ProTECting: An Application of Local Differential Privacy for IoT at the Edge in Smart Home Scenarios

Israel C. Vidal[1], André L. C. Mendonça[1], Franck Rousseau[2], Javam C. Machado[1]

[1]Laboratório de Sistemas e Banco de Dados – Universidade Federal do Ceará (UFC)
Fortaleza – CE – Brazil

[2]Université Grenoble Alpes, CNRS, Grenoble INP, LIG
Grenoble – Rhône-Alpes – France

{andre.luis, israel.vidal, javam.machado}@lsbd.ufc.br

Franck.Rousseau@imag.fr

Abstract. *With the growth of the Internet of Things (IoT) and Smart Homes, there is an ever-growing amount of data coming from within people's houses. These data are valuable for analysis and to discover patterns in order to improve services and produce resources more efficiently, e.g., using smart meter data to generate energy with less waste. Despite their high value for analysis, these data are intrinsically private and should be treated carefully. IoT data are fundamentally infinite, and this property makes it even more challenging to apply conventional models to achieve privacy. In this work, we propose a differentially private strategy to estimate frequencies of values in the context of Smart Home data, considering the infinite property of the data and focusing on getting better utility than state of the art.*

1. Introduction

With the popularization of the Internet of Things (IoT) and the greater availability of various kinds of sensors in the market, there is an increasing amount of data being generated. It is expected that by the year 2021, the amount of data generated by IoT devices, people, and machines reaches the magnitude of zettabytes [Networking 2016]. These data can be beneficial for improving services, for example, by using Smart Meters data to gain a better understanding of the energy usage in a city or using health sensors to recognize activities through health monitoring.

Most of the IoT data arrives as streaming data, which brings some challenges due to its intrinsic characteristics, i.e., the data is potentially unbounded and happens in a non-predictable order. Besides that, it requires fast processing, which turns the traditional solution of sending data to be processed at a cloud server unfeasible. An alternative for cloud computing is to process data at the edge of the network, which motivates the emergence of edge computing [Shi et al. 2016].

Although these data can be beneficial for improving services, careful attention to privacy issues is becoming more urgent. The lack of care with the individual's privacy can lead to severe problems, such as a malicious edge server eavesdropping sensitive data. As shown in [Molina-Markham et al. 2010], through relatively simple statistical methods, an adversary is able to identify crucial private information, such as if the household has left her child home alone or even if her last breakfast was eaten hot or cold.

Many works in the literature implement privacy through a trusted third party entity that has access to the raw data of a set of users. However, in real-world scenarios, it is often not reasonable to depend on such an entity for privacy reasons. The third-party entity may be malicious or be hacked and expose users' personal information, for example. More recent works focus on the local perspective of privacy, where the privacy process is performed closer to the user without depending on a trusted third party entity.

In Differential Privacy (DP) [Dwork 2006], a mechanism $\mathcal{M}$ is said to be differentially private if the probability of any output of $\mathcal{M}$ does not vary significantly, by a threshold of ε, independent of the input. DP was initially proposed to work as an interactive model [Dwork et al. 2006], responding privately to statistical queries in a database. In this interactive scenario, a trusted entity that has access to the raw data is necessary. However, in more recent work, such as [Erlingsson et al. 2014], there has been significant interest in the local setting of this model, called Local Differential Privacy (LDP), where a randomization process is done locally to ensure the definition of DP.

In this paper, we describe ProTECting, an application of privacy protocols that guarantees the Local Differential Privacy properties for estimating frequencies of values in the context of Smart Homes at the edge over infinite streams. One of the reasons why infinite streams make it more challenging to achieve the Differential Privacy property is that a priori, the privacy budget should be consumed by each interaction of the randomization mechanism, which could make it inapplicable for multiple uses unless one works through this issue. Differently from state of the art, ProTECting reaches better utility levels by leveraging optimized privacy protocols. In this context, utility is measured by how close the estimated frequencies are to the real ones. An optimized protocol is one that minimizes the variance of the outputs, therefore resulting in outputs with less error and better utility.

The outline of this paper is defined as follows. Section 3 discuss the state of the art, focusing on the most similar approaches regarding DP, IoT, and edge computing. Section 2 explains the theoretical background about the networking environment, related to IoT and edge computing, and the DP settings. In Section 4, we present ProTECting as a solution to ensure privacy in IoT scenarios at the edge. To evaluate the performance of ProTECting, Section 5 reports the preliminary experimental results, applied on real sensor data from Smart Meters [UK Power Networks 2015]. Finally, Section 6 presents conclusions and gives future research directions.

