k-Anonymity in Databases with Timestamped Data

Sergio Mascetti Claudio Bettini
DICo, University of Milan, Italy

X. Sean Wang
Dept of CS, University of Vermont, VT

Sushil Jajodia
CSIS, George Mason University, VA

Abstract

In this paper we extend the notion of k-anonymity in the context of databases with timestamped information in order to naturally define k-anonymous views of temporal data. We also investigate the problem of obtaining these views. We show that known generalization techniques, despite being applicable under certain conditions, have some limitations, and propose a new generalization algorithm based on the hierarchy of time granularities.

1. Introduction

There are many databases that store timestamped data, such as those storing bank transactions, medical exam results, audit logs, insurance records, etc. These data may need to be transferred among organizations, or different branches of the same organization, and sometimes may even need to be available to the general public once appropriately anonymized. A common consensus is that the release of specific stored data, even if anonymized, is in many situations preferable to the release of a statistical summary of the same data.

The notion of k-anonymity, proposed by Samarati [5], addresses the problem of releasing microdata while safeguarding the anonymity of the respondents to which the data refer. In this approach, a database relation provides k-anonymity if any attempt to link identifying information to its content maps the information to at least k individuals. k-anonymity is based on the identification in the relation schema of the set of attributes whose values may be used, possibly together with external information, to re-identify the data. For example, even if data about the ZIP code, date of birth and sex do not explicitly identify an individual, they may be linked to external information (e.g., public voter lists) to obtain name, address and city. These attributes are called *Quasi-Identifiers* (QI). Relations can be made k-anonymous either by dropping attributes that uniquely identify the data respondents or generalizing values.

Table 1. Relation with timestamped data.

Address	date and time	Exam
$Addr_1$	2005-11-22T09:30	Ex_1
$Addr_2$	2005-11-22T10:30	Ex_2
$Addr_2$	2005-11-22T11:30	Ex_3
$Addr_1$	2005-11-23T09:00	Ex_2
$Addr_1$	2005-11-24T10:00	Ex_5

Several papers have recently investigated k-anonymization techniques (see [3] for a taxonomy of k-anonymization models). It is generally assumed that the database relation to be anonymized has a single tuple for each respondent. We argue that in databases with timestamped data this assumption is not reasonable and a trivial conversion of data in terms of a different schema to satisfy this requirement may not be effective. For example, consider a hospital's database containing information about patients and their medical exams. Suppose that some information about medical exams needs to be released after being appropriately anonymized. Table 1 represents information about the patients address, the date and time of the exam and the exam type, possibly together with the exam results and other sensitive information. Clearly, the hospital also knows the actual identity of each patient undertaking a specific exam, but this is not to be revealed.

In order to enforce k-anonymity of this relation with known techniques, the relation should be first transformed in a different one satisfying the condition of having a single tuple for each patient. If we assume that the QI is composed by the attribute $address$ only, and if we assume that, in Table 1, the second and the third tuples belong to the same patient and the remaining tuples to a different one, then one way to achieve the goal is to convert the data using a different schema, as shown in Table 2.

Although the appropriateness of using timestamps in

Table 2. A different relation schema enabling the representation of a single tuple for each patient.

Address	Exam@ 2005-11-22 T09:30	Exam@ 2005-11-22 10:30	...
$addr_1$	Ex_1	NULL	...
$addr_2$	NULL	Ex_2	...

attribute names may be questionable[1], this would enable standard k-anonymization techniques to be applied. However, consider the case that the attacker can acquire spatio-temporal information about the patients, for example from a mobile phone service provider or from a location-based service provider. Since the attacker knows the users' location at specific times and knows the location of the hospital, he knows when a user is at the hospital. In order to protect anonymity, in this case we must consider time information (the date-time attribute) as part of the QI together with the address attribute.

In this case, the conversion of the relation schema performed above poses a serious problem. Indeed, in Table 2 the dimension of QI becomes greater than the number of possible values of the temporal attribute. Since in existing approaches (e.g., [3, 5, 6]) the complexity of the anonymization algorithm is exponential in the size of the QI, and since the number of timestamps is usually large, this preprocessing phase simply makes the algorithms ineffective.

In order to avoid the problems described above, in this paper we propose a slight revision of the main definitions of k-anonymity allowing anonymization techniques to be applied even on relations having multiple tuples associated to the same respondent. While this extension is very much needed to deal with databases with timestamped data, it is also quite convenient for general databases. Our approach is a proper extension of the formalism proposed in [3].

