Analyzing Continuous k_s-Anonymization for Smart Meter Data*

Short Paper

Carolin Brunn, Saskia Nuñez von Voigt, and Florian Tschorsch

Distributed Security Infrastructures, Technische Universität Berlin, Berlin, Germany
{c.brunn, saskia.nunezvonvoigt, florian.tschorsch}@tu-berlin.de

Abstract. Data anonymization is crucial to allow the widespread adoption of some technologies, such as smart meters. However, anonymization techniques should be evaluated in the context of a dataset to make meaningful statements about their eligibility for a particular use case. In this paper, we therefore analyze the suitability of continuous k_s-anonymization with CASTLE for data streams generated by smart meters. We compare CASTLE's continuous, piecewise k_s-anonymization with a global process in which all data is known at once, based on metrics like information loss and properties of the sensitive attribute. Our results suggest that continuous k_s-anonymization of smart meter data is reasonable and ensures privacy while having comparably low utility loss.

1 Introduction

The suitability of data anonymization techniques, such as k-anonymity [10], must be evaluated in the context of a dataset to make meaningful statements. In particular, the data types, the granularity, and distribution have an impact on the efficiency of data anonymization and affect the fundamental trade-off between data privacy and data utility.

For smart meter data, the efficiency of data anonymization remains unclear as the application scenario and the data pose a challenge. While smart meters (SMs) become increasingly important to enable dynamic resource management of various energy sources, the type of data differs from other relational data sources. SMs generate a data stream derived from continuous sensor data, measuring consumption of electric energy, gas, and water. SM data, therefore, comprises sensitive, personal data that require privacy protection. In addition, the application scenario dictates a distributed architecture with distributed data sources.

In this paper, we investigate the continuous anonymization of SM data and assess the efficiency of k_s-anonymity for the anonymization in this scenario. The concept of k_s-anonymity is an extension of k-anonymity for data

* Supported by the Federal Ministry of Education and Research of Germany (Project 16KISA034)

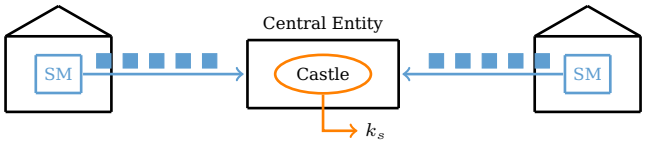

Fig. 1: Centralized architecture with smart meters forwarding measurements to a central entity for anonymization.

streams [2]. In particular, we use the widely recognized algorithm for stream anonymization, CASTLE [2], and study its characteristics and suitability. For our study, we consider a typical SM architecture in which distributed SMs send their data to a central entity (CE). We evaluate the suitability of k_s-anonymity for SM data based on metrics such as information loss and range of the sensitive attribute, and compare the performance of continuous piecewise anonymization with an idealized anonymization as baseline.

Our results suggest that k_s-anonymity is a reasonable choice for anonymizing smart meter data. Based on our metrics, the performance of continuous data anonymization appears to be comparable to our baseline. Further analysis of the diversity of consumption measurements shows that in most clusters, the values of the sensitive attribute are distributed over a wide range and are not clustered around a single consumption value. Additionally, we note that the prioritization of attributes during the anonymization process depends on the different magnitudes of the attribute ranges. This should be taken into account in any case, but can also be exploited to shape the process to a certain degree.

The paper is organized as follows. After introducing our problem statement as well as k_s-anonymity and CASTLE in Sec. 2 and 3, we present our evaluation in Sec. 4. In Sec. 5 we conclude the paper.

2 Problem Statement and Related Work

Problem Statement. Our goal is to analyze whether continuous k_s-anonymization is suitable for SM data. Since the data type differs from other relational data sources in some crucial characteristics, this is everything but obvious. The data points generated by SMs are discrete measurements of user consumption, e.g., electricity consumption from a continuous data stream. Different strategies can be applied to discretize the data stream, such as using the current consumption value or aggregating the entire consumption between two measurements. Thus, SM data has different characteristics, such as the temporal granularity of the measurements.

