Sensitivity-based Anonymization of Big Data

M. Al-Zobbi, S.Shahrestani, C. Ruan

School of Computing, Engineering & Mathematics., Western Sydney University, NSW, Australia

**Abstract**

Big data compels interest in allowing users to analyze the data. Analytics is an old technique that was proposed to abstract beneficial information from data. Analytics technique is now increasingly used in big data. The massive increase of data volume leads users to abstract more information for research or commercial purposes. However, this increased demand for big data analytics has increased data disclosure and privacy violation accordingly. Hence, a need for a privacy protection framework is becoming more essential than before.

In this paper, we present a framework for anonymizing the analyzed data, by using k-anonymity method. The framework is able to resolve the performance and data loss that manifests in the previous anonymization models. The framework is compatible with the parallel distributed environment of *map/reduce* pattern. Our goal is to find an optimal solution with a gradual access control.

# Introduction

Big data is the future trend in virtual environments. Human create an average of 4 Zettabyte (1,000,000 Petabyte) of data worldwide. The term big data refers to the massive amount of digital information. Many factors help data faster growth, such as; Internet speed, cheap storage, mobiles, tablets, cameras, cloud technology, and social websites [1].

There is no rigorous definition of big data. However, it is possible to define the initial reason that urged researchers to propose the big data expression. Generally, data is beneficial when it is analyzed, so users gain more information, and are able to understand the bigger picture of the business activities. Hence, the term data analytics is involved with the structured data. The computer’s hardware development does not cope with the rapid increase in data, which causes technical obstacles on dealing with a large quantity of data. For instance, the large data quantity is no longer fit into the computers memories used by processes. Therefore, there was a need to revamp the methods and tools that process big data [2].

Two major features should be considered in designing any big data tools, which are volume and velocity. The volume can be resolved by a technique that has the ability to handle a large quantity of data, and velocity can be enhanced by a high-performance technique to speed up the process [3]. These two features can be gained by using a distributed environment, where many computers process the data in a parallel time.

Big data needs to be stored, retrieved and analyzed. Thus, data analytics is one part of big data processes. Also, storing and retrieving a large size of data is a cumbersome for users, since processing such transactions may overwhelm the memory and cause more delay. Thereby, a need for a fast and reliable tool is essential. Big data can be distinguished from conventional data by Online Transaction Processing scaling (OLTP), and Online Analytical Processing (OLAP). OLTP presents the storing and retrieving. While OLAP presents data analytics [4].

This paper is divided into six sections. The first section introduces the big data definition and a comparison between big data and conventional data. The second section describes general specifications of big data anatomy and the successful anonymization model conditions. Section three looks into the current anonymization models adopted in big data. The sections four to six propose the adoptive Multi-Dimensional Sensitivity-Based Anonymization model (MDSBA).

# Anonymization Models Specifications

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

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

## Equivalency Increase

Theorem 1: *since k-anonymity value is constant, and does not change with the increasing number of records. Hence, the percentage of equivalent records proportioned extrusive with the increasing number of records.*

The increasing number of records can help the least frequent 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.

We can prove this theorem experimentally. Before we do so, let us define the following probability equations. Any attribute can be represented 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 generally defined by using the following assumptions: Let a domain of data D, contains m number of quasi-identifiers (qid’s)[5] and one sensitive attribute. Suppose D contains a number N records, and the k-anonymity = ǩ . Based on k-anonymity model [6], the probability of records occurrences in the domain D is described as:

(1)

Where P[qid] is the probability for each qid, and P[S] is the sensitivity probability.

There are correlations of n = 1 / β, where n denotes the maximum number of qid’s combinations in N. This means that any qid 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 ǩ X n records to gain the k-anonymity for each combination value. Since n = 1 / β, then we can represent a minimum number of records for anonymity by Nmin.

