Steered Microaggregation as a Unified Primitive to Anonymize Data Sets and Data Streams

Josep Domingo-Ferrer, *Fellow, IEEE,* Jordi Soria-Comas and Rafael Mulero-Vellido

***Abstract*—As data grow in quantity and complexity, data anonymization is becoming increasingly challenging. On the one side, a great diversity of masking methods, synthetic data generation methods and privacy models exists, and this diversity is often perceived as unsettling by practitioners. On the other side, most of the anonymization methodology was designed for static, structured and small data, whereas the current landscape includes big data and, in particular, data streams. We explore here a unified and conceptually simple anonymization approach, by presenting a primitive called steered microaggregation that can be tailored to enforce various privacy models on static data sets and also on data streams. Steered microaggregation is based on adding artificial attributes that are properly initialized and weighted in order to guide the microaggregation process into meeting certain desired constraints. To demonstrate the potential of this type of microaggregation, we show how it can be used to achieve k-anonymity, t-closeness, l-diversity and ϵ-differential privacy in the context of static data sets; furthermore, we discuss how it can be used to achieve k-anonymity of data streams while controlling tuple reordering. Beyond its flexibility and theoretical appeal, steered microaggregation can drastically reduce information loss, as shown by our experimental evaluation.**

***Index Terms*—Privacy, anonymization, statistical disclosure control, microaggregation, k-anonymity, t-closeness, l-diversity, ϵ-differential privacy, data streams.**

I. Introduction

BIG data have become a reality with the new millennium. Almost any human activity leaves a digital trail that someone collects and stores: internet-of-things sensors, social media, machine-to-machine communication, streaming video, etc. As a result, data from many independent sources are available, which can be fused and analyzed to extract content. Big data differ from traditional data sets in at least three ways: volume, variety —data from several sources in several formats— and velocity —not only static but also dynamically harvested data, often as continuous streams.

Even though big data are extremely valuable to advance research and hence the welfare of humankind in many fields, they have also an important downside: they invade more and more the privacy of the subjects on whom they are collected (take the famous teenager pregnancy guess reported in [16]). The fraction of (big) data made available to the general public is called open data. The motivation to exchange data is to support collaboration between two or more organizations, like joint data mining (*e.g.* see [9] for an open data apology). However, open data that are not properly anonymized are very likely to lead to reidentification [20].

The authors are with Universitat Rovira i Virgili, Dept. of Computer Engineering and Mathematics, UNESCO Chair in Data Privacy, CYBERCAT-Center for Cybersecurity Research of Catalonia, Av. Països Catalans 26, 43007 Tarragona, Catalonia. E-mail {josep.domingo, jordi.soria, rafael.mulero}@urv.cat

Anonymization is usually carried out by modifying the original data using a method for statistical disclosure control (SDC, [24]). The traditional approach to anonymization in official statistics was (and still largely is) to first try an SDC method with parameters that degrade utility only mildly, and then measure the disclosure risk on the anonymized data [42], [25], [19]. If necessary, re-anonymize with more aggressive parameters until the risk is low enough. In computer science, a more formal approach was introduced based on a *privacy model*, that is, a privacy condition dependent on a parameter that guarantees an upper bound on the risk of reidentification disclosure and perhaps also on the risk of attribute disclosure by an intruder. The model can be enforced using one or several SDC methods whose parameters derive from the model parameters. Examples of privacy models include k-anonymity [35] and its extensions (l-diversity [29], t-closeness [26], (n,t)-closeness [27], crowd-blending privacy [21], β-likeness [6] and others), as well as ϵ-differential privacy [17].

A. Contribution and plan of this paper

As mentioned above, anonymization has become quite a challenging endeavor in the last decades, both due to the appearance of new types of data (in particular, dynamic vs static data) and due to the multiplicity of privacy models. In this paper, we seek to offer a primitive, called steered microaggregation, that is general enough to be able to satisfy several privacy models on static data sets and also on data streams. As its name suggests, our primitive is based on microaggregation, which is an SDC family of methods based on partitioning the data set into groups of k or more records and replacing records in each group by the group centroid; microaggregation was shown to yield k-anonymity if used on quasi-identifier (QI) attributes [15].

The goal in steered microaggregation is to define a general control mechanism to guide microaggregation algorithms. That is, rather than just generating clusters of a given cardinality that are as homogeneous as possible, steered microaggregation allows taking into account additional constraints. This is achieved by introducing artificial attributes designed in specific ways.

After giving some background in Section II, we present steered microaggregation in Section III. Then, we first focus on a static data set, which is the scenario that most privacy models implicitly assume. As the first application of steered microaggregation, in Section IV we show how it can be

used to achieve t-closeness, one of the privacy models in the k-anonymity family offering strictest privacy guarantees. In Section V, we sketch how the algorithm for t-closeness can be modified to yield l-diversity. Using the relation between t-closeness and ϵ-differential privacy, in Section VI we describe how the latter can be attained by steered microaggregation in certain cases. We then move to a usual big data scenario: data that change dynamically, either by periodic updates or continuously. Anonymizing incremental updates with notions inspired by k-anonymity has been addressed by [31], [45], [4], [33], [38], [43], [37], [13]. Nevertheless, scaling these approaches to *continuously* updated data is not obvious. Anonymization in continuous data publishing was treated by [5], who proposed methods to achieve k-anonymity, l-diversity and t-closeness for data streams. As another application of steered microaggregation, in Section VII we show how it can be used to satisfy k-anonymity in data streams, with the novel feature of keeping under control the amount of tuple reordering. While the algorithms in [5] are *ad hoc* and relatively complex, we will show that stream anonymization based on steered microaggregation can be quite compact and simple. Experimental work evaluating the performance of the proposed algorithms in terms of the utility-risk trade-off is presented in Section VIII. Finally, conclusions and future research lines are gathered in Section IX. The proofs of some results are given in the Appendix. A preliminary and shorter version of this work appeared in the conference paper [14].

II. Background

A. k-Anonymity, l-diversity and t-closeness

Assume a data set D from which direct identifiers have been suppressed, but that contains QI attributes and sensitive attributes; examples of the latter are health condition, salary, etc. The intruder's goal is to determine the identity of the individuals to whom the values of sensitive attributes in records of D correspond (*identity disclosure*). A natural attack strategy is for the intruder to use the QI attributes to link records in D with records in some external database containing direct identifiers [24].

Definition 1. *A data set D is said to satisfy k-anonymity ([35]) if each combination of values of the QI attributes in it is shared by at least k records.*

We will use *equivalence class* to refer to a maximal set of records that are indistinguishable with respect to the QIs. k-Anonymity protects against identity disclosure: given an anonymized record in D, an intruder cannot determine the identity of the individual to whom the record (and hence the sensitive attribute values in it) corresponds. The reason is that there are at least k records in D sharing any combination of QI attribute values.

k-Anonymity is attained by modifying the values of the QI attributes, usually either by a combination of generalization and suppression or by using microaggregation.

Unfortunately, k-anonymity does not protect against *attribute disclosure*: if the values of a sensitive attribute in all the records of an equivalence class are the same, then the intruder learns the value of that sensitive attribute for the target individual even if he cannot determine the individual's record within the class. l-Diversity [29] is an extension of k-anonymity that tries to fix this weakness.

Definition 2. *An equivalence class is said to satisfy l-diversity if there are at least l "well-represented" values for the sensitive attribute. A data set is said to satisfy l-diversity if all its equivalence classes satisfy l-diversity.*

Several notions of "well-representedness" have been proposed. In this work, we focus on *distinct l-diversity*, which requires equivalence classes to have at least l different values for the sensitive attribute.

t-Closeness [26] is another extension of k-anonymity which also tries to solve the attribute disclosure problem.

Definition 3. *An equivalence class is said to satisfy t-closeness if the distance between the distribution of the sensitive attributes in this class and the distribution of the sensitive attributes in the whole data set is no more than a threshold t. A data set is said to satisfy t-closeness if all its equivalence classes satisfy t-closeness.*

The specific distance used between distributions is central to evaluate t-closeness, but the original definition does not advocate any specific distance. The Earth Mover's distance (EMD) [34] is the most common choice (and the one we will adopt in this work), although other distances have also been explored [32], [12]. $EMD(P, Q)$ measures the cost of transforming one distribution P into another distribution Q by moving probability mass. EMD is computed as the minimum transportation cost from the bins of P to the bins of Q, so it depends on how much mass is moved and how far it is moved. For numerical attributes, the distance between two bins is based on the number of bins between them.

B. ϵ-Differential privacy

Differential privacy (DP) was proposed by [17] as a relative privacy model that limits the knowledge gain between data sets that differ in one individual.

Definition 4. *A randomized function κ gives ϵ-DP if, for all data sets D_1, D_2 such that one can be obtained from the other by modifying a single record, and all $S \subset Range(\kappa)$, it holds that* $\Pr(\kappa(D_1) \in S) \leq \exp(\epsilon) \times \Pr(\kappa(D_2) \in S)$.

Although initially the focus was on the interactive setting, DP has also been extended for microdata releases in [3], [18], [22], [8], [40], [36].

C. Microaggregation

Microaggregation is a family of perturbative methods for SDC of microdata releases. Multi-dimensional microaggregation was proposed and formalized in [10] and its application to k-anonymity was described in [15]. It consists of: i) partitioning the data into clusters of at least k records so that records in each cluster are as similar as possible; and ii) replacing records in each cluster by the centroid record. For numerical data, one can use the mean record as centroid;

for categorical data, one can resort to the median or some other average operator defined in terms of an ontology (*e.g.* see [11]).

If microaggregation on the QI projection is used to attain k-anonymity, instead of replacing QIs by the QI cluster centroid one can present QI values in several alternative ways: i) generalize the cluster QI values to a common value; ii) decouple QIs and sensitive attributes by permuting the sensitive attribute values within each cluster (as proposed in [39]), or iii) use the cluster partition separately on QIs and sensitive attributes, and link QI clusters and sensitive attribute clusters via a cluster identifier (this is the Anatomy method [44]).

If the records within each cluster are maximally homogeneous, the information loss is minimized because replacing very homogeneous records by their centroid entails almost no information loss; therefore utility is maximized. A common information loss metric is the SSE (sum of squared errors). Finding an optimal partition (that maximizes within-cluster similarity) in multi-dimensional microaggregation is an NP-hard problem [30]; therefore, heuristics are employed to obtain an approximation with reasonable cost. An example heuristic for the partition step of microaggregation is MDAV [15].

