29th INTERNATIONAL CONFERENCE ON DATA ENGINEERING WORKSHOPS YEAR 2013

Privacy against Aggregate Knowledge Attacks

Olga Gkountouna #1, Katerina Lepenioti #2, Manolis Terrovitis *3

#*National Technical University of Athens (NTUA)*
Athens, Greece
1olga@dblab.ece.ntua.gr
2ge03119@mail.ntua.gr

**Institute for the Management of Information Systems (IMIS)*
Athens, Greece
3 mter@imis.athena-innovation.gr

***Abstract*— This paper focuses on protecting the privacy of individuals in publication scenarios where the attacker is expected to have only abstract or aggregate knowledge about each record. Whereas, data privacy research usually focuses on defining stricter privacy guarantees that assume increasingly more sophisticated attack scenarios, it is also important to have anonymization methods and guarantees that will address any attack scenario. Enforcing a stricter guarantee than required increases unnecessarily the information loss.**

Consider for example the publication of tax records, where attackers might only know the total income, and not its constituent parts. Traditional anonymization methods would protect user privacy by creating equivalence classes of identical records. Alternatively, in this work we propose an anonymization technique that generalizes attributes, only as much as needed to guarantee that aggregate values over the complete record, will create equivalence classes of at size k. The experimental evaluation on real data shows that the proposed method produces anonymized data that lie closer to the original ones, with respect to traditional anonymization algorithms.

I. Introduction

It is often the case that the data published by an organization or company are too detailed to expect attackers to have accurate partial knowledge. Still, an attacker might have some aggregate or abstract knowledge of a record. Examples like this often arise in practice. For example IMIS is currently involved in anonymizing tax related data from Greece. Each record in this data collection has hundreds of fields, that trace financial activity often in a very detailed level. When publishing or sharing such data, we expect that the major threats come from attackers who will be able to identify records using aggregate knowledge e.g. total taxable income and not the exact values of fields that are hard to acquire as background knowledge, e.g., exact sum of expenses in agricultural financial activities.

The same case can appear in several other application areas; when publishing movement data an attacker might know how long a trip took, but not detailed information on the duration of each stop. Also when publishing medical data, an attacker might know a previous diagnosis for a patient that corresponds to a certain value range on a combination of indicators, but it is unlikely that he can have exact partial knowledge about exam results. Anonymizing such data under a traditional anonymization framework would guarantee privacy, but it would cause unnecessary distortion on the data, since we need only to create groups of similar records with respect to the abstract knowledge of the attacker (an aggregate function over the record fields).

Consider the motivating example of Table I which contains tax data of individuals. A realistic scenario is that an attacker may only know the approximate total income of a target, but not more detailed information. Thus, it is not the exact values of attributes that act as quasi-identifiers, but the aggregate information on them.

Assume that an attacker knows that Paul's total income ranges from 260 to 270. If Table I is published, after removing direct identifiers (names), the attacker can estimate the total income for each record and identify the one that belongs to Paul's. The 2-anonymous Table II is an anonymization of Table I. The last column (*Total Income*) is not published, it is inferred

Name	*Salary*	*Capital Gains*	*Other income*	(*Total Income*)
John	10	20	100	130
Nick	15	15	105	135
Paul	30	40	200	270
Mark	40	30	210	280

TABLE I
ORIGINAL TAX DATA

Id	*Salary*	*Capital Gains*	*Other income*	(*Total Income*)
1	[10-15]	[15-20]	[100-105]	[125-140]
2	[10-15]	[15-20]	[100-105]	[125-140]
3	[30-40]	[30-40]	[200-210]	[260-290]
4	[30-40]	[30-40]	[200-210]	[260-290]

TABLE II
CLASSIC 2-ANONYMOUS TABLE

Id	*Salary*	*Capital Gains*	*Other income*	(*Total Income*)
1	[10-15]	20	100	[130-135]
2	[10-15]	15	105	[130-135]
3	30	40	[200-210]	[270-280]
4	40	30	[200-210]	[270-280]

TABLE III
AGGREGATE 2-ANONYMOUS TABLE

by the attacker using the rest of the values. Any attacker with aggregate knowledge will not be able to identify less than 2 records. However, the same privacy can be ensured in Table III where records {1, 2}, as well as {3, 4}, are indistinguishable with reference to the aggregate function. As we can observe, not all values are generalized and a smaller information loss is achieved. In both cases Paul is indistinguishable from Mark. Nonetheless, In table III more detailed information about them is available without violating privacy with reference to the attacker's possible knowledge.