(2)

Referring to k-anonymity, the equivalency q is defined as the number of equivalent records q ≥ ǩ for each occurrence. For instance, if ǩ =5, then each distinguished record must appear five times in N before gaining the k-anonymity. However, not all records are equivalent in N. Some qid records may appear less frequently, and they probably need a larger N to reach the k-anonymity equivalency. Based on Equation 2, we can generalize the minimum value of N to include the number of combinations ñ appear in N:

(3)

Where ñ denotes the number of combinations appear in N, ñ ⊂ n

Equations 2 and 3 suppose that each record has an equal number of appearances to the other records. However, in the real data, this is not a common case. Thus, some records appear less frequently than the others, which makes some records reach the equivalency, while others fail. However, equation 3 describes only one scenario. Nevertheless, any scenario should consider the variable ñ, and the three constants of (ǩ, β, n). The probaility value of variable ñ remains between the stability and increase, and it never decreases. However, in the real world, the value of ñ usually increase, while the stability scenario is less probable.

Besides, the equivalency percentage Q is calculated by dividing the number of equivalent values over N:

Where Q is the equivalency percentage, and q is the frequency number of equivalent values in N records.

The records frequency is added to frequency number q when it is equal to or larger than ǩ. Based on Equation 4, there is a direct proportion between the frequency and N. Also, equation 3 shows that *Nmin* will be higher, if there is any increase in ñ. Equations 3 and 4 prove that a higher number of records, will lead to a higher number of q, and this is true in both of ñ stability or increase. Equation 4 is more general and doesn’t depend on equivalency frequency ñ, which proves that N α q.

**Example 1**: in adult database from the UCI Machine Learning Repository [7]. The database describes adult’s age, occupation, marital status, education, sex, hours per week, race, native country, and salary. We considered the salary attribute is the sensitive data, and we assigned 3 qid attributes; age, education, and sex. The experiments are conducted using MatLab simulator [8].. We experimented the above theorem for a small, medium and large size of records.

Experiment 1: Number of records N =10,000. Ǩ = 10. 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, P[S] = P [<=50K, >50K] = 0.5, β = 0.01 X 0.125 X 0.5 = 0.000313 ≈ 0.0003.

The n = 1/β = 3195 value describes the total number of combinations. While the number of appearing combinations in 10,000 records was ñ = 1741, which presents around 50% of the probable appearances. The number of equivalent records in k-anonymity, where Ǩ = 10 is q=6272, which presents around Q=60% of the total number of records.

Experiment 2: The n=1/β=3195 value describes the number of combinations. While the number of appearing combinations in 20,000 records was ñ = 2196, which presents around 69% of the probable appearances. The number of equivalent records in k-anonymity, where Ǩ = 10 is q=14828, which presents around Q=75% of the total number of records.

Experiment 3: The n=1/β=3195 value describes the number of combinations. The number of appearing combinations in 32,561 records was ñ = 2498, which presents around 78% of the probable appearances. The number of equivalent records in k-anonymity, where Ǩ = 10 is q=26846, which presents around Q=82% of the total number of records.

Experiments results: The three experiments showed an increase in equivalency percentage Q. Figure 1 shows the increase of Q between [10,000 -32561] records.

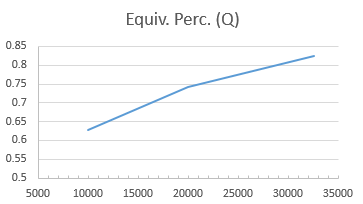


Figure 1. Equivalency percentage increase in Adult data

Also, the three experiments showed an increase value of ñ, as shown in Figure 2.

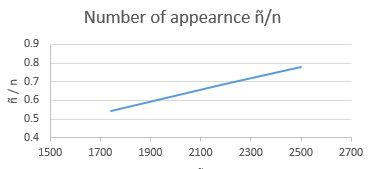


Figure 2. Number of combinations increase

## The Information Gain Decrease.

Data is usually disturbed after anonymization, and information gained is negatively affected accordingly. The hidden data on anonymization may affect the user’s queries results. The impact of anonymization can be measured by Laplace equations using InfoGain and Scores. The scores= InfoGain / AnonyLoss.

Current anonymization models were designed for conventional data. Therefore, the grouping process is the major task on anonymizing data. Data is usually grouped into equivalent records, known as compression. This technique supports masking operations. 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 large data loss; if the equivalent records were not handled and anonymized properly. The current anonymization processes start by generalizing qid’s, then grouping, and finally specializing. This technique provokes a large size of equivalency, and involves a useless anonymization process for unneeded equivalent data.