D. c-Means clustering

Microaggregation can be viewed as form of clustering where the constraints are on the minimum cardinality of clusters rather than on their number. Besides microaggregation, in this work we will use standard clustering that sets in advance a number c of clusters [2]. In particular, we will employ the online Lloyd's algorithm [23] for c-means clustering[1] specified in Algorithm 1, that iteratively refines an initial set of clusters by computing the distance between each record and each cluster (usually represented by its centroid) and reassigning each record to the closest cluster.

Algorithm 1. *[Online Lloyd's algorithm]*
Let D *be a data set*
Let $Clusters = \{C_1, \ldots, C_c\}$ *be an initial clustering of* D
Repeat
  For each $r \in D$
    Compute the distance between r *and the clusters in* $Clusters$
    Assign r *to its closest cluster*
  End for
until $Clusters$ *does not change*

III. Steered microaggregation

We define *steered microaggregation* as a form of microaggregation in which the clustering process is steered (guided) by introducing one or several artificial attributes, plus one or several microaggregation parameters. A more formal definition follows.

[1]This algorithm is known as k-means, but we call it here c-means to avoid confusion with the minimum cluster size k of k-anonymity and microaggregation.

Definition 5 (Steered microaggregation)**.** *A steered microaggregation algorithm is an algorithm* $SM(D, k, M, \mathcal{A}, \Omega)$*, where* D *is an original data set,* k *is the minimum cluster size,* M *is a microaggregation algorithm,* $\mathcal{A}$ *is a set of artificial attributes each with its corresponding values for all records in* D*, and* Ω *is a set of microaggregation parameters.*

Then, SM *operates as follows:*

1) *Augment the data set* D *by adding to it the attributes in* $\mathcal{A}$ *(thereby increasing the dimensionality of* D*). Call the augmented data set* D'*. In this way, each record in* D' *has now one additional value per additional attribute.*
2) *Run the microaggregation algorithm* M *on* D' *using the microaggregation parameters in* Ω*.*
3) *Return the microaggregated data set after removing from it the attributes in* $\mathcal{A}$*.*

The advantage of steered microaggregation over regular microaggregation is that the former can go beyond the usual goal of maximizing the within-cluster similarity of records; specifically, the steering process is meant to satisfy additional constraints (like l-diversity, t-closeness, etc.). The nature of such additional constraints determines the way of choosing the artificial attributes in $\mathcal{A}$ and the microaggregation parameters in Ω.

IV. t-Closeness via steered microaggregation

k-Anonymity can be attained via microaggregation of the QI attributes. Since k-anonymity only protects against reidentification disclosure, extensions to k-anonymity (such as t-closeness) are needed to achieve protection against attribute disclosure. However, running regular microaggregation algorithms on QIs cannot ensure the level of variability of the sensitive attribute required by such extensions, because attribute disclosure depends on the sensitive attribute.

The problem is thus how to take into account the sensitive attribute in the microaggregation process. Performing *ad hoc* modifications of specific microaggregation heuristics to satisfy each specific k-anonymity extension is certainly an option. Steered microaggregation offers a more general alternative by steering regular microaggregation algorithms into enforcing the specific constraints of the chosen k-anonymity extension. We focus on t-closeness in this section and we sketch the way to attain l-diversity in Section V.

t-Closeness requires the distribution of the sensitive attribute in each k-anonymous class to be similar to the distribution of the sensitive attribute in the overall data set. Thus, to attain t-closeness, we must make sure that the sensitive attribute values of the records in each microaggregation cluster are sufficiently spread across the values in the overall data set. This can be ensured by: (i) splitting the n records of the data set into b equal-sized buckets $B_1, \ldots, B_b$, where B_1 contains the n/b smallest sensitive attribute values, B_2 contains the following n/b values, and so on; and (ii) steering the microaggregation algorithm into forming clusters such that each cluster contains one record from each of the previous buckets.

The value chosen for parameter b (number of buckets) determines the level of t-closeness that can be reached. In particular, as t decreases (stronger protection), b must increase

(the sensitive attribute in microaggregation clusters must be more spread). For a given level of t-closeness, we need to determine the required value for b. If the sensitive attribute Z has *no repeated values* (as it is plausible if it is continuous numerical), Proposition 2 in [41] states that $b \geq n/(2(n-1)t+1)$. In case of Z having *repeated values* (as it is likely if it has a finite number of possible values, such as in categorical or ordinal attributes), we have the following lower bound for b, proven in the Appendix.

Proposition 1. *Let D be a data set with n records and let the sensitive attribute Z of D have a finite number of possible values. Then to achieve t-closeness the number of buckets must satisfy $b \geq n/(nt+1)$.*

If we want t-closeness on top of k-anonymity, then the lower bounds on b are:

$$b \geq \begin{cases} \max\left(k, \left\lceil \frac{n}{2(n-1)t+1} \right\rceil\right) & \text{if } Z \text{ has no repetitions;} \\ \max\left(k, \left\lceil \frac{n}{nt+1} \right\rceil\right) & \text{if } Z \text{ may have repetitions.} \end{cases} \quad (1)$$

A first option is to steer the microaggregation by introducing an artificial attribute that incorporates the previous information about the conditions needed to satisfy t-closeness by focusing on a set of target clusters. See Algorithm 2, where we assume for simplicity that all QIs are numerical. Note 1 further below discusses how to deal with the non-numerical case.

Algorithm 2. *[t-closeness via steered microaggregation]*

Let $Q_1, \ldots, Q_m$ *be the QI attributes and Z be the sensitive attribute of D.*
Let k *be the expected level of k-anonymity, t the expected level of t-closeness, n the number of records in D, and ω_z the weight that signals the importance of t-closeness.*

1) *Normalize $Q_1, \ldots, Q_m$ for their range to be in $[0, 1]$. Let $\hat{Q}_1, \ldots, \hat{Q}_m$ be the normalized attributes.*
2) **Set** *b to meet the suitable lower bound in Expr. (1).*
3) *Define buckets $B_1, \ldots, B_b$ clustering the values of Z, in such a way that all buckets contain the values of the same (or a similar) amount of records and the values within each bucket are maximally homogeneous. Without loss of generality, let B_1 be the bucket with the largest number of records having values of Z in it, let B_2 be the bucket with the second largest number of records, and so on.*
4) *Add a bucket attribute $B(Z)$ and initialize its value to 0 for all n records.*
5) **For** $i = 1$ **to** $|B_1|$ **do**
   a) *Take as the i-th record one of the records such that Z falls in B_1 and $B(Z) = 0$.*
   b) **Set** $B(Z) := i$ *for that record.*
6) **For** $j = 2$ **to** b **do**
   a) **Let** $\mathbf{V}$ *be a vector with $|B_{j-1}|$ binary positions all initialized to 0.*
   b) **For** $i = 1$ **to** $|B_j|$ **do**
      i) *Take as the i-th record one of the records such that Z falls in B_j and $B(Z) = 0$.*
      ii) *Look for the record i' in B_{j-1} with $\mathbf{V}[i'] = 0$ and with closest values of normalized attributes $\hat{Q}_1, \ldots, \hat{Q}_m$ to the ones of the i-th record in B_j.*
      iii) **Set** $\mathbf{V}[i'] = 1$.
      iv) **Set** *$B(Z)$ for the i-th record in B_j to the value of $B(Z)$ for the i'-th record in B_{j-1}.*
7) *Normalize $B(Z)$ for its range to be in $[0, \omega_z]$. Let $\hat{B}(Z)$ be the normalized bucket attribute.*
8) *Microaggregate (*e.g. *using MDAV [15]) with group size b the projection on attributes $\hat{Q}_1, \ldots, \hat{Q}_m, \hat{B}(Z)$. Call $Q'_1, \ldots, Q'_m, B(Z)'$ the microaggregated attributes.*
9) *De-normalize $Q'_1, \ldots, Q'_m$ in the records returned by the microaggregation heuristic and output the records (where, for each record, the values of $Q'_1, \ldots, Q'_m, Z$ are output).*

Note 1 (On non-numerical attributes)**.** If some of the QI attributes are not numerical, then Algorithm 2 must be modified, because non-numerical attributes cannot be normalized. The normalization and de-normalization operations in Steps 1, 7 and 9 must be suppressed. Rather than enforcing attribute weights via normalization, weights must be enforced by the distance used in the microaggregation. For instance, for nominal categorical QIs we can only compare attribute values for equality. In this case, we can use the Hamming distance and weight it appropriately (according to the number of QIs) to make sure that it is between 0 and 1. In the case of hierarchical categorical QIs, we can use a weighted variant of the S-distance proposed in [11].

Algorithm 2 seeks homogeneity for the values of $Q_1, \ldots, Q_m, B(Z)$. Attribute $B(Z)$ is an artificial attribute, with weight ω_z, that introduces an additional constraint in the microaggregation process. $B(Z)$ is initialized in such a way that it sequentially numbers the records whose sensitive attribute value falls in each bucket: if there are $|B_1|$ records with values of Z in B_1, then $B(Z)$ for those records takes values $1, 2, \ldots, |B_1|$; if there are $|B_2|$ records with values of Z in B_2, then $B(Z)$ for those records takes values $1, 2, \ldots, |B_2|$; and so on. Hence, if the weight ω_z attached to $B(Z)$ is large enough, clusters containing records with the same value for $B(Z)$ will be favored, which means clusters such that the values of Z in the records of each cluster all fall in different buckets. Spreading Z over all buckets within each cluster is a way to obtain a within-cluster distribution of Z that resembles the distribution of Z in the overall dataset. If ω_z is large enough for the value of t, this should yield t-closeness.

The following proposition (proven in the Appendix) quantifies the weight ω_z needed to make sure that all values of Z in a k-anonymous class fall in different buckets. This is a sufficient condition for t-closeness.

Proposition 2. *If $|B_1|$ is the size of the largest bucket and the Euclidean distance is used to calculate the distance between records and $\omega_z > |B_1|\sqrt{m}$, then Algorithm 2 ensures as much as cluster cardinality constraints allow that, in any microaggregation cluster, the values of Z for all records fall in different buckets.*

By taking the weight ω_z as in the previous proposition, we strive to strictly satisfy t-closeness. However, by using a lower ω_z we can trade off t-closeness for utility preservation, that is, look for clusters whose QI values are more homogeneous.