2. Smart Homes and Data Privacy

2.1. Edge Computing and Internet of Things

Internet of Things (IoT) is a recent communication paradigm related to the interconnection of everyday objects to the Internet, which has strengthened with the evolution of advanced wireless technologies. The basic idea behind IoT consists of a variety of physical objects provided with embedded systems, capable of interacting with each other and with the users, which turns the Internet even more immersive and pervasive.

The use and interaction of these interconnected objects, such as home appliances, surveillance cameras, monitoring sensors, machines, and more, which produces and transports data, leads to the development of a variety of applications that make use of this enormous amount of generated data. This variety of applications beforehand mentioned may

find heterogeneous domains, such as home and industrial automation, energy and traffic management, and more.

However, in this complex scenario, the enormous amount of information generated by objects and transported through the network alongside the heterogeneous fields of application leads to challenging issues. These issues are comprised not only by networking ones but also by privacy issues regarding the users, which could have their privacy breached by an adversary, e.g., a malicious cloud server, that collects and analyzes data produced by objects in a house in order to know when the house is empty. Examples of networking issues are scalability and complexity in the system perspective due to the need for a well-established network to maintains the IoT, services, and devices network altogether. Regarding privacy concerns, it is closely related to the lack of care within the gathered information related to daily human activities, which, combined with relatively simple statistical methods, could reveal crucial private information, as previously mentioned in Section 1.

Smart homes are a particular scenario of IoT, where network-connected objects are located inside a house. A well-known application over smart homes consists of the use of smart meters to measure and collect the energy consumption in a house. However, since the information provided by devices are very sensitive, the household's daily activities and behaviors can be revealed. It is desired that the smart meters report the households' bills without revealing how the energy was used [Molina-Markham et al. 2010], as it could breach excessive information that allows the energy provider, or an adversary, to discover, for example, if any household watched TV on a given night. Nevertheless, smart meter data collection may be beneficial to the costumers since the provider may learn from collective energy use to distribute and generate energy more efficiently. Providers also need to periodically get the actual energy consumption of each house to bill the customer. Therefore our approach does not interfere in the consumption information collected by providers to charge costumers. Instead, we apply our data privacy techniques to the streaming collection of energy use for data analysis by energy providers.

Edge Computing is a computing paradigm that brings the data processing closer to where it is needed, at the edge of the network. In other words, instead of processing data inside the cloud, all data processing is done locally at the edge [Shi et al. 2016]. In smart home scenarios, edge computing can be seen in a home appliance that gathers and process the data coming from all connected smart devices within the house. All data gathered are processed through a sanitation algorithm before being sent to the entity of interest. This flow is strictly recommended, as it does not depend on a hypothetical trusted entity responsible for treating the raw data. In section 4, the home appliance will be called edgeBox. These two terms are used interchangeably in this paper.

2.2. Differential Privacy

Differential Privacy (DP) is a mathematical model proposed by Dwork [Dwork 2006], which gives strong privacy guarantees. It ensures that the probability of any output of $\mathcal{M}$ does not vary significantly, by a threshold of ε, independent of the presence or absence of any individual in the data set [Dwork et al. 2014] that is, the addition or removal of an individual will not substantially affect the outcome of any statistical analysis performed in the data set [Domingo-Ferrer et al. 2016]. Thus an adversary should not be able to

learn anything about a specific individual that he could not have learned without access to the data set. Differential Privacy is defined as follows:

Definition 1. A randomized algorithm (mechanism) $\mathcal{M}$, gives ε-Differential Privacy (ε-DP) if for all data sets D_1 and D_2, differing on at most one individual, and all $S \subseteq$ Range($\mathcal{M}$),

$$\frac{Pr[\mathcal{M}(D_1) \in S]}{Pr[\mathcal{M}(D_2) \in S]} \leq e^{\varepsilon},$$

where the probability is taken over the randomness of $\mathcal{M}$.

There are several ways to achieve DP [Dwork 2008], that is, making the outputs of two different neighboring databases (differing on at most one individual) computationally indistinguishable, as stated in Definition 1. It is often based on the addition of noise to the real answer. A mechanism $\mathcal{M}$ which satisfies ε-Differential Privacy is independent of the computational power or even external information acquired by an adversary, offering then a very powerful and strong privacy guarantee.

This model was initially proposed to work in an interactive way [Dwork et al. 2006], responding privately to statistical queries in a database. In this interactive scenario, a trusted entity that has access to the raw data is necessary. However, in more recent work, such as [Erlingsson et al. 2014, Wang et al. 2017], there has been significant interest in the local setting of this model, known as Local Differential Privacy (LDP), where the randomization process is done locally by the user to ensure the definition of DP.