The second part of the paper investigates anonymization techniques considering in particular the case when the temporal attribute is part of QI. After showing that known generalization algorithms, despite being applicable under certain conditions, have limitations regarding the anonymization of the temporal component, we propose a new generalization algorithm based on the hierarchy of time granularities.

The main contributions of this paper are the following:
(i) The original definition of k-anonymity of the view of a database relation is extended to relax the condition that a relation has only a single tuple for each respondent;
(ii) We provide a new generalization algorithm specific for achieving k-anonymity by generalizing the temporal attribute. We show that the algorithm computes the least k-anonymous generalization accordingly to a specific metric. Formal properties of the algorithm are investigated including a time complexity analysis.

The rest of the paper is organized as follows. In Section 2 we provide some formal notions, including the revised definition of a k-anonymous relation. In Section 3 we investigate anonymization techniques, and propose in Section 4 a new algorithm. Section 5 concludes the paper.

[1] This approach has some analogies with the attribute timestamped temporal database model [7].

2. Extended notion of k-anonymity relations

In the following we denote with R the database relation containing the data to be anonymized.

We call *UID* a single hidden attribute of *R* that uniquely identifies the respondent of each tuple, i.e., the user to whom the tuple values refer to.

In [5] the UID attribute, if present, is removed at the beginning of the generalization process; in our approach, the attribute is used during the generalization process and is suppressed just before the data is released. Therefore, in this paper, when not differently stated, we assume that R is a database relation whose schema is composed by the attribute UID, a set $QI = \{Q_1, \ldots, Q_n\}$ of attributes for the quasi-identifier and a set $DATA$ of other attributes. Note that the values of the DATA attributes are assumed not to be externally available in combination to any information on data recipients, otherwise the attributes should have been included in QI.

We define a function that, given a combination of values of QI, returns the set of respondents associated with tuples containing that combination.

Definition 1 *Given a relation R with D_U the domain of UID and D_Q the Cartesian product of the domains of $Q_1, \ldots, Q_n$, the function $u_R : D_Q \rightarrow 2^{D_U}$ is defined as*

$$u_R(\langle q_1, \ldots, q_n \rangle) = \pi_{UID}(\sigma_{(Q_1=q_1 \wedge \ldots \wedge Q_n=q_n)}(R)).$$

Note that in this paper we use the relation algebra definition of the projection (π) operation, whose result does not contain duplicate tuples. The notion of frequency set (analogous to the one defined in [3]) can now be easily formalized.

Definition 2 *Let t be a tuple of R and $\langle q_1, \ldots, q_n \rangle = \pi_{QI}(t)$. The frequency set of $\langle q_1, \ldots, q_n \rangle$ in R is defined as $f_R(\langle q_1, \ldots, q_n \rangle) = |u_R(\langle q_1, \ldots, q_n \rangle)|$.*

In the following, when no confusion arises, we use $u()$ and $f()$ without indicating the database relation R.

Table 3. A relation that is not 2-anonymous.

UID	QI		data
	Q	T	
u_1	q_1	2006-01-03	d_0
u_2	q_1	2006-01-03	d_1
u_1	q_1	2006-01-11	d_2
u_4	q_1	2006-01-12	d_3
u_5	q_2	2006-02-07	d_4
u_6	q_2	2006-02-10	d_5

In SQL, the frequency set can be obtained with the query:
SELECT COUNT (DISTINCT UID) FROM R
WHERE $Q_1 = q_1$ AND ... AND $Q_n = q_n$.

Definition 3 *Relation R is said to be k-anonymous if, for each tuple t of R, $f_R(\pi_{QI}(t)) \geq k$.*

Example 1 *Consider the relation represented in Table 3. The relation is not 2-anonymous. Indeed, consider the third tuple and its projection on the QI: $\langle q_1, 2006\text{-}01\text{-}11\rangle$. Since it is the only tuple with this value for the QI, $f(\langle q_1, 2006\text{-}01\text{-}11\rangle) = 1$.*

Definition 4 *Given a relation R, a view V_f is called a k-anonymization of R if there exists a view V_a of R such that: (i)V_a is k-anonymous, (ii) V_a is obtained from R by suppressing tuples or suppressing, modifying or distorting data in any attributes but UID, and (iii) $V_f = \pi_S(V_a)$ where $S = Att(V_a) \setminus \{UID\}$.*

In the definition above, V_a is a k-anonymous view of R; it should not be released because it contains the UID. On the contrary, V_f does not contains the UID and can therefore be revealed; note that V_f is intuitively k-anonymous since it is a projection of V_a, but, according to Definition 3, k-anonymity cannot be checked since the absence of the attribute UID does not allow to apply the function $f()$.

