence. For instance, if k =5, then each distinguished record must appear five times in N before gaining the *k-anonymity*. In the real data, the number of Q-IDs combinations appearances are usually less than *n*. Let us call the number of actual combinations appear is $\bar{n}$. Based on our assumption of one time appearance for each combination; we can calculate the minimum value of N as:

$$N_{min} = \mathrm{k} \times \bar{n} \quad (2)$$

Where $\bar{n}$ denotes the number of actual combinations that appear in *N*, $\bar{n} \subset n$

Equation 2 assumes that each record has an equal number of appearances to the other records, which concludes that $q=N_{\min}$. However, in the real data, this is not a typical case. Thus, some records appear less frequently than the others, which makes some records reach the equivalency, while others fail. However, Equation 2 describes only one scenario. Nevertheless, any situation should consider the variable $\bar{n}$. The probability value of variable $\bar{n}$ remains between the stability and increase, and it never decreases. In reality, the value of $\bar{n}$ usually increase, while the stability scenario is less probable. Besides, the equivalency q is proportioned extrusive with N, and can be described as q ≤ N, so q α N. This can be presented by the increase percentage of equivalency Q = q / N.

The positive relationship q ≤ N can be proven experimentally. We have conducted three experiments by using the adult database from the UCI Machine Learning Repository [22]. The database describes the age of an adult, their occupation, marital status, education, sex, hours worked per week, race, native country, and salary. We assumed the salary attribute is the sensitive data, and we assigned three Q-ID attributes; age, education, and sex. The experiments are conducted using MatLab simulator [23], by choosing three groups of N records small, medium and large.

During our experiments we assumed that k = 10, and the Q-IDs probabilities are calculated as; P[age] = P [1-100] =0.01, P[education] = P [Y5-6, Y7-8, Y9, Y10, Y11, Y12, HS-grade, Some-college] = 0.125, P[sex] = P [Male, Female] = 0.5, and P[S] = P [<=50K , >50K] = 0.5. Hence, the maximum number of combinations is calculated as $n = \frac{1}{\prod_{i=1}^{m} P[q\_id_i]} = 3195$.

In the first experiment; we started by N =10,000 records. The number of the actual appearing combinations in 10,000 records was $\bar{n}$ = 1741, which presents around 50% of the probable appearances. The number of equivalent records is q=6272, which presents around Q=60% of the total number of records.

In the second experiment, we increased the number of records N= 20,000. The number of actual appearing combinations $\bar{n}$ = 2196, which presents around 69% of the probable appearances. The number of equivalent records is q=14828, which presents around Q=75% of the total number of records.

In the third experiment; we further increased the number of records N=32,561. The number of actual appearing combinations is $\bar{n}$ = 2498, which presents around 78% of the probable appearances. The number of equivalent records is q=26846, which presents around Q=82% of the total number of records.

The three experiments showed an increase in both of the equivalency percentage Q and the actual appearing combination $\bar{n}$. Fig. 1 and Fig. 2 show both values increase for three different volumes of records.

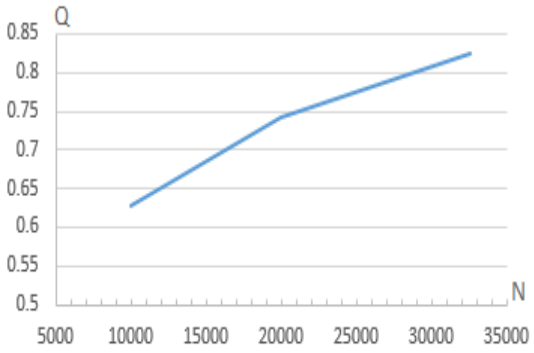

Fig. 1. Proportionality of equivalent records with increased record numbers

.

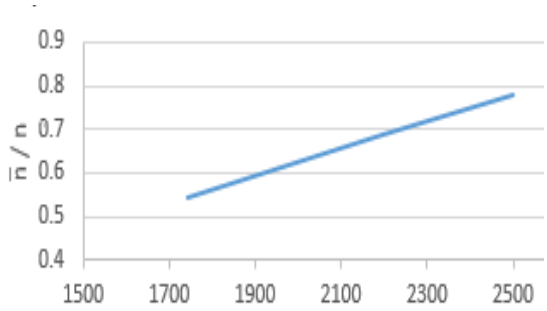

Fig. 2. The effect of increasing the number of actual combinations with increased number of records

B. The Information Gain

Current anonymization methods are mostly adaptations of approaches that were designed for conventional data [7]. Therefore, the grouping process is the primary task for anonymization, which supports the masking operations [24]. As explained before, data equivalency increases extrusive with the data size increase, which concludes a larger group of equivalent records. As a result, this will end up with a massive information loss; if the equivalent records were not handled and anonymized properly. An example of the current anonymization processes is MDTDS [16], which starts by generalizing Q-IDs, then grouping them, and finally going through a specializing step. This technique provokes a large size of equivalent records and involves an inefficient use of anonymization process, which is can be avoided for equivalent data.