Definition 2. A randomized algorithm (mechanism) $\mathcal{M}$, gives ε-Local Differential Privacy (ε-LDP) if for all pairs of values v_1 and v_2 and all $S \subseteq$ Range($\mathcal{M}$),

$$\frac{Pr[\mathcal{M}(v_1) \in S]}{Pr[\mathcal{M}(v_2) \in S]} \leq e^{\varepsilon},$$

where the probability is taken over the randomness of $\mathcal{M}$.

Notice that LDP is a particular case of DP where the inputs of $\mathcal{M}$ are values, instead of data sets. Therefore, two neighboring inputs are two possible values, instead of data sets differing on at most one individual. LDP mechanisms, also known as protocols, are often based on the Randomized Response (RR) process [Warner 1965], which was formerly proposed in the context of surveys. RR has as objective to provide plausible deniability to questions that may harm participants in a survey. RR was later proved to guarantee the definition of ε-LDP [Dwork et al. 2014] and has been used as a foundation for almost every protocol in the context of Local Differential Privacy [Erlingsson et al. 2014, Wang et al. 2017].

3. Related Work

The authors of [Molina-Markham et al. 2010] tackle the problem of privately charging energy consumption. In addition to proposing a statistical procedure capable of identifying house activities in fine-grained measurements, showing that there must be a meticulous

privacy procedure to make use of smart meter data, they describe a protocol that allows smart meters to report a bill without revealing how the energy was used. The procedure uses cryptography and zero-knowledge proof to guarantee that the company will not have access to the information from which house comes a given data, even though the company will be able to charge for the energy used. A downside of this work is that the server still has access to what they call *blinded data*, which consists of the data from all houses with the identification removed, and this is not enough for guaranteeing privacy [Sweeney 2002].

The work [Ács and Castelluccia 2011] uses differential privacy to deal with the problem of using consumption data to learn privately about users. The approach is based on the Laplace Mechanism, which adds a noise sampled from a Laplace distribution to the result of a numerical query. The authors propose a Distributed Laplace Mechanism (DLM). The information they want to learn privately in this work is the consumption summation of M houses in a given time. To do this, each house adds a small noise n to its consumption c (which is not enough to guarantee DP) and encrypts the report $r = c + n$ in a way that the server is not able to decrypt a report alone, but it is able to decrypt the summation of the reports. The server, then, has $S = \sum_{h=1}^{M} c_h + n_h = \sum_{h=1}^{M} c_h + N$, where N is a noise that follows the Laplace distribution as previously presented, i.e., that is enough to guarantee DP. This strategy is useful for learning the summation of consumption, but cannot be used to learn more information than that. This strategy also does not consider the infinite property of IoT data, focusing on finding a solution for a given timestamp.

IoT data often appear as data streams. Works that deal with the problem of guaranteeing privacy in the context of streaming data deal with additional complexity because streaming data is potentially unbounded and continuously generated at rapid rates. The work [Leal et al. 2018] proposes a strategy to estimate the sensitivity and also presents a microaggregation algorithm that is capable of enhancing the utility for publishing differentially private data using the Laplace Mechanism in the context of streaming data. This work depends on a trusted third party entity to achieve its privacy, which may not be acceptable in the context of IoT data and smart homes.

The work [Cao and Yoshikawa 2015] uses differential privacy to publish statistics about the streaming of trajectories. The objective is to publish, for a defined set of possible locations, how many people are in each location at a given time. As in the context of streaming data the order in which the data appears is unpredictable, it is not possible to know beforehand in which timestamps a given trajectory will appear. It is also not possible to know how long a trajectory can least. To overcome these difficulties, the authors use the concept of a l-trajectory, i.e., a trajectory of size l. They show that it is possible to guarantee that a l-trajectory is ε-DP. On the other hand, this work, besides depending on a third party trusted entity, need to know beforehand the set of locations, which may not be reasonable in real-world scenarios. The solution also does not consider the infinite property of IoT data, using a windowed strategy to simplify the problem.

The work [Erlingsson et al. 2014] proposes RAPPOR, a strategy to achieve ε-LDP even when a client reports infinite times over a true value. In order to guarantee ε-LDP, even if a value is reported multiple times, it uses two rounds of randomization. Suppose a client wants to report a value v. It first encodes v into a Bloom Filter B, which

is then randomized using a protocol that guarantees that the result B' is ε_∞-LDP. Then, B' is memoized and reused every time the value v needs to be reported. The first randomization process, known as Permanent Randomization, is responsible for guaranteeing what the authors call *Longitudinal Privacy*. After the memoization step, B' is randomized again, using another protocol that guarantees ε_1-LDP every time the value v is reported. This step is called Instantaneous Randomization. However, the authors show that even if multiple Instantaneous Randomization steps are executed, an attacker could learn at most the true value of B', which itself is also protected by an ε_∞-LDP mechanism. The authors do not argue how the protocols used for the Permanent and Instantaneous Randomization were chosen.