3. Anonymization Techniques

In the following of this paper we focus on the techniques that can be used to enforce k-anonymity assuming that the time attribute is part of the QI, i.e., $QI = \{Q_1, \ldots, Q_n, T\}$. We first show that, although we relax the assumption about a single tuple for each respondent, existing k-anonymization strategies can still be used in our approach. However, in Subsection 3.2 we observe that the existing algorithms make an assumption that constrains the number of temporal granularities that can be used to express a temporal attribute. Since a larger set of granularities can lead to more desirable generalization we propose an alternative anonymization strategy, assuming that the time attribute of each relation can be expressed in terms of a temporal granularity chosen from an arbitrarily rich calendar.

3.1. Adopting known generalization algorithms

Several anonymization techniques have been proposed in the literature ([4, 5, 6, 3]). Among others, LeFavre et. al. ([3]) defined an algorithm that computes all the possible k-anonymous generalizations of a given relation. A *domain generalization relationship* ($\leq_D$) is defined between the domains of the attributes of QI: intuitively, $D_i \leq_D D_j$ indicates that the values in the domain D_j are the generalization of the values in the domain D_i. The algorithm proposed in [3] is based on three properties and one assumption about the domain generalization relationship. The properties intuitively state that: (i) if a relation is k-anonymous, then all its generalizations are k-anonymous (Generalization property); (ii) if a view V is generalized into a view V', then the frequency set function of V' can be computed from the frequency set function of V (Rollup Property) and (iii) if a view is k-anonymous with respect to a quasi-identifier QI, then it is also k-anonymous with respect to any set of attributes $P \subset QI$ (Subset property). In the formulation of the algorithm, it is also assumed that the domain generalization relationship is a total order.

When temporal information is used in a database relation, it is generally represented in a date format like `2005-12-25` or `2005-12-25 15:35`. A standard date format generalization of these values consists of dropping characters at the right of the string, obtaining, for example, `2005-12`.

Adopting a standard date format generalization, the algorithm presented in [3] can also be used in our approach for the following three reasons: (i) the three properties and the assumption about the domain generalization relationship are verified; (ii) relaxing the assumption about a single tuple for each respondent only affects the definition of frequency set, and (iii) the algorithm proposed in [3] does not deal with the computation of the frequency set.

3.2. Using granularities to generalize temporal data

A date format generalization as the one presented above is necessarily based on a total order of time granularities, and it is usually limited to the most common ones (e.g., year, month, day, hour). Hence, it may sometimes lead to timestamps that are being generalized too much, possibly making data unusable.

A different approach consists in changing the time attribute in order to reflect the fact that its values denote granules of a specific time granularity. A larger set of granularities can help enhancing the tradeoff between generality (to guarantee k-anonymity) and specificity (to guarantee minimality). For example, a generalization of `days` in terms of `weeks` could be preferred to a generalization in terms of `months` if k-anonymity can be guaranteed with both of them. By adopting the framework proposed in [2], the set of granularities to be used in the generalization can be arbitrarily defined, taking into account the specific domain of data. For example, it is easy to define a granularity G that partitions the hours of the day into morning, afternoon, and night. Standard granularity operations could be used to present the result in a user-friendly way.

Using granularities to generalize the values of the temporal attribute requires a different algorithm with respect to the one presented in [3]. Indeed, we show that, if granularities are used to perform the generalization, the three properties and/or the total order assumption are not always guaranteed.

In the following we need to go into some technical details involving granularities in order to illustrate the specific generalization process. We follow the formal notions defined in the consensus glossary [1]. The definitions essential to the comprehension of this paper are reported in Appendix A. We indicate with G the granularity used to express the temporal attribute of R and with $\mathcal{G}$ the set of granularities that can be used to express the temporal attribute of R. We also denote with R^H the *generalization* of R that is obtained using H to express the temporal attribute of R. If we denote with $G(i)$ the granule of G with index i, relation R^H can be obtained by computing, for each index i appearing in the temporal attribute of R, the index j of the granule of H that covers $G(i)$. This computation can be performed with the $\lceil\cdot\rceil$ operation: $j = \lceil i \rceil_G^H$.

