Barriers to the implementation of k-anonymity and related microdata anonymization techniques in a realworld application

Barriers to the implementation of k-anonymity and related microdata anonymization techniques in a realworld application

Andreas Wiegnand, 1878334
Ludwig Schallner, 1850413

Abstract

ToDo Abstract

1 Introduction

Nowadays data are a key factor in almost every domain. It is comparable to the gold rush of the 19th: century 19^{th} century [11].]. Furthermore, storage space and network ability become increasingly affordable [13]. This is leading to an open-source community where the created and stored data are not only useful to the original data holder, but also to other researchers. In some cases the data are only useful when they are combined and analyzed together with other data. However, those data may contain some personal or sensitive information; thus the data should only get released if their privacy is secured [8].

Table 1. Basic example

SSN	Age	Postcode	Problem
680-90-2665	25	4568	procrastination
008-07-4179	34	4567	stress
391-05-7998	48	4569	stomach cancer
078-36-3853	39	4568	obesity
411-71-9290	42	4561	stomach ulcers
527-59-1948	27	4568	stress

The data presented *Table 1* must first be anonymized before their release is approved. A very common technique to achieve this goal is the so-called k-anonymity process, which prevents the danger of private data leakage. This paper aims to show the barriers to the implementation of k-anonymity. Section 1 will present the required theoretical background to understand k-anonymity and its purpose. Section 2 will discuss the underlying barriers of k-anonymity, while Section 3 takes into consideration the possible attacks of k-anonymity, which might function as barriers to this process. Section 4 explains how multiple algorithms can be implemented k-anonymity. A summary of the implementations of this process and its possible barriers will be provided in the last section of this paper.

2 Basics

In order to understand the process of k-anonymity one should be familiar with its basic concepts. This is possible by presenting and explaining in detail the theoretical background. The first term to be introduced is *microdata*. The data being processed contain records of information about individuals. Microdata is much more aggregate than usual data. These microdata are naturally accessible, as that everyone who has these data can run his/her own statistics from them [1]. Another important term is *Identifier*. These are attributes which can detect explicitly the record owner without any other attribute. For example, the owners full name (first and last name), telephone number, social security number, and even more specific private data [5].

In order to anonymize the data, *identifier* is removed from the published data. However, there is another process that are retained are the so-called *Quasi-identifier*. These are attributes which non-explicitly identify the record owner. When they are combined with other non-explicit attributes or other tables, they can re-identify the record owner. In such cases these combinations of attributes are called quasi-identifier, such as *gender*, *age*, *postcode*, *weight* and *height* [4]. An example of this process is to be found in figure 1 (the quasi-identifier of figure would be ZIP (postal code), birth date and sex).

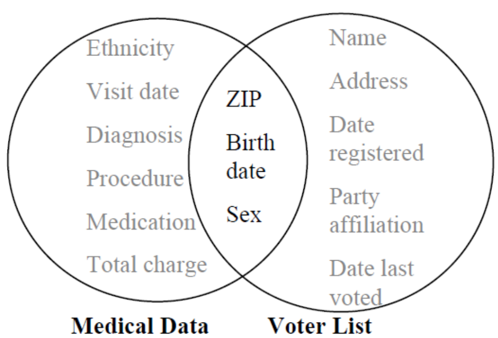

Fig. 1. Quasi-identifiers

The most important type of data worth mentioning is *sensitive data*. This type of data is useful to researchers but they contain private information and should not be publicized or be accessible to strangers. Record owners do not want to be linked to these type of data. Next to data types there are 3 ground knowledge types: *Background knowledge*, *Instance-level background knowledge*, *Demographic background knowledge*. *Background knowledge* deals with the uncertainty of the amount of data access the attackers has. The developer has to consider that the attackers might have access to a table, since they know that in order to achieve k-anonymity tables are generalized. Furthermore, the attackers are aware of the domain attributes. Instance-level background knowledge, then, reveals that the adversary possesses comprehensive knowledge of the targets specific details. For example, Alice (the adversary) knows that Bob does not suffer from a disease, because he has no symptoms. In this case, Alice can conclude of what actually Bob suffers from. Lastly, in the demographic background knowledge the adversary is informed about general facts, for example P (t[condition] = cancer— t[Age] 40). With these information the attackers can access and interference the records [9].

