# A General Algorithm for $k$-anonymity on Dynamic Databases

Julián Salas[1(✉)] and Vicenç Torra[2]

[1] Internet Interdisciplinary Institute (IN3), Universitat Oberta de Catalunya (UOC), Center for Cybersecurity Research of Catalonia (CYBERCAT), Barcelona, Spain
jsalaspi@uoc.edu

[2] School of Informatics, University of Skövde, Skövde, Sweden
vtorra@his.se

**Abstract.** In this work we present an algorithm for $k$-anonymization of datasets that are changing over time. It is intended for preventing identity disclosure in dynamic datasets via microaggregation. It supports adding, deleting and updating records in a database, while keeping $k$-anonymity on each release.

We carry out experiments on database anonymization. We expected that the additional constraints for $k$-anonymization of dynamic databases would entail a larger information loss, however it stays close to MDAV's information loss for static databases.

Finally, we carry out a proof of concept experiment with directed degree sequence anonymization, in which the removal or addition of records, implies the modification of other records.

**Keywords:** Big data privacy · $k$-anonymity · Graph anonymization
Geo-spatial data anonymization · Microaggregation
Dynamic data privacy

## 1 Introduction

Dynamic publication of databases and combining data from diverse sources increases privacy risks, any privacy model must satisfy requirements such as linkability, composability and computability to be useful for big data anonymization [1,2]. Composability means that the privacy guarantees of the model are preserved (possibly to a limited extent) after repeated independent application of the privacy model. In [3], it was proved that linking two $k$-anonymous datasets does not imply that the obtained data set is $k$-anonymous for any $k > 1$. That is, $k$-anonymity in general does not guarantees composability.

However, in this paper we show that composability may be achieved considering that the data is managed by only one central holder as in the case of a dynamic database. Thus, providing a general algorithm for $k$-anonymity of dynamic data.


J. Garcia-Alfaro et al. (Eds.): DPM 2018/CBT 2018, LNCS 11025, pp. 407–414, 2018.
https://doi.org/10.1007/978-3-030-00305-0_28

The concept of $k$-anonymity was defined in [4] and [5]. This model assures that any individual in the dataset is indistinguishable from at least other $k-1$ individuals in terms of quasi-identifier attributes values (QI).

The definition of $k$-anonymity for graphs can be restated considering that the attacker knows a specific property $\mathcal{P}$ of a graph, see [6]. In this case, the structural property $\mathcal{P}$ of the graph is the equivalent to a QI in a database. An example of this property $\mathcal{P}$ is the degree of the nodes [7].

Graph modifications to guarantee $k$-degree anonymity have additional restrictions, for example, the $k$-anonymous degree sequences must be graphic, i.e., they must correspond to the sequence of degrees of a graph. Some theoretical conditions for degree sequences to be graphic and applications to $k$-degree anonymization and edge randomization can be found in [8,9].

In this paper we provide a general algorithm based on microaggregation, considering that the tuples of dynamic databases are represented as points in metric spaces, and the databases are updated and published several times. We present examples of the application of our algorithm for databases and degree sequences of directed graphs.

### 1.1 Related Literature

There are several papers that provide $k$-anonymity for multiple publications of databases by means of generalizations. In [10], $k$-anonymity is guaranteed on incremental updates. The authors use generalization as the method for aggregation of the records and reduce the generalization granularity as incremental updates arrive. Their approach guarantees the $k$-anonymity on each release, and also on the inferred table using multiple releases, by full-domain generalization, using multidimensional partitioning with Mondrian algorithm [11].

Sequential anonymization of a given release $T_1$ in the presence of a previous release $T_2$ is considered in [12]. So, the authors consider the case when releasing new attributes associated to same set of individuals. They use generalization/specialization to guarantee $(X, Y)$-anonymity on sequential releases by leveraging the fact that generalizing join attributes makes more matches, cf. [12].

Shmueli et al. [13] extended the framework that was considered in [12], considering also $k$-linkability and $k$-diversity, and achieve them by local recoding (in contrast to Wang et al. global recoding). They expressed the constraints for $k$-anonymization in sequential release with continuous data publishing scenario, as an $R$-partite graph, where $R$ is the number of releases. Then, to compute properly the level of linkability or diversity, it is needed to identify all the $R$-cliques that are part of a perfect matching in the $R$-partite graph. This was shown to be NP-hard for $R > 2$ in [13].