Example 2 *Consider the relation R presented in Table 3. The temporal attribute is expressed in terms of `day`. We can compute $R^{\texttt{week}}$ by applying the $\lceil i \rceil_{\texttt{day}}^{\texttt{week}}$ operation to each value i of the temporal attribute. Table 4 shows the relation $R^{\texttt{week}}$. Note that, differently from R, relation $R^{\texttt{week}}$ is 2-anonymous.*

The $\lceil\cdot\rceil_G^H$ operation is not always defined. For example, $\lceil i \rceil_{\texttt{week}}^{\texttt{month}}$ is not defined if week i starts in a month and ends in the following. Hence if i is a timestamp value in R and $\lceil i \rceil_G^H$ is not defined, then a generalization R^H is not feasible. In order to avoid this situation we assume that for each $H \in \mathcal{G}$, G is *finer* than H ($G \preceq H$). This guarantees the function $\lceil i \rceil_G^H$ is always defined. In practice, the generalization process disregards any granularity in $\mathcal{G}$ not satisfying this condition.

Note that the $\preceq$ relation has the same role that the $\leq_D$ relation has in the approach proposed in [3]. Indeed, if we impose that the $\preceq$ relation is a total order in $\mathcal{G}$, then all the three properties and the assumption of [3] are verified, and hence that anonymization strategy can be used. However, in general, $\preceq$ is not a total order. For example, if we want to generalize a temporal attribute expressed in terms of `day` we may want to use the granularities `week` and `month`. However, `week` $\not\preceq$ `month` and `month` $\not\preceq$ `week`. Hence, if we impose that that the $\preceq$ relation is a total order in $\mathcal{G}$, then we cannot have both `week` and `month` in $\mathcal{G}$.

Table 4. Generalization of the temporal attribute of Table 3 in terms of `week`

UID	QI		data
	Q	T	
u_1	q_1	2006-week 1	d_0
u_2	q_1	2006-week 1	d_1
u_1	q_1	2006-week 2	d_2
u_4	q_1	2006-week 2	d_3
u_5	q_2	2006-week 6	d_4
u_6	q_2	2006-week 6	d_5

The following result shows that, with the additional assumption that the granularities in $\mathcal{G}$ have the same *image* of G, if the $\preceq$ relation is not a total order in $\mathcal{G}$ then no total order exists preserving the Generalization Property. Note that this is a reasonable assumption; Indeed H *covers* G follows from $G \preceq H$ and G covers H may be desirable since, intuitively, if G does not cover H, then H generalizes more than required without enhancing k-anonymity. We have a more involved version of Theorem 1 that also admits the case that G does not cover H.

Theorem 1 *If the $\preceq$ relation is not a total order in $\mathcal{G}$, then there does not exist a relation $\leq_D$ that is a total order in $\mathcal{G}$ and such that for each relation R and granularities $H, H' \in \mathcal{G}$ such that $R_H \leq_D R'_H$, R^H is k-anonymous implies $R^{H'}$ is k-anonymous.*

From Theorem 1 follows that the strategy proposed in [3] cannot be applied, since it requires that $\leq_D$ is a total order such that the Generalization property holds.

4. An algorithm for time-based anonymization

In this section we propose an algorithm for the anonymization of the private table based on the time attribute only. Such an algorithm may be useful in many cases, based on the following considerations:

- the QI is generally composed by more than one attribute, but several researchers have pointed out that a

uniform generalization of all the attributes in the QI may not be desirable.

- if we assume that QI is minimal (as assumed in [3]), then it is sufficient to generalize one of the attributes in order to obtain k-anonymity;
- the particular semantics associated with the time attribute makes it a candidate for being generalized in many application domains.

4.1. Least temporal generalization

Our goal is to define an algorithm that, given a relation R and a set of granularities $\mathcal{G}$, returns a granularity H such that R^H is k-anonymous. In general, there exists a set $S \subseteq \mathcal{G}$ of granularities such that for each $H \in S$, R^H is k-anonymous. In this case, the algorithm should return the granularity $H \in S$ such that R^H is the "least general" among all the $R^{H'}$ for each $H' \in S$. Therefore, we define a metric $gen()$ that is used to quantify the generalization of R; then, a total order relation $\leq_g$ is used to compare $gen(R^H)$ and $gen(R^{H'})$ for each pair of granularities $H, H' \in \mathcal{G}$.

Intuitively, the finer than relationship also defines an order on the generality of the relations expressed in terms of different granularities. Indeed, it is easily seen that if $H \preceq H'$ then R^H k-anonymous implies $R^{H'}$ k-anonymous. Hence, the following property on the $gen()$ metric is desirable.