The MDTDS is a popular anonymization method adapted in MapReduce. The method adapts *k-anonymity* and requires n times of iterations to find the best score on specialization rounds [18]. The iterations create n rounds between the *Map* and the *Reduce*. Since *Map* and *Reduce* may be connected through the network on separate computers; therefore, an unknown number of iteration times may create a high delay. Also, the iteration locks both servers till the end of the process, which will disturb the parallel computing principle. Also, the number of the *Map* and the *Reduce* computers is not always equal [25]. Most network structures increase the number of the *Map* servers on the account of the *Reduce* servers. This is because of a vast number of *Map* tasks in comparison with the *Reduce* tasks.

Any big data anonymization method is supposed to split the large tasks into small limited functions. This is essential to utilize the MapReduce slave servers. Hence, each node spends less time on each process, and before transferring the rest of the job to another server. Splitting tasks method is not implemented in MDTDS. Besides, MDTDS splits the large size of data into small chunks. This technique negatively affects the information gain, increasing the information loss.

Eventually, grouping data based on equivalency is an acceptable procedure if it was handled correctly on data masking. This evolves a better masking method by skipping the equivalent records, and applying masking on semi or non-equivalent data only.

C. The Parallel Distributed Environment and Multilevel Access

A parallel distributed environment handles big data. The multi-task processes should be considered in any anonymization method used for big data. This can be implemented by splitting tasks into sub-tasks, and distributing them among multi-computers to cope with the massive data volume [26].

In non-distributed environments, data must be divided into small chunks. This technique is essential in a limited hardware resources environment, with a single computation point. In this case, dividing data will prevent hardware overwhelming by a large size of data. In big data environment, splitting data into small chunks will negatively affect the information gain. This is inasmuch the previous equivalency increase, which is described in subsection A. For instance, a data chunk of 10,000 records will be extensively anonymized more than a data chunk of 30,000 records, which leads to a higher information loss. This is because of the lower equivalency in the lower number of records.

However, parallel distributed environments have a limited size of handling data on each time retrieval, such as in Hadoop *Map/Reduce*. This size of data retrieved can be pre-configured on Hadoop File System (HDFS). A trade-off between the maximum size and information gain should be studied carefully to determine the best fit size. Hence, we will further investigate this concern in our future works.

A large number of users may require a robust access control method for proper management of the variety of the user privileges [9]. The access control method can be granularly integrated with the distributed environment to manage multiple levels of access without affecting the analytics performance.

IV. MULTI-DIMENSIONAL SENSITIVITY-BASED ANONYMIZATION METHOD (MDSBA)

Multi-Dimensional Sensitivity-Based Anonymization (MDSBA) method is developed to resolve three main issues. These are user access disparity, implementing Role-Based Access Control (RBAC) in MapReduce environment, and proposing an anonymization method with a subtle performance in MapReduce. MDSBA adapts a multi-dimension technique for performing a high level of computation for MapReduce.

The *MDSBA* method mandates to define the privacy method and masking pattern for each access level. Data owners determine a subset of attributes as Q-ID, and a sensitive attribute S, then, the level of sensitivity is determined by *MDSBA* equations. *MDSBA* process is operated within RBAC environment. The Sensitivity Level of the attribute S is denoted by ψ, and the ownership level of a user is indicated by $\bar{k} = k\text{-}i$, where $i = \{k\text{-}1, \ldots, 1\}$ and $\bar{k} < k$. A lower ownership level $\bar{k}$ implies a higher sensitivity factor, denoted by ω.