For our evaluation, we use a realistic architecture in which distinct SMs measure the consumption and forward the data directly to a trusted CE, e.g., the energy provider. Figure 1 visualizes this architecture. To facilitate

further processing by third parties, e.g., for district management, the data is collected and anonymized centrally before it is forwarded.

Related Work. There are several approaches to avoid accurate profiling and disclosure of information based on smart meter measurements. For instance, load balancing and shaping prevent characteristic traces in consumption data, while other approaches focus on achieving privacy by design with specific architectures. Another focus is on protecting privacy by anonymizing consumption data, e.g., with k-anonymity [10] or differential privacy [3]. One algorithm to achieve k-anonymity for streaming data is the one analyzed in this paper—CASTLE. Several other algorithms exist, some of which also address challenges of CASTLE, such as [4, 7, 8, 11]. However, to the best of our knowledge, there are no studies that evaluate the suitability of k_s-anonymity specifically for smart meter data.

3 k_s-Anonymity and CASTLE

We focus on k_s-anonymity [2], which is an extension of k-anonymity [10] for streaming data. The main idea is to modify and group data items in such a way that groups comprise at least k entries that are indistinguishable from each other—an Equivalence Class (EQ). k_s-anonymity [2] extends this idea and requires that a published anonymized stream comprises EQs with at least k distinct individuals, not just k entries.

CASTLE [2] is an established algorithm to achieve k_s-anonymity by assigning incoming data points, called tuples, to clusters that represent their generalization. The tuples are specified in a metric space defined by the so-called Quasi-identifiers (QIs) [10]. The clusters are EQs, where all data points share the same generalized values for each QI attribute. Each cluster must contain at least k distinct individuals. CASTLE either creates a new cluster or assigns the tuple to an existing cluster by minimizing the information loss. As information loss metric, CASTLE uses the Generalized Loss Metric [5]. For cluster generalization, QI attributes either form intervals, in the case of continuous attributes, or they are generalized to their lowest common ancestor with respect to their corresponding domain generalization hierarchy (DGH) for categorical attributes. A DGH is a directed tree structure that defines hierarchical values for such categorical attributes. CASTLE also uses a delay constraint δ that specifies the maximum time that can pass before a tuple needs to be generalized and published. The clusters that were anonymized with CASTLE can then be published and used for further processing, e.g., by third-party data processors.

We can already observe that during the anonymization process, the QIs are used for the generalization. Accordingly, we expect that their different magnitudes will play a crucial role. At the same time, please note that the

sensitive attribute is not considered in the process. This could enable attacks if users in a cluster have different consumption ranges that differ significantly from each other.

4 Evaluation

Methodology. For our evaluation, we use a dataset of electricity consumption measurements that is publicly available at UCI[1]. The dataset consists of consumption data from 370 clients, measured every 15 minutes between 2011 and 2015. Based on the consumption profiles in the dataset, we infer that the set comprises data of individual households, and larger consumers such as schools, hospitals, or small industry[2]. The original dataset contains only measurement data and no additional information about clients. We therefore added synthetic addresses, modeling a district in Berlin, where zip code, street, and house number are encoded in an integer value. Furthermore, we adapted the format of the timestamps. Due to its size, we sampled the dataset (weeks 46 & 47 of November 2014) resulting in $164\,102$ tuples.

We use the publicly available CASTLEGUARD implementation [9]. When disabling the differential privacy feature (which we did), it resembles the CASTLE algorithm. Since we identified potential bugs, we made some minor adaptations to the code[3], e.g., in the function `merge_clusters`. We provide our code including the respective changes as well as the dataset on GitHub.[4]

We simulated a distributed k_s-anonymization with CASTLEGUARD for different δ and k, and compare it to a global k_s-anonymization process in which all data tuples are known in advance and then clustered all at once. The latter is simulated using the ARX anonymization tool [1].