These approaches were improved in [14] with the guarantee that an adversary cannot link any quasi-identifier tuple with any sensitive value with probability greater than $1/\ell$. Their application scenario is of sequential release publishing in which the set of tuples is fixed, while the set of attributes changes from one release to another.

Byun et al. [15] consider record insertions and provide guarantees of $\ell$-diversity, by delayed publishing and maintaining published histories on dynamic databases.

The first study to address both record insertions and deletions in data republication, is proposed in [16]. It proposes a new privacy notion called $m$-invariance, if a record $r$ has been published in subsequent releases $R_1, \ldots, R_i$, then all $QI$ groups containing $r$ must have the same set of sensitive values. They add "counterfeit tuples" and use generalization for anonymization. Moreover, Bu et al. [17] show that the same guarantee of $m$-invariance, may be used for attribute disclosure.

The problem of $k$-anonymization of data streams was studied in [18], in which a data stream is modeled as an infinite append-only sequence of tuples with an incremental order that stores also information about when the data have been collected. In that case, the delay in which data is published is relevant, hence they add a constraint that considers the maximum allowed time of a tuple staying in memory before its output.

It is important to note that in all previous cases the method for anonymization was based on generalization, while we will consider microaggregation, which to our knowledge, has not been used before for $k$-anonymizing dynamic data, except for $k$-anonymization of documents in [19]. Moreover, our method may be used for additions, suppressions and updates.

## 2 Proposed Method

We represent by $D_t$ the publication of database $D$ at timestamp $t$.

To maintain the generality, we denote the elements of the database as pairs $(x_j, t)$, in which $x_j$ represents the QIs of individual $j$ at timestamp $t$. Thus, $x_j$ are vectors in a metric space of QIs.

Our algorithm for dynamic anonymization (Algorithm 1) works as follows:

From database $D$ we obtain a $k$-anonymous database $\tilde{D}$, by applying the MDAV microaggregation method [20,21]. We obtain the groups $C_1, \ldots, C_m$ with $k_1, \ldots, k_m$ elements (all $k_i \geq k$) and centroids $c_1, \ldots, c_m$.

Now, each element $x$ of the database $D$ is represented by some $c_i$ with $i \leq m$ in the anonymized database $\tilde{D}$. Since we are assuming that the space of QIs is a metric space, then we can obtain the Voronoi tessellation of the set of points, that is, we partition the space with respect to the points $C = c_1, \ldots, c_m$ as follows: $P_i = \{x \in D : d(x, c_i) \leq d(x, c_j)$ for all $j \leq m\}$ therefore we obtain the partition $P = P_1, \ldots, P_m$ of the space $D$.

Starting from this partition, when modifying the database by adding a record $x$ in timestamp $t$, denoted as $add(x, t)$, we calculate the centroid with minimum distance to $x$, $d(x, c_i) \leq d(x, c_j)$, assign $x$ to the corresponding set $P_i$, and update the count $k_i$ to $k_i + 1$. If $k_i + 1$ equals $2k$, then all the elements in $P_i$ are used to recalculate new cluster centroids $c_i'$ and $c_{m+1}$ to replace former centroid $c_i$. Note that, the other assignments of records in groups $P_j \neq P_i$ remain unchanged.

If a former element $x \in D$ is removed on timestamp $t$, denoted $remove(x, t)$ $= \emptyset$, the count $k_i$ of the partition $P_i$ that contained $x$ is updated ($k_i = k_i - 1$), whenever $k_i = 0$ the centroid $c_i$ is removed.

For updating an element at timestamp $t$, we are removing the original value $remove(x, t)$ and adding the updated value $add(x, t)$. When making several updates at the same time, the algorithm works in a similar way.

A simple example is depicted in Fig. 1. We consider $k$-anonymization of data on two variables and $k = 2$. In this example, adding the green nodes allows us to update the centers (red triangles) in the right partition. Note that the left size center is not updated, otherwise this will give information about the newly added point.