In this paper, we present ProTECting, a strategy that makes use of the Edge Computing paradigm to apply Local Differential Privacy (LDP) in the context of IoT data. ProTECting uses two rounds of randomization and memoization to guarantee that even if a user reports infinite values, their responses are still protected by the concept o LDP. In order to obtain a better utility level, ProTECting uses optimized protocols. The strategy will be detailed in Section 4.

4. ProTECting

Our approach, called ProTECting (**Pr**ivacy for **IoT** and **E**dge **C**ompu**ting**), was thought to work over an edgeOS, i.e., a specialized operating system that runs in an edge gateway, from now on called edgeBox, and manages smart devices. In this paper, we have omitted the full architecture of an edgeOS, but [Shi et al. 2016] can be checked for more details. For our proposal, what is important about an edgeOS is that there is a data abstraction layer in it that gathers data produced by devices inside a house.

Our solution works between the data abstraction layer and all external communication to guarantee that all data that goes outside the house is private. A possible exception for this is that for the Service Provider (SP) to be able to charge for the consumption, it may need to have access to coarse-grained measurements. As shown in Section 3, there are possible strategies to charge privately. Figure 1 shows the interactions between each edgeBox and an SP that wants to learn privately with data generated by devices inside the users' houses. This figure shows the Privacy Gateway layer in which ProTECting runs over.

In a recent work [Vidal et al. 2019], we have proposed a different strategy to guarantee ε-Local Differential Privacy in the context of Smart Homes, which uses the same architecture shown in Figure 1. In this previous work, we use a window-based strategy to simplify the problem in order to be able to achieve ε-LDP. The given guarantee is that every data generated inside a sliding window of a defined size of w is ε-LDP. As this strategy uses LDP, it does not depend on a trusted third party entity, which is close to the goal of the present paper. On the other side, the window-based strategy used may not be reasonable in real-world scenarios, since it does not consider the infinite property of streaming data, which is essential in Smart Homes scenarios. To attack this problem, we propose ProTECting, which considers the important property of infinite data streams and focus on utility. Notice that ProTECting and [Vidal et al. 2019] could not be directly compared, since the latter guarantees ε-LDP for a defined sliding window, while the former guarantees it for infinite reports, but since they are inserted in the same context, we

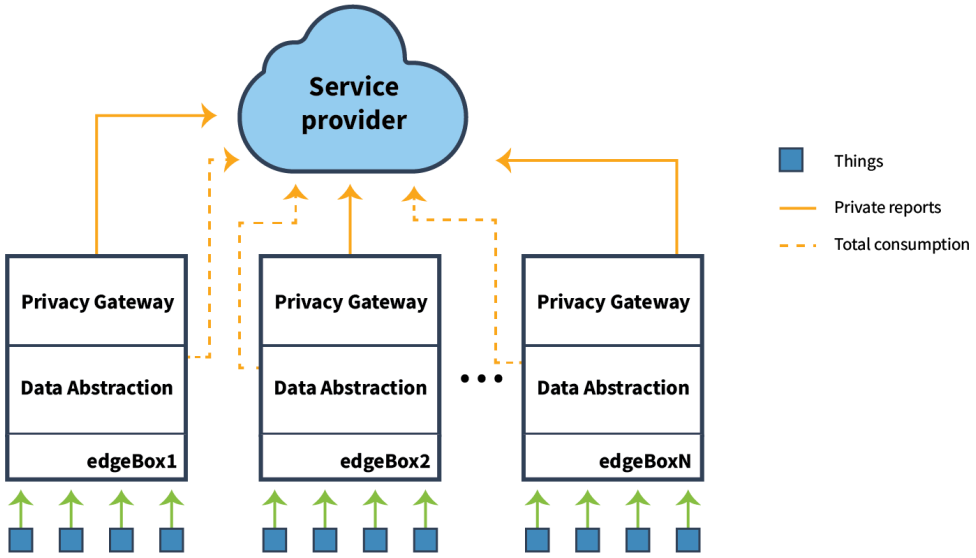

Figure 1. Communication between the Service Provider and the houses

present some of the results achieved in the previous work in Section 5.

ProTECting is built over the idea of using two randomization processes proposed in [Erlingsson et al. 2014] in order to achieve privacy over time, and has as the primary goal to improve utility. In order to obtain better utility levels, ProTECting uses optimized privacy protocols.