Information Loss. For our evaluation, we use information loss as utility metric. Specifically, we use the Generalized Loss Metric (GLM) [5], which is also used for estimating the information loss in CASTLE. Here, the cluster range of a generalized attribute is compared to the overall range of this attribute. For each entry, the information loss of an attribute is defined as $\frac{u_i - l_i}{U - L} \in [0, 1]$, where u_i or l_i is the upper or lower limit of entry i's attribute generalization, and U or L is the overall upper or lower limit of this attribute, respectively. For our evaluation, we calculate the average information loss across all clusters per attribute.

Figure 2 shows the average information loss of all clusters for varying k and δ of the *address* (left plot) and *time* (right plot) attribute, respectively.

[1] `https://doi.org/10.24432/C58C86`

[2] The magnitude of consumption values suggests that the values are given in Watt instead of kW as noted in the description of the dataset.

[3] We have reached out to the developers to discuss the bugs/changes.

[4] `https://github.com/carolin-brunn/dpm-castle-analysis`

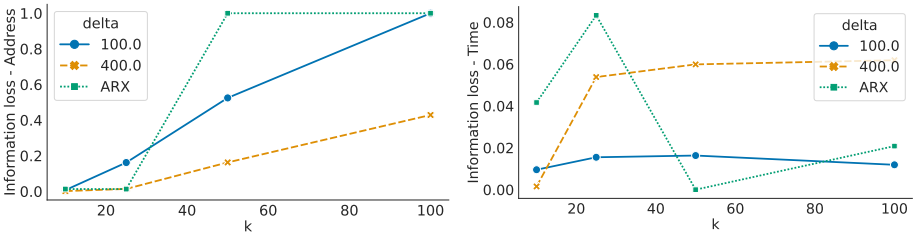

Fig. 2: Average information loss for quasi-identifying attributes.

We observe that the information loss is highest for the *address* attribute in most settings. The clusters must always contain k distinct individuals that all have different addresses, thus, whenever a cluster is created, the *address* attribute needs to be generalized. The information loss increases for the *address* with an increasing k. This is expected since an increasing k requires more distinct individuals with different addresses. We also observe that for CASTLE, the information loss increases as the ratio between k and δ increases. Presumably, CASTLE is forced to join very different clients, if many clients have to be extracted from a relatively small sliding window. Overall, it is noticeable that the *address* information loss is comparable for ARX and CASTLE. For lower k, ARX has a lower information loss than CASTLE's sequential generalization with $\delta = 100$. However, for larger k and δ the advantage of ARX fades. For $\delta = 400$, CASTLE consistently finds better clusters that result in a lower information loss when compared to ARX.

For the *time* attribute, the information loss of ARX and CASTLE is comparable. Note, however, that in the beginning ARX has a higher information loss than CASTLE. Presumably, this is caused by ARX' anonymization strategy: ARX chooses the same generalization level for all values of the same attribute. Consequently, one cluster that requires a higher level of generalization may cause all other clusters that could be formed with a lower generalization to be published with the unnecessary generalization.

In general, the results suggest that CASTLE is a reasonable alternative to a global generalization with ARX especially for larger δ. Nevertheless, attribute ranges seem crucial for the prioritization when generalizing attributes. Consequently, analyzing the exact behavior of CASTLE with attributes of different magnitudes and diverse parameter settings is necessary to find optimal settings for the anonymization of smart meter data, which we will investigate in the remainder.

UID Diversity. Next, we compare the size of the published clusters and the diversity of unique identifier (UID) values in these clusters. The k value is the required minimum number of UIDs per cluster. Therefore, a larger UID diversity means a larger number of distinct individuals that protect each other from information disclosure. In contrast, very large clusters with a

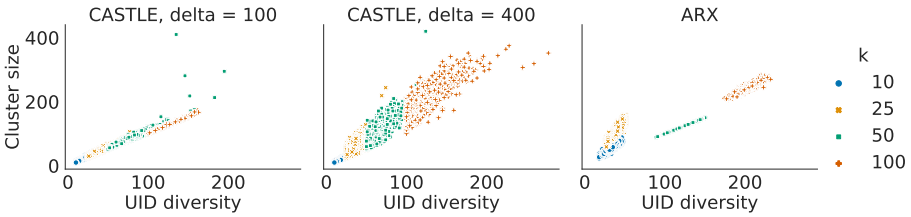

Fig. 3: Cluster size [tuples] in relation to the UID diversity.

low UID diversity indicate that many data tuples correspond to the same individuals. This could compromise privacy as a person may have similar consumption values, resulting in low diversity of consumption values and potentially disclosing information.

