Towards Optimal Sensitivity-Based Anonymization for Big Data

Datasets containing private and sensitive information are useful for data analytics. Data owners cautiously release such sensitive data using privacy-preserving publishing techniques. Personal re-identification possibility is much larger than ever before. For instance, social media has dramatically increased the exposure to privacy violation. One well-known technique of *k-anonymity* proposes a protection approach against privacy exposure. MDSBA adapts *k-anonymity* method with developing a better concept to apply a wider framework that can manage and control the user’s authorization access in a fine-grained approach and a granular concept. This approach may reduce the personal re-identification in a comprehensive framework for big data. However, this may lessen the usefulness of information gained. The value of *k* should be carefully determined, to compromise both security and information gained. Unfortunately, there is no any standard procedure to define the value of *k*. The problem of the optimal *k*-anonymization is NP-hard [1]. In this chapter, a greedy-based heuristic approach was suggested to provide an optimal value for *k*. The approach evaluates the empirical risk concerning Sensitivity-Based Anonymization method. The approach is derived from the fine-grained access and business role anonymization for big data, which forms the framework.

A Quasi-identifier (Q-ID) refers to a subset of attributes that can uniquely identify some tuples in a table. Incautious publication of quasi-identifiers will lead to a privacy leakage. Moreover, choosing a small number of Q-IDs may negatively affect the security, and may leave more chances of re-identifying personal information. Nowadays, re-identifying a person is much easier than ever before. This refers to the internet revolutionary with smart-phones, social media, and services automation. This indicates the risk of the auxiliary information that may lead to identifying a person, and regardless the relationship between the adversary and the victim. This recalls a need for increasing the number of Q-IDs. Any personal information as little as simile should be considered as a Q-ID. Guessing the personal information that may cause re-identification is impossible. The current anonymization methods assume a limited number of Q-IDs, usually up to seven Q-IDs. MDSBA increases the number Q-IDs and merges them in a small number of attributes, between two to four for each group. Each group is denoted by G(QID), and each G group can be a combination of attributes, as discussed later in this chapter [2].

## Previous Solution to Find the Optimal *k* Value

Eliminating the re-identification through rigorous inference in data is NP-Complete. This is due to functional and multi-valued dependencies. Hence, preventing the privacy violation is eventually impossible [3] [4]. Data is redundant with multi-owner, and the same individual’s private information can be available with multi-owner. Hence, linking released data to other external data may carry hundreds of probable inferences. Moreover, the current social media revolution provides even a higher probability of identifying a person. Knowing few personal information can be facilitated and assisted by searching the individual’s social media public profile on Facebook, Twitter, or Linked-in to identify the age, address, current place, and possibly other personal information. This inference combination increases the privacy violation and derives unlimited possible types of attacks [5]. Data owners may fail to protect all kinds of attacks. Therefore, security advisers need to improve rules and techniques to protect privacy continuously. However, this does not guarantee a complete security protection but may reduce the probability of security threats [6].

The currently proposed *k-anonymity* models do not provide access control frameworks for big data analytics [7]. The core method of any access control is identifying analyzers’ needs of data analytics and interests. Researchers implement the de-identification technique that modifies the data such that no combination of quasi-identifiers (Q-ID) smaller than *k* [8]. This technique evolves the organizing of attributes in equivalent groups or domains to gain the *k-anonymity*. However, the key part of the de-identification is assigning proper Quasi-identifiers and the *k* numbers by data owners [9]. Hence, the amount of information gained does not only depend on the method of anonymization, but also, depends on the chosen number of *k* value, and the way of determining the Q-ID attributes.

There are no any direct instructions to assist business owners in selecting Q-IDs or *k-anonymity* values. Few studies highlight some thoughts about finding the optimal values of *k* and Q-IDs nomination [10]. Contemporary studies have proven that finding optimal *k* value is NP-hard [11]. The hard part of finding *k-anonymity* value is not about identifying any random number, but about finding the optimal known value that is said to be the best. Researchers proposed different techniques to find the best *k* value[12]. Proposed techniques implement either one of two methods; generalizing or specializing techniques. Generalization technique suppresses or generalizes all data, computes the cost metrics for finding the best cut, and assigning the optimal *k* value by recursively examining the best specializing level [13]. In the specializing technique, proposed researchers follow the greedy-based heuristic approaches by implementing the entropy equations or by adopting crowdsourcing answers before and after anonymization [14]. There are few studies suggested techniques for finding the optimal Q-IDs. A distinct Q-ID and a tuple separator are computed to determine the optimal Q-ID and *k* value [15]. However, proposed methods of finding *k* value are expensive solutions, and data owners need to scan datasets numerous times to enable the optimal *k* finding. Also, proposed methods did not study the increasing number of Q-ID attributes on improving the de-identification. A solution for finding the optimal value of *k* in *k-anonymity* method is proposed in this chapter. The solution approach is based on the MDSBA for big data analytics. The framework provides data analytics granularity for multi-domain access.

## MDSBA Grouping and the Gradual Access

This chapter proposes a guidance for data owners on assigning *k-anonymity* parameters. The suggested proposal is related to the role-based anonymization control framework. The framework provides a fine-grained access control by dividing Q-ID attributes into vertical groups, with two to four attributes for each cluster. The core method of the granular access is introduced by three approaches; the probability value of Q-ID attributes, the ownership level, and the grouping method of Q-IDs. The probability of Q-ID is calculated by counting the number of unique values in the specified attribute. The ownership level is the key point of distinguishing user’s access levels. The value is an element of *k* in *k-anonymity* method, which determines the number of equivalent records on anonymization, where 2≤ ≤ *k*.