When Algorithm 2 is used to strictly satisfy t-closeness, the way the artificial attribute is initialized steers the microaggregation to a set of target clusters that satisfy t-closeness but do not attempt to maximize utility (because they do not take into account the within-cluster homogeneity of QIs). We present next an alternative that avoids the above shortcoming by trying to maximize within-cluster homogeneity given a target t-closeness level.

We will use Algorithm 1 for clustering, which iteratively modifies an original set of clusters to reduce the clustering cost until convergence to a (possibly local) minimum of the cost function is reached. Usually, the cost function is set to be the SSE of the clustering and the clusters are refined by assigning each record to the cluster with the closest centroid.

We need to alter the parameters of Lloyd's algorithm to make sure that t-closeness is satisfied. Our cost function measures the information loss on the QI attributes, that is, we compute SSE on the projection of records on the QI attributes. However, the distance function in Algorithm 1 is the one that determines how the clusters are modified. Therefore, we should define this distance appropriately to enforce t-closeness. We proceed as follows:

- Add an artificial attribute A to the data set D whose value is set to the rank of each record with respect to the sensitive attribute. Let D' be the result of augmenting D with A.
- Define an initial set of clusters that satisfy t-closeness.
- Extend the distance measure to make sure that moving a record between clusters does not violate the t-closeness requirement.

We can choose an arbitrary set of initial clusters, *as long as they satisfy t-closeness*. Following the procedure described in [41] (and used in Algorithm 2), we choose a cluster size appropriate to the desired level of t-closeness and select the records in each cluster to be as spread as possible with respect to the sensitive attribute. It might seem that, to minimize the clustering cost, one should create as many clusters as possible, which would require taking the cluster size as close as possible to the lower bound defined by Equation (1). However, doing so leads to sub-optimal results. The main reason is that, when the clusters are small, we do not have enough freedom to select records that are close to each other in terms of the QI attributes. One might still hope that, if larger clusters are preferable, the clustering algorithm will eventually converge to them. However, this does not happen because increasing the size of a cluster requires decreasing the size of another cluster (note that Lloyd's algorithm does not change the number of clusters), and making a cluster smaller is hard to reconcile with the cluster continuing to satisfy t-closeness. In conclusion, it is not possible to determine the optimal number of initial clusters analytically; it has to be done empirically.

To make sure that t-closeness is satisfied in every iteration of the clustering algorithm, we need to avoid adding/removing a record to/from a cluster if doing so violates t-closeness. This is attained by defining the extended distance $d_{D'}(r, C)$ for clustering the augmented data set D' as:

$$\begin{cases} 0 & \text{if } r \in C \text{ and } EMD(C \setminus \{r\}, \{1, \ldots, n\}) > t; \\ \infty & \text{if } r \notin C \text{ and } EMD(C \cup \{r\}, \{1, \ldots, n\}) > t; \\ d_D(r, C) & \text{otherwise.} \end{cases} \tag{2}$$

In the above distance definition, EMD parameters actually are the empirical distributions of the respective sets. The first and second cases prevent, respectively, removing and adding a record if that violates t-closeness. The remaining case uses the distance $d_D(r, C)$ between r and C based on the projection of records on the QI attributes.

If we want to ensure k-anonymity in addition to t-closeness, we need the initial set of clusters to satisfy both properties and then we can use the following extended distance $d_{D'}(r, C)$ instead of the one in Expression (2):

$$\begin{cases} 0 & \text{if } r \in C \text{ and } (EMD(C \setminus \{r\}, \{1, \ldots, n\}) > t \\ & \quad \text{or } |C \setminus \{r\}| < k); \\ \infty & \text{if } r \notin C \text{ and } EMD(C \cup \{r\}, \{1, \ldots, n\}) > t; \\ d_D(r, C) & \text{otherwise.} \end{cases} \tag{3}$$

Note 2 (On the use of extended distances)**.** For the extended distance in Expression (2) to ensure t-closeness and the extended distance in Expression (3) to ensure t-closeness and k-anonymity, the initial set of clusters must satisfy those properties. Assuming that they do, the extended distances make sure the clusters obtained in successive iterations continue to satisfy the properties. However, if the initial clusters did not satisfy the desired properties, the extended distances would be unable to lead to clusters satisfying them.

Algorithm 3 formalizes the previously described steps. Depending on whether the extended distance in Expression (2) or (3) is used, the algorithm can be termed just steered c-means (because it does not guarantee k-anonymity) or steered microaggregation, respectively. In the algorithm, we only describe the generation of the set of clusters that satisfy t-closeness. QIs can then be aggregated, generalized, decoupled or anatomized as mentioned in Section II-C.

Algorithm 3. *[t-closeness via steered c-means/microaggregation]*
Let $Q_1, \ldots, Q_m$ *be the QI attributes and* Z *be the sensitive attribute of* D.
Let t *be the expected level of* t*-closeness and* n *the number of records in* D.

1) *Obtain an augmented data set D' by adding to D an artificial attribute A whose value for each record is the record's rank with respect to the sensitive attribute.*
2) *Select c to satisfy the suitable lower bound in Expression (1).*
3) *Create the initial clusters $\{C_1, \ldots, C_c\}$ as $C_i = \{r_j : A[r_j] \mod c + 1 = i\}$ for $i = 1, \ldots c$.*
4) **If** *one wants just t-closeness* **then let** $d_{D'}$ *be the extension of the distance to D' defined in Expression*

(2) **else if** *one wants t-closeness plus k-anonymity* **then let** $d_{D'}$ *be as in Expression (3).*

5) **Call** *Algorithm 1 on D' with the initial clusters defined in Step 3 and the distance defined in Step 4.*
6) **Return** *the generated clustering.*

V. Steered microaggregation and l-diversity

l-Diversity [29] is an extension of k-anonymity that can be regarded as a predecessor of t-closeness. Rather than requiring the distribution of the sensitive attribute value within each k-anonymous class to mimic the distribution of the sensitive attribute in the entire data set, l-diversity merely requires a certain variability l of the sensitive attribute values within each k-anonymous class. It is not surprising, therefore, that Algorithms 2 and 3 can be easily adapted to satisfy l-diversity.

To adapt Algorithm 2 to yield k-anonymity and l-diversity, the artificial attribute $B(Z)$ must be initialized as follows. Take all records from $i = 1$ to n; if the value of Z in the i-th record has appeared j times in previous records, set $B(Z) = j + 1$ for the i-th record.

With the previous initialization of $B(Z)$, two records with the same value for Z will have different values for $B(Z)$. If the weight ω_z is large enough, two records with different values of $B(Z)$ will not be microaggregated into the same cluster. Rather, clusters will tend to contain records with the same value of $B(Z)$, which means records with the j-th repetitions of *different values of* Z, for some j.

While simple, the previous adaptation of Algorithm 2 to l-diversity may be regarded as a tad too coarse: for clusters to be diverse, one does not really need to cluster together records with *precisely* the j-th repetitions of different values of Z; it would be enough to cluster records with whatever repetitions of different values of Z. Hence, it may be more interesting to try to adapt Algorithm 3 for l-diversity. This can be done with the following steps:

1) Split the records in the data set into as many buckets as there are different values for the sensitive attribute. The i-th bucket B_i must contain all the records where the sensitive attribute takes the i-th different value. Define an artificial attribute A whose value is set to i for all the records in bucket B_i.
2) Generate an initial set of clusters such that each of the clusters contains at least l different values of A.
3) Use a suitable extended distance that ensures l-diversity (and also k-anonymity if one views l-diversity as an extension of k-anonymity). See details below.

Regarding the distance measure, analogously to Expression (2) we can define an extended distance $d_{D'}(r, C)$ to make sure that moving a record between clusters does not violate the l-diversity requirement, as follows:

$$\begin{cases} 0 & \text{if } r \in C \text{ and } |\{A[x] : x \in C \setminus \{r\}\}| < l; \\ d_D(r, C) & \text{otherwise.} \end{cases}$$

The first case in the distance definition prevents removing a record from a cluster if that results in the cluster containing less than l different values of A. The alternative case resorts to the distance between QIs of records in D.

If we want the clusters to be k-anonymous in addition to l-diverse, the extended distance $d_{D'}(r, C)$ should be:

$$\begin{cases} 0 & \text{if } r \in C \text{ and } (|\{A[x] : x \in C \setminus \{r\}\}| < l \\ & \quad \text{or } |C \setminus \{r\}| < k); \\ d_D(r, C) & \text{otherwise.} \end{cases}$$

Whatever the extended distance used, analogous remarks to those made in Note 2 also apply here: the modified Algorithm 3 will only yield l-diversity, respectively l-diversity and k-anonymity, if the initial clusters satisfy those properties.

VI. Steered microaggregation and differential privacy

Consider the following distance between two distributions.

Definition 6. *Given two random distributions F_1 and F_2 taking values in a discrete set $\{x_1, x_2, \cdots, x_N\}$, we define the distance between them as* $d(F_1, F_2) = \max_{i=1,2,\cdots,N} \{\Pr_{F_1}(x_i)/\Pr_{F_2}(x_i), \Pr_{F_2}(x_i)/\Pr_{F_1}(x_i)\}$.

In [12], it is proven that, if an anonymized static data set satisfies $\exp(\epsilon/2)$-closeness when the distance between distributions mentioned in Definition 3 is the one in Definition 6, then the data set satisfies DP in the sense stated in the following proposition.

Proposition 3. *Let $k_I(D)$ be the function that returns the view on subject I's sensitive attributes given a data set D. If D satisfies $\exp(\epsilon/2)$-closeness when using the distribution distance in Definition 6, then $k_I(D)$ satisfies ϵ-DP. In other words, if we restrict the domain of k_I to $\exp(\epsilon/2)$-close data sets, then we have ϵ-DP for k_I.*

Thus, given a protected data set output by Algorithm 2, one needs to find the minimum ϵ for which the data set satisfies $\exp(\epsilon/2)$-closeness with the distance of Definition 6. Then one obtains ϵ-DP as per Proposition 3.

