This is a pre-print version, the original appeared in IEEE International Workshop on Computer-Aided Modeling Analysis and Design of Communication Links and Networks (IEEE CAMAD 2014),
published by IEEE, http://dx.doi.org/10.1109/CAMAD.2014.7033231

Aggregation and Perturbation in Practice: Case-Study of Privacy, Accuracy & Performance

Henrich C. Phls, Max Mssinger, Benedikt Petschkuhn, Johannes Rckert

Chair of IT-Security
Institute of IT-Security and Security Law (ISL),
University of Passau, Germany
hp@sec.uni-passau.de

Abstract. We analyse accuracy, privacy, compression-ratio and computational overhead of selected aggregation and perturbation methods in the Internet of Things (IoT). We measure over a real-life data set of detailed energy consumption logs of a single family household. We modelled privacy by simple, threshold-driven machine-learning algorithms that extract features of behaviour. The accuracy of those extraction is used as privacy metric. We state for different parameters of the aggregation, reduction and perturbation if the output still allows detections, as this follows the EU's data protection principle of "minimisation": increased privacy due to less detailed data, but still good enough accuracy for the purpose. The result is that many detections for sensible predictions and intelligent reactions are still possible with lower quality data.

Keywords: IoT, Data Aggregation, Perturbation, Privacy

1 Introduction

The IoT opens up a whole new dimension of data gathering, processing and utilisation. The temporal and spatial coverage is foreseen to be on an unprecedented scale. The amount of data which could be accumulated in an environment enriched with ubiquitous, unobtrusive computing devices is virtually immeasurable [6, p.336]. Aggregation of measurements before transmitting is a technique commonly proposed to limit not only the amount of data that is transferred onwards to the next hop, but also this techniques has been suggested to decrease the privacy sensitivity of data.

Provided that a lot of Internet-of-Things (IoT) devices currently reaching the markets are focussed on the Smart Home [1], this work is concerned with the possibility to deduct behavioural patterns from the energy readings gathered for in-house circuits. Current market ready IoT deployments gather data at a few central places, e.g., energy consumption at smart meters, needing only the deployment of few devices. Still, new applications shall be able to evolve based on top of that data, e.g., provide an intelligence and self-adapting home environment learning from the energy patterns.

This case study is mainly motivated by the fact that under EU privacy laws the data gathered must be "necessary for the performance of a contract to which the data subject is party" [2]. We wanted to know if we really need the high precision in which the IoT could gather data. We applied and evaluated different parameters for aggregation and perturbation on a real-life data set in order to find what level of reduced data quality and hence additional privacy we could achieve. Alongside, aggregation yields compression. Privacy, in this paper is not geared towards disguising the identity of the datas subject, but rather towards lowering data quality to the bare "necessity" [2] to suit a given purpose of an application. Purpose is based on European legislation, e.g., [5], meaning that getting data such that an application can learn and forecast behavioural patterns, like detecting and then deducting that you are usually at home between 12-16 on saturdays and sundays, but away on weekdays, can be a legitimate purpose, e.g. to adjust your heating system and schedule your parcel delivery. Hence, a data subject could give their informed consent to just that purpose. However, the question 'How low can the granularity and data quality become such that the application still works?' was still unanswered [10].

Scope and Methodology: In this work we present a case study on electrical energy consumption data. According to M. Jawurek [7, p. 80], aggregation can be applied on three different dimensions: spatial, temporal or arbitrary. We therefore gathered several detailed energy consumption profiles of several in-house circuits of one family household and hence we will focus on temporal aggregation. As we have detailed self observation logs from the family about timing of actions, e.g. using the microwave to heat milk for the morning coffee, additionally we know what devices each circuit contains. From this we devised threshold machine learning algorithms (see Sect. 3.1) that correlate energy measurements with actions (e.g., sleep, wake, watching TV, vacation) performed. These algorithms allow identifying behavioural patterns in traces and later make assumptions on the privacy gained by aggregation and perturbation methods.

Data Set: Data was gathered in one household of a family, using in-circuit 'smart meters' measuring the energy consumption of devices connected to each electrical circuit. Each in-circuit-smart-meter sends a 'tick' on every consumed Watt hour (1 Wh) that is recorded together with a timestamp[1]. The data set contains separately the energy consumption of several circuits: (a) *living room* with a TV (approx. 100W) and several independent lights (150W in total), (b) *study room* with computers and a TV (approx. 40-70W). The data was collected over a period of seven months with around 926,000 entries.

2 Aggregation, Perturbation and Reduction

We classified the different aggregation methods we have been using into three different categories: (a) Aggregation over time, (b) Perturbation of the data with noise, (c) Reduction.

[1] based on `volkszaehler.org`

2.1 Aggregation

Aggregation is a mechanism to increase privacy by merging different single data points. It is not geared towards disguising the identity of the data's subject, but attempts to enhance privacy by lowering the accuracy of data, hereby limiting the possibility to deduce private information. According to M. Jawurek ([7],p.80), aggregation can be applied on three different dimensions: spatial, temporal or arbitrary. For this first instance of the case study we calculated the harmonic and the arithmetic mean over different time intervals.