The Multi-Dimensional Top-Down Specialization (MDTDS) is an example of a popular anonymization model adopted in MapReduce. The model adopts k-anonymity, and requires n times of iterations to find the best score on specialization rounds. The iterations create n rounds between *map* and *reduce*. Since, *Map* and *reduce* may be connected through the network on separate computers; therefore, a 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 *map*, and *reduce* computers is not always equal. Most network structures increase the number of *map* servers on the account of the *reduce* servers. This is because of a large number of *map* tasks in comparison with the *reduce* tasks.

Any big data anonymization model, is supposed to split the large tasks into small limited tasks. 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 junks of data. This technique negatively affects the information gained, and increases the data loss.

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

## The Parallel Distributed Environment

Big data is handled by a parallel distributed environment. The multi-task processes should be considered in any Anonymization model 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.

In non-distributed environments, data must be split into small junks. This technique is essential in a limited resources environment, with a single computation point. In this case, splitting data will prevent hardware overwhelming by a large size of data. However, Splitting data into small junks will negatively affect the information gained. This is inasmuch the previous equivalency increase theorem. For instance, a data of 10,000 record will be extensively anonymized more than a data of 30,000 records, which leads to a higher data loss. This is because of the lower equivalency in the lower number of records.

However, parallel distributed environments have limited size of handling data on each time retrieval, such as in Hadoop *map/reduce*. This size of data retrieved can be pre-configured in Hadoop file system HDFS. A trade-off between the maximum size and information gained should be studied carefully to determine the best fit size. Hence, we need to further investigate this concern in our future work.

## Gradual Access

Big data nature is public. Big data is prone to external attacks more likely than the conventional data. Many users from many organizations may enquire big data analytics. The large increase number of users may require a robust access control model to manage a properly discriminated access for variants of user’s privileges [9]. The access control model can be granularly integrated with the distributed environment to manage gradual levels of access, and without affecting the analytics performance.

# Anonymization Models Adopted in Big Data

Two methods are proposed to protect data privacy, these are perturbation and k-anonymity. Many models were developed based on one of these two methods. The models were initially developed for conventional data, and before the big data manifestation. Recently, data analytics is becoming more prominent in big data. MapReduce is a parallel distributed process has gained a popularity in managing big data. This leads the researchers to alter the previous privacy models, so it fits the new distributed environment.

Both privacy methods become less effective in big data. Most k-anonymity models permit a large amount of data loss. Also, Perturbation becomes less effective in a large quantity of data, because it is possible to estimate the original data from the anonymized data when the data volume becomes massive. Besides, the anonymization exaggeration may be cross-referenced with the other available data following de-anonymization techniques [10].

A popular anonymity model is known by K-anonymity was proposed. The model suggests an anonymization for quasi-identifier (QID) [11] [6]. One of the main reasons behind the large data loss in k-anonymity model was the single dimensional operation. K-anonymity adheres one group for all data, which considerably reduces the gained information. This concern was resolved by proposing the top-down specialization model. The TDS is capable of endorsing the multi-dimensional operation. Eventually, the TDS model was proposed based on LKC model in a multi-dimensional operation [12]. Hence TDS is known by multi-dimensional TDS (MDTDS) [13] [14].

However, the MapReduce transaction method is different from the classical transaction method in analytics process. MapReduce segregates 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 models operations. The anonymization process should be amended to fit the reading, shuffling and reducing of data, as per MapReduce environment.

Few privacy models were altered to fit the MapReduce framework, by performing 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, reducers combine the values belonging to each distinct key according to some functions and write the result into an output file. The framework ensures fault-tolerant execution of mappers and reducers while scheduling them in parallel on any node in the system [15].

Since the MapReduce operations include; split, map, shuffle and reduce, therefore, any practical security solution should take these operations into consideration. Any tweaking in the available algorithms should consider the milestones of the scale-up efficiency and the data privacy [16].

The recently developed model in k-anonymity was MDTDS. The model is segregated into two-phase steps, known by Two-Phase TDS or TPTDS[17]. In perturbation, Airavat is the most popular model [18]. Besides, PINQ and GUPT [19].