Property 1 *Given a relation R, a metric function $gen()$, a total order $\leq_g$ on the co-domain of $gen()$, and granularities $H, H' \in \mathcal{G}$, if $H \preceq H'$, then $gen(R^H) \leq_g gen(R^{H'})$.*

A first candidate metric is the one returning the minimum value of the frequency set of R for each value of the quasi-identifier.

Definition 5 *Given a relation R, we define $gen_m(R) = min_{(i \in \pi_{QI}(R))} f_R(i)$.*

Note that this is equivalent to say that $gen(R)$ is the maximum value $k \in \mathbb{N}$ such that R is k-anonymous. We prove that Property 1 holds using the metric $gen_m()$ and the order $\leq$ on the natural numbers.

Theorem 2 *Given a relation R and two temporal granularities H and H', if $H \preceq H'$ then $gen_m(R^H) \leq gen_m(R^{H'})$.*

Another desirable property for a metric is to preserve k-anonymity, as formally stated by Property 2.

Property 2 *Given two relations R and R', a metric function $gen()$, and a total order $\leq_g$ on the co-domains of $gen()$, if $gen(R) \leq_g gen(R')$ then, if R is k-anonymous, then R' is k-anonymous.*

It is easily seen that metric $gen_m()$ preserves k-anonymity; Indeed, Property 2 holds using the metric $gen_m()$ and the order $\leq$ on the natural numbers.

As illustrated by Example 3, this metric has some drawbacks. Since only the minimal value is considered, in many situations, we can have $gen_m(R^H) = gen_m(R^{H'})$ while, intuitively, R^H is less general then $R^{H'}$. To overcome this problem, a metric considering the average generalization of values with respect to the original relation can be introduced.

Definition 6 *Given a relation R, we define $gen_s(R) = \sum_{(t \in R)} f_R(\pi_{QI}(t))$.*

The metric returns the sum, for each tuple t in the (possibly generalized) relation R, of the frequency set function of the QI of t. While in principle the metric should divide the sum by the number of different timestamps in the original relation, this amount is constant for all generalizations and can be ignored.

We prove that Property 1 holds using the metric $gen_s()$ and the order $\geq$ on the natural numbers.

Theorem 3 *Given a relation R and two temporal granularities H and H', if $H \preceq H'$ then $gen_s(R^H) \geq gen_s(R^{H'})$.*

Note that Property 2 does not hold using the metric $gen_s()$ and the order $\geq$ on the natural numbers. However, the metric $gen_s()$ shows a higher discrimination power than $gen_m()$ while preserving the generalization intuition.

To exploit the benefits of both metrics it is possible to compose them.

Definition 7 *Given a relation R, we define $gen_c(R) = \langle gen_m(R), gen_s(R) \rangle$.*

The idea of the metric $gen_c()$ is that, when two relations R^H and $R^{H'}$ are compared, first $gen_m(R^H)$ is compared with $gen_m(R^{H'})$ and, if this metric cannot order the two relations, then the metric $gen_s()$ is applied.

To order the results of the $gen_c()$ metric, we define a total order $\leq_c$ on pairs of numbers such that $\langle a, b \rangle \leq_c \langle a', b' \rangle$ if $(a < a') \vee (a = a' \wedge b \geq b')$.

Theorem 4 *Let R be a relation, H, H' be two granularities. If $H \preceq H'$, then $gen_c(R^H) \leq_c gen_c(R^{H'})$.*

Example 3 *Consider relation R and $R^{\texttt{week}}$ shown in Tables 3 and 4, respectively. It is possible to generalize R with the relation $R^{\texttt{month}}$ as shown in Table 5.*

Note that $R^{\texttt{month}}$ is 2-anonymous. Since both $R^{\texttt{week}}$ and $R^{\texttt{month}}$ are 2-anonymous, and `month` and `week` are not ordered by the $\preceq$ relation, a metric should be applied. Applying $gen_m()$ we obtain $gen_m(R^{\texttt{week}}) = gen_m(R^{\texttt{month}}) = 2$. On the contrary, if we apply the

UID	QI		data
	Q	T	
u_1	q_1	2006-month 1	d_0
u_2	q_1	2006-month 1	d_1
u_1	q_1	2006-month 1	d_2
u_4	q_1	2006-month 1	d_3
u_5	q_2	2006-month 2	d_4
u_6	q_2	2006-month 2	d_5

Table 5. Generalization of the temporal attribute of Table 3 in terms of month

$gen_s()$ metric, we obtain $gen_s(R^{week}) = 2 + 2 + 2 = 6$ and $gen_s(R^{month}) = 3 + 2 = 5$ identifying R^{week} as "less general" than R^{month} which is intuitively correct. It can also be easily seen that $gen_c(R^{week}) \leq_c gen_c(R^{month})$.