Related to Definition 3.1 in chapter three, number of G(QID) groups can be described mathematically, by denoting a number n of Q-IDs, where Q-ID={qid1, qid2, ..qidn} in a table T. The sensitive attributes C are denoted by the number s, C={c1, c2, …,cs}. Each two to four Q-IDs are grouped in a group G, so the number of created groups, denoted by ϒ, is related to the total number of Q-IDs (n), and can be presented by Each non-overlapped G(QID) group consists of a number of Q-IDs and one class, which is usually mapped to one or more business roles R. Let us also denote U as a user, and O as an organization. The role-based anonymization control is presented as; {G 🡪 R}(many to many relationship), {R 🡪 O}(many to many relationship), and {O 🡪 U}(many to many relationship). As described, users are given permissions to access Q-IDs through their own organizations. Next, a *k* value of *k*- anonymity is assigned to each G group, while is assigned to each organizational R. This means that *k* is a fixed value, and given only once to the G groups regardless user’s privileges, while is a dynamic changing value and given differently to each role of each organization. Hence, similar role R is given different values for various organizations. To setup the core method of the granular access; data owners create G groups and assign a proper *k* value for each group. On the other hand, Organizations are delegated some G groups as per business needs. Every organization is given a set of roles with an authorization level of percentage value () for each role. Theis decided based on the service level of agreement between the data owner and the organization, and is given within the range between [0.1 – 1]. The value is calculated as. Organizations are delegated with the requested business roles. It is the organization’s mission to assign its own users to the proper business roles.

It is evident that personal re-identification increases parallel with the number of permitted Q-ID increase. It is believed that any information belonging to individuals may support the personal re-identification. It was proven that the knowledge of additional attributes other than Q-IDs, can also, raise the problem of unique identifiers [16]. Hence, additional attributes are preferable to be accommodated within G(QID) groups. Therefore, MDSBA considers each auxiliary attribute as a potential Q-ID. The proposed technique, of finding *k* value, determines the optimal *k* value used in *k-anonymity* method. In general, finding the optimal *k* value requires many experimental trials. Finally, the data owner determines the lowest *k* value that leads to the least information loss. In big data, the case is even worse and requires more computation time and costs. In MDSBA, the optimal *k* value is determined with the minimal computation time and costs, as described in section 6.4.

MDSBA method is operated by MapReduce operations. Its approach implements iterated split and filter techniques for data records with mapping, shuffling, and reducing. The first split aggregates data based on sensitive class value C. Each group, concludes with one class value C, is anonymized separately. This kind of split supports the parallel operations of MapReduce. The anonymization then is applied to the lowest probability attribute for each split. For instance, suppose that a G group contains two Q-ID attributes, that is G(QID)={sex, USA\_State}. The probability of sex is P(sex)=0.5, if the values of the attributes are only (male, female), while the P(USA\_State)=1/50=0.02. In this case, the anonymization is applied by grouping the sex records and anonymizing the USA\_state records. The amount of anonymization applied on sex records is related to the value of , where larger value of promote a higher anonymization level. Further details about mathematical computations and processes stages is explained in chapter 3 [17].

MDSBA divides the process into four to five stages. The zero stage filters the *obvious guess* records, as described in section 5.3.5. Stage one filters the fully equivalent records. All Q-ID attributes must be equivalent so they can be excluded from stage two process. This is stated by [group all(qid)]. The rest of the non-fully equivalent records are further filtered by semi-equivalency in stage two. The semi-equivalency is measured by grouping all Q-ID attributes but the lowest Q-ID probability attribute and stated by [group all(qid)-1]. The lowest Q-ID attribute is anonymized by either taxonomy tree, interval, or suppression. Stage three follows similar steps, by grouping all Q-ID attributes but the lowest Q-ID probability attributes, and stated by [group all(qid)-2]. The least probability attributes are anonymized and merged with the rest of the groups. In the final stage, grouping is only applied to one Q-ID attribute, which is the highest probability one.

## Possible Attacks against MDSBA

Sweeney has addressed few attacks against *k-anonymity* method [16]. The attacks may occur by multiple queries of analytics, so anonymization is applied on different Q-ID attributes on each anonymization time. Adversaries may request data several times with multi-queries, so the anonymization process may apply anonymity on the first Q-ID attribute in the first trial, and on the second Q-ID attribute in the second trial. Hence, linking chance between the two anonymized tables is high.

The previously mentioned attack is not possible in MDSBA. This refers to the consistency of anonymity. In MDSBA, Q-IDs anonymization refers to the attribute probabilities. The anonymization always starts with the lowest probability attribute. Hence, amending queries does not switch the anonymization process to other Q-ID attributes. Also, the class attribute, in MDSBA, must gain the *k-anonymity* equivalency principle. However, other possible attacks against MDSBA can be summarised into two types of attacks; *obvious guess* and Across Groups Unique Identifiers (AGUI). In the *obvious guess*, an adversary may be able to guess the sensitive attribute (class), if the Q-ID attributes are known to the adversary. This violation may appear when a group of equivalent records has one class value. In the previous chapter, it was experimentally proven that increasing the may reduce the obviously guessed records. The *obvious guess* impact may facilitate the security breach and increase the re-identification threat. For instance, if a group of patients has one value of the class ‘Diabetes=positive’ and they share the same race, age, and state, then the intruder can obviously guess the diabetes state of the patient. To avoid this breach, an initial filtration on stage zero is implemented, and before splitting data into class value groups. The *obvious guess* records can reduced by increasing the value of to a certain extent. However, preventing the appearance of *obvious guess* records is possible by filtering the records in a zero stage, by using UDF program. This resolution was further discussed in chapter 5 / section 5.3.7. The second possible attack against MDSBA is caused by splitting Q-ID attributes into small groups. This Across Groups Unique Identifier’s security threat is lower than the uniqueness appearing in each G(QID) group. The attacker needs to know almost everything about the victim’s background. This is believed to be a background knowledge attack. As proven experimentally in the previous chapter, resolving AGUI can be implemented by increasing the value of , so the number of AGUI records decreases. AGUI maximum security threat appears when users are given exactly two G(QID) groups. Giving users more than two groups may reduce the impact of AGUI.