Considering *K-Anonymity*, the main goal of making a k-anonymized table is to have at least (k-1) tuples of each identical tuple by taking the corresponding quasi-identifiers into account [13, 8]. For example, the k-anonymized version of *Table 1* in the introduction section would be the following table:

Table 2. Basic example 2-anonymized

SSN	Age	Postcode	Problem
*	2*	456*	stress
*	3*	456*	stress
*	4*	456*	stomach cancer
*	3*	456*	obesity
*	4*	456*	stomach ulcers
*	2*	456*	stress

The set of all tuples with the identical quasi-identifiers of a table are referred as equivalence class [8].

It is also important to note that there are two kinds of disclosure, *identity disclosure*, during which an individual is linked to a particular record. Due to this *attribute disclosure* might occur, once new information about an individual are revealed. For example, Bob is linked to his record in *Table 2*, because of some attack (see Section 3.2). The adversary discovers that he suffers from stress [13].

The last two terms concerning the basic concepts of k-anonymity are *Global recoding/domain generalisation* and *local recording*. The first refers to a very common generalization technique during which once an attribute value is generalized then all occurrences of that value are replaced by the generalized one [13, 12, 8, 7]. The later entails coding strategies, which work differently from the ones described above. *Local recording* generalizes the attribute values in cells. Consequently, these types of strategies do not over generalize the table and the data distortion is significantly lower [8].

3 Underlying Barriers

In the following section, the paper will present the basic and most challenging barriers to the implementation of k-anonymity. The barriers, which can emerge during the implementation of k-anonymity, will be explained further. These are the so-called *distortion of data*, or as mentioned in some papers *data loss*.

3.1 Distortion of data as Barrier

A basic underlying barrier of k-anonymity is the process of measuring whether an implementation has been successful or whether it leads to a satisfying result. This can be measured by a simple calculation. The *modification rate* denotes the fraction of cells which are modified within the attribute set of the quasi-identifier [8].

Table 3. a: original table, b: example for local recording, c: example for domain generalization

a			b			c		
Gender	Birthday	Problem	Gender	Birthday	Problem	Gender	Birthday	Problem
male	13.08.1962	stress	male	13.08.1962	stress	*	196*	stress
male	28.10.1967	obesity	male	28.10.1967	obesity	*	196*	obesity
male	20.01.1977	stress	*	197*	stress	*	197*	stress
female	15.09.1973	obesity	*	197*	obesity	*	197*	obesity
female	15.03.1985	stress	female	15.03.1985	stress	*	198*	stress
female	28.05.1986	obesity	female	28.05.1986	obesity	*	198*	obesity

As can be observed in *Table 3b*, the modification rate is $33{,}\overline{33}\%$ (4 out of 12 quasi-identifier are changed) but for *Table 3c* the rate is 100% (12 out of 12 quasi-identifier got changed). As it can be seen in this simple example, the modification rate calculation is an unsatisfying procedure. Due to this, Li, Wong, Fu and Pei introduced the **weighted hierarchical distance**. In order to calculate the weighted hierarchical distance of a cell which is generalized from level p to level q, the following formula is used

$WHD(p,q) = \frac{\sum_{j=q+1}^{p} \omega_{j,j-1}}{\sum_{j=2}^{h} \omega_{j,j-1}}$ [8].

Let the hierarchy of birth date be {D/M/Y, M/Y, Y, 10Y, C/T/G/P, *}. Where D/M/Y stands for day.month.year, 10Y a 10 years interval and C/T/G/R for Child/Teen/Grownup/Pensioner.

Another example can be shown by having a uniformed weight $w_{j,j-1} = 1$ where $2 \leq j \leq h$ [8]. By Using the aforementioned example Birthday is generalized from D/M/Y to 10Y, which corresponds into $WHD_{Birthday}(6,3) = \frac{3}{5} = 0,6$. The Gender generalization would be $WHD_{gender}(2,1) = \frac{1}{1} = 1$. In such, in order to generalize 5 cells of age from D/M/Y to 10Y, one has to do the same data distortion as when 3 cells of gender are generalized from Male/Female to *. This calculation presents a much better way to address the distortion of data than the modification rate. However, this does not take into account how close is the generalization to the root (which would be *).