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 a subtle model 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.

# Multi-Dimensional Sensitivity-Based Anonymization model (MDSBA)

Multi-Dimensional Sensitivity-Based Anonymization (MDSBA) model is developed to resolve three main issues, these are; gradual access for multi-level users, implementing Role-Based Access Control (RBAC) in MapReduce environment, and proposing an anonymization model with a subtle performance on the new features of Hadoop. MDSBA adopts a multi-dimension technique for performing a high level of computation for MapReduce.

Data is split horizontally instead of vertically. The split is based on attributes values instead of using a small junk of data records. This technique serves the increase percentage of data equivalence theorem. Data is split twice into four different groups with two levels, which enables a better multi-task approach in the distributed environment. Moreover, data is categorized into three different categories; equivalent, semi-equivalent, and non-equivalent. Equivalent data is defined as the number of equivalent records that is higher than or equal to ǩ value in k-anonymity. Equivalent data can’t be anonymized, while anonymization is applied on semi and non-equivalent only.

The semi-equivalent is defined by at least two qid values equivalency. The semi-equivalent is a middle case between fully and none equivalent data. The semi-equivalent records are grouped separately, and this is completed by grouping the equivalent qid’s only. The non-equivalent records are grouped with one qid only, which is the highest qid probability value.

MDSBA can reduce the data 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 model is proposed to mimic the MapReduce environment, where a master server controls the slave nodes or (workers). Each node is configured to run map, and/or reduce. In the recommended cluster structure; the map tasks are conducted on multi data nodes (slaves). The data nodes read directly from the repository file, and process the query. The data node output is emitted to the reduce task nodes over the network.

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 job 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.

## MDSBA Definition

The *MDSBA* method mandates to define the privacy model and masking pattern for each access level. Data owners determine a subset of attributes as Quasi Identifiers QID, and a sensitive attribute S, then, the level of sensitivity is determined by *MDSBA* equation. *MDSBA* process is operated within RBAC environment.

**Definition 1**: Sensitivity Level (ψ) implies a scale of data anonymization prominence, so the anonymised data T delegates a multi-level of distorted data.

**Definition 2**: The k-anonymity is the maximum number of equivalent records for the ownership level ǩ. Hence, Ǩ= k-i, where i= {k,k-1, …,1} and Ǩ < k.

Referring to definition 2; the lowest ownership value is Ǩ= k-1. This implies a higher user access level, denotes by ω. Hence, a higher value of Ǩ leads to a lower value of ω, which is denoted by .

## MDSBA and MapReduce

MDSBA is specifically proposed for the MapReduce structure. MDSBA divides the anonymization into multi jobs including; reading, filtering, grouping, and filtering data again, to create SG and NG groups. The multi jobs are shown in Figure 3. The master server divides the user’s query into the multi-job process, and each job 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 junk, instead, the split occurs in the HDFS level. Hence, the retrieved data size is pre-configured in HDFS.



Figure 3. Pig Latin syntax to create SG and NG groups

The prominent aim of our model is creating two levels of grouping. As shown in Figure 3, the first grouping level depicts the number of sensitive values, and segregates 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 segregates G tuples into SG or NG groups. The segregation is conducted in reduce processes. Each group of G is divided into three categories, equivalent, semi-equivalent, and non-equivalent. Both of equivalent and semi-equivalent groups are combined in one group donated by SG, while the non-equivalent groups are combined in one group denoted by NG. The definition of equivalent mandates a complete qid similarity between records. The semi-equivalency mandates a minimum of two qid similarities between records. Finally, anonymization process is applied on each group separately, and the output of each process is merged in one output file.