Figure 3 shows the cluster size in tuples against the UID diversity for different values of k and δ. For ARX, we observe that the UID diversity of most clusters is between $2 \cdot k$ and $2.5 \cdot k$. Moreover, the clusters generated with ARX are about the size of their UID diversity.

For CASTLE, we observe that δ significantly influences the cluster sizes. For better visibility, we excluded a few clusters that were larger than 500, which were most likely caused by an unfavorable combination of tuples due to an expiring δ. In Figure 3, the cluster size increases with larger δ, while the range of UID diversity remains about the same. We suspect that this is caused by the nature of the dataset. The extracted sample includes about one-third of the available data points, i.e., measurements of about 120 clients per time point, and each client appears on average 1-2 times per hour. One hour corresponds to approx. 490 data points. Thus, for $\delta = 100$, each client that appears has about 1 data point in the sliding window when the clusters are created. Consequently, the cluster size and UID diversity are about the same. For $\delta = 400$, the sliding window can contain several tuples per client. In this case, multiple time points belonging to the same client, are mostly included in the same cluster, resulting in larger clusters with the same UID diversity. This is also reflected by our information loss analysis of the *time* above.

Consumption Range. Next, we analyze the diversity and distribution of the sensitive attribute, i.e., electricity consumption. Our initial analysis showed that almost all settings have a diversity of the sensitive attribute that is at least approximately equal to the UID diversity suggesting a high level of privacy protection. However, we do not include the results in this paper since diversity was designed for categorical, but not numerical attributes. It particularly does not take the range or similarity of values into account as was previously described in [6].

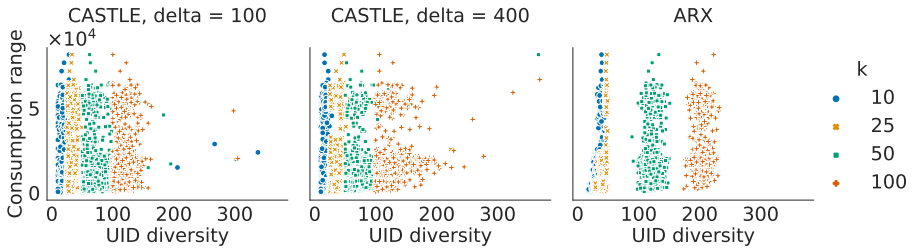

Fig. 4: Consumption range against UID diversity per cluster.

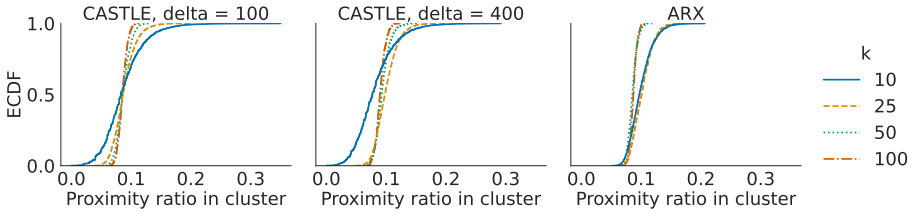

Fig. 5: Average proximity ratio of tuples in clusters.

Instead, we consider the range e of the sensitive attribute in the clusters inspired by (k, e)-anonymity [12]. Figure 4 shows the consumption range against the UID diversity. We see that k and the UID diversity only slightly influence the consumption range for CASTLE. Indeed, a certain UID diversity exhibits all different ranges of the sensitive attribute.