## Finding the optimal *k* value

As discussed previously, finding the optimal *k* value was proven to be NP-Hard. It was also mentioned that small values of *k*-anonymity are inadequate in multi-Q-ID groups. Moreover, MDSBA framework relies on the granular access of users, which requires a gradual increase or decrease in value. A greedy-based heuristic approaches is suggested to find the optimal values of *k*. The obtained k value should be as large as possible. We need to consider some factors that may help *k* value estimation. The aim of determining the optimal *k* value is the tradeoff between security level and gained information level. We need to be aware of the security risk on assigning a low value of *k*, especially, when datasets contain multiple G(QID) groups. However, larger values of *k* may negatively affect the usefulness of gained information. Therefore, keeping the degradation of data to the minimal level is possible by choosing the optimal value of k.

It is obvious that the minimum permitted number of equivalent records is =2. This minimum value achieves the safe threshold value to prevent a unique identification. Based on this fact, the value of *k* can be assigned as per the value. As described before, the minimum permitted value of=0.1. Since = *k*, and the minimum =2, then the value of *k* should be at least *k*=2/0.1=20. Considering the minimum *k*=20, we may increase the value of *k* starting from *k*=20, because dropping *k* below 20 may eliminate the equivalency concept, and ruin the *k-anonymity* principle. In some cases, we may need larger values of *k*, so *k*=20 is the smallest possible value. Datasets with multiple G(QID) groups should be assigned with larger values of *k*. This is referring to the dramatic increase of AGUI values in G(QID) groups. In order to choose a larger *k* value for each G(QID) group, we first need to insure that the chosen *k* should not negatively affect the information gained to a great extent. To do so, let us study the cumulative frequency equation that presents a bench mark or a reference point on measuring the data contrast. The cumulative frequency can be calculated by the disruption equation, as described in the next section.

### Cumulative Frequency

The term frequency in statistics means the number of times a given datum occurs in a dataset. Two types of frequencies are defined in statistics, relative frequency and cumulative relative frequency. The relative frequency measures the fraction of times for a value occurrences. The total percentage of each occurrence must equal to 100%. The cumulative relative frequency represents the accumulation of the previous relative frequencies. The frequency always represents the occurrence of a particular discrete value or occurrences of the value in an interval of a data. This means all types of frequencies require intervals to find out the frequent values [18].

In archived data, it is possible to calculate the anonymization loss starting from *k*=20. MDSBA adapts a naïve equation to calculate the anonymization loss, known as disruption. The disruption equation measures the data disturbance occurred after applying anonymization. The disruption increases monotonically with the increasing value of *k*. The disruption value was previously described in section 4.5. The equation of calculates one block of anonymized data, where M denotes the number of the records in this lock, and N denotes the total number of records for all dataset. The total disruption for all dataset is calculated by the summation of all values of Ɗ[total]. However, this kind of calculation can be a stick in the wheels, since the value of *k* should be calculated before the anonymization process is conducted, while the disruption equations can only be calculated after the anonymization completion. To resolve this issue, a rough estimation for the disruption value is calculated before the start of the anonymization process. The estimation is conducted by finding four checkpoints of *k*.

To assign the checkpoints; data owners need to define the *k* value range first. This depends on the security level of the internal organizational policy. Data owners may choose a range between [20 – 80], or even higher. Data owners, then choose the four checkpoints, which are {} The symbol of represents the first value of the *k* range, which is =20. The second symbol of represents the last value of the *k* range, which can be =80 or even higher. The initial suggested *k* value is =20. The data owner needs to investigate the new value =30. If the CF of the new value is far away from the reference line, then *k* will remain 20. To save the calculation time for disruption values, it is better to count the number of non-equivalent records when *k*=20, and then when *k*=30. This rough calculation accelerates the process of finding the optimal *k* value.

Practically, finding the number of non-equivalent records is straight forward steps in Pig Latin script. A fast computation script groups all equivalent records, SG= group data by (QID0, QID1, QID2), and then filters the grouped records, (frequency= FILTER SG by *k* <= 20); the number of non-equivalent records for each value of *k* is accumulative. Hence, the cumulative frequency (CF) can be calculated based on the number of non-equivalent records. To state the CF mathematically, let us denote the disruption values for {} by { respectively. The values of represent the lowest and highest disruption for minimum and maximum *k*, while the values represent the current and new inspected disruption for the current and new inspected *k*. Eventually, the number of inspected values of *k* are denoted by I. The cumulative frequency is measured by disruption values as:

(1)

The inspected is approved if the 0.9 ≤ CF ≤ 1.1. A range of 0.2 is allowed for data dispersion in comparison to the linear increase of data disruption. This equation is formulated to reduce the information loss that occur as a result of anonymizing data with larger values of *k*. To demonstrate Equation 1, let us study the Seer data records, which concludes three G(QID) groups with classes in bold, as shown in Table 6.1.

A data owner has decided an interval of *k* between [20 -80], so the seven inspected values of *k* are {20, 30... 80}. It is apparent that *k* value increases by 10’s units to avoid the overhead computation cost. The initial value of *k* is (20), while the inspected value for the three G(QID) groups is *k*=30. The lowest disruption value for G(QID)1 is, the disruption is calculated based on the lowest probability attribute, which is (Age=0.012). Recalling Equation 1, the 0.93. Since the CF ≥ 0.9, then the new value of *k*=30 is accepted. Next, we need to inspect *k*=40, so 0.54, which is < 0.9. Therefore, the new *k* value is rejected, and the value of *k* remains as *k*=30. Figure 6.1 illustrates the CF of the G(QID)1, and the diversion when *k*=40 from the reference point. The reference point is found by calculating the linear disruption increase based on the minimum and maximum D values only.

