Open Access

Open Journal of Information Systems (OJIS)
Volume 1, Issue 1, 2014

http://www.ronpub.com/ojis
ISSN 2198-9281

Pattern-sensitive Time-series Anonymization and its Application to Energy-Consumption Data

Stephan Kessler, Erik Buchmann, Thorben Burghardt, Klemens Böhm

Institute for Program Structures and Data Organization, Karlsruhe Institute of Technology (KIT),
Am Fasanengarten 5, 76131 Karlsruhe, Germany,
{stephan.kessler,erik.buchmann,thorben.burghardt,klemens.boehm}@kit.edu

ABSTRACT

Time series anonymization is an important problem. One prominent example of time series are energy consumption records, which might reveal details of the daily routine of a household. Existing privacy approaches for time series, e.g., from the field of trajectory anonymization, assume that every single value of a time series contains sensitive information and reduce the data quality very much. In contrast, we consider time series where it is combinations of tuples that represent personal information. We propose (n, l, k)-anonymity, geared to anonymization of time-series data with minimal information loss, assuming that an adversary may learn a few data points. We propose several heuristics to obtain (n, l, k)-anonymity, and we evaluate our approach both with synthetic and real data. Our experiments confirm that it is sufficient to modify time series only moderately in order to fulfill meaningful privacy requirements.

TYPE OF PAPER AND KEYWORDS

Regular research paper: *privacy, time-series, anonymity, smart-meter, smart grid*

1 INTRODUCTION

The anonymization of time series is an important concern. Time series such as GPS trajectories, energy consumption data or records of physical activities reveal many personal details about an individual. In many situations, such data should be published, e.g., to give way to scientific insights or to foster innovations. For example, effective regulation of energy production and consumption will only be possible if energy-consumption time series of households are available to the parties involved [32]. Thus, there is an antagonism between privacy concerns on the one hand and the need to publish time series data on the other hand.

In a nutshell, time-series data tends to be either what we call point-sensitive or pattern-sensitive. In point-sensitive time series, every single (time, data)-point might reveal sensitive information. For example, each (time, position)-tuple in a GPS track may reveal where an individual lives, works, etc. Existing privacy measures [30, 10, 23] and privacy-enhancing technolog -ies [27] typically try to make sets of point-sensitive time series indistinguishable as a whole, e.g., by computing their averages. This causes a severe loss of information, e.g., when the values averaged are dissimilar.

This paper studies pattern-sensitive time series where combinations of (time, data)-tuples represent personal information. An example is energy-consumption data. Figure 1 indicates that the daily routine and the appliances used in the household can be inferred from patterns contained in such data. It requires several values

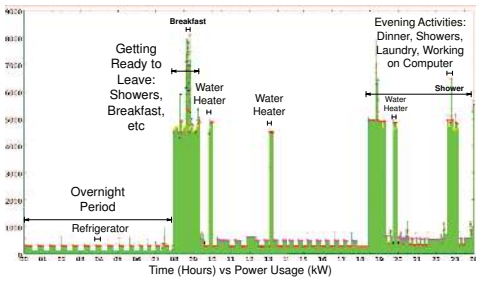


Figure 1: Example of Smart Meter data, reprinted from [26] with author permission

to detect a certain pattern, e.g., the switching period of the thermostat of a water heater. However, knowing the consumption value at one specific point of time typically is not informative and does not violate the privacy. For pattern-sensitive time series, a sufficient degree of privacy might be obtainable without making time series entirely indistinguishable. In particular, it might be acceptable to expose a few data points if the information loss caused by the anonymization is much smaller. The distinction between point-sensitive and pattern-sensitive is not always clear-cut. For example, GPS trajectories of commuters might be pattern-sensitive and point-sensitive at the same time. This is because trajectories reveal the commuting route (pattern) as well as the places of work and living (points). Still, there usually is a tendency towards one category, and studying the implications of this differentiation on information exposure is worthwhile.

Given a set of time series with patterns containing personal information, an adversary having certain limited amount of (time, value)-tuples as external knowledge of an individual may find out the values at other points of time. We refer to this as inference. This paper studies the relationship between the degree of anonymization and the number of data points that can be inferred for time series with patterns as personal details. Approaches for trajectory anonymization, e.g., [1] and [27], provide privacy guarantees under the worst case assumption of exhaustive external knowledge, i.e., an adversary knows entire time series. This is a theoretical limit – an adversary with such knowledge does not need to break any anonymization, since he already knows everything. Guarantees for this theoretical extreme case require to reduce the data quality very much.

In this paper, we investigate how to anonymize a database of time series with minimal information loss, assuming that an adversary knowing a limited number of (time, data)-tuples from a time series is allowed to learn a few tuples from the same time series that were unknown to her so far. However, an adversary must not learn the entire time series. This is challenging, for two reasons: (1) Anonymity is hard to obtain without making many households indistinguishable, e.g., by generalization, so that the data quality is low. Otherwise, a stakeholder with access to time series and external knowledge about a few tuples of a certain individual could single out candidate time series belonging to this individual. (2) Being anonymous does not necessarily prevent an adversary from gaining information about an individual. For example, it is sufficient to know that an individual is the originator of one element of a set of similar time series to learn further information. Since time series data are identifying and sensitive at the same time, it is not possible to use approaches for micro databases, e.g., [23].

In this paper, we use time series of smart meter data [11] as a prominent example of data containing sensitive patterns to motivate and evaluate our approach. We make the following contributions:

- We introduce (n, l, k)-anonymity, a privacy measure that allows to specify a degree of anonymity and an upper bound of the information exposed, given the extent of external knowledge of an adversary. To our knowledge, the idea of having such an upper bound for information exposure has not been investigated for time-series anonymization yet.
- We propose several heuristics that transform a set of time series so that it is (n, l, k)-anonymous. Our heuristics strives to minimize the information loss caused by the transformation. We propose and test three heuristics that differ regarding the way the data is modified.
- We evaluate our approach by extensive experiments both with real-world smart meter data and with synthetic data. Our evaluation with the real-world data shows that it is sufficient to modify each value by less than 10% on average to ensure that each time series is indistinguishable to a high degree. In other words, even though the indistinguishability is many times higher compared to the original data set, only slight modifications suffice.

Paper structure: Section 2 discusses the technical background and related work. Section 3 introduces (n, l, k)-anonymity, followed by our anonymization method in Section 4. Section 5 is our evaluation, and Section 6 concludes.

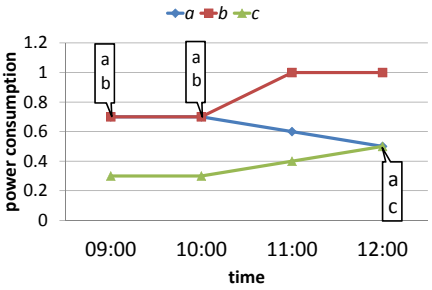
2 BACKGROUND

In this section, we briefly describe the smart grid and explain how it threatens the privacy of households. Furthermore, we review related privacy approaches.

2.1 The Smart Grid

The smart grid is an initiative to save energy, based on consumption forecasts, the optimization of energy consumption, fine-grained resource planning and seamless integration of decentralized energy sources. On the consumer side, the smart grid strives for flexible tariff models which motivate consumers to reduce peak loads and shift consumption to periods when more energy is available, e.g., from fluctuating renewable energy sources [14, 25]. Smart meters are an important part of the smart grid. They record energy consumption with a high resolution and transfer the readings automatically to a measuring point operator. Advanced smart meters measure energy consumption, active power, reactive power and other parameters [13] in small time intervals and are able to collect other data of the household in addition, e.g., water, gas or heat consumption. Furthermore, smart meters can communicate with other appliances as part of the "smart home" vision [11]. Numerous initiatives support the smart grid deployment: e.g. "European Smart Grid Technology Platform" [12] or the NIST Framework/Roadmap for smart grids [28]. In some countries (e.g., in Germany), the installation of smart meters is required by law for new or reconstructed buildings [4]. Thus, smart meters are relevant for large parts of society.

Example 1 (*Smart meter data set*): Let Alice (a), Bob (b) and Carol (c) be three persons/households with smart meters installed. Figure 2b contains the consumption data, the corresponding chart is in Fig. 2a. There are three time series (a, b and c) consisting of four tuples each. $(09{:}00, 0.7kwh)$ is a tuple that is part of two time series a and b. □



(a) graph

id	point of time	consumption
a	09:00	0.7 kwh
b	09:00	0.7 kwh
c	09:00	0.3 kwh
a	10:00	0.7 kwh
b	10:00	0.7 kwh
c	10:00	0.3 kwh
a	11:00	0.6 kwh
b	11:00	1.0 kwh
c	11:00	0.4 kwh
a	12:00	0.5 kwh
b	12:00	1.0 kwh
c	12:00	0.5 kwh

(b) tabular representation

Figure 2: example data

2.2 Privacy Threats

Since smart meters collect data with a high level of detail, the data measured allows to infer a lot of personal information, as follows.