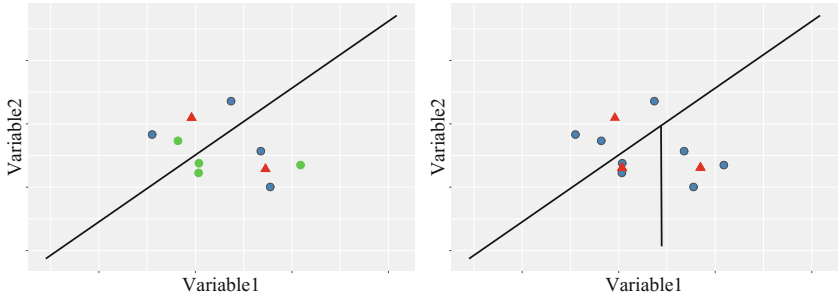


**Fig. 1.** Improving utility by adding records

## 3 Empirical Evaluation

If an individual's record belongs to multiple databases, even when it belongs to $k$-anonymous groups on each of them, his anonymity may be reduced to a value lower than $k$ when the following property does not hold.

**Property 1.** *In the case of multiple releases of the same database, if an individual $x$ is known to belong to a set $S_1$ of $k$-elements on release $t_1$ and is also known to belong to a set $S_2$ of $k$-elements on release $t_2$, then $x$ is known to belong to a set of $|S_1 \cap S_2|$ which may be less than $|S_1|$ and $|S_2|$, unless $S_1 \subset S_2$ or vice versa.*

When we add records to a database, following our Dynamic algorithm we guarantee that this property holds by assigning the new records to groups of at least $k$ records. Only when a set $S_1$ has at least $2k$ elements, we can divide it in sets $S_2, S_3 \subset S_1$ without breaking Property 1. Hence, our Dynamic algorithm maintains $k$-anonymity.

Deleting records may be more problematic because if a group has $k$ records, deleting one node and publishing the remaining would decrease the anonymity set to $k - 1$, this is the reason of not deleting any node until the entire group of $k$ has been deleted in our approach.

**Algorithm 1.** Algorithm for dynamic $k$-anonymity

**Input**: $k$-anonymous database $D$, centroids $C = c_1, \ldots, c_m$, partitions $P = P_1, \ldots, P_m$, counts $\mathcal{K} = k_1, \ldots, k_m$, timestamp $t$, operation $\sigma$.
**Output**: $k$-anonymous database $D_t$, updated centroids $C_t$, and counts $K_t$
**if** $\sigma = add((x,t), D)$ **then**
  $b = argmin_i d(x, c_i)$ (Add $x$ to group $C_b$)
  $k_b = k_b + 1$
  **if** $k_b + 1 = 2k$ **then**
    $(P'_b, P_{m+1}) =$ Apply MDAV to the points in $P_b$
    $C = C \setminus c_b$
    $C = C \cup \{c'_b, c_{m+1}\}$
    $P_b = \emptyset$
  **end**
**end**
**if** $\sigma = remove((x,t), D)$ **then**
  $b = argmin_i d(x, c_i)$ (assign $b$ to buffer of removals $R_b$)
  $k_b = k_b - 1$
  **if** $k_b = 0$ **then**
    $C = C \setminus c_b$
    $R_b = \emptyset$
  **end**
**end**
**return** $(D_t,\ C_t,\ K_t,\ P_t)$

For measuring the information loss, we use the average Euclidean distance to the anonymized records:

$$IL(D, \tilde{D}) = \frac{1}{n} \sum_{1 \le i \le n} d(x_i, \tilde{x}_i)$$

Here we are considering $x_i$ the original record, $\tilde{x}_i$ its corresponding anonymized record, and $d$ the Euclidean distance.

We apply our method to a database and a graph, to test it under two different assumptions, only adding records, or deleting and updating. Since there is no other microaggregation algorithm for dynamic data, we must compare our algorithm to MDAV that is designed for static data.

We use a subset of 4000 records from the census-income dataset from UCI repository [22] which has 40 attributes. We choose these 4000 records such that at least 5 of their 7 continuous attributes are different from 0. These 7 attributes correspond to age, wage per hour, capital gains, capital losses, dividends from stocks, number of persons that worked for employer and weeks worked in the year.

We start with the first 2000 records, obtain a $k$-anonymous version of the database and the centroids $c_1, \ldots, c_m$. Then, we add the records one by one and recalculate the information loss measure $IL$ every time we add a record. In Fig. 2, we plot dynamic $k$-anonymizations for $k = 2, 5$ and compare them to applying the MDAV algorithm for the static dataset with $2000, 2250, 2500, \ldots, 4000$ records.