The authors of [Wang et al. 2017] proposes a novel protocol, called Optimized Unary Encoding (OUE), which optimizes the parameters p and q used in the Unary Encoding (UE) protocol having as objective to minimize the variance. The semantic of these parameters will be explained later in this Section. The UE protocol consists of transforming a value v into a bit array representation of it in which all bits are set to zero, except for the one representing v which is set to one. The definition of UE can be found in Definition 3.

Definition 3. $UE(v, d) = B = [0, \ldots, 1, \ldots, 0]$, where B is a bit array of size d where only the v-th position is 1.

The motivation for choosing the Unary Encoding protocol instead of encoding it in a Bloom Filter is that in this work, we focus on utility, and the justification for using Bloom Filters is to reduce communication cost at the cost of reducing accuracy [Wang et al. 2017].

The randomization process performed in the Unary Encoding protocol consists in bit-wise perturb the bit array B in the following manner:

$$B'[i] = 1 \begin{cases} \text{with probability } p, & if\ B[i] = 1 \\ \text{with probability } q, & if\ B[i] = 0 \end{cases}$$

Therefore, we can conclude that the conditional probabilities of $B'[i]$ given $B[i]$ are:

$$Pr[B'[i]|B[i]] = \begin{cases} Pr[B'[i] = 1|B[i] = 1] = p \\ Pr[B'[i] = 1|B[i] = 0] = q \\ Pr[B'[i] = 0|B[i] = 1] = 1 - Pr[B'[i] = 1|B[i] = 1] = 1 - p \\ Pr[B'[i] = 0|B[i] = 0] = 1 - Pr[B'[i] = 1|B[i] = 0] = 1 - q \end{cases}$$

As shown in [Wang et al. 2017], to minimize the variance of the UE protocol, while still guaranteeing the ε-LDP property, the values of p and q should be: $p = 0.5$ and $q = \frac{1}{e^{\varepsilon}+1}$. The proof that the Unary Encode protocol satisfies ε-LDP is known in the literature and can be found in [Erlingsson et al. 2014, Wang et al. 2017]. The Optimized Unary Encoding protocol, which consists of the UE with the probabilities set to the previously mentioned values, is a special case of UE and, therefore, follows the same proof.

As presented in Definition 3, the UE strategy can be directly applied for integer values, which may not be true in the context of IoT, since IoT data is likely to be composed of real numbers. In order to work around this issue, we use a discrete representation of the values, which can be seen as a histogram representation of them. For having this discrete representation, it is necessary to have a defined number of bins (d) and a range of values (min_value, max_value). Therefore, $Hist\_UE$ shown in Line 3 of Algorithm 1 outputs a bit-array of size d with the corresponding bin of value v set to one. The range values are supposed to be known in advance, and the choice of (d) could impact the utility obtained.

Algorithm 1: PROTECTING

Input: $v, \varepsilon_1, min\_value, max\_value, d$
Output: S

1 $p \leftarrow 0.5$
2 $q \leftarrow \frac{1}{e^{\varepsilon_1}+1}$
3 $B[v] \leftarrow Hist\_UE(v, min\_value, max\_value, d)$
4 **if** $B'[v] == \emptyset$ **then**
5 $\quad$ PERMANENT-RANDOMIZATION$(B[v], p, q)$
6 $S \leftarrow$ INSTANTANEOUS-RANDOMIZATION$(B'[v], p, q)$
7 **return** S

The Algorithm 1 is performed for each value that needs to be sent to a cloud server, and is detailed as follows. It receives as input a value v, the privacy budget ε_1, the range of values *min_value*, *max_value* and the number of bins d used in the UE representation. The output S is an anonymized bit-array of v. Lines 1-2 sets the optimal probability values of p and q regarding ε_1 based on [Wang et al. 2017]. Line 3 converts the input value v to its UE histogram representation. Line 4 checks if already exist the anonymized memoized version $B'[v]$ of $B[v]$, i.e., the same value v has previously been reported for the first time, otherwise $B'[v]$ must be generated once through the Permanent Randomization step. Finally, Line 6 calculates the output S through the Instantaneous Randomization step, which uses the memoized bit-array B', and returns it in Line 7.

Notice that the Permanent Randomization step is ε_1-LDP because it uses the OUE protocol. Next, we show that the Instantaneous Randomization (IR) step guarantees ε_2-LDP.