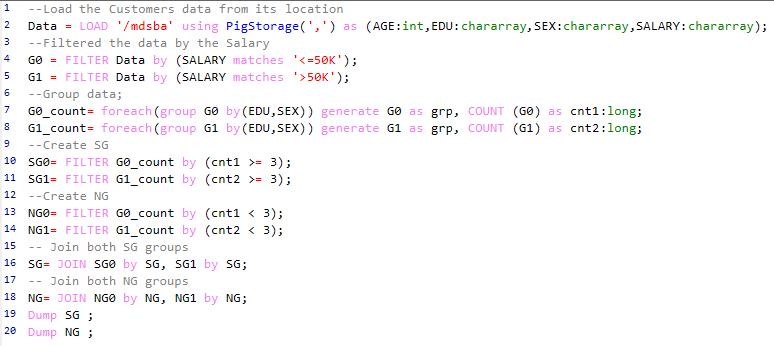


Figure 4. MapReduce process for MDSBA

Figure 4 illustrates a Pig Latin [20] script example for two jobs created to generate SG and NG groups. Line 2 reads the data sample for anonymization. Lines 4-5 create G0 and G1 groups. Lines 7-8 group each G group by two QIDs. The grouping counts the number of similar or semi-similar records, and group them in one line. Lines 10-11 filter again the number of records that is ≥ ǩ, where ǩ denotes the ownership level for the user. Here in this example, we suppose k=5, and the ownership level ǩ = 3. Finally, we join both SG0, and SG1 in one group of SG, and we apply similar join to the NG groups. MapReduce applies the above script by creating two separate jobs, as a reason for the number of Dump(s) in Lines 19-20. Each Dump or Store syntax is computed in a separate job, therefore, two separate jobs are created to output two files, SG and NG.

# MDSBA Sensitivity Calculation

The value of ω can be calculated by finding the probability of the minimum and the maximum values in a quantity of n qid’s. Hence, the maximum probability is defined as the highest probable value among qid’s, or:

And the minimum probability is defined as the product of all qid’s probabilities, or:

(6)

Based on equations (5) and (6); the value of ω can be found between ωmin and ωmax, as in equation (3):

Equation (8) collates both terms of ω and τ to conclude the sensitivity equation ψ. The object’s sensitivity degrades with the data age. The aging factor τ affects the sensitivity reversely. The older objects are less sensitive if compared with the newer ones. Hence, two factors determine the sensitivity level, the ownership level ω, and the aging factor τ.

(8)

Where

Ǩ value refers to definition 2. Equation 8 is used with the NG and SG domain. The equation measures the sensitivity level for objects and 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.

## The Object Aging Sensitivity (τ)

Recalling equations 8, 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 9. 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.

Data owners may set ρ to 0% if their data objects don’t mutate with the time factor.

## Sensitivity Examples

Three masking operations are applied in anonymization process, suppression, taxonomy tree or cut, and interval. The sensitivity values are calculated by equation 8. The sensitivity value is reflected on the chosen level or interval. The following examples illustrate the three masking operations with sensitivity values reflect.

**Example 2**: Let an object with 3 qid 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, qid0 represents IQ interval from 50-150, qid1 represents student ethnicity as a taxonomy tree with the following levels; qid1-level 1= {German, French, Saudi, Persian, Japanese, Spanish, Chinese, Kenyan, American…}, and qid1-level2= {Caucasian, Asian, Middle Eastern, African, Red Indian…}, and finally qid1-level3= {human}. The qid2 represents the student mark at the school qid2={A+, A, A-, B+, B, B-, C+, C, C-, D+, D, D-, F}.

The probability of each qid is P(qid0)=1/(150-50)=0.01, P(qid1-level1)=1/150=0.007, and P(qid1-level2)=1/200=0.005, and finally P(qid2)=0.077.

Based on the above attributes, calculate the sensitivity values ψ for the ownership value ǩ=10.

Recalling equations 5 & 6 to find out the value of ω, the values of ωmax=max (0.01, 0.005, 0.125) =0.125. While the minimum value of ωmin=0.01 X 0.005 X 0.125 = 6.25 X 10-6. The value of ω as per equation 7; ω=6.25 X 10-6 + (20-10) X (0.125-6.25 X 10-6)/20 = 0.063

Recalling equation 10, the aging sensitivity τ=-0.7 X 0.063 X 0.9 X (2-12/10) =0.032

Now we can calculate both sensitive values for NG and SG domain. **ψ** = 0.009 X 0.063 - 0.032 **= 0.032**.