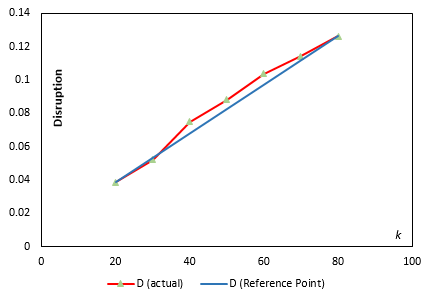


Figure 6.1- Cumulative Frequency for Seer G(QID)1 disruption.

Table 6.1- Seer dataset with three G(QID) groups.

|  |  |  |
| --- | --- | --- |
| G(QID)1 | G(QID)2 | G(QID)3 |
| Age | Year | CS Tumor |
| Sex | Race | Radiation |
| County | Grade | Marital Status |
| State (class) | Diagnosis (class) | Survival months (class) |

### Linear Regression

Regression Analysis is another statistical modeling that is related, in somehow, to the frequency distribution. This type of modeling helps to understand the relationship between an independent variable with a regular increase, and dependent variable changes. The regression analysis is being used for prediction and machine learning [19]. In MDSBA, linear regression analysis may give a general model for the anonymization loss. In anonymization concept, the regression analysis can be applied to measure the level of anonymization loss. It is known that increasing the *k* value will decrease the number of fully-equivalent records. This, in turn, will increase the anonymization loss, and as a result, will reduce the gained information. In another word, the anonymization loss increases parallel with the *k* value increase. The *k* value can be considered as an independent value, because it increases regularly without depending on any other variables. The anonymization loss depends on the *k* value, which means it is a dependent variable that changes increasingly parallel with the *k* value increase [20].

The previous cumulative frequency equation can be used in archived data to determine the optimal *k* value. The same equation can also be used in live data, if accuracy was ignored. Live data increases with the time, and the disruption values keep changing accordingly. The disruption value decreases proportionally with the increase number of data records. Actually, the previous CF equation is just a naïve equation derived from the linear regression. Finding the first and last disruption values to draw the regression line is an easy task but inaccurate. More accurately method is drawing a regression line and finding out the largest diverted values.

The regression line is found by the equation, where, and. Replacing the anonymization parameters imposes that the disruption line *d* replaces, and *k* replaces. The general regression equation is given by:

(2)

The value of the slope is formulated as:

(3)

(4)

Equations 2, 3 and 4 are used to calculate the regression line for the disruption values. This equation can be used in both archived and live data, to determine the largest disruption values in a specific range of *k* values for a certain size of the dataset. However, this equation is intensely needed in live data, since the regression line keeps changing parallel with the data growth. More data records impose a disruption decline with a minor disruption disparity between the various *k* values. For instance, if the disruption value for *k*=30 is *D*=0.001, then the disruption value for *k*=40 may be D=0.012. This small disparity value between two values *k* may lead to a high error rate, if the previous CF equation was used. Hence, using a more accurate equation is strongly recommended to determine the optimal value of *k*.

In the previous example, of applying CF on Seer data has resulted of *k*=30. The diagram of Figure 6.1 has shown a quite large disruption when *k*=40, hence, the successful value was *k*=30. If the regression line was replaced by CF, then Equations 2 and 3 calculate the *d* value as shown in Table 6.2. The values of D represent the actual disruption values, which are found by grouping and investigating the number of anonymized records. The regression lines of d and D values are plotted in Figure 6.2.

Table 6.2- Seer data / G(QID)1 results after line regression calculation

|  |  |  |  |
| --- | --- | --- | --- |
| ***k*** | ***d (reg. line)*** | ***D*** | ***D-d*** |
| 20 | 0.040879 | 0.03865 | -0.00223 |
| 30 | 0.055709 | 0.051998 | -0.00371 |
| **40** | **0.070539** | **0.074965** | **0.004426** |
| 50 | 0.085369 | 0.088056 | 0.002687 |
| 60 | 0.100199 | 0.103597 | 0.003398 |
| 70 | 0.115029 | 0.114283 | -0.00075 |
| 80 | 0.129859 | 0.126033 | -0.00383 |

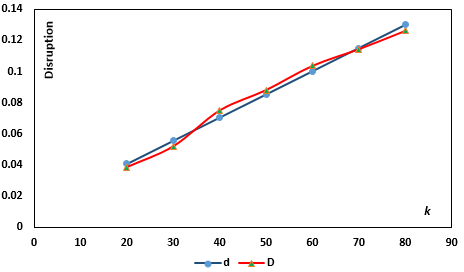


Figure 6.2- Regression calculation for Seer G(QID)1 disruption.

As noticed in Table 6.2, similar results to CF are found when *k*=40. Table 6.2 shows the subtraction result of (*D-d*). The table shows the largest contrast value between the regression line and the actual line, when *k*=40. It is essential to remember that if the *D* values are dropped below the regression line, then this value indicates a very low disruption, so it is accepted. Table 6.2 shows some negative values when *k*={20,30,70,80}. The largest disruption value must be positive. The largest value prevents *k* from gaining a higher value. In the previous example, the largest disruption value has occurred when *k*=40. The initially chosen value was *k*=20, then moved up to *k*=30. The *k* value progression was inhibited when *k*=40.