Usage of electrical appliances: There are several proposals for the non-intrusive detection of electrical appliances present in a household and their usage periods [17, 26]. Figure 1 shows an example: It displays the power consumption of a household annotated with the detected appliances in use. Depending on the temporal resolution of the data, it is possible to identify the appliances used, e.g., oven, microwave or television [29]. With advanced smart meters, it is even possible to distinguish individual devices, e.g., different game consoles [19].

Personal details: Information on the usage times of appliances allows deep insight into the household's habits. Based on the amount of energy used during a specific time, it is possible to determine the daily routine, e.g., when residents take their breakfast, leave or return home [6]. An adversary can draw conclusions, e.g., if individuals are shift workers or go to church on Sundays. The daily power usage also gives evidence regarding the lifestyle, i.e., how many people live in a household, how long the individuals are at home, or if the households prepare meals in the oven or in the microwave.

Re-identification: Since energy consumption reflects many personal details of the households, smart meter data can be assumed to be inherently identifying. In particular, a set of values from a time series of smart meter data can be a *quasi-identifier* [30]. These values allow to assign the time series to an individual household. The process of linking anonymous data to an individual is called *re-identification*. Re-identification needs external knowledge on the power consumption of the household, as we will explain in section 3.2.

Note that the privacy threats described are a result of inferring information from several values of energy-consumption data, i.e., one consumption value at a specific time is neither sufficient to identify devices nor

habits. In other words, smart meter produce pattern-sensitive time series.

2.3 Privacy Approaches

In this section we discuss some recent privacy approaches for different use cases.

Relational anonymity criteria: k-anonymity [30] defines anonymity as being indistinguishable amongst $k-1$ other records, with respect to a quasi-identifier, in a relational data set. This principle and its improvements (l-diversity [23] and t-closeness [20]) usually discern the attributes as quasi-identifying and sensitive. Since this distinction is impossible for time series where the values are identifying and contain sensitive information at the same time, k-anonymity and its successors cannot be directly applied to time series.

Differential privacy: Differential privacy [10] is an approach for anonymizing query results, e.g., on trajectory data [7] or smart meter data [2]. The approach guarantees that a query result does not change much, if a record about a particular person is appended to the database. However, such strict privacy guarantees require total exante knowledge about all queries that are executed on the database. Furthermore, the approach perturbs the data set very much [22]. In contrast, we strive for an approach that reduces the amount of perturbation by taking sensitive patterns into account. Furthermore, we want to publish data without restricting the queries that are allowed on the data set.

Anonymity approaches for transaction data: Time series of transactions, e.g., from Internet shops or workflow systems, contain private information. A recent approach for anonymization of transaction data is ρ-uncertainty [5], which divides transaction data into sensitive and non-sensitive one and exploits the hierarchical structure of transactional data to generalize information, e.g., in a shopping cart scenario "diapers" $\rightarrow$ "baby goods". [33] extends this concept by considering an upper bound for the external knowledge for transaction data, but still distinguishes between sensitive and non-sensitive items. [31] does not depend on such a distinction. However, those approaches cannot be applied to our case, because time series do not have a hierarchical structure that can be exploited for anonymization.

Anonymity for moving object data: Moving object databases store trajectories, i.e., sequences of (time, position)-tuples. Privacy approaches for this kind of data, e.g., [9, 15, 18, 34, 27, 8], assume parts of the trajectory to be quasi-identifiers [35]. A popular approach is to transform trajectory sets into equivalence classes of at least k members [34]. [8] extends this concept by considering an upper bound for external knowledge. All approaches assume that parts of the trajectory can be clearly identified as quasi-identifiers for each individual, and this does not change over time, e.g., the path between the workplace and home. However, time series of smart meter data do not contain such "ideal" identifiers. Instead, identifying parts may be repeated at different points of time. [27] renders sets of trajectories indistinguishable to at least k others by using clustering, i.e., the approach assumes that each time series is a quasi-identifier as a whole covering the theoretical extreme case of an adversary having the complete time series as external knowledge. Since this assumption is undue for pattern-sensitive time series, the approach modifies such time series too much.

Anonymity for smart meter data: A recent approach [11] for smart meter privacy assumes that only consumption values measured with a high temporal resolution contain private information. This is motivated by the fact that it requires a high metering frequency to clearly identify electrical appliances (cf. Figure 1). The approach proposes an architecture where high resolution data is assigned to pseudonyms, while low resolution data is assigned to identifiers for, say, accounting. However, it is possible to map energy consumption data identified by pseudonyms to households, i.e., to break the anonymization. This is called re-identification [3]. It makes use of patterns in the energy consumption that are characteristic for a single household. Such patterns may appear in consumption data metered with any frequency. For example, vacation weeks can be as characteristic for a household as the morning routine. Thus, a separation in high- and low-resolution data is not general enough for our purpose.

Adversaries and external knowledge: Finally, the impact of aggregated external knowledge like "the average age of the individuals in a database is 48" on anonymization has been studied, e.g., in [21, 24]. However, none of the approaches we are aware of considers exact knowledge of some parts of the database or allows to specify an upper bound on information exposure suitable for a set of pattern-sensitive time series.

3 (n, l, k)-ANONYMITY

In this section, we introduce our terms and assumptions, we formalize our adversary model and we describe (n, l, k)-anonymity, our privacy measure for sets of pattern-sensitive time series. Intuitively, (n, l, k)-anonymity allows to specify a degree of anonymity, a limit on the information an adversary can learn about a household, and an upper bound on the external knowledge the adversary might possess. First, we state the following assumptions:

- The database contains a number of time series that is sufficient for anonymization. Intuitively, the (n, l, k)-anonymity parameters must not require each time series to be indistinguishable from more time series than contained in the database, and an adversary cannot know more values of a time series than the database actually contains.

- We assume that all values of the time series are equally sensitive. In other words, we consider the most general case where each value poses the same potential privacy threat.

- All smart meters measure the energy consumption at the same points of time and with fixed time intervals. While our approach can be extended for flexible points of time, we do not address this issue here.

3.1 Terms and Definitions

Let $\mathbb{T}$ be a set of points of time, e.g., from time series of power consumption values measured by a smart meter. $\mathbb{M}$ is the range of values measured. Thus, we model a time series as a set of (t, m) tuples where $t \in \mathbb{T}$ is a timestamp, and $m \in \mathbb{M}$ is the consumption value measured. Such a set contains exactly one tuple for each $t \in \mathbb{T}$. Thus, a time series implies a function f: $\mathbb{T} \rightarrow \mathbb{M}$. For a given $\mathbb{T}$ and $\mathbb{M}$, a database $DB = \{f_1, \cdots, f_n\}$ is a set of such functions. Each time series $f \in DB$ is assigned a random identifier, i.e., there is no direct relation between the time series and the households $\mathbb{H}$ that have produced the time series.

$\mathbb{V}_t^{DB}$ refers to the existing values in a data set at t: $m \in \mathbb{V}_t^{DB} \Leftrightarrow \exists f \in DB : f(t) = m$. Table 1 shows all symbols used. Our approach can be extended to multi-dimensional time series, e.g., smart meters measuring power, water and gas consumption. However, to ease presentation, we use a one-dimensional numerical range in this paper, i.e., $\mathbb{M} = \mathbb{R}$.

3.2 Adversary model

In our scenario, an adverary has access to the anonymized database DB', which is a copy of DB that has been modified using (n, l, k)-anonymity. Furthermore, the adversary knows a limited number of (t, m) tuples from a certain household $h \in \mathbb{H}$, which he knows as well (external knowledge $\mathbb{K}$). The objective of the adversary is to learn more tuples from the same household in order to observe patterns that reveal personal information. In the following, we formalize the notions of external knowledge and of an attack.

Definition 1 (external knowledge $\mathbb{K}$): External knowledge $\mathbb{K}$ is a set of tuples (t, m) (with all t pairwise different). □

Intuitively, $\mathbb{K}$ contains a limited number of tuples an adversary knows about a specific household $h \in \mathbb{H}$. It depends on the anonymization scheme if those tuples match tuples from none, one or multiple time series in the anonymized database DB'.