A last example can be shown with height weight $w_{j,j-1} = 1/(j-1)^{\beta}$ where $2 \leq$ j $\leq$ h and $\beta = \mathbb{R} \geq 1$ [8]: β can be chosen by the user. For example $\beta = 1$. For $WHD_{Birthday}(6,3) = \frac{0,\overline{33}+0,25+0,20}{1+0.5+0,\overline{33}+0,25+0,20} \sim 0,3431$. For $WHD_{gender}(2,1) = \frac{1}{1} = 1$. The distortion of almost 3 changed cell of birthday from D/M/Y to 10Y have the impact on the distortion as only one cell from Female/Male to * gets generalized.

Coming to a conclusion the researcher demands the information provided in the tables, as shown in these examples. Therefore, it is very important that the least possible information are lost during the anonymization process. In order to understand the importance of this aspect, consider another example. You a table with survivors from a disaster beyond all expectations. Researchers will try to discover the long-term effects of this disasters and whether the victims are more likely to live a long and happy life by measuring their distance from disasters location. If there is a mass generalization of the by location, it might be useless or non-significant for researchers to work with this information.

3.2 Attacks as Barrier

Furthermore, also attacks have to be considered as barriers for the implementation, because if the implementation ignores the weaknesses which the attacks use, k-anonymity will be useless. It is absolutely necessary that an attacker, under no circumstances, can learn about whatsoever target if he is studying the published database. Not even if the attacker has background knowledge from any other sources [3]. Unfortunately like Dwork showed 2006 that such safety is impossible because of the impossibility to predict what the attacker may know. Therefore its important and necessary that the implementation takes possible attacks into account and implement countermeasures, but because attacks are not the main part of this paper it will be only a short introduction.

Homogeneity attack As an example, let Alice be the adversary and let be Bob her target. They are neighbors, some day Bob get transported with an ambulance to an hospital. Assume the hospital published the table **??**, where all current patients with them Nationality, Age, ZIP, and Problem are listed, but this table got 4-anonymized before release. Alice knows that Bob is a 31 old, American who lives in ZIP Code 02239. She can conclude that either he is entry 3, 5,6, or 11. Furthermore, all of these entry have the same Problem, Cancer. Alice can conclude Bob is suffering from Cancer even if the table the table got 4-anonymized [13, 9]. To counter such attacker **diversity** is needed [9].

Such method is the so-called l-diversity which will not addressed further in this paper.

Table 4. Homogeneity attack

	Nationality	Age	ZIP	Problem	Nationality	Age	ZIP	Problem
1	American	42	02135	Viral Infect	*	≥40	021**	Viral Infect
2	Japanese	41	02133	Hearth disease	*	≥40	021**	Hearth disease
3	Germany	38	02238	Hearth disease	*	3*	0223*	Cancer
4	Japanese	29	02139	Fever	*	≤30	021**	Fever
5	Indina	37	02232	Viral Infection	*	3*	0223*	Cancer
6	Native-american	34	02236	Cancer	*	3*	0223*	Cancer
7	Russia	53	02138	Viral Infection	*	≥40	021**	Viral Infection
8	China	23	02139	Cancer	*	≤30	021**	Cancer
9	American	23	02141	Short of breath	*	≤30	021**	Short of breath
10	Indian	46	02139	Viral Infection	*	≥40	021**	Viral Infection
11	American	31	02239	Vomiting	*	3*	0223*	Cancer
12	American	28	02130	Viral Infection	*	≤30	021**	Viral Infection

Background knowledge attack This attack use the demographic background knowledge, which got explained in the basics, of an adversary. Assume Alice have a college, which get also to the same hospital. This college is 32 years old, Japanese and have the ZIP 93607. Everyone with the same quasi-identifiers (Age = 3* and ZIP = 936**) have a cancer or a hearth disease. Because she knows that Japanese have a very low risk of a hearth disease she conclude her college has cancer [9].