Note that the order $gen_c(R^{week}) \leq_c gen_c(R^{month})$ depends on the values of R. Indeed, it is possible that, for a certain R, $gen_c(R^{month}) \leq_c gen_c(R^{week})$. Intuitively, this happens when most of the values of the temporal attribute of R are in the same week but in different months.

Based on a metric it is possible to define which relations provide a least generalization while guaranteeing k-anonymity.

Definition 8 *Given a relation R, a metric function $gen()$ with a total order $\leq_g$ on its co-domain, a set $\mathcal{G}$ of temporal granularities, an integer k, and $H \in \mathcal{G}$, we say that R^H is a* least k-anonymous generalization of R with respect to $\mathcal{G}$ *if R^H is k-anonymous and $gen(R^H) \leq_g gen(R^{H'})$ for each $H' \in \mathcal{G}$ such that $R^{H'}$ is k-anonymous.*

Note that, depending on the metric, it is possible that the least k-anonymous generalization of R is not unique. This can happen with all the three metrics considered above.

In the rest of the paper we use the metric $gen_c()$, and we apply the total order $\leq_c$. However, our results are independent from this choice.

4.2. Time-based anonymization algorithm

Our goal is to design an algorithm that computes $G' \in \mathcal{G}$ such that G' is a least k-anonymous generalization of R with respect to $\mathcal{G}$.

A straightforward strategy to solve the problem consists in computing, for each granularity $H \in \mathcal{G}$, the metric $gen_c(R^H)$ after checking the k-anonymity of R^H, hence identifying a granularity G' such that $R^{G'}$ is a least k-anonymization of R with respect to $\mathcal{G}$.

However, a smarter strategy can be adopted to strongly improve the performance of the algorithm: it is possible to exploit Property 1 to avoid considering granularities $H' \in \mathcal{G}$ if a granularity $H \in \mathcal{G}$ with $H \preceq H'$ has been considered, and R^H is k-anonymous. Algorithm 1 presents this strategy. The main body of the algorithm simply initializes the values for the recursive Procedure 1. The metric $gen_c()$ and k-anonymity are computed by Procedure 2.

Algorithm 1 leastGeneralization

- **Input**: a relation R, whose schema is composed by an attribute UID, a set of attributes $DATA$ and a set of attributes $QI = \{Q_1, \ldots, Q_n, T\}$ where T is a temporal attribute expressed in terms of a granularity G; a finite set $\mathcal{G}$ of granularities such that $\nexists H, H' \in \mathcal{G}$ s.t. H is shift equivalent to H' and $\forall H \in \mathcal{G}$, $G \preceq H$; an integer value $k > 1$.
- **Output**: $G' \in \mathcal{G}$, if exists, s.t. $R^{G'}$ is a least k-anonymous generalization of R with respect to $\mathcal{G}$; **null** otherwise.
- **Method**:

1: $G' =$ **null**;
2: $minimum =$ **null**;
3: A = select $UID, Q_1, \ldots Q_n, T$ from R order by $Q_1, \ldots, Q_n, T$;
4: leastGenRec(A, G, $\mathcal{G}$, k, $minimum$, G');
5: **return** G';

Since the algorithm exploits the $\preceq$ relation to prune some granularities, it must consider the granularities of $\mathcal{G}$ in the partial order imposed by the $\preceq$ relation. It is possible to represent the granularities and the $\preceq$ relation by a directed graph where the nodes are the granularities, and there is an edge from H to H' if and only if $H \preceq H'$ and $\nexists H'' \in \mathcal{G} | H \preceq H'' \preceq H'$. If there is an edge from H to H', we say that *H' is directly coarser than H*.

In Algorithm 1 we assume that there are no granularities in $\mathcal{G}$ that are *shift equivalent* (see Appendix A) ; thanks to this assumption, there are no $H, H' \in \mathcal{G}$ such that $H \preceq H'$ and $H' \preceq H$, therefore the graph is acyclic. Note that, since the $\preceq$ relation does not depend on R, the graph can be precomputed once for each set $\mathcal{G}$.

Procedure 2 shows how to compute k-anonymity and the $gen_c()$ metric for the relation A^H. Note that relation A^H is not explicitly constructed but the temporal values are computed on-the-fly by applying the $\lceil \cdot \rceil_G^H$ operation.