Based on the above calculations, the anonymization for one qid could be as follows:

Qid0 [interval] = 1/0.03 ≈ 33. The cut for qid1 [cut] = P (qid1-level1) or P (qid1-level2). This depends on the k-anonymity, hence qid1-level or qid1-level2 are both < 0.032, and both are correct cut.

Qid2 [sup] =0.077 > 0.03. This means qid2 can’t anonymize data independently. Hence, there must be another anonymized qid to reduce the anonymization value to ≤ the sensitivity value.

For NG anonymization, there must be two out of three qid’s in-charge of anonymization, for instance if qid1, and qid2 are anonymized then; qid0 [interval] X qid1 [cut] ≤ 0.032. The anonymization values are the highest cut and interval value, which is P (qid1-level1) X P [interval=5] = 0.007 x 0.2 = **0.0014 < 0.032**.

# RBAC Mapping

The *MDSBA* model adopts Role-Base Access Control model. This model 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) [21-23]. The idea is mapping roles between the service provider (SP), which stores the data over 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 given based on the minimum ownership level ǩ=k-*x*, where, and *x= 1, 2…, k.* Each data object is given an access level for each user. For instance, if a user was given a minimum ownership value as ǩ=6 of an object o, and k-anonymity=10; then the user is permitted to use the anonymized o object with ǩ=6, 7, 8, or 9.

Most UNIX-based operating systems adopt RBAC. RedHat, SELinux, SUSE, SOLARIS, and other operating systems embed RBAC as a built-in access control for users and roles. RBAC is a fine-grained level used for controlling user access to tasks that would be normally restricted to root role. RBAC is an alternative solution for superuser group that contains root and other administrator roles in UNIX [24]. 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 to others. In MDSBA, there is a need for some superuser privileges, but not all, to run certain tasks.

RBAC provides authorization, which is a discrete right that can be granted to a role or a user. Authorization enforces policy at the user application level, while privilege enforces policy in the kernel level. Privileges and RBAC provide a compelling alternative model to the traditional superuser model. Privileges and RBAC are used in our framework to protect user’s processes, and any other attacks might be launched by user’s malicious programs. Attacks may occur against client machine, or any node of Hadoop infrastructure; this includes; master and secondary name node, slave data nodes, HDFS, Hadoop directories and job trackers. The mentioned nodes are joint in one domain, which increases the attack probability. The attack may be against processes, file systems, network configuration, and files and folders. Processes, network configuration and systems can be protected by using RBAC and privileges. On the other side, files and folders for read/write access can be protected by applying UNIX file system permissions or (chmod).

MDSBA implements RBAC in both SP, and FS sides. SP provides roles for managing Hadoop infrastructure, these roles vary depending on the SP hierarchy structure. However, four primary roles should be considered in Hadoop environment these are; hadoop manager, system manager; network manager, and data manager. The analyser can be subdivided into sub-roles like; data owner, software developer, database administrator, and analyst. Figure 5 illustrates an example of roles in SP.

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.



Figure 5. Suggested RBAC authorization structure for SP

In the federation service, many organizations store their own user’s authorizations and authentications details. In this paper, we will not discuss the details of FS side, and we will leave this for a future work. However, what we aim in this paper is the transferring of the xml file throughout the SAML tokens. SAML is an XML base single sign-on (sso) standard, which provides authentication and authorization mechanism, with an interoperability between different security services in distributed environments [25].

SAML standard can be implemented in different scenarios, this depends on the business needs and limitations. However, all scenarios follow close similar procedures as in [26, 27]. SAML 2.0 divides the XML file into 8 sections these are: response ID, Issuer ID, Status (success or Fail), Assertion ID, Signature key, Conditions, Authentication statement, and Attribute Statement. The last section contains unlimited names and values of attributes. Developers use this section to pass any authorization attributes and values [28]. In MDSBA, we need to inform the RBAC system with the ownership and access levels. This can be applied by embedding one or more attributes with the XML response file in the Attribute section [29].

The system sequence diagram, shown in Figure 6, describes steps of user’s access between SP and FS. The steps start from user’s authentication with the FS, and ends with the analytics query dispatch to the SP. The diagram briefly describes the RBAC mapping in the SP side. Hence, the algorithm diagram, illustrated in Figure 7, specifies the RBAC mapping procedure. Starting from receiving the xml token, creating a new user, or finding an available user, creating user’s own profile, and finally providing the user with an authentication for permitting access to Hadoop system.