Table 5. Background Knowledge Attack

	ZIP Code	Age	Disease	ZIP Code	Age	Disease
1	93677	29	Liver Disease	936**	≤30	Liver Disease
2	93602	22	Liver Disease	936**	≤30	Liver Disease
3	93909	52	Cancer	9390*	≥40	Cancer
4	93906	47	Flu	9390*	≥40	Flu
5	93673	36	Hearth Disease	936**	3*	Hearth Disease
6	93607	32	Cancer	936**	3*	Cancer

Unsorted matching attack against k-anonymity This attacks is based on the very common strategy to release two tables separately. For example assume a two column weight table (a). This table get separated in two tables (b, c). Table b will contain Age completely generalized but ZIP ungeneralized, table c will have Age ungeneralized but Zip Generalized. The adversary just will merge both tables and will get table (a), and get access to sensitive information. This weakness can be fix via random sorting. [10].

Table 6. My caption

a		**b**		**c**	
Age	ZIP	Age	ZIP	Age	ZIP
42	91058	*	91058	42	91050
44	91058	*	91058	44	91050
50	27785	*	27785	50	27780
52	27785	*	27785	52	27780
20	32105	*	32105	20	32100
21	32105	*	32105	21	32100
31	67676	*	67676	31	67670
32	67676	*	67676	32	67670

Conclusion After showing possible attacks on k-anonymity it should be clear that before implementation an application with k-anonymity, these attacks should be tested and the application should be secure against any possible attack.

3.3 NP Hard

Meyerson and Williams analyze the production of an optimal K-anonymity solution in their complexity and found out that it is an NP-Hard Problem. Which means that the problem is at least NP-Complete but maybe harder. That can result that the Algorithm which should produce optimal K-anonymity will maybe not find a solution. For the real world application, this means that we are not sampled of producing a k-anonymity solution with less information loss as possible which results in worse datamining and maschine Learning Applications. However the show an approximation algorithm for k-anonymizing, which will take polynomial time and will use suppression the most O(k log k) [13]. The problem with suppression is the high information loss. So someone had to choose between time complexity and information loss as a barrier for the implemenation of k-anonymity [6].

4 Algorithm

This section will show some algorithms which goals is to archive k-anonymity through generalization.

4.1 The KACA Algorithm

This algorithm idea is to archive k-anonymity by clustering attribute hierarchical structures. The algorithm choose a random equivalent class, which is smaller than k. The next step is to form a lager equivalent class by merging the chosen one with the closest equivalent class. Which is resulting in a larger combined equivalent class. Through repeating this process the final result is that each equivalent class consists of at least k tuples [8].

Algorithm 1: K-Anonymization by Clustering in Attribute hierarchies (KACA) [8]

form equivalence classes from the data set
while *there exists an equivalence class of size* $< k$ **do**
    randomly choose an equivalence class C of size k k
    evaluate the pairwise distance of C and all other equivalence classes
    find the equivalence class C' with the smallest distance to C
    generalise the equivalence classes C and C'
end

This algorithm has a runtime of $O(nlogn+|E|^2)$. Li, Wong, Fu, and Pei have shown that their KACA-Algorithm is resulting in a 5.57 times smaller amount of distortion as the well known Incognito algorithm. The reason is lying in the technique which Incognito is using. Its a global recoding algorithm, which is resulting in a over-generalized table [8].

4.2 The OLA Algorithm

The OLA Algortihm was original produced for the field of health data anonymization. Also he wants to prodcue optimal k-anonymity. That means producing the usual definition of k-anonymity with the differents to produce less information loss as possible. In LUDWIG KAPITEL was shown one possibilty of measuring the information loss. The Algortihm can use 3 different Kind of Information loss metrices which result in different anonymization result. Information loss can result in loss of statistical power, inaccurate analysis result and inefficient use of data. The alogrithem works with supression and generalization of the data. Supression can result in drastical information loss due to the fact that a hole attribute gets raced. generalization will performed on all potential Quasi identifiers.

For different kind of data their a different kind of genralization techniks. So for string the rightmost char can be deleted. For numerical data we can produce intervals which will include the generalicate attribute. For date we can reduce the specifikation from days to months till years. Generalization includes Supression. The hightes attribute on each generalication lattice is the supressed version of the attribute in includes no information at all.