In this work we aim at providing a form of k-anonymity to prevent attacks against identity disclosure. We propose a local-recoding generalization approach that preserves utility by generalizing the least number of values necessary to form equivalence classes of size k (or more) with respect to the aggregate function. Compared to classic k-anonymity, even for local-recoding methods, we achieve better utility as we do not create classes of completely identical records.

The basic idea of our method is to form groups of records that have similar aggregate function values of their quasi-identifiers. To achieve this we perform local generalizations independently within each group. We limit our discussion to numerical values, but our method can be extended to categorical if aggregate functions are defined over them.

Our main contributions include the following:

- We define the problem of anonymizing data with respect to aggregate information;
- We define k^f*-anonymity* guarantee to satisfy privacy against aggregate-knowledge attacks;
- We propose a utility-preserving anonymization algorithm;
- We evaluate our methods with real-world data and compare our results to Mondrian, a multidimensional local-recoding k-anonymity algorithm.

II. Problem Definition

Let D be a relational table with numerical attributes $QI_1, QI_2, ..., QI_n$ of the same domain $\mathcal{I}$. Let f be an aggregate function defined on $\mathcal{I}^n \rightarrow \mathbb{R}$.

We consider attackers whose knowledge is limited to the value of $f(qi_1, qi_2, ..., qi_n)$ of a target record t. They can use their knowledge to identify records of D that have rare or unique values of f, and further discover detailed information about them, such as exact attribute values.

We provide a novel privacy guarantee to address these attacks that are limited to aggregate knowledge on the attributes of the target record.

Definition 1: (aggregate privacy guarantee) A database D is considered k^f-anonymous if any attacker knowing the value of an aggregate function f on the attributes for a record $t \in D$, is not able to use this knowledge to identify less than k records in D.

Our guarantee extends k-anonymity [1] for this attack scenario. Using the *Total income* (f=sum) to infer tax details of an individual, or using the GPA (f=average) to infer a student's grades are realistic examples of this attack.

Often, the attacker does not know the exact aggregate value with 100% precision but rather an approximation of it. For instance, she may not know the exact *Total Income* of a target is 32,128.22. Instead, she may know the income is in the range 30,000 to 35,000. This allows us to propose a more flexible version of the k^f-anonymity guarantee which preserves higher utility, as we show in our experimental evaluation.

Definition 2: (relaxed aggregate privacy guarantee) A database D is considered $k^{(f,d)}$-anonymous if any attacker knowing the value of an aggregate function f on the attributes for a record $t \in D$, there exists positive $d < 1$ such that at least k records in D have aggregate values $f \cdot (1 \pm d)$.

If a database D is not k^f-anonymous it can be transformed to $D^\star$, by generalizing attribute values, so that $D^\star$ satisfies the k^f*-anonymity*. Transformations of the records involve generalization of those values that are necessary to make groups of at least k records that have identical f values or identical value ranges of f, as in table III. A generalization is a set of rules in the form $v \rightarrow [a, b]$, which map a value v of the original data to a range that includes it.

There can be many possible anonymizations of a dataset that satisfy k^f-anonymity for a given function f, as shown in tables III and II. For instance, anonymizing all values to the domain $\mathcal{I}$ is a possible solution, but it would introduce the highest information loss, as the microdata would be practically useless.

The problem of k^f-anonymization of a dataset is to find the set of generalizations rules that satisfy k^f-anonymity and preserve as much data utility as possible.