The central idea of this procedure is that, since the tuples of A are ordered according to the QI, then the tuples of A^H are ordered according to the same attributes. Indeed, the values of the attributes $Q_1, \ldots, Q_n$ are the same in A and A^H and the order of the temporal attribute is guaranteed by

Procedure 1 leastGenRec

- **Input**: a relation A, whose schema is composed by an attribute UID and a set of attributes $QI = \{Q_1, \ldots, Q_n, T\}$ where T is a temporal attribute expressed in terms of a granularity G; a finite set $\mathcal{G}$ of granularities such that $\nexists H, H' \in \mathcal{G}$ s.t. H is shift equivalent to H' and $\forall H \in \mathcal{G}$, $G \preceq H$; granularities $H, G' \in \mathcal{G}$; a value $minimum$; an integer value $k > 1$.
- **Output**: the output is returned by possibly updating the value of the variables G' and $minimum$ given in input.
- **Method**:

1: $n = gen(A, H, k)$;
2: **if** ($n \neq$ **null**) **then**
3: **if** ($minimum \neq$ **null** $\vee\ n \leq_c minimum$) **then**
4: $G' = H$;
5: $minimum = n$;
6: **end if**
7: **else**
8: **for** (each $H' \in \mathcal{G}$ directly coarser than H) **do**
9: $\mathcal{G} = \mathcal{G} \setminus \{H'\}$
10: leastGenRec(A, H', $\mathcal{G}$, k, $minimum$, G')
11: **end for**
12: **end if**

the properties of the $\lceil \cdot \rceil_G^H$ operation: it can be easily proved that if $i < j$ are indexes of the granularity G, then $\lceil i \rceil_G^H \leq \lceil j \rceil_G^H$. Since in A^H the tuples with the same values for the QI are contiguous, then it is possible to compute the k-anonymity of A^H and the value of $gen_c(A^H)$ by processing the tuples in order and checking when the value of the QI changes.

The correctness of Algorithm 1 is stated by the following result.

Theorem 5 *The Algorithm leastGeneralization eventually terminates and the result is a granularity $G' \in \mathcal{G}$ such that $R^{G'}$ is a least k-anonymous generalization of R with respect to $\mathcal{G}$.*

An accurate analysis of the complexity of the algorithm leads to the following result.

Theorem 6 *The worst case time complexity of Algorithm 1 is $O(n \cdot |R| \cdot log_2|R| + |\mathcal{G}| \cdot |R| \cdot (n + u))$ where n is the number of attributes in the QI and $u = |\pi_{UID}(R)|$.*

Intuitively, time $O(n \cdot |R| \cdot log_2|R|)$ is required to order relation R according to the n attributes of the QI and time $O(|R| \cdot (n + u))$ is required to execute the gen procedure that, in the worst case, is executed $|\mathcal{G}|$ times.

Procedure 2 gen

- **Input**: a view A, whose schema is composed by an attribute UID and a set of attributes $QI = \{Q_1, \ldots, Q_n, T\}$ where T is a temporal attribute expressed in terms of a granularity G and that is ordered by $Q_1, \ldots, Q_n, T$; a granularity H s.t. $G \preceq H$; an integer value $k > 1$.
- **Output**: the value of $gen_c(R^H)$ if R^H is k-anonymous, **null** otherwise.
- **Method**:

1: t = first tuple of A;
2: $sum = 0$;
3: $U = \emptyset$;
4: $m = |\pi_{UID}(R)|$;
5: t' = next tuple after t;
6: **while** ($t' \neq$ **null**) **do**
7: **if** ($\pi_{Q_1,\ldots,Q_n}(t) == \pi_{Q_1,\ldots,Q_n}(t') \wedge \lceil \pi_T(t) \rceil_G^H == \lceil \pi_T(t') \rceil_G^H$) **then**
8: $U = U \cup \{\pi_{UID}(t')\}$;
9: **else if** ($|U| \geq k$) **then**
10: $m = min(m, |U|)$;
11: $sum = sum + |U|$;
12: $U = \{\pi_{UID}(t')\}$;
13: **else**
14: **return null**;
15: **end if**
16: $t = t'$;
17: t' = next tuple after t;
18: **end while**
19: **if** ($|U| \geq k$) **then**
20: $m = min(m, |U|)$;
21: $sum = sum + |U|$;
22: **return** $\langle m, sum \rangle$;
23: **else**
24: **return null**;
25: **end if**

If an index is used to keep R ordered according to the attributes of the quasi-identifier, then the worst case time complexity is linear in the cardinality of R times the cardinality of $\mathcal{G}$.