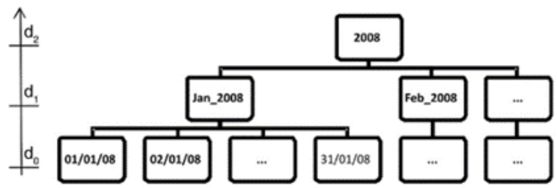

Fig. 2. Generalization

Algorithm 2: The OLA Algorithm works in 3 Steps:

---

1 **while** *not all generalization strategies are compared* **do**
2   For every generalization, strategy builds a binary search to find all k-anonymous nodes in the different strategies.
3   For every generalization strategy that includes k-anonymous nodes save the one with the least Information loss in the hole generalization strategy, this is referred to a local option k-anonymous solution.
4   Now compare the local optimum solutions to respect of the information loss. The one with the lowest Information loss of all local optimum solutions is the global optimum solution.
5   **return** *global optimal solution*
6 **end**

The most time consuming operation is finding the all the K-anonymous notes with and compare them to each other with respect to their information loss. To get a better performance at the Programm step 1) the OLA algorithm works with Predictive Tagging that boost the process. This Tagging take advantage of 2 Theorems of the generalization Lattice. That every k-anonymous note in the same generalization lattice on hight n. All notes above n and in the same generalization strategy are also k-anonymous. So the algorithm only has to find the first k-anonymous note in the strategy and tag all above as k-anonymous.

4.3 Cloaking Algorithm

Moving object data poses new challenges to a traditional database, data mining, and privacy-preserving technologies due to its unique characteristics: it is time-dependent, location-dependent, and is generated in large volumes of high-dimensional stream data. The following algorithm shows an example of privacy production.The Cloaking Algorithm tries to produce Anonymity on location-based data for users of Location Bases Services(LBS). The Cloaking Algorithm is installed on a location protection broker on a trusted server and anonymize messages which will afterward send the LBS. K-anonymity prevents such a privacy breach by ensuring that each individual record can only be released if there is at least k 1 other (distinct) individuals whose associated records are indistinguishable from the former in terms of their quasi-identifier values. there a two possible attacks to get the identity of a sender of a message. At the Restricted Space Identification, the attacker A observes that message M is sent from location L afterward he gets the background knowledge that L belongs to someone specific. For example, if Mr. Bob the owner of a flat sends a message and the attacker observes this message. He can re-link the identity of Bob. Another Attack is Observation Identication. If A has observed the current location L of subject S and nds a message M from L then A learns that S has sent M. To prevent this leaking of information the cloaking algorithm works with Spatial Cloaking and Temporal Cloaking. Spatial Cloakings goal is to increase the location of m in such a way that there are more messages in it. So that there is not only one message at a time in an area. Temporal Cloaking extends the sending time until more messages are in one area.

4.4 High-Dimensional Transaction Data

Transaction dada is typical high-dimensional. Shopping sites like Amazon.com got millions of catalog items which all could be a potential QID and therefor must be k-anonymitats before publishing this kind of data [14]. Aggarwal shown in his paper "On k-Anonymity and the Curse of Dimensionality" that this task is impossible to archive. They work with a clustering K-anonymity algorithm like The KACA Algorithm and try to archive 2-anonymity on a 3108 dataset. He shows with the increase of the dimension of the dataset and the goal to archive k-anonymity the information loss increase rapidly till a point of not practical for applications like data mining tools. The reason for that is the curse of high dimensionality which doesn't allow the clustering of points because of too much space between them. Xu introduced a method to get at least a bit of practical usage. He proclaimed that an attack knows at least n-different transactions of a victim and concludes that some information can get [15]. So he said that that the backgroundknowlwege of an attacker in bounded [2].