Example 2 (*External knowledge*): Suppose that an adversary has access to the data illustrated in Figure 2. His aim is to get additional information on a specific individual. The adversary only knows the content of the table. In particular, he does not know the mapping from random identifiers in DB' to households $\mathbb{H}$ .Without additional information, he cannot decide whether time series `a`, `b` or `c` belong to the household he is interested in. In the following we call these time series candidates, and an adversary cannot determine which one belongs to the household in question. Given the candidates a, b and c, he is uncertain regarding the consumption values at 11:00. On the other hand, if he knows that a specific household consumes 0.7 at 09:00 and 10:00, he can exclude household `c`. Uncertainty at 11:00 now only is between `a` and `b`. Finally, if an adversary knows the consumption value of `c` at 09:00, he learns the one from 10:00 as well. □

In the following, we assume that all tuples in $\mathbb{K}$ relate to the same household. Note that this is the most specific knowledge an adversary might have. Examples of less specific knowledge include cases where time series depend on each other, e.g., if an adversary possesses consumption values from several households and knows that these households have breakfast and lunch roughly at the same time. The objective of an adversary is to know at least l tuples from h in total to observe patterns in the time series, e.g., to identify breakfast time or the usage frequency of the microwave oven. We assume that it is sufficient for the adversary to learn that tuples belong to a specific household with a probability $P > 1/k$, i.e., exact knowledge is not required. Thus, we specify our adversary as follows:

Definition 2 (Adversary $\mathcal{A}_{\mathbb{K}}$): The adversary $\mathcal{A}_{\mathbb{K}}$ possesses $n = \|\mathbb{K}\|$ tuples (with $n < l$) from a specific household $h \in \mathbb{H}$, and he has access to the anonymized database DB'. The adversary wants to assign a set of at least $l - n$ tuples in addition to those n ones to h with probability $P > 1/k$. Formally, the adversary is successful if he learns a data set S:

Symbol	Description
DB	Data set of time series ($DB = \{f_1, \cdots, f_n\}$)
DB'	Anonymized data set of time series
$\mathbb{H}$	Set of households the smart meter data originates from
$\mathbb{K}$	External knowledge
n	Anonymization parameter for external knowledge
l	Upper bound for information exposure (includes knowledge)
k	Anonymity parameter for the size of indistinguishable sets
$\mathbb{I}_{DB}^{\mathbb{K}}(t)$	Set of indistinguishable time-series at t with external knowledge $\mathbb{K}$
$\mathbb{M}$	Set containing possible (power consumption) values, mostly range
(t, m)	referred as tuple if a time series f for which $f(t) = m$ applies exists
$\mathbb{T}$	Set containing points of time, mostly used as domain of time series
$\mathbb{V}_t^{DB}$	Set containing actual (power consumption) values at t

Table 1: List of frequently used abbreviations

$$\exists S \subseteq \{(t,m) | f \in DB' : f(t) = m\} : \\ \|S\| \geq (l-n) \wedge S \cap \mathbb{K} = \emptyset$$

And for S it holds that:

$$\forall (t,m) \in S : P((t,m) \text{ was generated by } h) > \frac{1}{k}$$

□

Note that we do not make any assumptions regarding the frequency or the appearance of sensitive patterns. Thus, we use a set of (t, m) tuples containing timestamps t and values m as a generalization of a pattern. According to our definition an attack is successful as soon as an attacker has uncovered a total of $l - n$ tuples different from the ones already known to him.

3.3 Anonymity in a data set of time series

For (n, l, k)-anonymity, we adapt the principle of k-anonymity for time series, i.e., we define anonymity as being indistinguishable amongst k individuals. Thus, we must prevent a tuple from being assigned to a specific household with a probability of more than $1/k$ (cf. Definition 2). External knowledge (see Def. 1) restricts the set of time series that may belong to the individual. For example, if $\mathbb{K} = \{(t_1, y_1), (t_2, y_2)\}$, the time series belonging to the individual must include both (t_1, y_1) and (t_2, y_2).

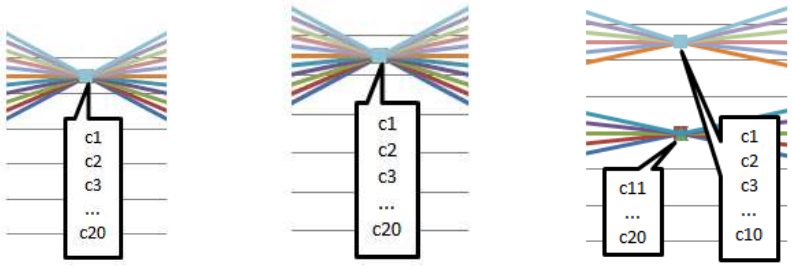
Definition 3 (candidate time series for $\mathbb{K}$ in data set DB: $CTS_{DB}(\mathbb{K})$): A candidate time series f for a given $\mathbb{K}$ is a time series with the following characteristic: For every tuple $(t, m) \in \mathbb{K}$, $f(t) = m$ holds. $CTS_{DB}(\mathbb{K})$ is the set of all candidate time series for $\mathbb{K}$ in a data set DB. □

Suppose that an adversary can constrain the time series to $CTS_{DB}(\mathbb{K})$. The fewer time series are in this set, the more delimiting is $\mathbb{K}$. We now determine the degree of indistinguishability/anonymity at points of time for which an attacker does not have any external knowledge. In Figure 3, the callout box points to the (time, data)-tuple contained in the time series listed. The figure illustrates that the set of candidate time series $CTS_{DB}(\mathbb{K})$ might or might not violate anonymity. In Figures 3a and 3b, the sets of candidates are different. But in both cases, an individual described by $\mathbb{K}_1$ or $\mathbb{K}_2$ respectively cannot be distinguished from 19 others. So the degree of anonymity is the same. In Figure 3c, the candidates have different values at the point of time in question. However, it is still impossible to distinguish the individual from 19 others. Thus, all three figures are equivalent in terms of anonymity. Based on this intuition, indistinguishability is the size of a set of time series at a specific point of time.

Definition 4 (Set of indistinguishable time series at point of time t for data set DB and external knowledge $\mathbb{K}$: $\mathbb{I}_{DB}^{\mathbb{K}}(t)$): $\mathbb{I}_{DB}^{\mathbb{K}}(t)$ includes all candidate time series as well as time series with the same value as a candidate at t. Formally, $\mathbb{I}_{DB}^{\mathbb{K}}(t) = \{f \in DB | \exists f' \in CTS_{DB}(\mathbb{K}) : f(t) = f'(t)\}$. □

The idea behind $\mathbb{I}_{DB}^{\mathbb{K}}(t)$ is that an adversary has an uncertainty between all time series assigned to tuples that also are assigned to a candidate time series. If time series have the same value at a point of time, one cannot distinguish them there. The following example illustrates this:

Example 3 (*Indistinguishability example*): Suppose an adversary knows the data set without personal identifiers from Figure 2b. Furthermore, he knows that the following tuple belongs to an individual: $\mathbb{K} = \{(11{:}00, 0.6)\}$. Since only for the time series a a value of 0.6 exists at



(a) $CTS_{DB}(\mathbb{K}_1) = \{c1, c2\}$ (b) $CTS_{DB}(\mathbb{K}_2) = \{c1, c2, c3\}$ (c) $CTS_{DB}(\mathbb{K}_3) = \{c1, c11\}$

Figure 3: Indistinguishability with different $CTS_{DB}(\mathbb{K}_i)$

11:00, the candidate time series is $CTS_{DB}(\mathbb{K}) = \{a\}$. Even if the adversary knows that a is the only possible pseudo-identifier, the number of indistinguishable time series at 10:00 and 12:00 is two: $\left\|\mathbb{I}^{\mathbb{K}}_{DB}(10{:}00)\right\| = \|\{a, b\}\| = 2$ respectively $\left\|\mathbb{I}^{\mathbb{K}}_{DB}(12{:}00)\right\| = \|\{a, c\}\| = 2$. □

The two principles behind that definition are as follows:

1. Privacy of an individual is better protected if several tuples may belong to the individual an adversary is interested in at a point of time.

2. The more time series are assigned to a tuple, the better the protection of privacy, since the less additional information is revealed to the adversary.

In the smart meter scenario, this can be illustrated as follows: Between $2am$ and $3am$, many time series have the same, low power-consumption value. This value is relatively "uninteresting" for an adversary because it is frequent at this point of time. Indistinguishability, which is related to the frequency of the value, is a characteristic of one certain point of time and is independent of other points. Thus, it is feasible to look at the indistinguishability of each point of time in isolation.

On the other hand, if the set of indistinguishable time series is small, one single point of time may reveal information to the adversary.

Definition 5 (Inferring point of time t): We say that t is inferred if the number of indistinguishable time series at t is below the user-defined k, i.e., $\left\|\mathbb{I}^{\mathbb{K}}_{DB}(t)\right\| < k$. □

Depending on the other user-defined parameters n and l, the adversary may infer certain values without breaching (n, l, k)-anonymity, as follows:

Definition 6 ((n, l, k)-anonymity): Let a data set DB and a number of n tuples of external knowledge $\mathbb{K}$ be given. DB is (n, l, k)-anonymous if there does not exist a set S' of at least $l - n$ points of time t where $\left\|\mathbb{I}^{\mathbb{K}}_{DB}(t)\right\| < k$. Therefore, a data set DB is (n, l, k)-anonymous if the following holds for an arbitrary set $\mathbb{K}$ of n tuples:

$$\nexists S' \subseteq \mathbb{T} : ||S'\| \geq l - n \wedge \forall t \in S' : \left\|\mathbb{I}^{\mathbb{K}}_{DB}(t)\right\| < k$$

□

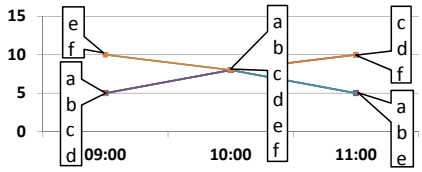


Figure 4: Example for a $(2, 3, 6)$-anonymous data set

Thus, an adversary $\mathcal{A}_{\mathbb{K}}$ cannot infer a set of at least $l - n$ tuples if he has access to the (n, l, k)-anonymous database DB' and n tuples of external knowledge $\mathbb{K}$. This is because the probability that the adversary can assign tuples from DB' to a household is greater than $\frac{1}{k}$ for less than $l - n$ tuples.