III. Anonymization algorithm

A. Possible Solutions

The solution space is the set of all possible generalization rules, as in the classic k-anonymity. However, the set of accepted solutions which satisfy our guarantee is much greater. The problem of optimal multidimensional k-anonymity was proven to be NP-hard [2]. In the worst case, i.e., an aggregate function f which takes a different value for every combination of the record attributes, the problem is the same. To deal with the complexity of the optimal anonymization problem we have opted for a heuristic solution. We cluster the records into equivalence classes, and perform *local-recoding generalization* on the values of each equivalence class separately.

B. Algorithm

We propose a local-recoding generalization algorithm. As shown in the pseudo-code, our method has two main phases. Phase one divides the records into groups (lines 1-4). We form equivalence classes with respect to the f function. First, all records are sorted with reference to their $f(qi_2, qi_2, ..., qi_n)$ value. Then, they are clustered into equivalence classes of sizes $k \leq |EC| \leq 2k - 1$. We limit the EC size to avoid overgeneralizing values.

In the second phase we consider each equivalence class separately, and perform generalizations to its values (lines 6-16). If all records in a class EC already have the same result

Algorithm 1 aggrAnon **AA**(D, f, k)

```
Input: $D$ {Original Dataset}, $f$ {aggregate function},
       $k$ {privacy parameter}
Output: $D^\star$ {k$^f$-anonymous Dataset.}
 1: for all tuples $t < qi_1, qi_2, ..., qi_n > \in D$ do
 2:     estimate $f(qi_1, qi_2, ..., qi_n)$
 3: sort tuples with reference to their $f$ values.
 4: form groups of size $\geq k$ and $\leq 2k-1$
 5: for every group $EC$ do
 6:     if all tuples have the same $f$ value then
 7:         add $EC$ to $D^\star$.
 8:     else
 9:         $Q = \{QI_1, QI_2, ..., QI_n\}$ //$Q$ contains all attributes
10:         $j = n - 1$
11:         while $Q$ not empty do
12:             estimate $f$ for all combinations of $j$ attributes
13:             Let $C_j$ be the combination with most similar $f$
                    for all tuples
14:             generalize the remaining attribute $QI_j = Q \setminus C_j$
                    to a common range $[v_{min}, v_{max}]$
15:             remove $QI_j$ from $Q$
16:             $j = j - 1$
17:             estimate $f$ for all tuples in $EC$
18:             if all tuples have the same $f$ value then
19:                 break
20:         add $EC$ to $D^\star$
21: return $D^\star$
```

for function f, then EC is directly added to the anonymous result $D^\star$. Otherwise, we find the least set of generalizations to make all records in EC indistinguishable with reference to f. Let n be the dimensionality of the dataset, i.e., the number of attributes in a record. We calculate f for all combinations of $n-1$ attributes, i.e., f is calculated for all but *one* attribute each time. Assume $\{Q = QI_1, .., QI_{j-1}, QI_{j+1}, ..., QI_n\}$ is the set of $(n-1)$ attributes for which the value of f is more similar among all tuples in EC. Then we generalize attribute QI_j to the range $[v_{min}, v_{max}]$, where v_{min} and v_{max} are the minimum and the maximum value that appears in QI_j in class EC. We then remove QI_j from Q and continue with the rest of the attributes of Q. We calculate f for all combinations of $n-2$ remaining attributes of Q and repeat the above process until all tuples in EC give the same value of f.

The choice of the attribute to generalize at each step is made by estimating $f(t)$ for every tuple t in EC, having excluded one attribute each time, as described above. We denote as f_i the value of f when attribute i is excluded form the estimation. We compute all the differences $f_i(t) - f_i(t')$, for all tuples $t, t' \in EC$ $(t \neq t')$ and estimate their average $avg(f_i(t) - f_i(t'))$. We select the attribute j which corresponds to the minimum $avg(f_j(t) - f_j(t'))$. For reasons of simplicity we consider all QI attributes as equally important, or equally sensitive. It is straightforward to adjust the proposed method for taking into account weights.