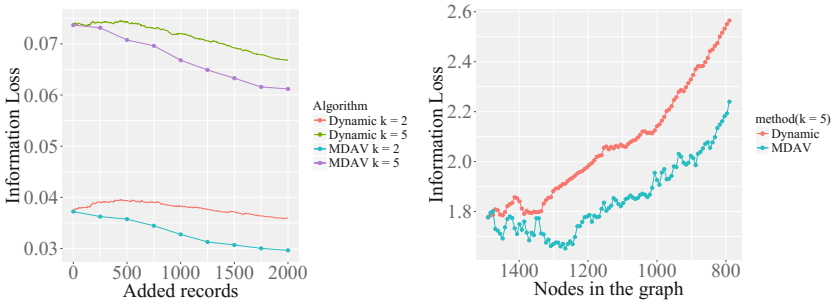


**Fig. 2.** Comparison of MDAV and dynamic algorithms when adding or updating nodes in a database and a graph

It is interesting to note that our Dynamic algorithm is not monotone since the subdivision step happens only when $2k$ values have been gathered on the same group, it increases the information loss locally until the subdivision, that improves it, see Fig. 2.

Next, we apply our method to the degree sequence of the polblogs directed network [23], which has 1490 nodes and 19090 edges, that represent political blogs in the US. In this case, deleting a node implies that all its relations are deleted, hence the degrees of its neighboring nodes are updated and consequently their corresponding records. The degrees of the nodes are represented as points in a 2-dimensional space where the coordinates represent the in-degree and the out-degree, and it is this set of coordinates that we anonymize. We deleted iteratively 7 nodes, until deleting 700 and remaining with a graph with 790 nodes, and made a comparison between MDAV and our algorithm for $k = 5$, see Fig. 2.

Note that the information loss is worse for Dynamic algorithm than for MDAV as the updates may generate additional nodes. Using microaggregation for degree anonymization has additional subtleties, for example, not all the degree sequences are graphic. More details and methods to obtain $k$-degree anonymous directed graphs are explored in [24].

## 4 Conclusions

We defined a general dynamic $k$-anonymity algorithm, that uses microaggregation and guarantees $k$-anonymity in a database with additions, deletions and updates of records. We compared our algorithm with the well-known MDAV algorithm, and found out that MDAV performs slightly better, suggesting that the restrictions of $k$-anonymity for dynamic databases, do not damage considerably the information loss.

As future work, we will apply our dynamic $k$-anonymity algorithm for anonymizing geolocated data and documents. Also, we would like to integrate further constraints such as $\ell$-diversity or $t$-closeness to the algorithm.

**Acknowledgements.** Julián Salas acknowledges the support of a UOC postdoctoral fellowship. This work is partly funded by the Spanish Government through grant TIN2014-57364-C2-2-R "SMARTGLACIS". Vicenç Torra acknowledges the suport of Vetenskapsrdet project: "Disclosure risk and transparency in big data privacy" (VR 2016-03346, 2017-2020).