VII. k-Anonymity for data streams via steered microaggregation

Although the bulk of the research in SDC refers to static data sets, dynamic data are gaining relevance in the big data context, as mentioned in the introduction. In this section we focus on data streams, a particular kind of dynamic data. Attributes relevant to anonymization are identifiers, QIs and sensitive attributes. Without loss of generality, we consider a single sensitive attribute (it can be the Cartesian product of several sensitive attributes). Thus:

- Let $S = \{s_i\}_{i \geq 0}$ be an original stream, that is, a sequence of continuously incoming tuples $s_i = (id, q_1, \ldots, q_m, z)$, where $s_i.id$ is the identity of the subject to whom s_i corresponds; $s_i.q_1$, …, $s_i.q_m$ are the values for QI attributes $Q_1, \ldots, Q_m$ and $s_i.z$ is the value of the sensitive attribute Z.
- Let $S' = \{s'_j\}_{j \geq 0}$ be the anonymized version of stream S, that is, a sequence of continuously output tuples $s'_j = (q_1, \ldots, q_m, z)$, where the subject's identity has been pruned, and $s'_j.q_1$, …, $s'_j.q_m$, $s'_j.z$ are suitably modified

versions of the QI values and the sensitive attribute value, respectively, in the original tuple corresponding to s'_j.

Anonymizing a stream may involve some delay and reordering of the tuples in the original stream. Such delay and reordering of the tuples is one dimension of the information loss inflicted by anonymization, the other dimension being related to how much the QI attribute values of the tuples are altered. We slightly generalize the delay constraint definition of [5].

Definition 7. *[Delay constraint] Let M be an anonymization mechanism that takes as input a data stream S and outputs an anonymized data stream S'. If δ is a positive integer, M is said to satisfy the delay constraint δ if and only if upon receiving any new tuple $s_i \in S$, M has already output in S' all anonymized tuples corresponding to tuples in S with position less than $i - \delta$.*

We next specify a formally refined version of the definition of data stream k-anonymity given in [5].

Definition 8. *[Stream k-anonymity] An anonymized data stream S' is a k-anonymized version of an original data stream S if for each tuple $s' \in S'$ there exists in S the corresponding original tuple s and, given any tuple $s' \in S'$, a group G of anonymized tuples exists such that: $s' \in G$, $|DS(G)| \geq k$ (where $DS(G)$ is the set of distinct subjects to whom tuples in G refer) and $\forall s'' \in G$, one has $s''.q_j = s'.q_j$ for $j \in [1, m]$. The group G is called the k-anonymous class of s'.*

In a static data set, using a regular microaggregation algorithm on the projection of the data set on the QIs leads to k-anonymity. However, regular microaggregation algorithms cannot be used in data streams, because they operate on the entire data set, whereas in a data stream all that is available at any moment is a certain number of incoming tuples. Hence, we require a microaggregation algorithm that is suitable for data streams.

Any microaggregation algorithm for data streams needs a set of tuples out of which the algorithm can generate clusters that are as homogeneous as possible. Therefore, the algorithm must buffer incoming tuples. The larger the buffer size, the more homogeneous the output clusters can be (and the less information loss is incurred), but the longer the delay (see Definition 7). Thus, the maximum affordable delay determines the buffer size.

Algorithm 4 keeps buffering tuples until there is a tuple that is about to expire (its delay is the maximum value δ). Since such a tuple must immediately be output, the algorithm generates a cluster around the expiring tuple by taking k tuples from the buffer, presents the QI values in the cluster in any of the ways described in Section II-C, and outputs the resulting cluster.

Algorithm 4. *[Stream microaggregation with maximum delay δ]*

Let *S be a data stream, δ the maximum delay allowed and k be the minimum cluster size.*
While $S \neq \emptyset$ **do***:*

1) *Read the next tuple s_i into the buffer $\mathcal{B}$.*
2) **If** *tuple $s_{i-\delta}$ is in $\mathcal{B}$* **then**
   If $|\mathcal{B}| \geq k$ **then**
   a) *Out of the tuples in $\mathcal{B}$, generate a cluster of k tuples including tuple $s_{i-\delta}$ that is maximally homogeneous in terms of the QI attributes.*
   b) *Remove the tuples in the generated cluster from $\mathcal{B}$.*
   c) *Present the QI values and sensitive attribute values in the tuples of the cluster in any of the ways described in Section II-C.*
   d) *Output the resulting cluster.*
   Else *Drop $s_{i-\delta}$.*

End while

Using a sliding window reduces the computational complexity of Algorithm 4 with respect to algorithms that deal with all records at once. Algorithm 4 has computational cost $\mathcal{O}(n\delta)$, where n is the total number of records of the data stream. In comparison, the cost of MDAV on a data set with n records is $\mathcal{O}(n^2)$. Thus a sliding window turns a quadratic cost into a linear cost in n.

While Algorithm 4 is simple and meets the delay constraint, it makes no attempt at preserving the order of any of tuples when outputting them: tuples are reordered as much as needed to satisfy maximum QI homogeneity within the delay constraint. This lack of regard for order preservation is a shortcoming shared by other stream k-anonymization proposals in the literature, such as the CASTLE algorithm [5].

However, preserving the order of tuples when anonymizing a data stream may also be regarded as a utility feature (the less reordering, the more utility). Steered microaggregation provides a way to integrate the order of tuples in the microaggregation process. This is done by adding an artificial attribute to the incoming tuples that contains the tuple's position number. As far as microaggregation is concerned, this artificial attribute is treated as an additional QI attribute. By attaching a suitable weight to the position attribute during the microaggregation process, we can trade off order preservation for homogeneity regarding the other QIs. Specifically, we propose Algorithm 5, where for simplicity we assume in the algorithm description that all QIs are numerical (see Section III for indications about steered microaggregation on non-numerical attributes):

Algorithm 5. *[Stream k-anonymity(S,δ,ω_p,k)]*
Let *S be a data stream with QIs $Q_1, \ldots, Q_m$ and sensitive attribute Z, δ be the maximum delay allowed, ω_p the weight attached to the position number and k the minimum cluster size.*

1) *Add a new attribute, P, containing the position of each tuple in S times ω_p, that is, $s_i.P = i \times \omega_p$.*
2) **Call** *Algorithm 4 taking $Q_1, \ldots, Q_m, P$ as QIs.*

To understand the rationale behind Algorithm 5 note that the partition step of microaggregation forms a cluster such that the within-cluster values of $Q_1, \ldots, Q_m, P$ are as homogeneous as possible, in order to lose as little information as possible when presenting the aggregated, generalized, decoupled or anatomized QIs within each cluster. Now, if the weight ω_p

is large, the distance between tuples used in the partition step basically amounts to the distance between the projection of the tuples on attribute P. Hence, to maximize within-cluster homogeneity, microaggregation will output clusters whose tuples have consecutive values of P, regardless of the values of attributes $Q_1, \ldots, Q_m$. Therefore, a large ω_p ensures that no tuple reordering occurs, although substantial QI damage can be incurred as a consequence of quite different values of $Q_1, \ldots, Q_m$ being averaged in the same cluster. On the other hand, if ω_p is small, then clusters are formed by microaggregation by prioritizing the homogeneity of within-cluster values of $Q_1, \ldots, Q_m$. Thus, substantial reordering and delays as large as δ can occur, because the position number is not taken into account when forming clusters: in fact, Algorithm 5 with very small ω_p is equivalent to Algorithm 4.

The following proposition quantifies the value of ω_p beyond which no reordering of the tuples takes place (see proof in the Appendix). The proposition assumes all attributes are numerical (see Note 3 below for the non-numerical case).

Proposition 4. *If attributes $Q_1, \ldots, Q_m$ are normalized to $[0, 1]$, attribute P is normalized to $[0, \omega_p]$, the Euclidean distance is used to calculate the distance between tuples and $\omega_p > (k-1)\sqrt{\frac{m}{2k-1}}$, then Algorithm 5 does not generate clusters containing tuples whose positions differ by more than $k-1$, which means that no reordering occurs.*

Note that the difference between the two most distant tuple positions in a cluster of k tuples is at least $k-1$. Specifically, $k-1$ is achieved when all tuples in the cluster are consecutive, which means there is no reordering.

Regarding computational complexity, Algorithm 5 amounts to Algorithm 4 plus the generation of the artificial attribute. Thus, the computational cost of Algorithm 5 is also $\mathcal{O}(n\delta)$.

Note 3. *[On non-numerical QIs]* If there are non-numerical QIs, the normalization assumed by Proposition 4 is not possible. However, in that case microaggregation weights $\omega_1 = \ldots = \omega_m = 1$ and ω_p can be enforced, respectively, for $Q_1, \ldots, Q_m, P$ as described in Note 1. Then a bound analogous to the one in Proposition 4 can be proven.

Note 4. *[Subjects with several tuples in the stream]* In Definition 8, stream k-anonymity requires that the tuples in each k-anonymous class refer to at least k different subjects. In Algorithm 5 and Proposition 4, we have implicitly assumed that all incoming tuples correspond to different subjects. However, in general, there may be several tuples referring to the same subject. To make sure k-anonymous classes refer to at least k subjects, we only need to modify Algorithm 4 so that, rather than generating a cluster of k tuples, it generates a cluster of $k' \geq k$ tuples such that they correspond to k different subjects. If such k' tuples cannot be found in $\mathcal{B}$, then $s_{i-\delta}$ has to be dropped.

VIII. Empirical analysis

We envision steered microaggregation as a general methodology to guide regular microaggregation or clustering algorithms into satisfying additional constraints. More specifically, we have proposed approaches to attain t-closeness, l-diversity and ϵ-DP for static data, as well as k-anonymity for data streams. Although steered microaggregation has simplicity as a key asset, we want it to be competitive in terms of information loss with existing methods.

A. t-Closeness evaluation

There is a variety of methods to attain t-closeness. The initial t-closeness algorithms were adaptations of previous k-anonymity algorithms that took into account the additional requirements of t-closeness (*e.g.* t-Incognito [26], t-Mondrian [27]). After those initial adaptations, several other algorithms were proposed (*e.g.* SABRE-KNN, SABRE-AK [7]). In this section, we compare Algorithms 2 and 3 against this previous literature.