5. Conclusions

In this paper we considered the problem of the anonymization of database relations with time-dependent tuples. We showed that current approaches cannot be straightforwardly applied to this type of databases and proposed an extension to the notion of k-anonymity to ac-

commodate this problem. We also propose a more effective generalization algorithm specifically designed for the anonymization of the temporal component. While outside the scope of this paper, our generalization technique may be integrated in algorithms for the concurrent generalization of the values of all the attributes in QI. These algorithms have the goal of obtaining domain-dependent least generalizations by tuning priorities and possibly threshold values for the generalization of the attributes in QI.

References

[1] C. Bettini, C. Dyreson, W. Evans, R. Snodgrass, and X. Wang. A glossary of time granularity concepts. In *Temporal Databases: Research and Practice*, volume 1399 of *Lecture Notes in Computer Science*, pages 406–413. Springer, 1998.

[2] S. X. W. Claudio Bettini, Sushil Jajodia. *Time Granularities in Databases, Data Mining, and Temporal Reasoning.* Springer-Verlag, 2000.

[3] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Incognito: efficient full-domain k-anonymity. In *SIGMOD '05: Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 49–60. ACM Press, 2005.

[4] A. Meyerson and R. Williams. On the complexity of optimal k-anonymity. In *Proceedings of the Twenty-third ACM Symposium on Principles of Database Systems*, pages 223–228. ACM Press, 2004.

[5] P. Samarati. Protecting respondents' identities in microdata release. *IEEE Trans. Knowl. Data Eng.*, 13(6):1010–1027, 2001.

[6] L. Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):571–588, 2002.

[7] A. Tansel, J. Clifford, S. Gadia, S. Jajodia, A. Segev, and R. Snodgrass. *Temporal Databases: Theory, Design, and Implementation*. Benjamin/Cummings, 1993.

A. Time granularities and temporal relations

A comprehensive formal study of time granularities and their relationships can be found in [2]. In this paper, for lack of space we only introduce notions that are essential to show our results.

Granularities are defined by grouping sets of instants into *granules*. For example, each granule of the granularity `day` specifies the set of instants included in a particular day. The whole set of time instants is called *time domain*, and for the purpose of this paper, the domain can be an arbitrary infinite set with a total order relationship, $\leq$.

Definition 9 *A granularity is a mapping G from the integers (the index set) to subsets of the time domain such that: (1) if $i < j$ and $G(i)$ and $G(j)$ are non-empty, then each element of $G(i)$ is less than all elements of $G(j)$, and (2) if $i < k < j$ and $G(i)$ and $G(j)$ are non-empty, then $G(k)$ is non-empty.*

The first condition states that granules in a granularity do not overlap and that their index order is the same as their time domain order. The second condition states that the subset of the index set that maps to non-empty subsets of the time domain is contiguous.

Each granule is identified by an integer. if we assume to map `day`(1) to the subset of the time domain corresponding to January 1, 2001, `day`(32) would be mapped to February 1, 2001, `b-day`(6) to January 8, 2001 (the sixth business day), and `month`(15) to March 2002. Independently, there may be a "textual representation" of each non-empty granule, termed its *label*, that is used for input and output. This representation is generally a string that is more descriptive than the granule's index. An associated mapping defines for each label a unique corresponding index. This mapping can be quite complex, dealing with different languages and character sets, or can be omitted if integers are used directly to refer to granules. For example, "August 2006" and "September 2006" are two labels each referring to the set of time instants (a granule) corresponding to that month.

The *image* of a granularity is the union of the granules in the granularity.

A granularity G *covers* a granularity H if the image of H is contained in the image of G.

Two granularities H and G are *shift equivalent* if there exists an integer j such that $G(i) = H(i + j)$ for every i in the index set.

If G and H are granularities, then G is said to be *finer than* H, denoted $G \preceq H$, if for each non-empty granule $G(i)$, there exists a granule $H(j)$ such that $G(i) \subseteq H(j)$.

When dealing with granularities, we often need to determine the granule (if any) of a granularity H that covers a given granule of another granularity G. For example, we may wish to find the month that includes a given week. This transformation is obtained with the *up* operation. Formally, for each index z, $\lceil z \rceil_G^H$ is undefined if $\nexists z'$ s.t. $G(z) \subseteq H(z')$; otherwise, $\lceil z \rceil_G^H = z'$, where z' is the unique index value such that $G(z) \subseteq H(z')$. The uniqueness of z' is guaranteed by the monotonicity of granularities.