Example 4 ($(2, 3, 6)$-*anonymity*): Figure 4 shows a $(2, 3, 6)$-anonymous data set. For instance, let $\mathbb{K} = \{(09{:}00, 10), (10{:}00, 8)\}$. Thus, $CTS_{DB}(\mathbb{K}) = \{e, f\}$. The number of indistinguishable households at 11:00 is six. This is because both tuples $((11{:}00, 5)$ and $(11{:}00, 10))$ are possible. Thus, the probability to assign the tuples to a specific household is $\frac{1}{6}$. Since it is impossible for an adversary $\mathcal{A}_{\mathbb{K}}$ with external knowledge of any two tuples to infer a third one, the data set is $(2, 3, 6)$-anonymous. □

We will propose techniques for ensuring (n, l, k)-anonymity in Section 4.

3.4 Data quality in anonymized time series

First of all, we define the function $anon_{(DB,DB')}(f)$ to ease our presentation later on.

Definition 7 ($anon_{(DB,DB')}(f)$): Let DB be the original data set of time series and DB' the corresponding anonymized version of the data set. The function $anon_{(DB,DB')}(f)$ maps the original time series to the anonymized one. □

Generally speaking, anonymizing a set of time series means modifying their values, e.g., if $f(t) = m$ is the non-anonymized value, the value of the modified time series might be $anon_{(DB,DB')}(f) = m'$. Thus, DB' contains the modified time series $anon_{(DB,DB')}(f) \in DB'$ for each time series $f \in DB$. In line with other researchers, we assume that, the larger the difference between the original values and the anonymized ones, the more information is lost. For instance, modifying each time series $f \in DB$ to a time series of zeros ($anon_{(DB,DB')}(f)(t) = 0, \forall t \in \mathbb{T}$) suffices (n, l, k)-anonymity, but leads to a loss of any information. However, to keep the utility of the anonymized data, we need a measure as feedback for the anonymization method that quantifies the loss of information.

The Euclidean information loss is appropriate to this end. Intuitively, it is the sum of all differences between the original and the anonymized time series at each point of time. However, our approach does not depend on this particular measure.

Definition 8 (Euclidian information loss): Let DB be the original data set and DB' the modified set. The loss of information at point of time t is: $IL_t(DB, DB') = \sum_{f \in DB} \left| f(t) - anon_{(DB,DB')}(f)(t) \right|$ This means for the information loss of a complete data set: $IL(DB, DB') = \sum_{\forall t \in \mathbb{T}} IL_t(DB, DB')$ □

3.5 Privacy Protection in an (n, l, k)-Anonymous Data-Set

After having explained (n, l, k)-anonymity, in this section we describe its impact on the privacy protection of individuals in more detail. Consider again the threats described in Section 2.2. They all have in common that an adversary needs several values of consumption data to extract information. For example, detecting the usage of a specific electrical appliance requires a specific sequence of time-value tuples. Suppose that it is necessary to know s exact power consumption tuples to detect a given appliance. In an (n, l, k)-anonymous data set with $l - n \leq s$, an adversary can infer at most $l - n$ tuples. For the remaining $s - (l - n)$ tuples that would be necessary for the detection there is an uncertainty of $\frac{1}{k}$ (k time series are indistinguishable). This also holds for the re-identification threat and the extraction of information on personal habits.

The choice of the values of n,l and k is a tradeoff between data quality and privacy. A better privacy protection is achieved, the higher the n, the lower $l - n$ and the higher k.

Usually, it is assumed that an adversary has a small number of tuples as external knowledge, compared to the total number of points of time. Although this is a realistic assumption, it also eases the privacy protection. To investigate privacy protection in more detail, we discuss a worst-case scenario assuming unlimited external knowledge of an adversary in the following.

3.5.1 Worst-Case Scenario

Suppose that an adversary has unlimited external knowledge. In our scenario this means that $n = \|\mathbb{T}\| - 1$, thus l is set to $l = \|\mathbb{T}\|$. This is an extreme case: First, an adversary having almost the actual data set as external knowledge usually does not need to extract any information from an anonymized version of the data. Second, since we have limited the complete knowledge of an adversary in the assumptions to the size of the data set this is the largest possible set of external knowledge. The adversary achieves complete knowledge inferring the value of a single point of time. The results are clusters of size k, since arbitrary external knowledge is possible. This means that each household is indistinguishable amongst $k - 1$ others at each point of time.

The example shows that the indistinguishability required for privacy protection is independent of the external knowledge, and this differs from other approaches. For example, even in the worst case $k = 2$ is applicable. For other scenarios where $n << \|\mathbb{T}|$, the number of values actually exposed $(l - n)$ is also independent of k.

4 An Approach for (n, l, k)-Anonymization

In what follows, a *cluster* C^t is a set of time series such that all elements of the cluster have the same value at t. We refer to t as the point of time of cluster C^t.

In this section, we propose heuristics for the computationally efficient transformation of a set of time series so that the result is (n, l, k)-anonymous. Our heuristic is structured according to three observations, which we will explain subsequently: (1) If an algorithm modifies the data set for each point of time so that it contains only clusters of at least k time series, the data set is (n, l, k)-anonymous already (Lemma 1). (2) In some cases, even clusters of less than k time series do not allow an adversary to infer values (Example 4). Furthermore, (n, l, k)-anonymity allows to infer $l - n$ points of time. (3) Building clusters of less than k time series at one point of time might influence other clusters at a different point of time (Example 5).

To obtain anonymization, our heuristic modifies values of a set of time series. The modified set of time series consists of clusters where all members of a cluster have the same value, the mean of the original values. We use this as the cluster representative.

It is sufficient that all clusters at all points of time consist of at least k time series to guarantee (n, l, k)-anonymity. Lemma 1 acknowledges this by defining a lower bound on the number of indistinguishable time series.

Lemma 1: For any point of time t, the set of indistinguishable time series $\mathbb{I}^{\mathbb{K}}_{DB}(t)$ contains at least as many time series as the number of time series assigned to any $v \in \mathbb{V}^{DB}_t$. Formally, let $count(t, m) = \|\{f \in DB | f(t) = m\}\|$ be the number of time series having value m at t. For $f \in DB$, $\left\|\mathbb{I}^{\mathbb{K}}_{DB}(t)\right\| \geq min_{f \in DB}(count(t, f(t)))$ holds.

Proof of Lemma 1: Suppose that $\mathbb{K}$ gives way to one candidate time series f. Having only one candidate left is the minimal possible uncertainty an adversary can have. (In contrast, having zero candidates left would be the maximum uncertainty, since the adversary would not even know if the individual is in the data set.) To calculate the indistinguishability at point of time t we have to count the time series assigned to (t, m), with $f(t) = m$. Let $k_t = min_{m \in \mathbb{V}^{DB}_t}(count(t, m))$ be the minimal number of time series assigned to a value at t. Thus, the indistinguishability at a specific point of time will always be greater than or equal to k_t, irrespective of the candidate time series. □

The more time series have to be modified in order to create a cluster, the more the data set is perturbed. While creating clusters of at least k time series guarantees (n, l, k)-anonymity, it may also be feasible to create smaller clusters and still have (n, l, k)-anonymity, as seen in Example 4.

(n, l, k)-anonymity does not allow many clusters having fewer than k time series. Because the anonymization at a point of time t_1 depends on the anonymization at another point t_2 if it is possible to infer point of time t_2. The following example shows that creating clusters with less than k time series may require to create clusters with k or more elements at other points of time.

Example 5 (*Requiring clusters of k time series*): Reconsider the data in Fig. 4: A cluster consisting of e and f at 11:00 would break the $(2, 3, 6)$-anonymity: An adversary knowing $(09{:}00, 10)$ can infer the cluster $\{e, f\}$ at 11:00. □

Generally speaking, in order to (n, l, k)-anonymize a data set, we have to generate clusters of time series. For this purpose, we come up with a heuristic consisting of two stages: **Clustering** and **Splitting**. Clustering creates clusters with size $\geq k$ for each point of time in isolation. Splitting generates clusters of size $< k$, but has to consider that clusters from different points of time depend on each other. The order of these two stages is not fixed; in principle it could be either way. However, unless the (n, l, k)-anonymity parameters allow the adversary to infer large parts of the database, clusters of fewer than k time series will occur infrequently. Thus, our heuristic does "divide and conquer" and solves the coarse problem of computing clusters of k time series first, before creating clusters with less than k time series by splitting larger ones.

4.1 Stage 1: Clustering

The objective of this stage is to come up with clusters of at least k similar time series at each point of time. In order to identify similar time series, our heuristic clusters the values of all time series for each point of time in isolation (see. Def. 4). Recall that our approach can be used for time series of multi-dimensional values. If $\mathbb{M}$ is one-dimensional, other approaches to create clusters of k time series, e.g., discretization, are feasible and are simpler than clustering.

For each point of time, this stage starts with a single cluster consisting of all tuples. We use an approach similar to hierarchical divisive clustering [16] to split this cluster successively into smaller clusters of at least k tuples. In order to limit the loss of information, we split between the two original values with the highest difference. 6.1 shows our algorithm in pseudocode. In this stage, more data may be changed than necessary to fulfill (n, l, k)-anonymity, cf. Example 4.