1) Datasets: We use same data sets as the SABRE paper [7], that contain between $100,000$ and $500,000$ records extracted from U.S. Census data. They specify the following 8 attributes: Age (numerical, 79 values), Sex (categorical, 2 values), Education (categorical, 17 values), Marital Status (categorical, 6 values), Race (categorical, 9 values), Work Class (categorical, 10 values), Birth Place (categorical, 83 values), and Salary (numerical, 50 values). Salary is the sensitive attribute. All other attributes are QIs.

2) Risk and Utility Metrics: When enforcing a privacy model, the tolerable disclosure risk level is set in advance and does not need to be measured: it relates to k for k-anonymity, and to t (and usually also k) for t-closeness. However, since Algorithm 2 allows trading off strict enforcement of the privacy model for improved data utility, we need to measure to what extent the privacy model has been satisfied. Specifically, to assess how close we are to reaching t-closeness, we use the following measures:

- *Maximum EMD.* This is the maximum EMD distance for the sensitive attribute between the clusters output by the algorithm and the overall data set. It represents the actual level of t-closeness reached.
- *Average EMD*. This is the average EMD distance for the sensitive attribute between the clusters output by the algorithm and the overall data set. We use this measure to show how near we are, in global terms, to reaching the expected level of t-closeness.

To compare our methods with t-Incognito, t-Mondrian, SABRE-KNN and SABRE-AK, we use the same information loss metric that was used in [7]: the *Relative Error (RE) in count queries.* The use of count queries has become a common way of measuring utility in SDC methods. Following [7], we evaluate information loss based on count queries of the form: `SELECT COUNT(*) FROM Microdata WHERE` $\texttt{pred}(A_1)$ `AND ... AND` $\texttt{pred}(A_\lambda)$ `AND` $\texttt{pred}(SA)$, where each A_i is a QI attribute and SA is the sensitive attribute. Each query involves $\lambda + 1$ attributes (λ randomly chosen QIs plus one sensitive attribute) and each $pred(A_i)$ has the form $(A_i = v_1$ `OR` $A_i = v_2$ `OR ... OR` $A_i = v_{b_i})$, where v_i are different random values in the domain of attribute A_i. For numerical or ordinal categorical attributes, it would make more sense to take a range of contiguous

values (rather than random values selected across the whole attribute domain). The value of b_i is computed as $b_i = \lceil |A_i| \times s^{1/(\lambda+1)} \rceil$, where $s \in [0,1]$ is the query selectivity (that is, the proportion of the data domain involved in the query).

The proposed methods form clusters of records that satisfy the desired privacy conditions. Having generated such clusters, there are several ways to present the protected data set to the user, discussed in Section II-C. For our experiments, we choose Anatomy [45], which vertically splits the records in each cluster into two parts, the QI attributes and the sensitive attributes; then, for each cluster, the user gets all the values of the QI attributes on the one side, and all the values of the sensitive attributes on the other side, but not their correspondence. This presentation approach is more utility-preserving than the others, as the original attribute values are maintained.

Since Anatomy only preserves the relation between QIs and sensitive attributes at the cluster level, when given a count query and an anatomized cluster of records we are uncertain about the exact count of records that satisfy the query predicate. Thus, we need to estimate this count, which is done as described in Algorithm 6.

Algorithm 6. *[Count estimation in an anatomized data set]*
Let D *be an anatomized data set*
Let q *be a count query*
Let $Clusters$ *be the set of clusters*
Set $estimated_count = 0$
For each *record* $r \in D$ **do**
  If r *satisfies the QI predicates of* q **then**
    Let p *be the proportion of records in the cluster of* r *that satisfy the sensitive attribute predicate of* q
    $estimated_count = estimated_count + p$
  End if
End for
Return $estimated_count$

If act is the actual query count on the original data set, and est is the query count estimated on the anatomized data set, the relative error is $RE = \frac{est-act}{est}$. As counts are computed for randomly selected predicates, to have a reliable result we need to compute the RE in many different queries and take some aggregated RE measure. In particular, we use $1,000$ different queries and, following [7], the reported RE is the median across all the queries.

3) Evaluation Results: In [7], several experiments were conducted to compare the SABRE methods to previously existent alternatives. Default parameters in those experiments were: data set size $n = 100,000$, query selectivity $s = 0.1$, level of t-closeness $t = 0.35$, number of QI attributes in the count queries $\lambda = 3$, and minimum cluster size $k = 6$. For comparison with our methods, we use the same information loss metrics and evaluation parameters.

In Figure 1, we show the evolution of RE as a function of the number λ of QIs. Following [7], the query selectivity is set to $s = 0.1$ and $t = 0.35$. For the reader's convenience, on the left of Figure 1, we include the chart that was published in [7] and on the right we display the RE of our methods. In essence, we observe that in all algorithms RE increases with λ. However, *our methods offer a radical RE improvement*. The two best methods in [7] (SABRE-KNN and SABRE-AK) achieve an RE that ranges between 10% (when $\lambda = 1$) and 100% (when $\lambda = 7$). That is a large RE that may have an important impact on the validity of statistical analyses. With our methods, *RE is drastically lower*: Algorithm 2 yields an RE less than 2.5% when $\lambda = 7$; Algorithm 3 performs even better: its RE is less than 0.25% when $\lambda = 7$.

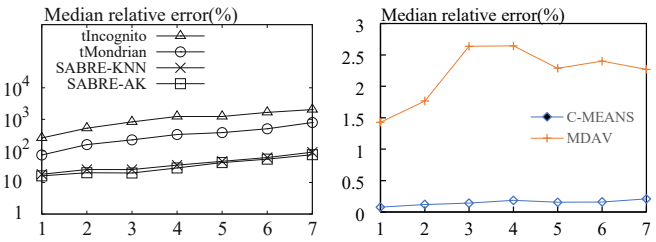


Fig. 1. t-Closeness. Comparison of the % relative error in count queries in terms of the number of λ of QI attributes involved in the query ranges (x-axis), when $t = 0.35$ and $s = 0.1$

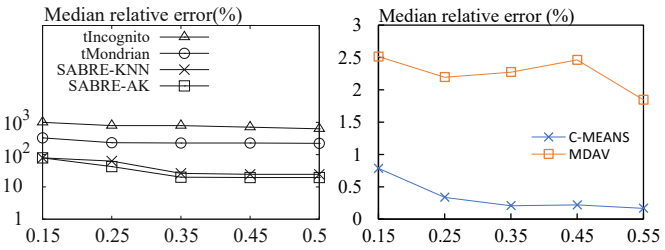


Fig. 2. t-Closeness. Comparison of the % relative error in count queries in terms of the level of t-closeness (x-axis), when $s = 0.1$ and $\lambda = 3$

In Figure 2, we show the evolution of RE as a function of t. Following [7], the query selectivity is set to $s = 0.1$ and $\lambda = 3$. On the left of Figure 2, we include the chart that was published in [7]. On the right, we display the RE of our methods. We observe that RE decreases as t increases. This makes sense because increasing t relaxes the EMD of equivalence classes. The two best methods in [7] (SABRE-KNN and SABRE-AK) achieve an RE between 100% (when $t = 0.15$) and 10% (when $t = 0.55$). These are large RE, and thus one must be wary when analyzing the protected data. When using the proposed steered microaggregation algorithms, *RE is reduced dramatically*: Algorithm 2 yields RE between 2.5% (when $t = 0.15$) and 1.75% (when $t = 0.55$); Algorithm 3 yields an RE between 0.8% (when $t = 0.15$) and 0.2% (when $t = 0.55$).

Figure 3 shows the evolution of the RE in terms of s, the query selectivity. Following [7], the level of t-closeness is set to $t = 0.35$ and $\lambda = 3$. On the left of Figure 2, we include the chart that appears in [7]. On the right, we display the RE of our methods. We observe that RE decreases as the query selectivity is increased. The rationale is that the greater the query selectivity, the greater the count of the query and the lower the relative error RE. Similarly to the previous cases,

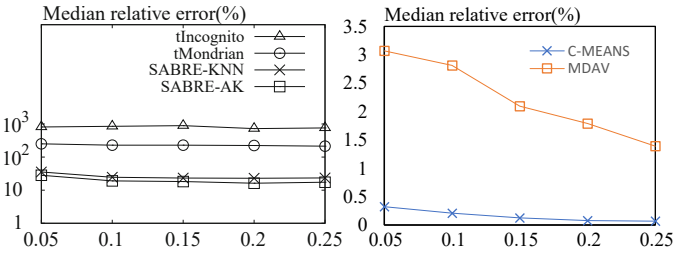


Fig. 3. t-Closeness. Comparison of the % relative error in count queries in terms of the query selectivity (x-axis), when $t = 0.35$ and $\lambda = 3$

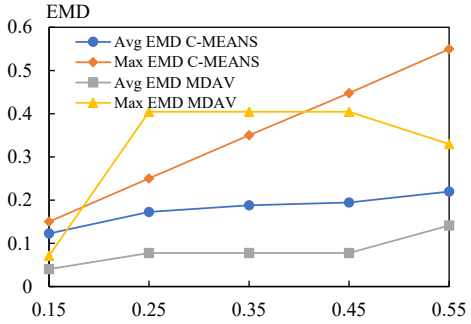


Fig. 4. t-Closeness. Average EMD and maximum EMD (achieved t-closeness level) in the protected data set, as a function of the required t-closeness level (x-axis). The weight of the artificial attribute in steered MDAV is set to $\omega = 1$.

our methods achieve a dramatical RE reduction.

The previous evaluations have shown that the information loss is much smaller in our algorithms and, particularly, in Algorithm 3. Now, we evaluate the level of protection that is attained. We should keep in mind that Algorithm 2 aims at minimizing the information loss in the microaggregation and that the attained level of t-closeness is a consequence of the artificial attribute and the weight attached to it. In Figure 4 we show the maximum and the average EMD of the clusters in the protected data set. We can notice that Algorithm 3 always spends the available privacy budget (the maximum EMD is always equal to the required level of t-closeness). On the contrary, the behavior of Algorithm 2 is more irregular: in some cases the level of t-closeness is more stringent than required, and in other cases the algorithm spends more privacy budget than is available.

Let us now compare the execution times as a function of the data set size. On the left of Figure 5, we include the execution times published in [7]. On the right, we display the execution times of our algorithms. We observe that our methods take about the same time as SABRE-KNN. However, they are between two and three orders of magnitude slower than SABRE-AK. If we prioritize low information loss for a certain t-closeness level, the longer computing time of our methods is fully justified. In contrast, if speed is the primary consideration (for example, because the data sets are very large), then SABRE-AK ranks first. However, we sketch below ways to speed up our methods that match the speed of SABRE-AK while offering lower relative error.