#### Three security levels

In linear regression, some disruption values are below the regression line, while others are above the regression line, as shown in Figure 6.2. This is apparent because the regression line presents the average middle line between various disruption values. Suppose that a number of disruption values, fall above the regression line, are three or four. The axiomatic question whether we consider any one of the three highest values or only the highest unique value as an inhibitor. For instance, if we study the disruption values for G(QID)2, as presented in Table 6.3. We find that the highest three disruption values appear when *k*=20, 30, and 60. The first two values have close similar values. In such cases, it is better to leave this for the data owner to determine the number of inhibitors. More inhibitors will hinder the *k* value progression, which means gaining smaller values of *k*, and as a result, less security applied.

Table 6.3- Seer data / G(QID)2 results after line regression calculation

|  |  |  |  |
| --- | --- | --- | --- |
| **Seer G(QID)2** | | | |
| ***k*** | **d** | **D** | **D-d** |
| 20 | 0.02468 | 0.027145 | 0.002465 |
| 30 | 0.03674 | 0.039215 | 0.002475 |
| 40 | 0.0488 | 0.048745 | -5.5E-05 |
| 50 | 0.06086 | 0.058712 | -0.00215 |
| 60 | 0.07292 | 0.076587 | 0.003667 |
| 70 | 0.08498 | 0.082143 | -0.00284 |
| 80 | 0.09704 | 0.098473 | 0.001433 |

High security imposes a higher value of *k*, so choosing one inhibitor can be used when data owners apply a high-security level on a specified dataset. Generally, we can decide three different levels of security {high, medium, and low}. Data owners may decide to go with any of the three security levels. In addition, other security options, such as the *k* value interval may support the security levels, as mentioned earlier. Briefly, we may summarize the security definition applied to any datasets by creating Table 6.4.

Table 6.4- Security levels options setup by data owners to decide the optimal *k* value

|  |  |  |
| --- | --- | --- |
| **Security-Level** | **Number of inhibitors** | ***k* value interval** |
| High | 1 | Example [20 – 200] |
| Medium | 2 | Example [20 – 150] |
| Low | 3 | Example [20 – 80] |

In Seer / G(QID)2 example, if data owner chose the security level=high, then the value of *k* for G(QID)2 is (***k* = 50**). This is because the highest distribution value was found on *k*=60. For the high-security level, we use only one inhibitor, which is *k*=60. Therefore, *k* should stop on the value before the inhibitor. If the security level=medium, then the value of *k* is (***k*=20**) because *k*=30 is the second inhibitor. Also, similar *k* value is gained for the medium security level.

## Finding the Optimal *k* percentage

It is important to remember that *k* value is given for each G(QID), and it is not related to user’s access and privileges. Regardless user’s access, *k* value is assigned to G groups, while is given as per user’s access level and privileges. For this reason, we need to identify a reasonable approach to assign the values for each permitted G(QID) group. The given value of can be determined based on the service level of agreement with organizations. However, a minimum value of should be determined to avoid security violation. The was clearly explain in the previous chapter/ section 5.2.2. The *k̄* is calculated as *k̄* =. Apparently, more given G(QID) groups manifest a higher personal violation probability. We need to investigate if this is correct, by experimenting various dataset groups. Experiments were conducted to measure the percentage of Across Groups Unique Identifiers (AGUI) appearance. The aim is to identify the factors that increase the AGUI records. This is essential to identify the best approaches for assigning the values. Two datasets of Adult and Seer were anonymized. Each dataset concludes three G(QID) groups as shown in Tables 6.1 and 6.5. The three G(QID) groups are mapped to two business roles (HR Manager, and Oncologist). In each dataset, the HR Manager is permitted to access G(QID)1 and G(QID)2, while the Oncologist is permitted to access G(QID)2, and G(QID)3, as shown in Figure 6.3. In the first experiment, an anonymization was applied on G(QID)1, and G(QID)2, which are mapped to (HR Manager). Every anonymized G(QID) was measured by disruption for different values of.

Table 6.5- Adult dataset with three G(QID) groups.

|  |  |  |
| --- | --- | --- |
| G(QID)1 | G(QID)2 | G(QID)3 |
| Age | Relation | Gain |
| Job | Race | Loss |
| Marital | Sex | Hrs-per-wk |
| Edu (class) | Work-class (class) | Salary (class) |

In the second experiment, an anonymization was applied on G(QID)2, and G(QID)3, and measured by disruption for differentvalues. In the third experiment, a user was given both roles; HR Manager, and Oncologist. The number of AGUI records are counted after completing the anonymization. The anonymization was applied with several values for both roles. The results are shown in Figures 6.4 and 6.5 for Seer and Adult datasets sequentially. The figures indicate that doubling the number of G(QID) groups will increase the disruption, which contributes to reducing the number of AGUI records. G(QID) groups, in both datasets, were given different values of *k*, range between [20 – 50]. The experiments show two factors may support the reduction of AGUI; the increased value of, and the increased number of G(QID) groups. However, the number of permitted G(QID) groups is related to the assigned roles, and we have no control over it. Thus, we prefer to control the. Users with one G(QID) permission group can be given any preferred value. On the other hand, the highest AGUI appearance occurs when the number of G(QID) groups=2. With the continuous G(QID) group’s increase, the number of AGUI keeps declining. This approach leads us to increase theto one, when the number of groups=2. The above diagrams show that AGUI number is reasonable when), and G(QID) groups are two.

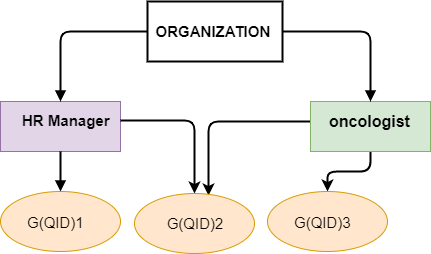


Figure 6.3- Groups mapping to business roles

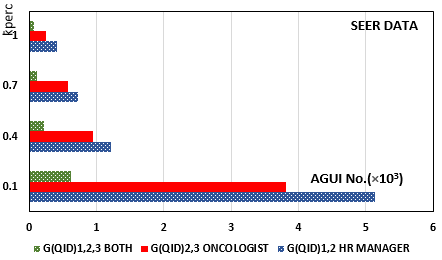


Figure 6.4. AGUI Num. for both roles in seer data.