4.2 Stage 2: Splitting

This stage splits the clusters from Stage 1 into clusters smaller than k. Splitting means dividing a cluster into two. The stage has to ensure that an adversary knowing n tuples of a time series cannot infer more than $l - n$ further data points. Thus, in this stage we consider time series at different points of time. The goal is to minimize the Euclidean information loss as a whole.

Intuitively, if a cluster is smaller than k, the number of inferable values from all time series might exceed the limit $l - n$. Thus, if a certain cluster at point of time t is split, another one at t' might be prohibited to split in order to not violate (n, l, k)-anonymity. The following example illustrates the difficulties of splitting.

Example 6 (*Alternative for $(2, 3, 6)$-anonymity*): Reconsider Example 4: This $(2, 3, 6)$-anonymous data set contains two clusters of three time-series at 09:00. If the same clusters were present at 10:00, the data set would not be $(2, 3, 6)$-anonymous anymore. However, Figure 5

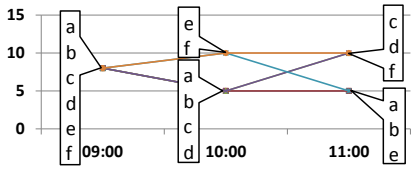


Figure 5: Alternative splitting example for a $(2, 3, 6)$-anonymous data set

shows another $(2, 3, 6)$ anonymous cluster configuration with two clusters at 10:00 and one single cluster at 09:00. If the clustering stage creates two clusters with all time-series at 09:00 and 10:00, the splitting stage can decide to split the clusters at 09:00 or 10:00 but not both. □

Definition 9 (Splitting of a cluster C^t between f and f'): Cluster C^t contains the neighbors f and f', with $f(t) > f'(t)$ in the original data set. "Neighbor" means that there is no time series between f and f': $\nexists h$: $f(t) > h(t) > f'(t)$. Then, the split between $f(t)$ and $f'(t)$ creates clusters C_1^t and C_2^t. C_1^t contains f and all time series $g \in C^t$ with $g(t) \geq f(t)$, while C_2^t contains all g with $g(t) \leq f'(t)$. Since f and f' are neighbors, C_1^t and C_2^t partition C^t. □

In the following, we present two heuristics for this optimization problem, *MostInformationLoss* and *MembersTimesHeight*. Both try to split as many clusters as possible.

4.2.1 MostInformationLoss (MIL)

The intuition behind MIL is to split clusters in the order of information loss, starting with the highest. Let DB' be the data set after Stage 1 has processed the original database DB. Therefore, DB' contains only clusters of at least k time series. MIL computes the information loss between the original database DB and the anonymized database DB' for each cluster C^t at each point of time t. However, a cluster is split only if the result still satisfies (n, l, k)-anonymity. 6.2.1 shows MIL in pseudocode.

4.2.2 MembersTimesHeight (MTH)

MTH uses two criteria to determine the order of cluster splits: The difference between the largest ($m_{max}^{C^t}$) and the smallest ($m_{min}^{C^t}$) value in the original data set of time series within a cluster C^t, and the number of time series $\|C^t\|$ assigned to that cluster. Thus, for each cluster C^t, MTH computes $Score(C^t) = (m_{max}^{C^t} - m_{min}^{C^t}) \cdot \|C^t\|$. The intuition is as follows: The more time series are assigned to a cluster, the more time series will probably be assigned to the splitted clusters, and the fewer time series have to be indistinguishable at other points of time. For instance, reconsider Figure 3c. If one of the clusters was smaller, more time series would have to be in $CTS_{DB}(\mathbb{K}_3)$ in order to prevent inference. The more time series the resulting clusters contain, the more candidate sets exist that keep the number of indistinguishable time series higher than k, giving way to further splits. This heuristic takes successive the clusters with the highest score and tries to perform as much splits as possible in the single clusters. The larger the distance between the highest and the lowest value (in the original data set) of the cluster members, the higher has been the information loss in Stage 1. For the multidimensional case, the difference between the minimum and the maximum has to be defined slightly differently, e.g., as the sum of the difference in each dimension. 6.2.2 shows MTH in pseudocode.

4.2.3 Validation

Splitting clusters of size greater than or equal to k results in new clusters of size less than k. Thus it may be possible that values at certain points of time can be inferred. Before conducting a split, the optimization heuristics in Stage 2 must validate that the data set is (n, l, k)-anonymous afterwards (see Algorithms 2 and 3).

A canonical solution would be to inspect all external knowledge that is possible and to compute what an adversary can infer. If there is no potential external knowledge based on which an attacker can infer at least $l - n$ tuples, the data set is (n, l, k)-anonymous. However, this solution is infeasible in practice. This is because for each individual $\binom{\|\mathbb{T}\|}{n}$ possible sets of external knowledge exist. We approach the problem from the opposite direction with the so-called fast validation, see Algorithm 5 and Section 6.3: Only points of time where clusters with less than k time series exist are inferrable. We can compute candidate sets of time series an adversary must be able to single out in order to infer those points of time:

- The candidate set creation considers only points of time with at least one cluster containing less than k time series. For such a point of time t candidates c_t are combinations of clusters at t with less than k time series in total. If the set of indistinguishable time series is a subset of such a combination, the point of time t is inferred.
- Candidate sets of different points of time (c_{t1} and c_{t2}) are combined by calculating the intersection of the two sets ($c_{t1} \cap c_{t2}$). If the set of indistinguishable households is a subset of $c_{t1} \cap c_{t2}$, $t1$ as well as $t2$ are inferred.
- Candidates for $l - n$ points of time are created by intersecting all candidates of $l - n$ different points of time.

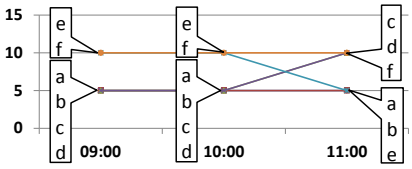


Figure 6: Example of a validation for a non $(2, 3, 6)$-anonymous data set

Algorithm 4 is the candidate creation. Example 7 illustrates this generation for a single point of time.

Example 7 (*Candidate time series*): Assume that there are three clusters at t: $\{a, b\}, \{c, d\}, \{e, f\}$. For $k = 5$ there exist three candidates: $\{a, b, c, d\}$, $\{a, b, e, f\}, \{c, d, e, f\}$ □

In order to test if this is possible we have to inspect clusters containing time series of these candidates in DB' and check if they single out one of these candidates. The following example illustrates the fast validation.

Example 8 (*Example for the fast validation*): Assume that we want to validate if the data set in Figure 6 is $(2, 3, 6)$-anonymous. In this case it is not. One candidate is the set $\{e, f\}$ at 10:00. The fast validation algorithm can choose up to two points of time, to find a set of indistinguishable time series that is a subset of $\{e, f\}$. Choosing the upper cluster at 09:00 already shows that the data set is not $(2, 3, 6)$ anonymous. □

4.3 Complexity Analysis

To keep the presentation simple, we define the variable $p = \|DB\| \cdot \|\mathbb{T}\|$, i.e., the product of the number of time series and the number of points of time.

4.3.1 Clustering

The **Clustering** Stage (see Algorithm 1) has complexity $O(p^2)$: Adding a time-value tuple to a cluster has constant complexity. Building the initial clusters has complexity $O(\|DB| \cdot \|\mathbb{T}\|)$. In the worst case, for $k = 2$, the algorithm has to split the initial clusters at each point of time $\frac{\|DB\|}{2}$ times to result in clusters of two elements. This leads to the following total complexity:

$$O(\|DB| \cdot \|\mathbb{T}\| \cdot \frac{\|DB\|}{2} \cdot \|\mathbb{T}\|) = O(p \cdot \frac{p}{2}) = O(p^2)$$

4.3.2 Validation

The validation if a data-set is (n, l, k)-anonymous is in the complexity class $O(p^3)$.

In the worst case, the validation algorithm has to check every combination of n tuples for every time-series at every point of time $\|DB\| \cdot \binom{\|\mathbb{T}\|}{n}$. This results in a complexity of $O(\|DB\| \cdot \binom{\|\mathbb{T}\|}{n}) = O(p^2)$. Given a combination of n tuples, the complexity for the validation of a single point of time is the number of clusters, with $O(p)$ as an upper bound. This results in the overall complexity $O(p^3)$. The same complexity also holds for the algorithm described in 6.3 that reduces the candidate set.

4.3.3 Optimization Heuristics

The complexity of MIL is $O(p^2 \cdot p^3) = O(p^5)$, as well as the complexity of MTH. Validation ($O(p^3)$) has the highest complexity of the optimization steps.

MIL (see Algorithm 2) behaves like the clustering in Stage 1 without the limit of a cluster size and with the validation of the (n, l, k)-anonymity. This leads to a complexity of $O(p^2 \cdot p^3)$. With MTH (see Algorithm 3), each cluster C^t is split at most $\|C^t\|$ times. The upper bound on the number of splits for all clusters is p. There are p clusters containing at most one time series. Thus, the upper bound for the complexity of MTH is $O(p^2 \cdot p^3)$.