To reduce the runtime of our algorithms, several simplifications are conceivable. A first approach is to run the algorithms

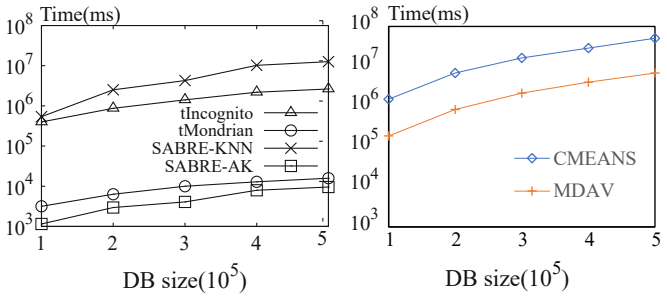


Fig. 5. t-Closeness. Execution time in milliseconds for data set sizes between 10^5 and 5×10^5

on a reduced sample of the collected data. In this way, runtime can be traded off for data utility. As seen in Figure 5, our methods can deal with samples of $500,000$ records in reasonable time; such large samples should be enough to get a good accuracy except perhaps for very complex data domains. When simple random sampling is not feasible, we can slightly alter the algorithms to make them significantly faster. In Algorithm 3, there are two easy simplifications:

- *Increase the convergence threshold.* Algorithm 3 keeps moving individual records to the closest cluster (provided that the privacy conditions are satisfied). In principle, if we want strict convergence, we should keep iterating until no more records move between clusters. However, most of the cluster reassignments are performed in the first iterations. Later iterations only reassign a small number of records. As running one iteration is costly, we may stop the process when the number of reassignments is below a given threshold without a significant impact on the utility of the data. For example, Figure 6 shows the relative error and the execution time of Algorithm 3 when the convergence threshold is set to $0\%, 0.1\%, \ldots, 1\%$ of the data set size (no. of records). We observe that setting the threshold to 1% halves execution time without any significant impact on data utility.
- *Increase the minimum cluster size.* Since the execution time of Algorithm 3 is quadratic in the number of clusters, we can significantly speed up the algorithm by increasing the size of the initial clusters (if smaller clusters can satisfy k-anonymity and t-closeness, so can larger clusters). Figure 7 shows the relative error and the execution time of Algorithm 3 when the minimal cluster size is set to a $0\%, 0.1\%, \ldots, 1\%$ of the data set size. For instance, by increasing the cluster size to 1% of the records, we reduce the execution time by a factor of 100 without any noticeable effect on the relative error.

B. l-Diversity evaluation

l-Diversity algorithms usually operate by generalizing the quasi-identifiers: given a set of generalization hierarchies (one for each quasi-identifier), they select an appropriate level of generalization for each attribute to make the resulting equivalence classes l-diverse. The usual evaluation of l-diversity algorithms is in terms of the information loss inflicted by

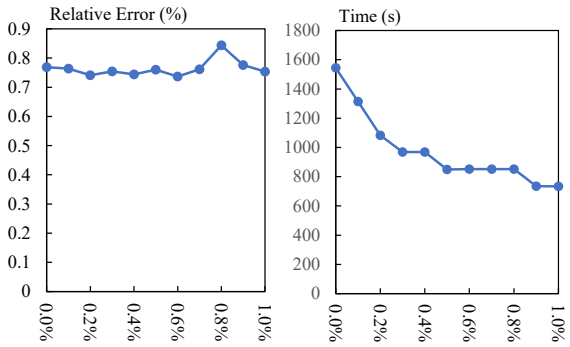


Fig. 6. t-Closeness. Evolution of RE and computing time (seconds) when the convergence threshold is increased to different percentages of the total number of records in the data set (x-axis)

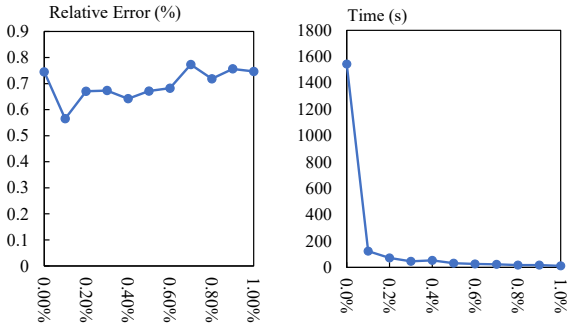


Fig. 7. t-Closeness. Evolution of RE and computing time (seconds) when the minimum cluster size is increased to different percentages of the total number of records in the data set (x-axis)

generalization on quasi-identifiers. Since our approach is based on microaggregation, such information loss metrics are not applicable. Hence, we measure utility as the relative error in count queries, as described in Section VIII-A.

For this evaluation, we used the anonymization software ARX [1]. To achieve l-diversity, we instructed ARX to use generalization while minimizing the average size of equivalence classes for the sake of utility. We used the same Census-based data sets described in Section VIII-A and we took the generalization hierarchies from [28].

In Figure 8 (left), we plot the relative error RE in count queries for different levels of query selectivity ($s \in \{0.05, 0.1, 0.15, 0.2, 0.25\}$). It can be seen that C-MEANS-L (the c-means based steered microaggregation algorithm described in Section V) *absolutely outperforms* ARX by achieving *a radically smaller RE*.

We attribute the drastic RE reduction for l-diversity and also for t-closeness to the fact that steered microaggregation does not rely on generalization. Indeed, generalization tends to yield large equivalence classes, because it is constrained by the levels available in the generalization hierarchies. In turn, large classes result in large information loss and hence large RE. As an example, although 5-diversity can in principle be achieved by a class with 5 records and in spite of having instructed ARX to minimize the size of classes, the average size of the classes output by ARX was 1,041.

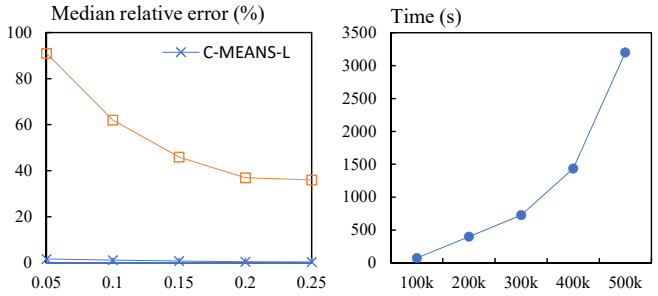


Fig. 8. l-Diversity. Left, comparison of the % relative error in count queries for a 5-diverse data set in terms of the query selectivity (x-axis). Right, computing time of C-MEANS-L under strictest convergence for 5-diversity and data set sizes between $100,000$ and $500,000$.

Regarding runtime, ARX took less than 1 second to generate the 5-diverse data sets, irrespective of the data set size. The C-MEANS-L runtime for data set sizes between $100,000$ and $500,000$ records is shown in Figure 8 (right). Clearly, C-MEANS-L takes much longer, but its runtime can be reduced with little RE impact using strategies similar to those discussed in Section VIII-A for t-closeness: increase the convergence threshold or the minimum cluster size. Furthermore, when anonymizing static data sets a small information loss is normally more important than a short runtime.

C. ϵ-Differential privacy evaluation

To evaluate steered microaggregation for ϵ-DP, note that the method described in Section VI only attains DP for the confidential attributes. This is different from the usual aim of generating ε-DP data sets. For the evaluation to be fair, we made sure that DP was applied to the same attributes.

In this evaluation, we compare the method described in Section VI with the method described in [40] in terms of the RE in count queries. We use the same data set that was considered in [40], namely a subset of the Census data set consisting of 30,162 records with attributes Occupation, Native-Country, Age and (working) Hours-per-Week. The first two attributes are categorical and play the role of quasi-identifiers, and the rest are numerical and are taken as confidential attributes. DP is enforced only on the two confidential attributes.

Figure 9 compares the median RE of C-MEANS-DP (the algorithm described in Section VI) and MDP (the algorithm described in [40]), for $\epsilon = 1$ and count queries with selectivity level $s \in \{0.05, 0.1, 0.15, 0.2, 0.25\}$. Algorithm MDP applies prior microaggregation to reduce the amount of noise required to attain DP; we have considered microaggregation group sizes 2, 250 and 500. Best results are obtained for size 500, so we have used that size in the figure. We observe that C-MEANS-DP *radically outperforms* MDP in achieving a low RE.

D. Stream k-anonymity evaluation

Regarding Algorithm 4 for stream k-anonymity, our evaluation will focus on the effect of the weight ω_p on tuple reordering in the anonymized stream. We replace the QIs in each cluster by their centroid.

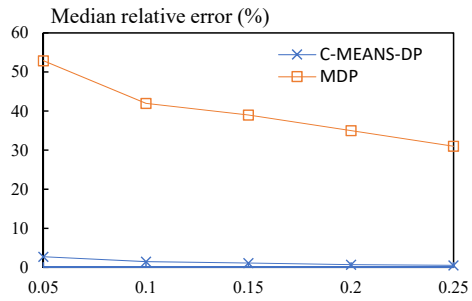


Fig. 9. Differential privacy. Comparison of the % relative error in count queries for DP with $\epsilon = 1$ in terms of the query selectivity (x-axis).

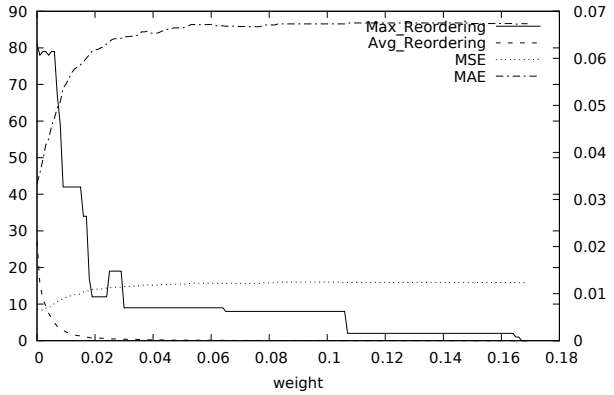


Fig. 10. Stream k-anonymity. Utility indicators with Algorithm 4 for $k = 10$ and $\delta = 100$ on the MCD data set in terms of the weight attached to the position attribute. The left-hand side ordinate scale refers to the reordering amount; the right-hand side ordinate scale refers to MSE and MAE.