5 Summary

Like we saw in section OLA algorithm there are possibilities of choosing between different information loss metrics which all compute different values to the same k-anonymous node. So the implementer has to choose which one fits the most in his data. Different Metrics have pros and contras and have to be compared. Another problem with Information loss is that you can measure the information loss on your dataset but what would be more is interesting the information loss compared to upcoming data mining tool or machine learning application. The information loss metrics cant know which information is important for the upcoming data mining step.suppression can also harm the quality of the data and should be chosen wisely.The production of k-anonymity is NP-Hard which results in a difficult implementation in real-time applications like the Cloaking - Algorithm had shown. This Complexity can result in problems of finding an optimal solution in real-world data sets and can make a practical implementation difficult. High Dimensional Data like transaction data with more than thousand of attributes are in practice not capable to produce k-anonymity for all attributes. The reason for that is the so called Curse of High Dimensionality which produces a metric space which is too large to produce k-anonymity solutions for all attributes. A Possibility is so called, bounded background knowledge that some attributes cant be used as a quasi-identifier because the effort of getting background knowledge which can be linked to the attributes is estimated too high. Like shown i chapter Attacks as Barrier attacks against k-anonymity are a threat for the anonymity of the user and can result in identity disclosure.There solutions regarding these kind of attacks we proposed in this paper. L-Diversity and T-Closeness will help against them. The cloaking algorithm shoves a good example of the connection between anonymity and usability. The anonymization of the data reduces the usability of location-based services. Which comes from the construction of the Spatial cloaking and Temporal cloaking boxes to have enough messages to constrain them and let the be k-anonymous. The Software gives the option that a user of the client can decide how much usability he wants to sacrifice to get anonymity. So the user is at least under the control of how much utility he wants to give up.

References

1. Ipumsl-confidentiality, https://web.archive.org/web/20070823010133/http://international.ipums.org/international/
2. Aggarwal, C.C.: On k-anonymity and the curse of dimensionality. In: Proceedings of the 31st international conference on Very large data bases. pp. 901–909. VLDB Endowment (2005)
3. Dalenius, T.: Towards a methodology for statistical disclosure control. Statistik Tidskrift 15, 429–444 (1977)
4. Dalenius, T.: Finding a needle in a haystack or identifying anonymous census records. Journal of official statistics 2(3), 329 (1986)
5. Domingo-Ferrer, J., Torra, V.: A critique of k-anonymity and some of its enhancements. In: Availability, Reliability and Security, 2008. ARES 08. Third International Conference on. pp. 990–993. IEEE (2008)
6. El Emam, K., Dankar, F.K., Issa, R., Jonker, E., Amyot, D., Cogo, E., Corriveau, J.P., Walker, M., Chowdhury, S., Vaillancourt, R., et al.: A globally optimal k-anonymity method for the de-identification of health data. Journal of the American Medical Informatics Association 16(5), 670–682 (2009)
7. LeFevre, K., DeWitt, D.J., Ramakrishnan, R.: Incognito: Efficient full-domain k-anonymity. In: Proceedings of the 2005 ACM SIGMOD international conference on Management of data. pp. 49–60. ACM (2005)
8. Li, J., Wong, R.C.W., Fu, A.W.C., Pei, J.: Achieving k-anonymity by clustering in attribute hierarchical structures. In: International Conference on Data Warehousing and Knowledge Discovery. pp. 405–416. Springer (2006)
9. Machanavajjhala, A., Gehrke, J., Kifer, D., Venkitasubramaniam, M.: l-diversity: Privacy beyond k-anonymity. In: Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on. pp. 24–24. IEEE (2006)
10. Meyerson, A., Williams, R.: On the complexity of optimal k-anonymity. In: Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. pp. 223–228. ACM (2004)
11. Rossi, B.: Data revolution: the gold rush of the 21st century, http://www.information-age.com/data-revolution-gold-rush-21st-century-2-123460039/
12. Sweeney, L.: Achieving k-anonymity privacy protection using generalization and suppression. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10(05), 571–588 (2002)
13. Sweeney, L.: k-anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10(05), 557–570 (2002)
14. Wang, K., Chen, R., Fung, B., Yu, P.: Privacy-preserving data publishing: A survey on recent developments. ACM Computing Surveys (2010)
15. Xu, Y., Fung, B.C., Wang, K., Fu, A.W., Pei, J.: Publishing sensitive transactions for itemset utility. In: Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on. pp. 1109–1114. IEEE (2008)