5 Evaluation

5.1 Experimental setup

We perform experiments both with real-world data and synthetic data. Experiments with synthetic data are necessary to investigate dependencies on exogenous parameters systematically.

Real-world setup For real-world experiments, 180 households have measured their power consumption every hour for two weeks. A metering point reflects the power consumption in the last 60 minutes. We have anonymized this data set to become (n, l, k)-anonymous, with different parameters. In the following we will present the most interesting results of these experiments.

$k = 10$ means that any individual is indistinguishable to nine others, there is only a 10% chance for an attacker to guess the original time series. The value of n reflects how hard it is for an attacker to get actual values. l depends on how sensitive a single value is in the specific scenario. We evaluate a broad range of parameters: n ranges from $n = 3$ to $n = 10$, and $l - n$ varies between 1 and 12. This is reasonable in terms of privacy: It means that an attacker has observed the consumption of a household for 3 to 10 hours, and he is allowed to have a total of 12 values at most in order to identify sensitive patterns that might have an impact on the privacy of the households.

Synthetic setup We generate time series in two different ways: randomly and sinus curves ($a + b \cdot sin(x \cdot c)$) with randomly chosen and equally distributed values for a, b, c. For each run, we set the number of households

to 1000 and the number of points of time to 168 (this equates to one week of hourly measurements). For splitting we have used MTH since it has performed better in preliminary experiments with the real-world setup. We report on the evaluation of two scenarios: (1) Randomly generated values in the $[10 \dots 30]$ range. (2) Sinus curves, with $a \in [30 \dots 50]$, $b \in [1 \dots 5]$ and $c \in [0.2 \dots 2]$.

5.2 Quality of anonymized data

5.2.1 Normalized divergence

To study how strongly anonymization changes the data, we define the normalized divergence as the ratio of the average value of the data points in the original data set and the Euclidean information loss for each data point:

$$NormDiv = \frac{\frac{IL(DB,DB')}{|\mathbb{T}|\cdot|DB|}}{\frac{\sum_{t\in\mathbb{T},f\in DB} f(t)}{|\mathbb{T}|\cdot|DB|}} = \frac{IL(DB,DB')}{\sum_{t\in\mathbb{T},f\in DB} f(t)},$$

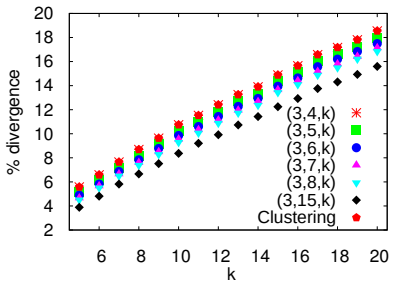
where DB is the original and DB' the anonymized version of the database.

A high normalized divergence means a high relative distance between the anonymized values and the original ones. Figures 7a and 7b graph the results of the clustering (Stage 1) and of the splitting (Stage 2). Since different n and l parameters affect only Stage 2, there is a separate curve for these configurations. Both graphs show that the resulting normalized divergence of a tuple is, even for $k = 20$, between 15% and 18%. Setting k to 20 means that Stage 1 reduces the number of tuples per point of time from 180 (the number of households/smart meters) to $180/20 = 9$. In other words, the number of distinct data points is reduced by 95%, and the normalized divergence is only 15 - 18%. If an indistinguishability of only $k = 10$ is required, and the number of distinct values is reduced by 90%, the divergence will only be around 10%. Our results on synthetic data in Figure 8a show an even smaller divergence, implying that comparable results can be achieved even with a higher k and many more households.

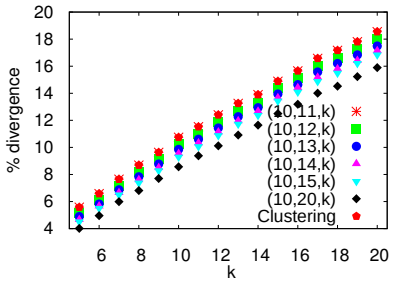
5.2.2 Standard deviation

The absolute difference between the standard deviation of values of the original data set and of the anonymized one tells us how strong the influence of the anonymization on the distribution of data points of the anonymized time series is. This is in contrast to the normalized divergence, which only reflects the change of single points. The standard deviation of a data set DB is as follows:

$$S(DB) = \sqrt{\frac{1}{\|\mathbb{T}\|\cdot\|DB\|}\sum_{\forall f\in DB,\forall t\in\mathbb{T}}(f(t)-\overline{DB})^2},$$



(a) $(3, l, k)$



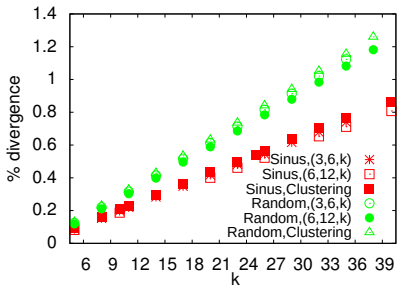
(b) $(10, l, k)$

Figure 7: Real-world scenario: normalized divergence of a data point

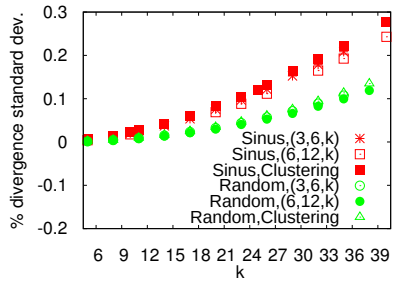
where $\overline{DB}$ is the mean of all values. Figure 9a and 9b show the shift of the standard deviation for different parameter values, it never exceeds 11%. For $k = 10$, it is only around 3%. This is rather low if we take into account that, for $k = 10$, a reduction of the number of distinct values in the original data set of 90% is necessary to create mostly clusters of 10 time series. Again, for the synthetic scenarios (Figure 8b), the results are similar.

5.2.3 Fraction of diverging points

Figure 10 shows the distribution of the divergence for different parameter settings. We choose $n = 7$ and $l = 10$ as average values of the previous experiments. For each setting, we have computed the divergence between each anonymized value and the original value, and we have categorized them into five classes, ranging from $0-2\%$ divergence to $30 - 50\%$. For instance, the figure shows that approximately 35% of the data points have a divergence of less than 2 % for $(7, 10, 10)$. Further, for a higher k, the fraction of points with a higher divergence increases. However, even for $k = 15$, more than 60% of the points have a divergence of less than 20%. Recall that executing Stage 1 with $k = 15$ and 180 data points (one for each household) results into $\frac{180}{15} = 12$ distinct data points/clusters. In a nutshell, Figure 10 shows that a high percentage of data points has a low pointwise divergence.

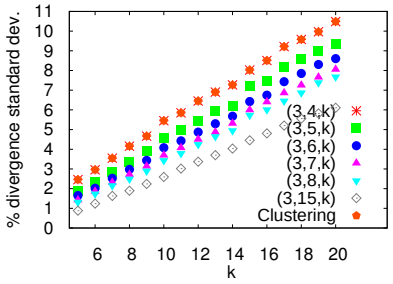


(a) Divergence

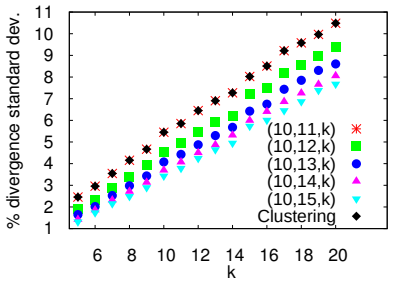


(b) Shift of standard deviation

Figure 8: Synthetic scenario



(a) $(3, l, k)$



(b) $(10, l, k)$

Figure 9: Real-world scenario: shift of standard deviation

Even if the number of distinct data points in the various time series has been reduced very much, the divergence of the data points and the standard deviation are low. Thus, (n, l, k)-anonymity gives way to high quality data. The higher the difference between l and n, the better splitting works.

5.2.4 Comparison of MTH and MIL

Finally, we have evaluated MTH and MIL with (n, l, k)-anonymity parameters ranging from $k = 3$ to $k = 8$, from $n = 3$ to 10 and from $l - n = 1$ to 12. Figure 11 summarizes our results. For each different k, we have calculated the average information loss. The figure shows that on average MTH reduces the information loss roughly twice as much as MIL. On the other hand, since MIL splits clusters with the highest information loss first, it tends to preserve outliers (c.f. 4.2). Thus, if the anonymized data set will be used for tasks like outlier mining, one should choose MIL.

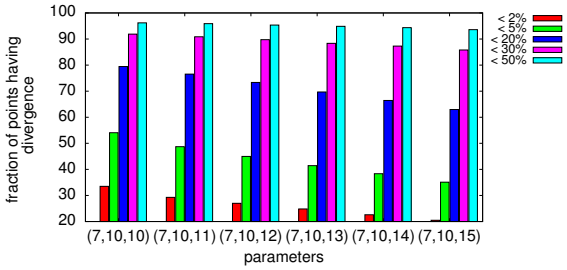


Figure 10: Fraction of data points with specific divergence

5.3 Computation Time

We have measured the computation time on an AMD Athlon 64 X2 Dual Core 4800+ Processor with Java 1.6 and heap space of 2GB and on the real-world data set. Figure 12 features the computation time for Stage 1 **Clustering**, and Figure 13 contains the run times for Stage 2. The clustering itself is much faster than the optimization because the validation is complex. The higher the difference between l and n, the more time takes the optimization: First, the heuristics can split more clusters the higher $l - n$ is. This is because more points of time can be inferred without violating (n, l, k)-anonymity. Second, if more clusters are split, the validation requires more computation time. This is because a larger number of candidate sets for inference exists (see Section 4.2.3). The MTH module is usually faster, since it results in fewer cluster splits.