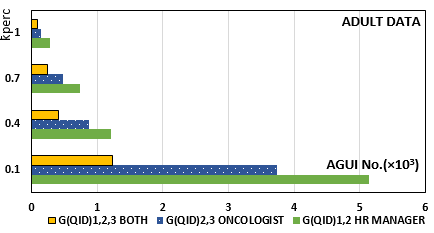


Figure 6.5. AGUI Num. for both roles in adult data.

The experiments showed that should be assigned based on the number of G(QID) groups. Referring to Figures 6.3 and 6.4, if an organization has been granted only Oncologist role with two G(QID) groups, then is fair enough to reduce number of AGUI to the minimal. If another organization was given both roles with three G(QID) groups, then is adequate to reduce the number of AGUI. The value of can be given to any role that is mapped to more than two G(QID) groups. However, the number of G(QID) groups is one factor, denoted by F (QID) that affects the estimation. Business owners may determine their own factors to conclude a comprehensive equation that estimates. In this chapter, only two factors are defined; F(QID) groups and the organizational trust level F(T). The F(QID) was explained earlier , as it is assigned by data owner. The F(QID) is given the value 1, when the number of G(QID)=2, and given the value less than 1, when the number of G(QID) > 2. The factor F(T) is given a value within the range between [0.1 -1], where 0.1 is the most trusted organization for the data owner, while the least trusted organization is given the value of 1. The final is found by calculating the average value of both factors. The final equation is. However, data owners may decide their own equations based on their own security policy.

In the previous example, if an organization has requested two business roles of Oncologist and HR Manager, and F(T)=0.2, then F(QID) can be given a value of 0.8, i.e. F(QID)=0.8. The final average is given by. In another example, if the same organization has requested only Oncologist role, then F(QID) can be given a value 1. This is because the two G(QID) groups manifest the highest AGUI value, hence,. Suppose that G(QID)1 is given *k*1=30, and G(QID)2 is given *k*2=40. The is calculated for the Oncologist by 1=300.6 = 18, and 2=400.6 = 24. If a user has requested an access with the Oncologist role, then a pair of values will be produced to authorize the user.

## Dynamic G(QID) groups

In some datasets, there are many attributes available in data schemas, while user’s inquiries and interests vary. In MDSBA, it is possible to control the amount of data that is permitted on the organizational level. Permitted data may consist of a small or large number of attributes. The G(QID) groups are pre-determined by data owners before mapping them to business roles. Roles later are delegated to organizations. Organizations need to look at each business role and its related mapped G(QID)’s, so they can decide the best-fit roles for their business nature. One of the obstacles that organizations may face, is the rigid G(QID) groups, where organizations really worry about attributes rather than groups. If, for instance, an organization requested only one attribute from a G(QID)1, and another attribute from G(QID)2. In this case, we may need to define new G(QID) groups for each organization, which is impractical. For this reason, dynamic G(QID) groups are preferable solutions, which suit different flavors of organizational queries.

To explain this approach, let us consider a set of attributes Vi for a dataset S= {V1, V2, V3, V4, V5, V6, V7}, if the G(QID) groups are determined by data owner as following G(QID)={G1(V1, V2, V3, V4 ), G2(V5, V6, V7)}, and the attached business roles to the G(QID) groups are R={R1(G1), R2(G2)}. Suppose that an organization has chosen to analyze the following attributes Ś= {V3, V4, V5, V6}. It is possible to enforce the organization to follow the G(QID) groups, but it will be more reliable if we have pre-defined G(QID) groups for such cases. The convenient solution can implemented by creating different patterns. Each pattern groups different attributes and assigned to separate business roles. In the previous example, we may define a second pattern for G(QID) group= {G3(V3, V4, V5, V6 )}, and we may map the second pattern to R={ R3(G3)}. However the third role (R3) should be exclusive to the second pattern, so each created pattern should have its own roles. Data owners may create two or three patterns for each dataset, and organizations may trade the best suit attributes. Next, the software application should be able to find the closest match pattern to the user’s choice.

One of the real examples that can be implemented by dynamic patterns, is a census and survey data. Census data consists of hundreds of attributes, and creating multi-pattern is the desired option for analyzers. One of the census examples is available in USA Ipums; the site provides many samples of data collected by survey or census [21]. Some datasets were downloaded from Ipums website, with the attributes shown in Table 6.6. The table shows chosen data after creating G(QID) groups and classes. The pattern divides attributes into five G(QID) groups, with one highlighted class for each group.

Table 6.6. Pattern 1 of census data presented by G(QID) groups.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PATTERN 1** | | | | |
| **G(QID)1** | **G(QID)2** | **G(QID)3** | **G(QID)4** | **G(QID)5** |
| REGION | FRIDGE | SCHOOL | NO\_CHILD | **EMPSTAT** |
| COUNTY | PHONE | HIGRADE | **RACESING** | LABFORCE |
| CITY | FUELHEAT | EDUC | BIRTHPLACE | LOOKING |
| **HOMELAND** | **VEHICLES** | GRADE\_ATT | YR\_IMMIG | WORKEDYR |
|  |  | **DEGFIELD** | SPEAKING |  |

The above attributes, also can be re-grouped in different patterns, and this depends on the user’s queries and demands. Table 6.7 shows another suggested pattern that data owners may follow.