Data is split horizontally rather than vertically. The division is based on attributes values rather than using a small chunk of data records. Data is split into four different groups with two levels, which enables a better multi-task approach in MapReduce. Moreover, data is categorized into three distinct categories, namely into equivalent, semi-equivalent, and non-

equivalent. Equivalent data is defined as the number of similar records that is higher than or equal to the $\bar{k}$ value in *k-anonymity*. Equivalent data cannot be anonymized, while anonymization is applied on semi and non-equivalent only.

The semi-equivalent is defined by at least two Q-ID values equivalency. The semi-equivalent is a middle case between fully and none equivalent data. The semi-equivalent records are generated by grouping two equivalent Q-ID records or more. The non-equivalent records are grouped with one Q-ID only. The equivalent and semi-equivalent groups are collated in one domain called Similar Group or SG, while the non-equivalent groups are joint in another domain called Non-similar Group or NG. The domains structure is specifically proposed for the MapReduce structure. Therefore, it divides the anonymization into multi jobs including; reading, filtering, grouping, and filtering data again, to create SG and NG domains. The MDSBA jobs are shown in Fig. 3. The master server divides the user query into the multiple task process, and each task is divided into multi-tasks. Tasks are conducted on data nodes slave servers. The slave servers are configured to be either *Map* or reduce. In MDSBA, data is not split into small chunks. Instead, the split occurs at the HDFS level. Hence, the retrieved data size is pre-configured in HDFS.

MDSBA can reduce the information loss by using two techniques; skipping the masking process on equivalent records, and distinguishing between semi and non-equivalent records on applying masking process. The masking of non-equivalent records induces extra penalty on anonymization. This penalty is necessary to generalize the diverted values in an interval or a taxonomy tree.

MDSBA is reliable and can be implemented by using Pig Latin, Hive, Spark, Java or any other scripting languages, or even a combination of them. The method is proposed to mimic the MapReduce environment, where a master server controls the slave nodes or (workers).

The master server may run the map processing on one node, and the key/pair value is emitted to another node. The master server creates a job, and each job contains three main tasks; map, shuffle and reduce. Users trigger the task by using a script, which contains queries, and each query may contain one or more tasks. The job tracker creates a job and divides tasks between nodes. Since each node is directly connected to the data or file repository; then each data node reads part of the file/data. As mentioned before, the data node reads a limited size of the data, and this can be determined by the HDFS accommodation size.

The prominent aim of our method is creating two levels of grouping. As shown in Fig. 3, the first grouping level depicts the number of sensitive values and divides tuples based on the sensitive value in domains G. For instance, three domains of G0, G1, and G2 are created for three different sensitive values. This grouping process is usually conducting in *Map* servers. The second grouping level depicts the number of equivalent records and separates G tuples into SG or NG domains. The separation is carried out in the *Reduce* processes.

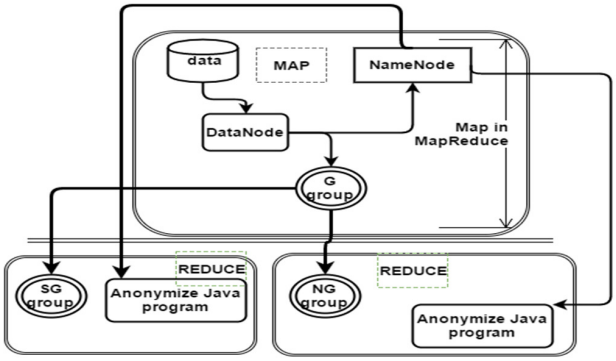

Fig. 3. MDSBA grouping method

Each domain of G is divided into three categories, equivalent, semi-equivalent, and non-equivalent. Both of equivalent and semi-equivalent groups are combined in the domain SG, while the non-equivalent groups are combined in domain NG. The definition of equivalency mandates a complete Q-ID similarity between records. The semi-equivalency mandates a minimum of two Q-ID similarities between records. Finally, anonymization process is applied on each domain separately, and the output of each process is merged into one output file.

The value of sensitivity factor ω can be calculated by finding the probability of the minimum and the maximum probability values in a quantity of m Q-IDs. Hence, the maximum probability value among Q-IDs is defined as:

$$\omega_{max} = \max\big(P(q\_id0), P(q\_id1) \dots P(q\_idm)\big) \quad (3)$$

By recalling Equation (1); the minimum probability is defined as the product of all Q-IDs probabilities, or:

$$\omega_{min} = \prod_{i=1}^{n} P[q\_id_i] \quad (4)$$

Based on Equations (3) and (4); the value of ω can be found between ω_{min} and ω_{max}, as in Equation (5):

$$\omega = \omega_{min} + (\mathrm{k} - \bar{k}) \left(\frac{\omega_{max} - \omega_{min}}{\mathrm{k}}\right) \quad (5)$$

Where ω denotes the sensitivity factor, and $\bar{k}$ denotes the ownership level.