Algorithm 1 can be modified to satisfy $k^{(f,d)}$-anonymity, by adjusting the termination condition at line 18, to require that all tuples to have aggregate function values in the range $f \cdot (1 \pm d)$.

More than one aggregate function. An interesting problem is when the attacker may know the values of more than one aggregate functions about a target tuple. Assume $f_1, \ldots, f_n$ are such functions. The attacker knows $f_1(t), \ldots, f_n(t)$ about a target tuple t. Algorithm 1 can be easily adjusted to this case by simply considering in all functions in the termination check of line 18. Moreover, the sorting and grouping of lines 3 and 4 has to take into account more than one function. This can be simply done by assigning a priority on how $f_1, \ldots, f_n$ are used in the sorting, or by employing a more advanced clustering algorithm.

Correctness. We can show that Algorithm 1 always produces a k^f-anonymous dataset, by showing that the output cannot contain any tuple t which shares the same value for function f, $f(t)$ with less than $k-1$ other tuples. This can happen if either its equivalence class is of size less than k or the class is not fully anonymized. The algorithm divides the tuples into equivalence classes of size $\geq k$ in the beginning (line 4) and each class is anonymized separately. Thus, it is impossible for a tuple to be in a class with size less than k records. The algorithm progressively generalizes each attribute inside its class to a common value, until the $f(t)$ value for every tuple in the class. Since the generalization hierarchy allows generalizing all domain values to common one and $f(t_1) = f(t_2)$ for $t_1 = t_2$, then there is always the trivial solution of generalizing all attributes in the equivalence class to the same value, thus creating identical tuples. Consequently, there cannot exist a tuple indistinguishable from less than k others in terms of the value of function f.

Note that in the worst-case scenario all equivalence classes will become totally k-anonymous. This happens if all attributes are generalized to their [min,max] range in every equivalence class. Thus, the upper limit of our information loss cost is the cost introduced by a local-recoding generalization k-anonymity approach.

C. Information Loss

To estimate the loss of utility introduced by the value generalizations we use Normalized Certainty Penalty (NCP) metric [3]. Let v be a value in original domain $\mathcal{I}$. Then:

$$NCP(v) = \begin{cases} 0, & v \textit{ not generalized} \\ |max_v - min_v|/|\mathcal{I}|, & \textit{otherwise} \end{cases}$$

where $[min_v, max_v]$ is the range to which v is generalized.

The total loss of an anonymous dataset D with $|D|$ records and n attributes, is the average NCP of all values:

$$NCP(D) = \frac{\sum_{i=0}^{|D|}\{\sum_{j=0}^{n} NCP(v_{i,j})\}}{n \cdot |D|}$$

where $v_{i,j}$ is the value of the j^{th} attribute in the i^{th} record.

IV. Experimental Evaluation

We evaluated experimentally the *aggrAnon* on real datasets from the UCI repository [4]. The implementation was done in C++ and the experiments were performed on a Core i7 at 2.13GHz with 8GB RAM, running Ubuntu Linux.

Algorithms. We compare our algorithm to *Mondrian* [5] a multidimensional local recoding generalization algorithm for k-anonymity. The code we used is from [6], implemented in java. Experimental results suggest that *aggrAnon* preserves greater data utility than *Mondrian*, when considering attackers whose knowledge is limited to an aggregate function of the attribute values. The aggregate function that we consider in our experiments is *sum()*. Since the implementations are different we do not compare the two algorithms in terms of execution time. However, we note that our algorithm was on average faster by an order of magnitude in all the experiments we performed.

Data. We use the IPUMS Census dataset from UCI data mining repository. We selected 7 numerical attributes which refer to different types of income.

Dataset	Records	Attributes	Domain size
ipums	233,584	7	1010000

TABLE IV
DATASET PROPERTIES

Parameters. We study the behavior of our algorithm with respect to various parameters: a) k parameter of anonymity, b) parameter d which quantifies the attacker's knowledge accuracy, and c) the dataset size $|D|$. In every experiment we vary one of these parameters keeping others fixed. The default setting of our parameters is k = 10, d = 0%, $|D|$ = 233,584.