1) Datasets: We evaluate Algorithm 4 on the Moderately Correlated Dataset (MCD), also used in [41]. It consists of $1,080$ records containing the numerical attributes TAXINC and POTHVAL (as QIs), and FEDTAX (as sensitive attribute). The correlation between the QI attributes and the sensitive attribute is around 0.5.

2) Utility Metrics: Algorithm 4 trades off the satisfaction of stream k-anonymity for improved data utility. In the experiments we evaluate how this trade-off evolves when changing the importance attached to the satisfaction of the delay constraint (as determined by ω_p). To this end, we use the following utility metrics:

- *Mean square error (MSE).* This is a usual information loss measure in microaggregation. We use it to evaluate the utility loss inflicted to QI attributes. MSE measures the dispersion of the records around their corresponding centroid. It is computed as the average of the squared distance between each record/tuple and its corresponding cluster centroid. For a data set D and a microaggregation partition P, MSE is computed as $MSE_P = \frac{1}{|D|}\sum_{x\in D} d(x, centr(C_x))^2$, where C_x is the cluster in P that contains record/tuple x, and $centr(C_x)$ is the centroid of C_x.
- *Mean absolute error (MAE).* MAE is similar to MSE, but it is less sensitive to outliers. It is computed as $MAE_P = \frac{1}{|D|}\sum_{x\in D} |d(x, centr(C_x))|$.
- *Maximum and average within-cluster tuple position differences.* These metrics measure the amount of tuple reordering in the case of stream anonymization.

3) Evaluation Results: Figure 10 shows the results obtained for the MCD data set (normalized to $[0, 1]$) with parameters $k = 10$ and $\delta = 100$. Note that $\omega_p = 1/\delta$ would normalize the artificial position attribute in the current window to a range of 1; that is, it attaches to the artificial position attribute the same importance as to the other attributes. In our case, $1/\delta = 0.01$. We observe that the most significant effect takes place for values of the weight attribute in the range $[0, 0.02]$. With $\omega_z = 0$, we only focus on minimizing MSE and MAE while keeping the delay below 100; in this case, the maximum and average reordering are quite large (80 and 25, respectively). With weight equal to 0.02, the maximum and average reordering are significantly reduced (down to 12 and 0.7, respectively) but there is a steep increase in MSE and MAE. The optimal value of the weight depends on how important is limiting the reordering vs the within-cluster homogeneity of QI values.

IX. Conclusions and Future Research

In this work, we have introduced the notion of steered microaggregation, which is a general framework to guide regular microaggregation algorithms into satisfying additional criteria (beyond cluster cardinality and maximal within-cluster homogeneity). This is done by adding artificial attributes with appropriate weights that influence the microaggregation. We have described two applications of steered microaggregation and we have presented experimental results on their performance:

- Enforcement of k-anonymity extensions and ϵ-DP in a static data set. This is done by introducing an artificial attribute that controls the within-cluster variability of the sensitive attribute. We have focused on the enforcement of t-closeness and l-diversity, but similar approaches are conceivable to satisfy other k-anonymity extensions. We have also pointed out that, thanks to the connection between t-closeness and ϵ-DP, steered microaggregation can satisfy the latter privacy model under specific conditions. Our experimental results show that the proposed steered microaggregation dramatically outperforms existent methods at reducing information loss, which is mainly due to using microaggregation instead of generalization.
- Generation of k-anonymous streams with improved preservation of the original order of the tuples. Since standard microaggregation algorithms for static data sets are not suitable for data streams, we have first proposed a basic microaggregation algorithm for stream k-anonymity that satisfies a maximum-delay constraint, but gives no order preservation guarantees. Then, we have shown how to use steered microaggregation to improve the preservation of the original order of the tuples in the anonymized data stream.

In the static data set setting, an interesting research avenue is to explore the use of steered microaggregation to satisfy

other privacy models. In what regards data streams, a challenge is whether and how steered microaggregation can satisfy k-anonymity extensions for data streams with moderate reordering and delay. Of particular interest is stream t-closeness, because satisfying it would also allow ϵ-DP queries to be answered based on t-close data streams.

ACKNOWLEDGMENTS AND DISCLAIMER

The following funding sources are gratefully acknowledged: European Commission (project H2020-700540 "CANVAS"), Government of Catalonia (ICREA Acadèmia Prize to J. Domingo-Ferrer and grant 2017 SGR 705) and Spanish Government (project TIN2014-57364-C2-1-R). The views in this paper are the authors' own and do not necessarily reflect the views of UNESCO or any of the funders.

APPENDIX

Proof. (Proposition 1). For t-closeness to hold, the EMD distance between the cluster-level distribution and the data set-level distribution of Z must be no more than t.

Let the possible values of Z be $\{v_1, \ldots, v_m\}$ sorted in ascending order. Let $B = \{B_1, \ldots, B_b\}$ be a partition of the records in D into b buckets of n/b records in ascending order of the attribute Z. Let C be a cluster that contains exactly one record from each bucket B_i, for $i = 1, \ldots, b$.

Consider value v_i. Due to the form of C (for each n/b records of D there must be one record of C), the maximum difference between the distribution of Z over C and the distribution of Z over D is $\frac{n/b-1}{n}$. This the case when D contains $n/b-1$ records with $Z = v_i$ and the cluster contains none of those records.

Thus, EMD is maximized when the difference between the accumulated probabilities reaches its maximum for all values $v_1, \ldots, v_{m-1}$, in which case EMD is $\frac{n/b-1}{n}$. By requiring this maximum distance to be no more than t, the proposition follows. □

Proof. (Proposition 2). Let us consider two records $\mathbf{r}_i$ and $\mathbf{r}_j$ that are maximally different in terms of $Q_1, \ldots, Q_m$, but have the same value for $B(Z)$ (which means their values Z fall in different buckets). Without loss of generality, we can assume that the values of the normalized attributes $\hat{Q}_1, \ldots, \hat{Q}_m, \hat{B}(Z)$ input for both records to the microaggregation are: $\mathbf{r}_i.\hat{q}_1 = \ldots = \mathbf{r}_i.\hat{q}_m = 0, \mathbf{r}_i.\hat{b}(z) = 0$; and $\mathbf{r}_j.\hat{q}_1 = \ldots = \mathbf{r}_j.\hat{q}_m = 1, \mathbf{r}_j.\hat{b}(z) = 0$.

Now consider another record $\mathbf{r}_{j'}$ that is maximally similar to $\mathbf{r}_i$ in terms of $Q_1, \ldots, Q_m$: that is, $\mathbf{r}_i$ and $\mathbf{r}_{j'}$ have identical values for $Q_1, \ldots, Q_m$. Hence, the normalized values of those attributes for $\mathbf{r}_{j'}$ input to the microaggregation are: $\mathbf{r}_{j'}.\hat{q}_1 = \ldots = \mathbf{r}_{j'}.\hat{q}_m = 0$. Furthermore, $\mathbf{r}_i$ and $\mathbf{r}_{j'}$ have values of Z that fall in the same bucket and we consider that: i) this is the largest bucket, B_1; and ii) $\mathbf{r}_i$ and $\mathbf{r}_{j'}$ are labeled with consecutive values of $B(Z)$. This case yields the minimum possible difference between the normalized values of $B(Z)$ for both records: $|\mathbf{r}_i.\hat{b}(z) - \mathbf{r}_{j'}.\hat{b}(z)| = \omega_z/|B_1|$.

Proving the proposition reduces to proving that, when using the Euclidean distance to calculate the distance between records, $\mathbf{r}_i$ and $\mathbf{r}_{j'}$ are found to be more distant than $\mathbf{r}_i$ and $\mathbf{r}_j$. This will mean that, no matter their similarity in terms of $Q_1, \ldots, Q_m$, two records with values of Z in the same bucket are regarded as less similar than two records having the same value $B(Z)$ (and hence with values of Z in different buckets).

Now, $|\mathbf{r}_i - \mathbf{r}_{j'}| = \sqrt{\omega_z^2/|B_1|^2}$, whereas $|\mathbf{r}_i - \mathbf{r}_j| = \sqrt{m}$. By the lower bound on ω_z assumed in the proposition, it holds that $|\mathbf{r}_i - \mathbf{r}_{j'}| > |\mathbf{r}_i - \mathbf{r}_j|$. □

Proof. (Proposition 4). Let us consider two tuples s_i and s_j that are maximally different in terms of $Q_1, \ldots, Q_m$ and P, within the maximum allowed position difference $k-1$. Without loss of generality, we can assume the normalized QI attributes for s_i and s_j to be as follows: $s_i.\hat{q}_1 = \ldots = s_i.\hat{q}_m = 0$ and $s_j.\hat{q}_1 = \ldots = s_j.\hat{q}_m = 1$.

On the other hand, the maximum difference between the normalized positions within a cluster that is compatible with no reordering is $k-1$. By normalizing, we get $|s_i.\hat{p} - s_j.\hat{p}| = \omega_p(k-1)/(k-1) = \omega_p$.

Now consider another tuple $s_{j'}$ that is maximally similar to s_i in terms of the QI attributes $\hat{Q}_1, \ldots, \hat{Q}_m$, but that minimally violates the allowed position difference p. That is, $s_{j'}.\hat{q}_1 = \ldots = s_{j'}.\hat{q}_m = 0$ and $|s_i.\hat{p} - s_{j'}.\hat{p}| = \omega_p k/(k-1)$.

Proving the proposition reduces to proving that, when using the Euclidean distance to calculate the distance between tuples, s_i and $s_{j'}$ are found to be more distant than s_i and s_j. This will mean that, no matter the similarity in terms of QI attributes $Q_1, \ldots, Q_m$, two tuples that violate the allowed position difference (even if only minimally) are regarded as less similar than two tuples meeting the position difference constraint. Hence, microaggregation will yield clusters whose tuples satisfy the position difference constraint.

Now, $|s_i - s_{j'}| = \omega_p k/(k-1)$, whereas $|s_i - s_j| = \sqrt{m + \omega_p^2}$. By the lower bound on ω_p assumed in the proposition, it holds that $\omega_p k/(k-1) > \sqrt{m + \omega_p^2}$. □

REFERENCES

[1] *ARX Data Anonymization Tool*, version 3.7.1, March 8, 2018. https://arx.deidentifier.org

[2] S. Äyrämö and T. Kärkkäinen. Introduction to partitioning-based clustering methods with a robust example. Reports of the Department of Mathematical Information Technology Series C. Software and Computational Engineering No. C. 1/2006.