Even with large differences between n and l, the total run time in our setup never has exceeded 12 minutes. This shows that our proposed method is applicable with acceptable runtime.

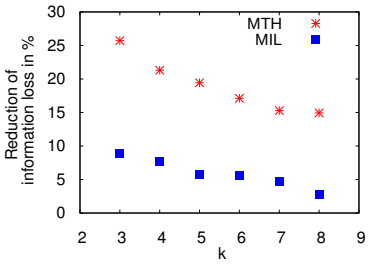


Figure 11: Average reduction of euclidean information loss depending on the optimization heuristic

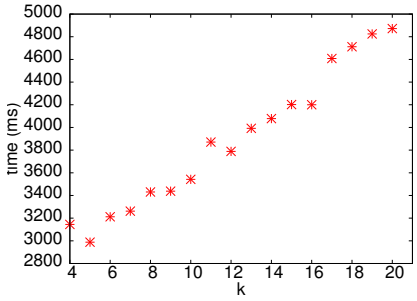


Figure 12: Execution time for stage 1 Clustering

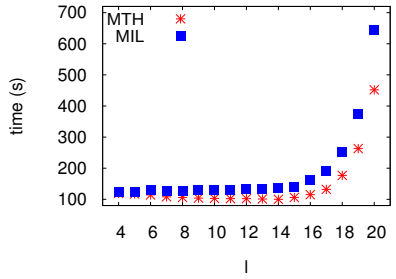


Figure 13: Execution time for both stage 2 optimization strategies ($(3, l, 4)$)

6 Conclusions

From one perspective, personal data should be widely available to facilitate scientific insights. In the smart grid scenario, this would mean that consumption data of households should be published. However, the data is personal and sensitive, and privacy of the individuals has to be protected.

Time series such as the energy consumption of households contain patterns of sensitive information. This means that several values are necessary to extract useful information. A definition of privacy taking those patterns into account has been missing so far. However, an important objective regarding anonymization is to keep the data quality as high as possible.

This paper has proposed (n, l, k)-anonymity, allowing a limited number of values to be inferred by an adversary. In addition, we have proposed various heuristics to anonymize time series into a (n, l, k)-anonymous version. Our evaluation has shown that the quality of (n, l, k)-anonymized data is high. Our evaluation has used domain-independent measures, indicating that our results might be applicable to a broad range of scenarios.

References

[1] O. Abul, F. Bonchi, and M. Nanni, "Never walk alone: Uncertainty for anonymity in moving objects databases," *ICDE 2008*, pp. 376 –385, April 2008.

[2] G. Acs and C. Castelluccia, "I have a DREAM! (DIffeRentially PrivatE smart Metering)," *Information Hiding*, pp. 118–132, 2011.

[3] E. Buchmann, K. Böhm, T. Burghardt, and S. Kessler, "Re-identification of Smart Meter data," *Personal and Ubiquitous Computing*, Mar. 2012. [Online]. Available: http://www.springerlink.com/index/10.1007/s00779-012-0513-6

[4] Bundesrepublik Deutschland, "Gesetz ueber die Elektrizitaets- und Gasversorgung," 2005.

[5] J. Cao, P. Karras, C. Raïssi, and K.-L. Tan, "ρ-uncertainty: inference-proof transaction anonymization," *Proceedings of the VLDB Endowment*, vol. 3, pp. 1033–1044, September 2010.

[6] A. Cavoukian, J. Polonetsky, and C. Wolf, "Smartprivacy for the smart grid: embedding privacy into the design of electricity conservation," *Identity in the Information Society*, vol. 3, pp. 275–294, 2010.

[7] R. Chen and B. C. M. Fung, "Differentially Private Transit Data Publication : A Case Study on the Montreal Transportation System Categories and Subject Descriptors," *KDD 2012*, pp. 213–221, 2012.

[8] R. Chen, B. C. Fung, N. Mohammed, B. C. Desai, and K. Wang, "Privacy-preserving trajectory data publishing by local suppression," *Information Sciences*, pp. 83–97, Jul. 2011.

[9] C.-Y. Chow, M. F. Mokbel, and W. G. Aref, "Casper*: Query processing for location services without compromising privacy," *ACM Transactions on Database Systems*, vol. 34, December 2009.

[10] C. Dwork, "Differential privacy," in *Automata, Languages and Programming*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2006, vol. 4052, pp. 1–12.

[11] C. Efthymiou and G. Kalogridis, "Smart grid privacy via anonymization of smart metering data," *First IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pp. 238 –243, oct. 2010.

[12] European Comission, *European Technology Platform SmartGrids: Vision and Strategy for Europe's Electricity Networks of the Future*, 2006.

[13] H. Farhangi, "The path of the smart grid," *Power and Energy Magazine, IEEE*, vol. 8, no. 1, pp. 18–28, January/February 2010.

[14] S. Gold, "Not-so-smart meters?" *Network Security*, vol. 6, pp. 9 – 11, 2009.

[15] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," 2003.

[16] A. Guénoche, P. Hansen, and B. Jaumard, "Efficient algorithms for divisive hierarchical clustering with the diameter criterion," *Journal of Classification*, vol. 8, pp. 5–30, 1991, 10.1007/BF02616245.

[17] G. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870 –1891, dec. 1992.

[18] H. Kido, Y. Yanagisawa, and T. Satoh, "An anonymous communication technique using dummies for location-based services," *International Conference on Pervasive Services*, pp. 88–97, 2005.

[19] H. Kim, "Unsupervised disaggregation of low frequency power measurements," *ICDE 2011*, 2011.

[20] N. Li, T. Li, and S. Venkatasubramanian, "t-closeness: Privacy beyond k-anonymity and l-diversity," pp. 106 –115, april 2007.

[21] A. Machanavajjhala, J. Gehrke, and M. Götz, "Data publishing against realistic adversaries," *Proceedings of the VLDB Endowment*, vol. 2, no. 1, pp. 790–801, Aug. 2009.

[22] A. Machanavajjhala, D. Kifer, J. Abowd, J. Gehrke, and L. Vilhuber, "Privacy: Theory meets practice on the map," *ICDE 2008*, pp. 277–286, 2008.

[23] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "L-diversity: Privacy beyond k-anonymity," *ACM Transactions on Knowledge Discovery from Data*, vol. 1, March 2007.

[24] D. Martin, D. Kifer, A. Machanavajjhala, J. Gehrke, and J. Halpern, "Worst-case background knowledge for privacy-preserving data publishing," *ICDE*, pp. 126 –135, april 2007.

[25] P. McDaniel and S. McLaughlin, "Security and privacy challenges in the smart grid," *Security Privacy, IEEE*, vol. 7, no. 3, pp. 75 –77, may-june 2009.

[26] A. Molina-Markham, P. Shenoy, K. Fu, E. Cecchet, and D. Irwin, "Private memoirs of a smart meter," *BuildSys '10*, pp. 61–66, 2010.

[27] M. Nergiz, M. Atzori, and Y. Saygin, "Towards trajectory anonymization: a generalization-based approach," *SPRINGL*, pp. 52–61, 2008.

[28] N. I. of Standards an Technology, *NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 1.0*, January 2010.

[29] E. L. Quinn, "Privacy and the New Energy Infrastructure," *SSRN eLibrary*, 2009.

[30] L. Sweeney, "k-anonymity: a model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, pp. 557–570, October 2002.

[31] M. Terrovitis, N. Mamoulis, and P. Kalnis, "Privacy-preserving anonymization of set-valued data," *Proceedings of the VLDB Endowment*, pp. 115–125, 2008.

[32] R. van Gerwen, S. Jaarsma, and R. Wilhite, "Smart Metering," *Distributed Generation (www.leonardo-energy.org)*, pp. 1–9, June 2006.

[33] Y. Xu, K. Wang, A. Fu, and P. Yu, "Anonymizing transaction databases for publication," *Proceeding of the 14th ACM SIGKDD*, pp. 767–775, 2008.

[34] R. Yarovoy, F. Bonchi, L. V. S. Lakshmanan, and W. H. Wang, "Anonymizing moving objects: how to hide a mob in a crowd?" *EDBT 2009*, pp. 72–83, 2009.

[35] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, "Mining interesting locations and travel sequences from gps trajectories," *WWW 2009*, pp. 791–800, 2009.

APPENDICES

6.1 (n, l, k)-Anonymization: Clustering

We use the following data structures: A $ClusterSet$ is the set of time series belonging to a specific $Cluster$. We describe a $Cluster$ as a tuple consisting of a $ClusterSet$ and the mean value as center of the cluster. $ClusterConfiguration$ is a set containing all clusters (represented by a $Cluster$ tuple) of the currently processed point of time t.