## References

1. Soria-Comas, J., Domingo-Ferrer, J.: Big data privacy: challenges to privacy principles and models. Data Sci. Eng. **1**(1), 21–28 (2016). https://doi.org/10.1007/s41019-015-0001-x
2. Torra, V., Navarro-Arribas, G.: Big Data Privacy and Anonymization, pp. 15–26. Springer International Publishing, Cham (2016). https://doi.org/10.1007/978-3-319-55783-0_2
3. Stokes, K., Torra, V.: Multiple releases of k-anonymous data sets and k-anonymous relational databases. Int. J. Uncertain. Fuzziness Knowl.-Based Syst. **20**(06), 839–853 (2012). https://www.worldscientific.com/doi/abs/10.1142/S0218488512400260
4. Samarati, P.: Protecting respondents identities in microdata release. IEEE Trans. Knowl. Data Eng. **13**(6), 1010–1027 (2001)
5. Sweeney, L.: k-anonymity: a model for protecting privacy. Int. J. Uncertain. Fuzziness Knowl. Based Syst. **10**(05), 557–570 (2002). https://www.worldscientific.com/doi/abs/10.1142/S0218488502001648
6. Chester, S., Kapron, B.M., Ramesh, G., Srivastava, G., Thomo, A., Venkatesh, S.: Why waldo befriended the dummy? k-anonymization of social networks with pseudo-nodes. Social Netw. Anal. Min. **3**(3), 381–399 (2013). https://doi.org/10.1007/s13278-012-0084-6
7. Liu, K., Terzi, E.: Towards identity anonymization on graphs. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, ser. SIGMOD '08, pp. 93–106. ACM, New York, NY, USA (2008). http://doi.acm.org/10.1145/1376616.1376629
8. Salas, J., Torra, V.: Graphic sequences, distances and k -degree anonymity. Discrete Appl. Math. **188**(C), 25–31 (2015). https://doi.org/10.1016/j.dam.2015.03.005
9. Salas, J., Torra, V.: Improving the characterization of p-stability for applications in network privacy. Disc. Appl. Math. **206**, 109–114 (2016). http://www.sciencedirect.com/science/article/pii/S0166218X16300129
10. Pei, J., Xu, J., Wang, Z., Wang, W., Wang, K.K.: Maintaining k-anonymity against incremental updates. In: 19th International Conference on Scientific and Statistical Database Management (SSDBM 2007), p. 5, July (2007)
11. LeFevre, K, DeWitt, D.J., Ramakrishnan, R.: Mondrian multidimensional k-anonymity. In: 22nd International Conference on Data Engineering (ICDE'06), p. 25, April (2006)
12. Wang, K., Fung, B.C.M.: Anonymizing sequential releases. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ser. KDD '06, pp. 414–423. ACM, New York, NY, USA (2006). http://doi.acm.org/10.1145/1150402.1150449
13. Shmueli, E., Tassa, T., Wasserstein, R., Shapira, B., Rokach, L.: Limiting disclosure of sensitive data in sequential releases of databases. Inf. Sci. **191**, 98–127 (2012). (Data Mining for Software Trustworthiness). http://www.sciencedirect.com/science/article/pii/S0020025511006694

14. Shmueli, E., Tassa, T.: Privacy by diversity in sequential releases of databases. Inf. Sci. **298**(C), 344–372 (2015). https://doi.org/10.1016/j.ins.2014.11.005
15. Byun, J.-W., Sohn, Y., Bertino, E., Li, N.: Secure anonymization for incremental datasets. In: Jonker, W., Petković, M. (eds.) SDM 2006. LNCS, vol. 4165, pp. 48–63. Springer, Heidelberg (2006). https://doi.org/10.1007/11844662_4
16. Xiao, X., Tao, Y.: M-invariance: towards privacy preserving re-publication of dynamic datasets. In: Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data, ser. SIGMOD '07, pp. 689–700. ACM, New York, NY, USA (2007). http://doi.acm.org/10.1145/1247480.1247556
17. Bu, Y., Fu, A.W.C., Wong, R.C.W., Chen, L., Li, J.: Privacy preserving serial data publishing by role composition. Proc. VLDB Endow. **1**(1), 845–856 (2008). https://doi.org/10.14778/1453856.1453948
18. Cao, J., Carminati, B., Ferrari, E., Tan, K.-L.: Castle: continuously anonymizing data streams. IEEE Trans. Dependable Secur. Comput. **8**(3), 337–352 (2011)
19. Navarro-Arribas, G., Abril, D., Torra, V.: Dynamic anonymous index for confidential data. In: Garcia-Alfaro, J., Lioudakis, G., Cuppens-Boulahia, N., Foley, S., Fitzgerald, W.M. (eds.) DPM/SETOP -2013. LNCS, vol. 8247, pp. 362–368. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54568-9_23
20. Domingo-Ferrer, J., Torra, V.: Ordinal, continuous and heterogeneous k-anonymity through microaggregation. Data Min. Knowl. Disc. **11**(2), 195–212 (2005). https://doi.org/10.1007/s10618-005-0007-5
21. Domingo-Ferrer, J., Mateo-Sanz, J.M.: Practical data-oriented microaggregation for statistical disclosure control. IEEE Trans. Knowl. Data Eng. **14**(1), 189–201 (2002)
22. Dheeru, D., Karra Taniskidou, E.: UCI machine learning repository (2017). http://archive.ics.uci.edu/ml
23. Adamic, L.A., Glance, N.: The political blogosphere and the 2004 u.s. election: divided they blog. In: Proceedings of the 3rd International Workshop on Link Discovery, ser. LinkKDD '05, pp. 36–43. ACM, New York, NY, USA (2005). http://doi.acm.org/10.1145/1134271.1134277
24. Casas-Roma, J., Salas, J., Malliaros, F., Vazirgiannis, M.: k-degree anonymity on directed networks. Knowl. Inf. Syst., (2018, to appear)