Figure 6. Sequence diagram for SAML Request and Response and MDSBA access.

The procedures, of mapping any external user to SP system is shown in the algorithm diagram, Figure 7. The mapping script parses the xml script, and reads the user\_id. This id is equivalent to the username in the system. Therefore, the script searches for the provided username. If the username is found, then a new password is generating and prompt user to automatically login to UNIX shell. If the username is not found, then a username will be created and added to RBAC role. In both cases, the user logs-in automatically and he/she doesn’t need to be authenticated again. On each time login, the password will be generated randomly and stored in UNIX files. For securing this created password, an encryption using SSH is invoked.



Figure 7. The algorithm used for mapping users and groups (roles).

# Conclusion

We have investigated the reason caused the increased value of data loss, when implementing the current anonymization frameworks in big data. We pointed out the anonymization models specifications for big data. In k-anonymity, we have experimentally proven that the percentage of equivalent records proportioned extrusive with the increasing number of records. We have proposed a novel model of multi-dimensional sensitivity-based. The model is able to cope with the *map/reduce* operations and engineering methods. Besides, the model promotes a gradual access, by implementing discriminated levels of k-anonymity. The user’s access and operations security is controlled by RBAC model. The RBAC model manages and controls all Hadoop operations and user’s data. We have explained the weak approaches in top-down specialization model. Our proposed model showed promising results in the parallel distributed environment.

Our future work will further experiment our MDSBA model. We need to depict a clear process of RBAC in both sides of federation service and service provider. We need to further investigate more complex cases in delegating other federation services and RBAC mapping.

# References

[1] J. K. Jinzhu Kong, "A Practical Approach to Improve the Data Privacy of Virtual Machines," ed, 2010, pp. 936-941.

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

[3] Y. Demchenko, P. Grosso, C. de Laat, and P. Membrey, "Addressing big data issues in Scientific Data Infrastructure," ed, 2013, pp. 48-55.

[4] M. a. Alam, *Oracle NoSQL database : real-time big data management for the enterprise*: New York : McGraw-Hill Education, 2014, 2014.

[5] 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.

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

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

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

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

[11] 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.

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

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

[17] 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.

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

[19] 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.

[20] P. Pasupuleti, *Pig Design Patterns*. Birmingham: Birmingham : Packt Publishing, 2014.

[21] A. Armando, R. Carbone, L. Compagna, J. Cuellar, and L. Tobarra, "Formal analysis of SAML 2.0 web browser single sign-on: breaking the SAML-based single sign-on for google apps," in *FMSE '08*, V. Shmatikov, Ed., ed: ACM, 2008, pp. 1-10.

[22] S. Force. (2010). *Example SAML Assertions*. Available: https://help.salesforce.com/apex/HTViewHelpDoc?id=sso\_saml\_assertion\_examples.htm&language=en

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

[24] Oracle. (2015). *Role-Based Access Control*. Available: https://docs.oracle.com/cd/E23824\_01/html/821-1456/rbac-1.html

[25] O. Org, "Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0," p. 46, 15 March 200 2005.

[26] Venkadesh.M and A. C. S. Dr, "Study on Improving Web Security using SAML Token," *International Journal of Computer Technology and Applications,* vol. 04, p. 917, 2013.

[27] J. Paschoud, "SHIBBOLETH AND SAML: AT LAST, A VIABLE GLOBAL STANDARD FOR RESOURCE ACCESS MANAGEMENT," *New Review of Information Networking,* vol. 10, pp. 147-160, 2004.

[28] O. Canovas, G. Lopez, and A. F. Gomez-Skarmeta, "A credential conversion service for SAML-based scenarios," *PUBLIC KEY INFRASTRUCTURE, PROCEEDINGS,* vol. 3093, pp. 297-305, 2004.

[29] J. B. Rosenberg, *Securing Web services with WS-Security : demystifying WS-Security, WS-Policy, SAML, XML Signature, and XML Encryption*. Indianapolis, IN: Indianapolis, IN : SAMS, 2004.