Table 6.7. Pattern 2 of census data.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PATTERN 2** | | | | |
| **G(QID)6** | **G(QID)7** | **G(QID)8** | **G(QID)9** | **ATTRIBUTES** |
| N\_CHILD | HOMELAND | SCHOOL | REGION | FRIDGE |
| EMPSTAT | BIRTHPLACE | HIGRADE | COUNTY | PHONE |
| **RACESING** | YR\_IMMIG | EDUC | CITY | FUELHEAT |
|  | **SPEAKING** | GRADE\_ATT | LABFORCE | VEHICLES |
|  |  | **DEGFIELD** | **LOOKING** |  |

It is necessary to establish separate business roles for each pattern. This is essential to avoid *k* value conflict between G(QID) groups. Organizations may choose the required attributes, and an automated algorithm can choose the best fit pattern. For instance, if an organization decided to analyze the following attributes (REGION, COUNTY, LABFORCE, LOOKING, EMPSTAT). The automated decision can determine the best pattern by creating a comparison matrix. The matrix lists all available patterns, and calculates the availability of each attribute within the G(QID) groups. Table 6.8 matrix determined that pattern 2 is closer to the user’s query.

Table 6.8. Matrix for choosing the best pattern.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | PATTERN 1 | Availability Perc. | PATTERN 2 | Availability Perc. |
| **REGION** | G(QID)1 | 0.5 | G(QID)9 | 0.8 |
| **COUNTY** | G(QID)1 | 0.5 | G(QID)9 | 0.8 |
| **LABFORCE** | G(QID)5 | 0.75 | G(QID)9 | 0.8 |
| **LOOKING** | G(QID)5 | 0.75 | G(QID)9 | 0.8 |
| **EMPSTAT** | G(QID)5 | 0.75 | G(QID)6 | 0.33 |
|  | TOTAL | 3.25 | TOTAL | 3.53 |

The matrix calculates the availability percentage of each attribute and chooses the highest total pattern. The availability percentage is computed by dividing the number of attributes appearances in each G(QID) group over the total number of attributes for the specified G(QID) group. For example, the availability percentage for (REGION and COUNTY), in the first pattern, is the result of two attributes appeared in G(QID)1 over four attributes, which is the total number of G(QID)1 group attributes, so 2/4= 0.5. Similarly, G(QID)9 is calculated by counting the number of appeared attributes (REGION, COUNTY, LABFORCE, LOOKING), while the total number of G(QID)9 attributes is five, and the division result is 4/5= 0.8.

Practically, the matrix can be stored in database and presented as a part of entity relationship diagram, as shown in Figure 6.6.

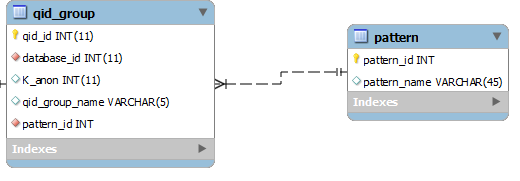


Figure 6.6- Part of ERD showing the relationship between patterns and G groups.

The matrix can be presented in an algorithmic structure by reading each pattern from the database, reading all related Q-ID groups, and finally, reading all attributes attached to each Q-ID group. Each G(QID) group is stored in a two-dimensional array, which stores the attributes and pattern number. For instance, in the previous example of the census, two patterns were created by the data owner. Each pattern consists of a set of G(QID) groups. An iterated algorithm counts the number of the available G(QID) for census dataset. In the previous example, the number of the available G groups is 9, so 9 arrays are stored in one three-dimensional. The array presents three values of; G(QID) number, pattern number, and Q-ID description as:

*array\_G\_QID [i][j][k]={(1,1, REGION), (1,1, COUNTY), (1,1, CITY), (1,1,HOMELAND) ,….,(2,9,REGION), (2,9,COUNTY), (2,9,CITY), (2,9, LABFORCE), (2,9,LOOKING)}.*

Next, the chosen attributes are also stored in a separate array. The array is iterated and compared with the G group arrays. With each comparison between the chosen values and the G arrays, mathematical calculations are conducted based on the number of match objects between both arrays. A complete algorithm is shown in Figure 6.7.

|  |
| --- |
| **Algorithm for finding the best fit pattern** |
| *Input: user chooses the needed attributes*  //the chosen attributes by users  // counts the number of patterns for database\_id =1111  //Loop for each pattern  //read all Q-ID IDs from the specified pattern  ”  //Loop for each G(QID)  ”  // Create three-dimensional array for all patterns  (k))  /\* The array is created and now it is ready for matching\*/  //the first loop for the patterns available  //The second loop for all groups in each pattern  //Number of Q-IDs in each G(QID)  //Loop for user’s attributes  //If statement to match attributes  //Number of equal objects between G(QID) and user’s attributes  //Calculate the probability between equal objects and the actual Q-IDs |

Figure 6.7-Algorithm for determining the best patters.

## Summary

It has been well-established that a major issue with *k-anonymity* relates to finding *k* value, which is an NP-hard problem. The previously proposed methods of finding *k* value are computationally expensive. Data owners need to experiment with various values of *k* to determine the best *k* value that concludes the highest information gain. These methods are even harder in big data anonymization. MDSBA provides a greedy-based heuristic approach to find the optimal values of *k*. The framework provides multi-technique to support the fine-grained access control over the MapReduce environment. The approach suggests an initial *k* value with a possibility to increase the value gradually based on the cumulative frequency (CF) of the data. CF equation is practically cost-effectively computation method, and is able to provide an approximate *k* value before commencing the anonymization process. Moreover, Linear Regression is another computation method to find the value of *k* accurately. CF is a special case of Linear Regression, where Linear Regression concludes a reference line to calculate the optimal *k* value with a reasonable computation time.