[3] A. Blum, K. Ligett and A. Roth. A learning theory approach to non-interactive database privacy. In: *Proc. of the 40th Annual Symposium on the Theory of Computing-STOC 2008*, pp. 609-618, 2008.

[4] Y. Bu, A. W. C. Fu, R. C. W. Wong, L. Chen and J. Li. Privacy-preserving serial data publishing by role composition. *Proc. of the VLDB Endowment* 1(1):845-856, 2008.

[5] J. Cao, B. Carminati, E. Ferrari and K.-L. Tan. CASTLE: continuously anonymizing data streams. *IEEE Transactions on Dependable and Secure Computing*, 8(3):337-352, 2011.

[6] J. Cao and P. Karras. Publishing microdata with a robust privacy guarantee. *Proc. of the VLDB Endowment* 5(11):1388-1399, 2012.

[7] J. Cao, P. Karras, P. Kalnis and K.-L. Tan. SABRE: a sensitive attribute bucketization and redistribution framework for t-closeness. *The VLDB Journal*, 20(1):59-81, 2011.

[8] R. Chen, N. Mohammed, B. C. M. Fung, B. C. Desai and L. Xiong. Publishing set-valued data via differential privacy. In: *Proc. of the 37th Intl. Conf. on Very Large Data Bases-VLDB 2011/Proc. of the VLDB Endowment*, 4(11):1087-1098, 2011.

[9] J. P. Daries, J. Reich, J. Waldo, E. M. Young, J. Whittinghill, A. D. Ho, D. T. Seaton and I. Chuang. Privacy, anonymity and big data in the social sciences. *Communications of the ACM*, 57(9):56-63, 2014.

[10] J. Domingo-Ferrer and J. M. Mateo-Sanz. Practical data-oriented microaggregation for statistical disclosure control. *IEEE Transactions on Knowledge and Data Enginering* 14(1):189-201, 2002.

[11] J. Domingo-Ferrer, D. Sánchez and G. Rufián-Torrell. Anonymization of nominal data based on semantic marginality. *Information Sciences* 242:35-48, 2013.

[12] J. Domingo-Ferrer and J. Soria-Comas. From t-closeness to differential privacy and vice versa in data anonymization. *Knowledge-Based Systems*, 74:151-158, 2015.

[13] J. Domingo-Ferrer and J. Soria-Comas. Anonymization in the time of big data. In *Proc of the Intl. Conference on Privacy in Statistical Databases-PDS 2016*. LNCS 9867, Springer, pp. 57-68, 2016.

[14] J. Domingo-Ferrer and J. Soria-Comas. Steered microaggregation: a unified primitive for anonymization of data sets and data streams. In *IEEE ICDM 2017 Workshops Proceedings-PAIS 2017*. [To appear].

[15] J. Domingo-Ferrer and V. Torra. Ordinal, continuous and heterogeneous k-anonymity through microaggregation. *Data Mining and Knowledge Discovery* 11(2):195-212, 2005.

[16] C. Duhigg. How companies learn your secrets. *New York Times Magazine*, Feb. 16, 2012.

[17] C. Dwork. Differential privacy. In: *Proc. of the 33rd Intl. Colloquium on Automata, Languages and Programming-ICALP 2006*, LNCS 4052, Springer, pp. 1-12, 2006.

[18] C. Dwork, M. Naor, O. Reingold, G. N. Rothblum and S. Vadhan. On the complexity of differentially private data release: efficient algorithms and hardness results. In: *Proc. of the 41st Annual Symposium on the Theory of Computing-STOC 2009*, pp. 381-390, 2009.

[19] E.A.H. Elamir and C.J. Skinner. Record level measures of disclosure risk for survey microdata. *Journal of Official Statistics* 22(3):525-539, 2006.

[20] M. Elliot, E. Mackie, S. O'Shea, C. Tudor and K. Spicer. End user licence to open government data? A simulated penetration attack on two social survey datasets. *Journal of Official Statistics* 32(2):329-348, 2016.

[21] J. Gehrke, M. Hay, E. Lui and R. Pass. Crowd-blending privacy. In: *Advances in Cryptology-CRYPTO 2012*, LNCS 7417, pp. 479-496, 2012.

[22] M. Hardt, K. Ligett and F. McSherry. A simple and practical algorithm for differentially private data release. Preprint arXiv:1012.4763v1, 21 Dec 2010.

[23] S. Har-Peled and B. Sadri. 2005. How fast is the k-means method?. In: *Proc of the 16th ACM-SIAM symposium on Discrete algorithms-SODA 2005*, pp. 877-885, 2005.

[24] A. Hundepool, J. Domingo-Ferrer, L. Franconi, S. Giessing, E. Schulte Nordholt, K. Spicer and P.-P. de Wolf. *Statistical Disclosure Control*, Wiley, 2012.

[25] D. Lambert. Measures of disclosure risk and harm. *Journal of Official Statistics* 9(3):313-331, 1993.

[26] N. Li, T. Li and S. Venkatasubramanian. t-Closeness: privacy beyond k-anonymity and l-diversity. In: *Proc. of the 23rd IEEE Intl. Conf. on Data Engineering-ICDE 2007*, pp. 106-115, 2007.

[27] N. Li, T. Li and S. Venkatasubramanian. Closeness: a new privacy measure for data publishing. *IEEE Transactions on Knowledge and Data Engineering* 22(7):943-956, 2010.

[28] J. Liu and K. Wang. On optimal anonymization for $l+$-diversity. *IEEE 26th International Conference on Data Engineering-ICDE 2010*, pp. 213-224, 2010.

[29] A. Machanavajjhala, J. Gehrke, D. Kiefer, and M. Venkitasubramaniam. l-Diversity: privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data* 1(1):art. no. 3, 2007.

[30] A. Oganian and J. Domingo-Ferrer. On the complexity of optimal microaggregation for statistical disclosure control. *Statistical Journal of the United Nations Economic Commission for Europe* 18(4):345-354, 2001.

[31] J. Pei, J. Xu, Z. Wang, W. Wang and K. Wang. Maintaining k-anonymity against incremental updates. In *SSDBM'07*, IEEE, p. 5-16, 2007.

[32] D. Rebollo-Monedero, J. Forné and J. Domingo-Ferrer. From t-closeness-like privacy to postrandomization via information theory. *IEEE Transactions on Knowledge and Data Engineering* 22(11):1623-1636, 2010

[33] D. Riboni, L. Pareschi and C. Bettini. JS-Reduce: defending your data from sequential background knowledge attacks. *IEEE Transactions on Dependable and Secure Computing* 9(3):387-400, 2012.

[34] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision* 40(2):99-121, 2000.

[35] P. Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. SRI International Report, 1998.

[36] D. Sánchez, J. Domingo-Ferrer, S. Martínez, and J. Soria-Comas. Utility-preserving differentially private data releases via individual ranking microaggregation. *Information Fusion*, 30:1-14, 2016.

[37] E. Shmueli and T. Tassa. Privacy by diversity in sequential releases of databases. *Information Sciences* 298(20):344-372, 2015.

[38] E. Shmueli, T. Tassa, R. Wasserstein, B. Shapira and L. Rocach. Limiting disclosure of sensitive data in sequential releases of databases. *Information Sciences* 191:98-127, 2012.

[39] J. Soria-Comas and J. Domingo-Ferrer. Probabilistic k-anonymity through microaggregation and data swapping. In: *FUZZ-IEEE 2012, IEEE International Conference on Fuzzy Systems*, IEEE, pp. 1-8, 2012.

[40] J. Soria-Comas, J. Domingo-Ferrer, D. Sánchez and S. Martínez. Enhancing data utility in differential privacy via microaggregation-based k-anonymity. *VLDB Journal* 23(5):771-794, 2014.

[41] J. Soria-Comas, J. Domingo-Ferrer, D. Sánchez and S. Martínez. t-Closeness through microaggregation: strict privacy with enhanced utility preservation. *IEEE Transactions on Knowledge and Data Engineering*, 27(11):3098-3110, 2015.

[42] V. Torra and J. Domingo-Ferrer. Record linkage methods for multidatabase data mining. In: *Information Fusion in Data Mining*, Springer, pp. 99-130, 2003.

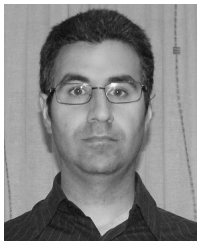
[43] K. Wang and B. Fung. Anonymizing sequential releases. In: *KDD'06*, ACM, pp. 414-423, 2012.

[44] X. Xiao and Y. Tao. Anatomy: simple and effective privacy preservation. In: *Proc. of the 32nd Intl. Conference on Very Large Data Bases-VLDB 2006*,pp. 139-150, 2006.

[45] X. Xiao and Y. Tao. M-Invariance: towards privacy-preserving republication of dynamic datasets. In *SIGMOD'07*, ACM, pp. 689-700, 2007.



Josep Domingo-Ferrer (Fellow, IEEE) is a distinguished professor of computer science and an ICREA-Acadèmia researcher at Universitat Rovira i Virgili, Tarragona, Catalonia, where he holds the UNESCO Chair in Data Privacy and leads CYBERCAT. He received the MSc and PhD degrees in computer science from the Autonomous University of Barcelona in 1988 and 1991, respectively. He also holds an MSc degree in mathematics. His research interests are in data privacy, data security and cryptographic protocols.



Jordi Soria-Comas is a postdoctoral researcher at Universitat Rovira i Virgili, Tarragona, Catalonia. He received an MSc in computer security (2011) and a PhD in computer science (2013) from Universitat Rovira i Virgili. He also holds an MSc in finance from the Autonomous University of Barcelona (2004) and a BSc in mathematics from the University of Barcelona (2003). His research interests are in data privacy and security.



Rafael Mulero-Vellido is a master's student at Universitat Rovira i Virgili, Tarragona, Catalonia. He received a BEng in Computer Science (2017) from the Technical University of Catalonia. He has collaborated as a developer in different research projects and his research interests are in data privacy and security.