Evaluation Metrics. We evaluate our method with respect to the execution time of our algorithm and the information loss of the produced anonymous data. The information loss metric that we use is NCP.

Information Loss. In Figure 1 we present the behavior of the algorithms in terms of information loss, which is caused by generalizations. The scale is logarithmic. As k increases, both algorithms increase sublinearly. However, *aggrAnon* (with d=0%) preserves significantly more utility than *Mondrian* for any k, as expected.

In the next graph of Figure 1, we vary the dataset size $|D|$. To perform this experiment we created two random samples of our dataset of sizes 100,000 and 50,000 records respectively. The latter was sampled from the former. We observe that information loss of *aggrAnon* decreases monotonically with $|D|$ and remarkably outperforms *Mondrian*.

As the percentage of the attacker's uncertainty (d) increases, the information loss falls dramatically, in the third graph. We could not compare to *Mondrian*, as there is no such relaxation parameter for this algorithm.

Execution Time. Figure 2 demonstrates the computational cost of our algorithm. Execution time is larger for small k values, but decreases super-linearly as k increases. This happens because the data is divided into fewer equivalence classes of greater size. Each EC is anonymized separately. Thus, fewer steps are made by the algorithm.

The scalability of our algorithm is shown in the second graph of figure 2. The curve grows linearly with the dataset size $|D|$. This happens because more data records form more equivalence classes of similar sizes, each examined independently of others.

Finally, in the last graph we observe that running time is almost insensitive to d parameter.

V. Related Work

L. Sweeney proposed *k-anonymity* guarantee to address this type of linking attacks [1]. A table is k-anonymous if each record is indistinguishable from at least $k - 1$ others with respect to the quasi identifier (QI) set [7], [1]. To achieve this, QI are transformed to form groups of records with identical QI values, called *equivalence classes*.

The objective of most anonymizing algorithms is to find an optimal recoding of the data that satisfies a given privacy guarantee and preserves as much data utility as possible. The latter is accomplished by minimizing a function which estimates the *information loss* that recoding introduces to the data. [2] proved that optimal k-anonymity for multidimensional QI is NP-hard, under both the generalization and suppression models. For the latter, they proposed an approximate algorithm that minimizes the number of suppressed values with the approximation bound $O(k \cdot logk)$. [8] improved this bound to $O(k)$, while [9] further reduced it to $O(logk)$.

Considering only global recoding can limit the search space [10], [11]. On the other hand, lower information loss can be achieved by considering multidimensional local recoding. Mondrian [5] partitions the space recursively across the dimension with the widest normalized range of values and supports a limited version of local recoding. [12] model the problem as clustering and propose a constant factor approximation of the optimal solution, but the bound only holds for the Euclidean distance metric. [3] propose agglomerative and divisive recursive clustering algorithms, which attempt to minimize the Normalized Certainty Penalty (NCP). [13] introduced the notion of k^m-anonymity for set-valued data. The authors limit the strength of the attacker by assuming she may know up to m items of a target. Moreover, attributes are not separated into quasi-identifiers and sensitive. Any item known to the attacker can act as a quasi-identifier, while if it is unknown it is considered sensitive.

VI. Conclusions

In this paper we studied the problem of anonymizing data in the presence of aggregate knowledge. To address this attack we proposed a relaxation of k-anonymity, that we call k^f-anonymity. We provided a utility-preserving algorithm which greedily selects a solution that satisfies our guarantee.