Equation (6) collates both terms of ω and τ to conclude the sensitivity equation ψ. The sensitivity of an object degrades with the data age. The aging factor τ affects the sensitivity reversely. The older the objects, the less sensitive they are considered to be. In other words, two factors determine the sensitivity level, the ownership level ω, and the aging factor τ.

$$\psi = |\boldsymbol{\omega} + \boldsymbol{\tau}| \quad (6)$$

$Where$ ψ denotes the sensitivity level, and τ denotes the aging factor

Equation 6 is used with the NG and SG domains. The equation establishes the sensitivity level for objects based on the user access level. The masking process tends to find a close similar or smaller than the sensitivity value. For instance, if ψ=0.5; then any value falls between 0-0.5 is accepted. However, finding the closer value to ψ is more appropriate. The aging factor creates a perturbation for the sensitivity value, and this manifests when the object is older than the obsolescence value, as explained in the next section.

Recalling Equations 6, the aging factor τ creates a perturbation for the sensitivity value. The aging value manifests the object age in comparison with the obsolescence value Ø. The object age τ is reversal with the sensitivity, whereas the older objects carry less sensitive information. The object aging calculation is mutable. Thus, two separate terms are expressed for the age y<Ø, and y≥ Ø, as described in Equation 7. The aging participation percentage in sensitivity is pre-determined by the data owners, and donated by ρ. The participation percentage ρ is constant when y< Ø, and linearly degrades when y≥ Ø.

$$\tau = \begin{cases} y < \emptyset & -\rho \times \omega \\ y \geq \emptyset & -\rho \times \omega \times 0.9 \times (2 - \frac{y}{\emptyset}) \end{cases} \qquad (7)$$

$$where\ \tau \geq 0\ , \quad \emptyset \epsilon\, \mathbb{Z}, \emptyset \geq 0\ , \quad 0 \leq \rho \leq 100\%$$

Data owners may set ρ to 0% if their data objects do not mutate with the time factor.

Three masking operations are applied in anonymization process, suppression, taxonomy tree or cut, and interval. The sensitivity values are calculated by Equation 6. The following examples illustrate the three masking operations with sensitivity values reflect. Let an object with three Q-ID attributes. The data owner intends to anonymize the data with k=20. Let the obsolescence value Ø=10, the aging participation ρ=70%, and object age y=13 years. The attributes values are students IQ test and described in Table 1.

Based on the given attributes, we can calculate the sensitivity level ψ for the ownership value $\bar{k}$=10.

Recalling Equations (3-5) to find out the sensitivity factor ω, the values of ω_{max}=max (0.01, 0.005, 0.125) =0.125. While the minimum value of $\omega_{min}=0.01 \times 0.005 \times 0.125 = 6.25 \times 10^{-6}$. The value of ω as per Equation 6 is **ω**= 0.063.

Recalling Equation 7, the aging sensitivity **τ**=-0.7 × 0.063 × 0.9 × (2-12/10) =-0.032

Now we can calculate both sensitive level for NG and SG domains. **ψ** = 0.063 - 0.032 **= 0.031**.

TABLE 1. THE THREE Q-IDS AND PROBABILITY

Q-id	Q-id type	Probability
Q-id0	Interval IQ_value=[50-150]	P(Q-ID0)=1/(150-50)=0.01
Q-id1	Taxonomy tree Student_Country_Level1 = {German, French, Chinese, Kenyan, American…} Student_Ancestry-Level2={Caucasian, Asian, Middle Eastern, African, Red Indian…} Q-id_level-3= {human}	P(Q-ID1-L2)=1/150=0.007 P(Q-ID1-L3)=1/200=0.005
Q-id2	Suppression Student_Grade={A+, A, A-, B+, B, B-, C+, C, C-, D+, D, D-, F}.	P(Q-ID2)=0.077

In SG anonymization, out of the three Q-IDs, only one will be anonymized, while the two Q-IDs must be equivalent. The anonymized Q-ID is chosen based on the highest equivalency frequency. For instance, if three grouping trials are conducted by using the pair of (Q-ID0, Q-ID1), (Q-ID0, Q-ID2), and (Q-ID1, Q-ID2), and the pair of (Q-ID1, Q-ID2) was the highest equivalency frequency; then Q-ID0 will be anonymized. Consider that Q-ID0 should be anonymized. Based on the above equations, the accepted interval can be between 33 to 100. The interval of 33 is the most appropriate range since P[Q-ID0] = 1/0.031 ≈ 33. Hence, the IQ value can be anonymized by using the following intervals [50 – 83[, [83 – 116[, and [116 – 150].