The same applies for ARX, independent of k the clusters exhibit all different ranges. The consumption of individual households is expected to be in smaller ranges typical for the number of members in a household. Compared to that, larger clients such as schools or industry have larger consumption with more variance. The results in Figure 4 suggest that different types of clients are included in many clusters for both processing strategies.

Consumption Proximity. Information about the range does not capture the distribution of the sensitive attribute. We therefore analyze the difference between neighboring consumption values in a cluster by analyzing their relative ϵ-neighborhood with $\epsilon = 0.2$, as described in [6]. We calculate the *proximity ratio* as the average percentage of tuples in a cluster that have other tuples in this cluster within 0.2-neighborhood. This could facilitate a proximity breach, which means that an attacker can infer that the sensitive attribute lies within a small interval [6].

Figure 5 shows the distribution of these values as empirical cumulative distribution plots. The larger the k, the fewer tuples are in 0.2-neighborhood of each other, indicating better privacy protection since the values of the sensitive attribute are less similar within a cluster. We observe no substantial difference between the results obtained with CASTLE and ARX, for $k = 10$,

the clusters generated by CASTLE show even less proximity than those of ARX. This means that the privacy obtained with the sequential k_s-anonymization is comparable to the global anonymization realized with ARX.

5 Conclusion

In this paper, we analyzed the suitability of k_s-anonymity for smart meter data in a centralized architecture. Our results suggest that the continuous k_s-anonymization with CASTLE is comparable to a global anonymization with ARX. Therefore, we consider k_s-anonymity as a reasonable approach for smart meter data anonymization. The exact influence of certain parameters, such as window size, require further research in order to find optimal settings for specific use cases. Additionally, the constraints of numerical data such as electricity consumption must be considered and suitable metrics for the evaluation of the privacy of anonymized data have to be chosen, for instance, the analysis of proximity instead of "pure" diversity.

References

1. Arx homepage, `https://arx.deidentifier.org/`, last accessed 14 June 2023
2. Cao, J., Carminati, B., Ferrari, E., Tan, K.: CASTLE: continuously anonymizing data streams. IEEE Trans. Dependable Secur. Comput. **8**(3), 337–352 (2011)
3. Dwork, C.: Differential privacy in new settings. In: SODA 2010 (2010)
4. Guo, K., Zhang, Q.: Fast clustering-based anonymization approaches with time constraints for data streams. Knowledge-Based Systems **46**, 95–108 (2013)
5. Iyengar, V.S.: Transforming data to satisfy privacy constraints. In: ACM SIGKDD 2002. pp. 279–288 (2002)
6. Li, J., Tao, Y., Xiao, X.: Preservation of proximity privacy in publishing numerical sensitive data. In: ACM SIGMOD 2008. p. 473–486 (2008)
7. Mohamed, M.A., Nagi, M.H., Ghanem, S.M.: A clustering approach for anonymizing distributed data streams. In: ICCES 2016. pp. 9–16. IEEE (2016)
8. Pallas, F., Legler, J., Amslgruber, N., Grünewald, E.: Redcastle: practically applicable k_S-anonymity for iot streaming data at the edge in node-red. In: M4IoT@Middleware 2021. pp. 8–13. ACM (2021)
9. Robinson, A., Brown, F., Hall, N., Jackson, A., Kemp, G., Leeke, M.: Castleguard: Anonymised data streams with guaranteed differential privacy. In: DASC/PiCom/CBDCom/CyberSciTech 2020. pp. 577–584. IEEE (2020)
10. Sweeney, L.: k-anonymity: A model for protecting privacy. Int. J. Uncertain. Fuzziness Knowl. Based Syst. **10**, 557–570 (2002)
11. Yang, L., Chen, X., Luo, Y., Lan, X., Wang, W.: Idea: A utility-enhanced approach to incomplete data stream anonymization. Tsinghua Science and Technology **27**(1), 127–140 (2022)
12. Zhang, Q., Koudas, N., Srivastava, D., Yu, T.: Aggregate query answering on anonymized tables. In: IEEE ICDE 2007. pp. 116–125 (2007)