The second part of the chapter is determining the access granularity, by choosing the best ownership level. This is implemented by dividing Q-ID attributes into small groups, and mapping the groups to business roles. Organizations, then, are delegated to the required roles with an authorization level of for each role. Theis determined by different factors. The experiments unveil two factors; trust percentage between data owners and organizations, and number of permitted G(QID) groups. The experiments recommended a value of 70% for, if the number of G(QID) groups are larger than two, and a value of 100% if the number of G(QID) group is two. The proposed method of finding helps reducing Across Groups Unique Identifiers (AGUI), while the proposed approach of finding *k* values helps reducing *obvious guess* risk.

The last part of this chapter is creating various G(QID) patterns by shuffling the Q-ID attributes as per organization’s desires. When the number of Q-IDs is quite large, then customers may choose different Q-ID attributes. Since access permissions are assigned on the G(QID) level, then users may permitted to access some unneeded Q-ID attributes. This randomness may create false access privileges, by allowing users to access unneeded attributes. To avoid such a security breach, several random G(QID) are created. The pattern is chosen automatically based on the chosen attributes. To do so a matrix is created to calculate the best value for the number of appearances.

References

[1] J.-W. Byun, A. Kamra, E. Bertino, and N. Li, "Efficient *k*-anonymization using clustering techniques," in *International Conference on Database Systems for Advanced Applications*, 2007, pp. 188-200.

[2] M. Al-Zobbi, S. Shahrestani, and C. Ruan, "Multi-dimensional sensitivity-based anonymization method for big data," in *Networks of the future: architectures, technologies, and implementations*, M. Elkhodr, S. Shahrestani, and Q. Hassan, Eds., ed Boca Raton: Francis, 2017.

[3] T. A. Su and G. Ozsoyoglu, "Controlling FD and MVD Inferences in Multilevel Relational Database Systems," *IEEE Transactions on Knowledge and Data Engineering,* vol. 3, pp. 474-485, 1991.

[4] M. Morgenstern, *Security and inference in multilevel database and knowledge-base systems* vol. 16: ACM, 1987.

[5] M. Smith, C. Szongott, B. Henne, and G. Von Voigt, "Big data privacy issues in public social media," in *Digital Ecosystems Technologies (DEST), 2012 6th IEEE International Conference on*, 2012, pp. 1-6.

[6] S. Yu, "Big privacy: Challenges and opportunities of privacy study in the age of big data," *IEEE access,* vol. 4, pp. 2751-2763, 2016.

[7] R. Lu, H. Zhu, X. Liu, J. Liu, and J. Shao, "Toward efficient and privacy-preserving computing in big data era," *IEEE Network,* vol. 28, pp. 46-50, 2014.

[8] J. P. Daries, J. Reich, J. Waldo, E. M. Young, J. Whittinghill, A. D. Ho*, et al.*, "Privacy, anonymity, and big data in the social sciences," *Communications of the ACM,* vol. 57, pp. 56-63, 2014.

[9] E. Kabir, A. Mahmood, H. Wang, and A. Mustafa, "Microaggregation sorting framework for *k-anonymity* statistical disclosure control in cloud computing," *IEEE Transactions on Cloud Computing,* 2015.

[10] R. Motwani and Y. Xu, "Efficient algorithms for masking and finding quasi-identifiers," in *Proceedings of the Conference on Very Large Data Bases (VLDB)*, 2007, pp. 83-93.

[11] H. Park and K. Shim, "Approximate algorithms for *k-anonymity*," in *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, 2007, pp. 67-78.

[12] A. Meyerson and R. Williams, "On the complexity of optimal *K-anonymity*," in *PODS '04*, C. Beeri, Ed., ed: ACM, 2004, pp. 223-228.

[13] R. J. Bayardo and R. J. Rakesh Agrawal, "Data privacy through optimal *k*-anonymization," ed. USA, 2005, pp. 217-228.

[14] R. Hariharan, C. Mahesh, P. Prasenna, and R. V. Kumar, "Enhancing privacy preservation in data mining using cluster based greedy method in hierarchical approach," *Indian Journal of Science and Technology,* vol. 9, 2016.

[15] Y. X. Rajeev Motwani, "Efficient Algorithms for Masking and Finding Quasi-Identifiers," 2007.

[16] A. Basu, T. Nakamura, S. Hidano, and S. Kiyomoto, "*k-anonymity*: Risks and the Reality," in *Trustcom/BigDataSE/ISPA, 2015 IEEE*, 2015, pp. 983-989.

[17] M. Al-Zobbi, S. Shahrestani, and C. Ruan, "Sensitivity-based anonymization of big data," in *Local Computer Networks Workshops (LCN Workshops), 2016 IEEE 41st Conference on*, 2016, pp. 58-64.

[18] B. I. Susan Dean. (2010, 29/12/2016). *Sampling and Data: Frequency, Relative Frequency, and Cumulative Frequency*. Available: https://[www.saylor.org/site/wp-content/uploads/2011/06/MA121-1.1.3-3rd.pdf](http://www.saylor.org/site/wp-content/uploads/2011/06/MA121-1.1.3-3rd.pdf)

[19] N. R. Draper and H. Smith, *Applied regression analysis*: John Wiley & Sons, 2014.

[20] S. Chatterjee and A. S. Hadi, *Regression analysis by example*: John Wiley & Sons, 2015.

[21] K. G. Steven Ruggles, Ronald Goeken, Josiah Grover, and Matthew Sobek, "Integrated Public Use Microdata Series: Version 6.0 [dataset]. Minneapolis," in *MN: University of Minnesota*, M. U. o. Minnesota, Ed., 6.0 ed. [www.ipums.org:](http://www.ipums.org:) IPUMS-USA, 2015.

[22] 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.