To the best of our knowledge, this is the first work treating aggregate information as potential attacker knowledge. In the

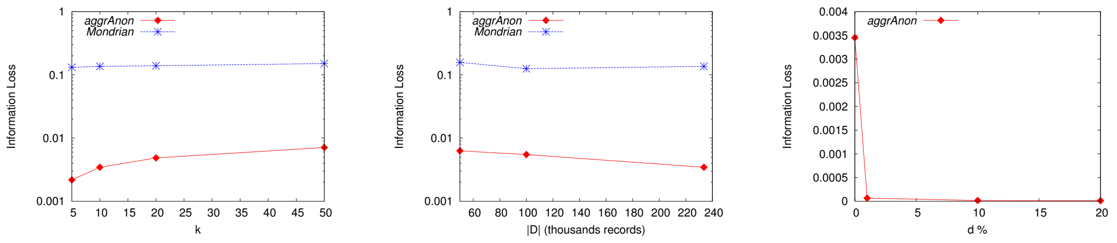

Fig. 1. Information Loss vs. k, $|D|$ and d.

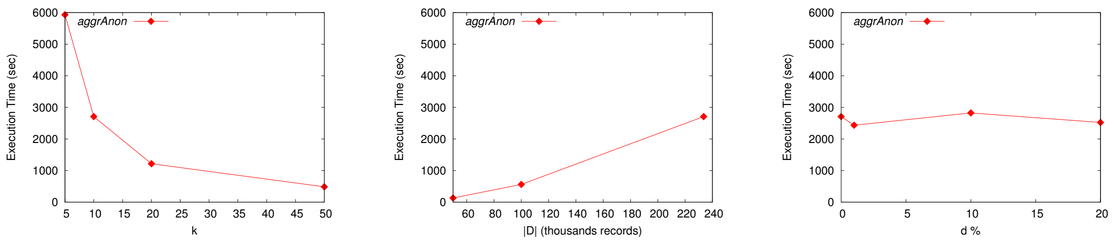

Fig. 2. Execution Time vs. k, $|D|$ and d.

future, we will extend our guarantee to provide a form of l-diversity. We also wish to examine more complicated functions as potential background knowledge and also attack scenarios where the attacker has knowledge of multiple aggregate values for a record. Finally, we will examine the scenario that an attacker has partial knowledge of a record, i.e., some attribute values, additionally to her aggregate-knowledge.

Acknowledgment

This research has been co-financed by the European Union (European Social Fund - ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: Heracleitus II. Investing in knowledge society through the European Social Fund.

References

[1] L. Sweeney, "k-Anonymity: A Model for Protecting Privacy," *IJUFKS*, vol. 10, no. 5, 2002.
[2] A. Meyerson and R. Williams, "On the Complexity of Optimal K-anonymity," in *PODS*, 2004, pp. 223–228.
[3] J. Xu, W. Wang, J. Pei, X. Wang, B. Shi, and A. Fu, "Utility-Based Anonymization Using Local Recoding," in *KDD*, 2006, pp. 785–790.
[4] "Uci repository," http://archive.ics.uci.edu/ml/datasets.html.
[5] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Mondrian Multidimensional k-Anonymity," in *ICDE*, 2006.
[6] "Utd anonymization toolbox," http://cs.utdallas.edu/dspl/cgi-bin/toolbox/.
[7] P. Samarati and L. Sweeney, "Generalizing Data to Provide Anonymity when Disclosing Information (abstract)," in *PODS (see also Technical Report SRI-CSL-98-04)*, 1998.
[8] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, and A. Zhu, "Approximation Algorithms for k-Anonymity," *Journal of Privacy Technology*, 2005.
[9] H. Park and K. Shim, "Approximate algorithms for k-anonymity," in *SIGMOD*, 2007, pp. 67–78.
[10] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan, "Incognito: Efficient Full-domain k-Anonymity," in *SIGMOD*, 2005, pp. 49–60.
[11] R. J. Bayardo and R. Agrawal, "Data Privacy through Optimal k-Anonymization," in *ICDE*, 2005, pp. 217–228.
[12] G. Aggarwal, T. Feder, K. Kenthapadi, S. Khuller, R. Panigrahy, D. Thomas, and A. Zhu, "Achieving anonymity via clustering," in *PODS*. New York, NY, USA: ACM, 2006, pp. 153–162.
[13] M. Terrovitis, N. Mamoulis, and P. Kalnis, "Privacy-preserving Anonymization of Set-valued Data," *PVLDB*, vol. 1, no. 1, 2008.