If the Q-ID1 must be anonymized, then either P (Q-ID1-L2) or P(Q-ID1-L3) can be used for anonymization. Both levels probabilities are smaller than the sensitivity level. However, Q-ID2 cannot be chosen for anonymization, since suppressing the student grade will be a higher probability than the sensitivity level.

V. RBAC INTEGRATION

The *MDSBA* integrates Role-Base Access Control method. This method is commonly used in big data for authorizing users. RBAC roles can be embedded in any assertion method such as; Security Assertion Markup Language (SAML) [27]. The idea is mapping roles between the service provider (SP), which stores the data in the cloud, and the federation service (FS). The FS withholds the authentication and authorization for users. The FS is authorized by the data owner and contains information about users who wish to participate in data analytics. Users sign an agreement with the data owners about the maximum level of data access. The access level is assigned to each data object, by determining the minimum ownership level $\bar{k}$=k-*x*, where $x \epsilon \mathbb{R}$, and *x*= 1, 2…, k-1.

RBAC is a fine-grained level used for controlling user access to tasks that would normally be restricted to root role. RBAC is an alternative solution for superuser group that contains root and other administrator roles in UNIX. Superuser members are permitted to conduct almost all tasks including, creating and killing processes, reading and writing to any file, running all programs and assigning privileges. In MDSBA, there is a need for some superuser privileges, but not all, to run certain tasks. RBAC is used in our framework for several reasons; including user authorization, managing the MapReduce environment and Hadoop files, and protecting user processes from any malicious attacks.

Users are authenticated before accessing the SP. The FS dispatches the user ownership number through SAML. This transfer occurs with an XML file, which contains, the user id, the ownership level, the organization id, the database schema id, and other essential variables. SP reads the insert from the XML file and creates a new user. The user id is deployed as a username, and a random password is created. Besides, the new username is added to the *Analyst>Ownership level-k* role. The username is created only once, and can be used on each time the user logs in. Each data owner has own RBAC roles and sub-roles. We only briefly described the RBAC mapping. Further experiments will be conducted in our future wok.

VI. CONCLUSIONS

The increased monitoring, processing and storage capabilities have lead to an explosive growth of big data. However, this is of value only when, for instance, through big data analytics, useful information can be securely extracted. This work presents some of the requirements of the anonymization process for implementation in the big data context to address part of the relevant privacy concerns. This is done through analysis of the contemporary anonymization approaches and identifying some of the reasons for their inefficiencies and potentials for high information loss. In particular, we show how the *k-anonymity* processes can be made more efficient by taking into account the increased proportion of equivalent records as a result of a high number of records in big data environments. This is the basis of a novel anonymization framework, MDSBA, reported in this paper. In this framework, the anonymization starts from the bottom going up through the records. It is done in this fashion as the top-down approaches generalize all of the records. These processes need to loop continuously through the records. In each loop, the information gain or loss and other parameters should be calculated for each attribute, even for equivalent records. Our proposed method uses data sensitivity in its anonymization process and does not generalize the equivalent records, making it more methodical and more efficient. It is suitable for operation in parallel distributed environments and is compatible with the MapReduce model. Our future works will expand our experiments to more complex settings and will establish a clear process for integration of RBAC with MDSBA.

VII. REFERENCES

[1] J. S. Ward and A. Barker, "Undefined by data: a survey of big data definitions," *arXiv preprint arXiv:1309.5821,* 2013.

[2] P. Russom, "BIG DATA ANALYTICS," *TDWI RESEARCH/IBM,* vol. 4, 2011.

[3] V. Mayer-Schönberger and K. Cukier, *Big data: A revolution that will transform how we live, work, and think*: Houghton Mifflin Harcourt, 2013.

[4] A. A. Cardenas, P. K. Manadhata, and S. P. Rajan, "Big data analytics for security," *IEEE Security & Privacy,* pp. 74-76, 2013.