Listing 1 shows our clustering approach. First, we put all time series into the same cluster (Line 5). The mean value of all tuples represents the data point for this cluster (Line 7). Let $m_1, ..., m_n$ be the sorted metering values of the original time series at point of time t. The

order implies that m_i is the neighbor of m_{i-1} and m_{i+1} (if existent) at t (Line 12). Regarding the loss of information, a good split would be between the two metering values with the highest distance and result in two clusters with more than k members (Line 12-24). A split leads to two new clusters represented by the mean value of the original data points.

6.2 Optimization Heuristics

6.2.1 MostInformationLoss (MIL)

Algorithm 2 contains the pseudo code of MIL. First, MIL computes the information loss between the original database DB and the anonymized database DB' for each cluster C^t at each point of time t (*sort* function in Line 6). MIL then iterates over each point of time, starting with the highest information loss (Line 8). For each point of time, it tries to split the clusters that incur the highest information loss (Line 11 - 18). Since the resulting clusters are smaller than k, this is not always possible. Thus, before a cluster is split into $f(t)$ and $f'(t)$, function $isNlkAnonymous(DB', t, f, f')$ checks if the resulting database of times series DB' still satisfies (n, l, k)-anonymity (Line 14).

6.2.2 MembersTimesHeight (MTH)

In contrast to the MIL algorithm, the MTH heuristic orders all clusters (regardless of their point of time) instead of iterating over the points of time and processing the clusters in that order. Thus, there are only two nested loops in the MTH algorithm on algorithm 3 in Line 6 and 8. The check if the split does not violate the (n, l, k)-anonymity remains the same (Line 9).

6.3 Validation Algorithm

Algorithm 5 contains pseudo code of the fast validation algorithm. It returns true if the data set is (n, l, k)-anonymous and false otherwise. First of all, it requires the building of candidate sets in Line 11. As we have explained in Section 4.2.3, we build a candidate set from points of time with combinations of clusters smaller than k and from their intersections with other points of time. Algorithm 4 is an implementation of this step.

After the creation of the candidates, the algorithm tries to build a set of indistinguishable households (that are a subset of one candidate) with at most n points of time as external knowledge. If this is possible, exactly $l - n$ points of time are inferred. In order to build such external knowledge, the algorithm takes a cluster as a starting point of possible time series (Line 17) and searches for other points of time that reduce this set (Line 23). If the algorithm finds external knowledge with at most n points of time that is a subset of a candidate set, it returns false since this violates the (n, l, k)-anonymity property.

Algorithm 1: Top Down Clustering

```
DB: Original data set
DB' = DB: Modified data set, initialized with copy

for each t ∈ 𝕋{   //Cluster point of time t
  ClusterSet C = ⋃_{f∈DB} {f}
  //Define Cluster as tuple: (representing Value, set of time series)
  Cluster c = (calcCenter(C, t), C)
  ClusterConfiguration CF = {c}
  // f and g are neighbors if no point is between (t, f(t)) and (t, g(t))
  List l_n = List of neighbors (f, g)

  forall((f, f') ∈ l_n in asc order of dist. between neighbors){
    if cluster including f, f' exists{
      ClusterSet C_1 = {f}
      ClusterSet C_2 = {f'}
      add all g ∈ C to C_1 or C_2 depending on g(t)

      if(|C_1| ≥ k and |C_2| ≥ k){
        CF = CF\(center, C)
        CF = CF ∪(calcCenter(C_1, t), C_1)
        CF = CF ∪(calcCenter(C_2, t), C_2)
      }
    }
}}

//Helper function: Calcs the average value of a set of time series at t
float calcCenter(Set of time series F, point of time t)
  return (Σ_{∀f∈F} f(t)) / ||F||
```

Algorithm 2: MostInformationLoss

```
DB: Original data set
DB': Data set after clustering in stage 1
sort(S, desc/asc, f): returns a sorted list of elements in set S, sorted descending/
    ascending by the expression f
getClusterConfig(DB, t): returns a set of clusters (represented as a set of time series)
     at point of time t

𝕋_sorted = sort(𝕋, desc, IL_t(DB, DB'))

foreach(d ∈ 𝕋_sorted){
  List l_neighbors = [(f, t), (f', t)), ...], f, f' are neighbors in DB

  foreach(c ∈ sort(getClusterConfig(DB', t), desc, IL_t^c(DB, DB'))){
    List l_neighbors = [((f, t), (f', t)), ...], f, f' ∈ c neighbors in DB
    foreach(((f, t), (f', t)) ∈ sort(l_neighbors, desc, |f(d), f'(d)|)){
      if(isNlkAnonymous(DB', d, f, f',(n,l,k))){
        split(DB', d, f, f')
      }
    }
} } }
```

Algorithm 3: MembersTimesHeight

```
DB: Original data set
DB': Data set after clustering in stage 1
sort(S, desc/asc, f): returns a sorted list of elements in set S, sorted descending/
    ascending by the expression f
getClusterConfig(DB): Returns a multi set of all clusters of the data set

foreach( c ∈ sort(getClusterConfig(DB'), desc, Score(c)){
    List l_neighbors = [((f,t),(f',t)),...], f,f' ∈ c neighbors in DB
    foreach(((f,t),(f',t)) ∈ sort(l_neighbors, desc, |f(t) − f'(t)|)){
        if(isNlkAnonymous(DB', t, f, f',(n,l,k))){
            split(DB', t, f, f')
        }
    } }
}
```

Algorithm 4: calculateCandidateSets(): Creation of candidate sets for the fast validation

```
DB: Data set
(n,l,k): Privacy parameters

// set containing set of candidates of single points of time
clusterCandidates = {};

foreach(t ∈ 𝕋){
  foreach(Combination of clusters at t C_t^i with ∑|C_t^i| ≤ k){
    CC = {⋃_∀i C_t^i}
    clusterCandidates = clusterCandidates ∪ CC;
  }
}

candidates = {};
foreach(Combination of l − n C_t ∈ clusterCandidates with different t){
  candidate = {C_t1 ∩ ... ∩ C_t(l−n)};
  candidates = candidates ∪ candidate;
}

return candidates;
```

Algorithm 5: isNlkAnonymous(): Algorithm for the (fast) validation

```
DB': Data set requiring validation
(n,l,k): Privacy parameters
t, f, f': Point of time and time series, that define the possible split

// copy and split in order to test if DB is still (n,l,k)-anonymous
DB = copy(DB');
DB = split(DB, t, f, f');

for each f ∈ DB{  //take every time-series of DB
    //creates candidate sets that infer l − n points of time
    CandidateSets candidateSets = calculateCandidateSets(DB, n, l, k);

    foreach(t ∈ 𝕋){
      Cluster C_f^t = Cluster at t containing f;
      foreach(CandidateSet s ∈ candidateSets){
        // pTS is the current knowledge of possible time series of an adversary
        Set pTS = C_f^t;
        //try to break the candidate set s
        Set diff = pTS\s;

        𝕂 = {t};
        while(‖𝕂‖ ≤ n){
          foreach(t2 ∈ 𝕋\t){
            if ∃ Cluster C^t2 containing an element of diff and excluding (at least)
                one of pTS{
              //add it to the knowledge
              𝕂 = 𝕂 ∪ {t2};
              //reduce the possible time series
              pTS = pTS\C^t2;
            }
            //is the candidate set already singled out
            if(pTS ⊆ s){
              //not valid
              return false;
            }
          }
        }
      }
    }
}
//valid
return true;
```

AUTHOR BIOGRAPHIES

Stephan Kessler received his diploma (M.Sc.) in computer science from the Karlsruhe Institute of Technology in 2010. Since 2011 he holds a scholarship of the "Information Management and Market Engineering" graduate school and works in the group of Erik Buchmann at the IPD Böhm. His research interests are privacy protection for time-series of smart meter data inlcuding its applications to local energy markets.

Erik Buchmann was born in Magdeburg, Germany in 1976. He received his Diploma (M.Sc.) in Business Informatics from the Otto-von-Guericke-University of Magdeburg, Germany in 2002. In 2006, he earned his Ph.D. in Computer Science (summa cum laude) from the same university. Since September 2006, he is a research associate of the IPD, Karlsruhe Institute of Technology. By 2007, he became head of the Young Investigator Group Privacy Awareness in Information Systems and its Implications on Society.

Thorben Burghardt earned his Ph.D. in Computer Science in 2010 from the Karlsruher Institute of Technology (KIT). His research interests include privacy protection of personal data and privacy mechanisms. He is currently head of the metering division of the LUCEBIT GmbH in Mannheim.

Klemens Böhm is full professor (chair of databases and information systems) at Karlsruhe Institute of Technology (KIT), Germany. Prior to that, he has been professor of applied informatics/data and knowledge engineering at University of Magdeburg, Germany, senior research assistant at ETH Zürich, Switzerland, and research assistant at GMD - Forschungszentrum Informationstechnik GmbH, Darmstadt, Germany. Current research topics at his chair are knowledge discovery and data mining in big data, data privacy and workflow management. Klemens gives much attention to collaborations with other scientific disciplines and with industry.