Theorem 1 (Instantaneous Randomization step satisfies ε_2- DP)**.**

Proof. To prove that the IR step is ε_2-LDP, we need the conditional probabilities of $S[i]$ given $B[i]$. Notice that to calculate these probabilities, we need to consider the Permanent Randomization step and its conditional probabilities given B. Those probabilities were presented earlier in this section. We also need $Pr[S[i]|B'[i]$, which is given by:

$$Pr[S[i]|B'[i]] = \begin{cases} Pr[S[i]=1|B'[i]=1] = p \\ Pr[S[i]=1|B'[i]=0] = q \\ Pr[S[i]=0|B'[i]=1] = 1 - Pr[S[i]=1|B[i]=1] = 1-p \\ Pr[S[i]=0|B'[i]=0] = 1 - Pr[S[i]=1|B[i]=0] = 1-q \end{cases}$$

Therefore, $Pr[S[i]|B[i]]$ is calculated as follows:

$$\begin{aligned} Pr[S[i]=1|B[i]=1] = Pr[B'[i]=1|B[i]=1] * Pr[S[i]=1|B'[i]=1] + \\ Pr[B'[i]=0|B[i]=1] * Pr[S[i]=1|B'[i]=0] \end{aligned} \tag{1}$$

$$\begin{aligned} Pr[S[i]=1|B[i]=0] = P[B'[i]=0|B[i]=0] * P[S[i]=1|B'[i]=0] + \\ P[B'[i]=1|B[i]=0] * P[S[i]=1|B'[i]=1] \end{aligned} \tag{2}$$

$$Pr[S[i]=0|B[i]=1] = 1 - Pr[S[i]=1|B[i]=1] \tag{3}$$

$$Pr[S[i]=0|B[i]=0] = 1 - Pr[S[i]=1|B[i]=0] \tag{4}$$

Let $p^* = Pr[S[i]=1|B[i]=1]$ and $q^* = Pr[S[i]=1|B[i]=0]$. In order to prove that this protocol satisfies Local Differential Privacy, we need to show that $\frac{Pr[S=s|B=B_1]}{Pr[S=s|B=B_2]} \leq e^{\varepsilon_2}$. This is shown as follows:

$$\frac{Pr[S=s|B=B_1]}{Pr[S=s|B=B_2]} = \frac{\prod_{i=1}^{d} Pr[s[i]|B_1[i]]}{\prod_{i=1}^{d} Pr[s[i]|B_2[i]]} \tag{5}$$

$$\leq \frac{Pr[s[v_1]=1|B_1[v_1]=1]Pr[s[v_2]=0|B_1[v_2]=0]}{Pr[s[v_1]=1|B_2[v_1]=0]Pr[s[v_2]=0|B_2[v_2]=1]} \tag{6}$$

$$= \frac{p^*(1-q^*)}{q^*(1-p^*)} = e^{\varepsilon_2} \tag{7}$$

Equation 5 comes from the fact that each bit in S is perturbed independently.

The bit arrays B_1 and B_2 come from the real values v_1 and v_2, respectively. Notice that these two arrays differ only in positions v_1 and v_2. Keeping the positions v_1 and v_2 of S as one and zero, respectively, maximizes the ratio in Equation 6.

Finally, Equation 7 comes from the probabilities presented in Equation 1-4. □

Observe that the Permanent and the Instantaneous Randomization steps both use the Optimized Unary Encoding protocol, having as the difference the input bit array, which gives to each protocol a different level o privacy (ε). Notice that ε_2 is limited by ε_1 since the IR uses as input the output of the PR.

The next and last step remains for the Service Provider to estimate the frequencies of each bin using the randomized reports received. The unbiased estimation is obtained using Equation 9, which was shown in [Wang et al. 2017] to yield an unbiased estimation. The frequency estimation will be better explained later in Section 5.

5. Experimental Results

This section describes the experiments conducted to evaluate the accuracy of ProTECting in terms of utility. We have considered the Histogram Intersection (Equation 8) between the real histogram and its privatized version as the utility metric.

$$Hist\_Intersec(Hist, Unb\_Hist) = \frac{\sum_{i=1}^{d} min(Hist[i], Unb\_Hist[i])}{\sum_{i=1}^{d} Unb\_Hist[i]} \tag{8}$$

Section 5.1 shows the experimental setup and detailed information about the data set used. Section 5.2 assesses the data utility of ProTECting by quantifying the impact of the ε parameter and, then, compares the obtained results with RAPPOR, the baseline. It also presents results for the window based strategy proposed in [Vidal et al. 2019]. As mentioned in Section 4, this strategy cannot be directly compared with the one proposed in this paper, since it presents privacy guarantees for a simplified problem. Therefore, its results are presented just as a guideline.

5.1. Experimental Setup

Our approach, ProTECting, the baseline, RAPPOR, and the window-based strategy were all implemented in Python 3.6, running on a desktop machine with Ubuntu 18.04 OS, Intel Core i5 (3.2 GHz) processor and 16GB of RAM. We have used real sensor (smart meters) data that consists of energy consumption readings from 5,567 London households generated between 2011 and 2014 as part of the Low Carbon London project, resulting in 167 million rows. We have used the attribute "KWH/hh (per half hour)" to report values.