[5] A. Machanavajjhala, D. Kifer, J. Abowd, J. Gehrke, and L. Vilhuber, "Privacy: Theory meets Practice on the Map," ed, 2008, pp. 277-286.

[6] K. Liu, H. Kargupta, and J. Ryan, "Random projection-based multiplicative data perturbation for privacy preserving distributed data mining," *Knowledge and Data Engineering, IEEE Transactions on,* vol. 18, pp. 92-106, 2006.

[7] S. G. Shui Yu, *Big Data Concepts, Theories, and Applications*: Springer, 2016.

[8] Q. Tran and H. Sato, "A solution for privacy protection in mapreduce," ed, 2012, pp. 515-520.

[9] S. Chaudhuri, "What next?: a half-dozen data management research goals for big data and the cloud," in *PODS '12*, M. Lenzerini, M. Benedikt, Kr, and M. tzsch, Eds., ed: ACM, 2012, pp. 1-4.

[10] L. Sweeney, "ACHIEVING -ANONYMITY PRIVACY PROTECTION USING GENERALIZATION AND SUPPRESSION," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems,* vol. 10, pp. 571-588, 2002.

[11] L. Sweeney, "ANONYMITY: A MODEL FOR PROTECTING PRIVACY " *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems,* vol. 10, pp. 557-570, 2002.

[12] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "L-diversity: Privacy beyond *k-anonymity*," *ACM Transactions on Knowledge Discovery from Data (TKDD),* vol. 1, pp. 3-es, 2007.

[13] H. Samet, *Multidimensional and Metric Data Structure*: Morgan Kaufmann, 2006.

[14] B. C. M. Fung, K. Wang, and P. S. Yu, "Top-down specialization for information and privacy preservation," ed. USA, 2005, pp. 205-216.

[15] X. Zhang, C. Liu, C. Yang, J. Chen, S. Nepal, and W. Dou, "A hybrid approach for scalable sub-tree anonymization over big data using MapReduce on cloud," vol. 80, ed, 2014, pp. 1008-1020.

[16] X. Zhang, C. Liu, S. Nepal, C. Yang, W. Dou, and J. Chen, "Combining top-down and bottom-up: scalable sub-tree anonymization over big data using MapReduce on cloud," in *Trust, Security and Privacy in Computing and Communications (TrustCom), 2013 12th IEEE International Conference on*, 2013, pp. 501-508.

[17] X. Z. Xindong Wu, Gong-Qing Wu,Wei Ding, "Data Mining with Big Data," *IEEE,* 2014.

[18] L. T. Y. Xuyun Zhang, Chang Liu, Jinjun Chen,, "A Scalable Two-Phase Top-Down Specialization Approach for Data Anonymization Using MapReduce on Cloud," *IEEE,* 2014.

[19] S. T. S. Indrajit Roy, Ann Kilzer, Vitaly Shmatikov, Emmett Witchel, "Airavat: Security and Privacy for MapReduce," *The University of Texas at Austin,* 2010.

[20] P. Mohan, A. Thakurta, E. Shi, D. Song, and D. Culler, "GUPT: privacy preserving data analysis made easy," in *SIGMOD '12*, K. S. Candan, uk, Y. Chen, R. Snodgrass, L. Gravano, and A. Fuxman, Eds., ed: ACM, 2012, pp. 349-360.

[21] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Mondrian multidimensional *k-anonymity*," in *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on*, 2006, pp. 25-25.

[22] R. K. a. B. Becker. (1996). *Adults Data*. Available: ftp://ftp.ics.uci.edu/pub/machine-learning-databases

[23] MatLab. (2014). *MatLab and SimuLink*. Available: https://au.mathworks.com

[24] G. Ghinita, P. Karras, P. Kalnis, and N. Mamoulis, "Fast data anonymization with low information loss," in *Proceedings of the 33rd international conference on Very large data bases*, 2007, pp. 758-769.

[25] L. Wang, J. Tao, R. Ranjan, H. Marten, A. Streit, J. Chen*, et al.*, "G-Hadoop: MapReduce across distributed data centers for data-intensive computing," *Future Generation Computer Systems,* vol. 29, pp. 739-750, 2013.

[26] A. Holmes, *Hadoop in practice*. Shelter Island, NY: Shelter Island, NY : Manning, 2012.

[27] K. D. L. a. E. Lewis, "Web Single Sign-On Authentication using SAML," 2009.