5.2. Utility Evaluation

As stated above, we have adopted the histogram intersection as the metric to evaluate our proposal, since it is a suitable strategy to understand how close the original data is from its private version. This metric is also robust to negative frequencies due to the

use of the Unbiased Estimator (Equation 9), which is used to decode, i.e., estimate, the frequencies from the reported values. In this estimator, $Hist'[i]$ denotes the frequency of the i^{th} bin from the reported values within an anonymization protocol. R is the total number of reports, given by $N * k$, and p^* and q^* are the probabilities in function of ε_1, as previously defined in Section 4. Thus, as we are interested in frequencies of values, negative frequencies do not make sense and, then, are set to zero, not affecting the utility measure.

$$Unb\_Hist[i] = max(0, \frac{Hist'[i] - R * q^*}{p^* - q^*}) \tag{9}$$

In order to better evaluate our proposal, we have sampled random rows from the data set to simulate a fixed number of $N = 1,000$ houses sending data collected from their edgeBoxes. We stated the number of reports k that each house sends as being 1,000. This value was arbitrarily chosen given that this work is inserted in the context of infinite streams, and, in real-world applications, each report may be sent hourly, weekly, monthly, or so on. Then, 1,000 reports is a reasonable number to simulate data streaming over Smart Homes.

The number of bins d used to define the size of the UE histogram, as shown in Definition 3, was fixed in 100. Setting a very low value for d ruins the quality of the data since almost all values will be represented by the same bit in the UE representation. Oppositely, a very high value for d excessively fragments the data, as the value representation using UE will become too sparse, being necessary massive quantities of reports to obtain useful information. Hence, choosing the proper value for d may be a challenging task. At last, the range of the reported values is comprised between zero and 10.76. This information is required to properly define the range of each bin from the UE histogram representation.

Figure 2 compares our approach, ProTECting, and the baseline, RAPPOR, in terms of utility by varying the privacy budget parameter ε_1. Since the IR step uses the same parameters p and q used in the PR step, the value of ε_2 of a report is defined in terms of ε_1 and can be calculated by Equation 7 in the proof of Theorem 1. The values of ε_2 for each ε_1 are presented in Table 1. Figure 2 also shows the utility for the window-based strategy, for a window of size 10.

ε_1	ε_2
1.00	0.23
2.00	0.82
3.00	1.63
4.00	2.55
5.00	3.51

Table 1. Value of ε_2 in terms of ε_1.

As can be observed, the histogram intersection obtained by ProTECting performed better utility levels in comparison to RAPPOR for all ε_1 values, meaning that the frequencies estimated through the unbiased estimator are closer to the real frequencies. This is

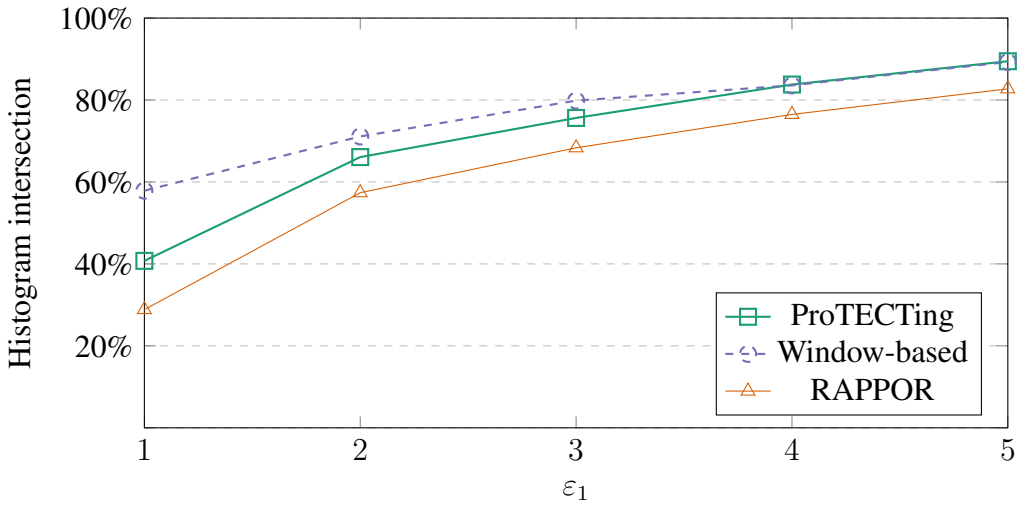

Figure 2. Histogram intersection varying ε_1 parameter.

expected since our approach makes use of an optimized protocol, the OUE, differently from RAPPOR. The OUE intuition is to maintain the maximum amount of bits in the UE representation of an input value unchanged after randomization steps. Since we are adopting the UE representation, where only one bit is set to one and the remaining $d-1$ bits are set to zero, the OUE uses probability values of p and q that maximizes the number of bits reported as zero that was initially zero.

Notice that the histogram intersection of both approaches increases as the ε_1 parameter increases. This is the expected behavior of the DP and LDP models. Higher values of ε_1 result in higher utility levels and, oppositely, lower values of ε_1 results in higher levels of privacy. Therefore, choosing the proper value of ε_1 may be a challenging task. Observe that as the value of ε_1 becomes higher, ProTECting's utility comes closer to the window-based strategy, which means that even though the problem solved in this paper is more complex than in the window-based one, it can still achieve the same level of utility as our guideline, while giving stronger privacy guarantees, since it does not depend on the window size. Remember that ProTECting would still hold the ε-LDP guarantee even when infinite reports are sent, while the window-based strategy only holds its guarantee for a defined number of reports.

6. Conclusion

This paper presented ProTECting, a practical solution for solving the privacy issue in Smart Homes, taking into consideration the infinite property of IoT data. ProTECting makes use of the Edge Computing paradigm and of the concept of Local Differential Privacy. It runs in the Privacy Gateway, over the Data Abstraction layer, and uses two differentially private optimized protocols to give a formal privacy guarantee even when infinite reports are sent by a user while keeping a good utility level. ProTECting achieves better utility than the baseline and, as the available budget increases, becomes closer to our guideline, which is the window-based strategy. As future work, we could mention the evaluation of ProTECting using different metrics and experimentation with other real-world sensor data.

Acknowledgments

This research was supported by CAPES (grant 88882.454571/2019-01), FUNCAP (grant IR7-0126-00041.01.00/17) and LSBD/UFC - Brazil.

References

Ács, G. and Castelluccia, C. (2011). I have a dream!(differentially private smart metering). In *International Workshop on Information Hiding*, pages 118–132. Springer.

Cao, Y. and Yoshikawa, M. (2015). Differentially private real-time data release over infinite trajectory streams. In *2015 16th IEEE International Conference on Mobile Data Management*, volume 2, pages 68–73. IEEE.

Domingo-Ferrer, J., Sánchez, D., and Soria-Comas, J. (2016). Database anonymization: Privacy models, data utility, and microaggregation-based inter-model connections. *Synthesis Lectures on Information Security, Privacy, & Trust*, 8(1):1–136.

Dwork, C. (2006). Differential privacy. In *Automata, Languages and Programming: 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II*, pages 1–12. Springer.

Dwork, C. (2008). Differential privacy: A survey of results. In *International Conference on Theory and Applications of Models of Computation*, pages 1–19. Springer.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006). Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer.

Dwork, C., Roth, A., et al. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407.

Erlingsson, Ú., Pihur, V., and Korolova, A. (2014). Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pages 1054–1067. ACM.

Leal, B. C., Vidal, I. C., Brito, F. T., Nobre, J. S., and Machado, J. C. (2018). δ-doca: Achieving privacy in data streams. In *Data Privacy Management, Cryptocurrencies and Blockchain Technology*, pages 279–295. Springer.

Molina-Markham, A., Shenoy, P., Fu, K., Cecchet, E., and Irwin, D. (2010). Private memoirs of a smart meter. In *Proceedings of the 2nd ACM workshop on embedded sensing systems for energy-efficiency in building*, pages 61–66. ACM.

Networking, C. V. (2016). Cisco global cloud index: Forecast and methodology, 2016–2021. *White paper. Cisco Public, San Jose*.

Shi, W., Cao, J., Zhang, Q., Li, Y., and Xu, L. (2016). Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5):637–646.

Sweeney, L. (2002). k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570.

UK Power Networks (2015). SmartMeter Energy Consumption Data in London Households. `https://data.london.gov.uk/dataset/`

smartmeter-energy-use-data-in-london-households. Accessed: 2019-06-28.

Vidal, I. C., Rousseau, F., and Machado, J. C. (2019). Achieving differential privacy in smart home scenarios. In *Anais do XXXIV Simpósio Brasileiro de Banco de Dados*, pages 211–216. SBC.

Wang, T., Blocki, J., Li, N., and Jha, S. (2017). Locally differentially private protocols for frequency estimation. In *26th {USENIX} Security Symposium ({USENIX} Security 17)*, pages 729–745.

Warner, S. L. (1965). Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69.