Harmonic Mean $A = \frac{n}{\sum_{i=0}^{n} \frac{1}{x_i}}$
It showed to be tolerant towards energy peaks and offers a good accuracy. Hence, we choose the harmonic mean for aggregation.

Arithmetic Mean $A = \frac{1}{n} \sum_{i=1}^{n} x_i$
Already few peaks negatively affected the accuracy of the aggregated result in many of our cases. Hence, we did not choose an arithmetic mean.

Aggregation over time interval For the aggregation we can use the different arithmetic functions mentioned. The time interval can be adjusted to suit the application. We ran with different intervals, i.e., 10 minutes, 1, 4, 8 and 24 hours.

2.2 Perturbation

Perturbation and the reduction of resolution both aim to abstract data to a level, on which the deduction of private information can hardly be performed. The basic method of perturbation relies on the introduction of random noise (i.e. data fragments) to the data items respectively the final aggregate, causing a distortion in the original values. Adding sufficient noise to prevent an attacker from deriving data items or patterns from the result while preserving the utility of the data is challenging [7, p.74-75]. In some cases, this challenge is difficult if not impossible to overcome. For example consider perturbation on an energy profile to avoid burglary when you are away. Perturbation needs to add enough noise to prevent an attacker from differentiating whether the inhabitants are present or absent. At the same time, exact data might be needed to perform certain computations, e.g. for the purpose of billing [7]. Consequently, perturbation is only applicable if the calculations don't need to be perfectly accurate. Furthermore, random perturbation carries the risk of revealing some kind of structure within the randomness, which could be used to compromise the original data set [9].

In this case study we tuned perturbation by adding different noise. First and foremost, the parameters to identify are a suitable maximum and minimum noise to be added. Secondly, the noise can be random, or pseudo-random, or following some specific distribution.

2.3 Reduction of Resolution on the Scale of Time

The reduction of resolution operates as the name implies by reducing the accuracy of the collected data, for example extending a time attribute from minutes to hours or even days. There is however a key difference in comparison to aggregation: In case of an aggregation over time, the mean of the values within a time interval is calculated and generalised over all entries within this interval. Reduction of resolution on the other hand doesn't change the values, but instead determines one timestamp within the observed time interval with which the timestamp of every entry is overwritten. So in contrast to aggregation, the measured values will be left untouched, yielding perfect accuracy. A conceivable use case would be reducing the resolution of consumption traces of a smart home before storing them externally, effectively limiting the amount of sensitive personal information that may be derived [7]. The interval is again the property that can be adjusted to suit the application.

3 Comparison Metrics and Evaluation

We compare using four metrics: (a) Feature Extraction, (b) Compression, (c) Accuracy, and (d) Computational Overhead.

3.1 Feature Extraction as one Metric fro Privacy

We used simple feature extraction algorithm to detect (1) if inhabitants are present and (2) if a certain device is used.

Presence Detection: The algorithm detecting presence on our energy consumption data set is based on comparing the average consumed energy over defined time intervals. It starts with a one week training phase over data for which the inhabitants indicated their presence. Then, it iterates over the whole data set using the defined interval as step-size. In each step the algorithm checks if a part of the interval features an average which is greater or equal to the average determined in the training phase. In case of a hit, we assume having detected presence and mark the interval accordingly. For example, we executed it with a target of a resolution of four hours, as Fig. 1 this would allow to forecast consumption at different time intervals a day, e.g. morning, lunchtime. Fig. 2 only targets to detect the presence on a daily basis.

As the presence within the household is not reasonably detectable by utilising the data of one circuit only, we applied the algorithm on an accumulated data set including the consumption of the living and the study room. We performed the detection over intervals of 4 hours and one day, on both the original data set, as well as on an accordingly aggregated data set. As the algorithm identified presence on the original data set almost flawless, we used these results as reference. Further comparison with the algorithm's results on the aggregated data set was based on the receiver operation characteristic notation: if both mark an interval, this is a true positive (TP), the opposite is a true negative (TN). If

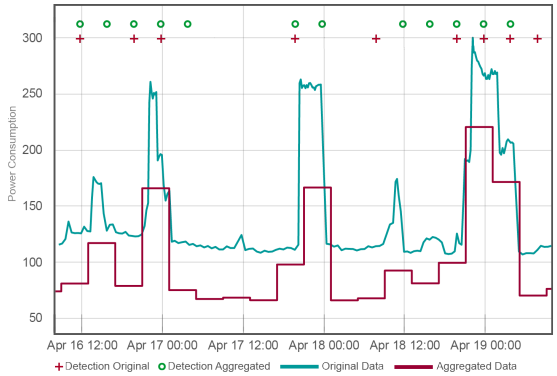

Fig. 1. Presence detection over an interval of 4 hours.

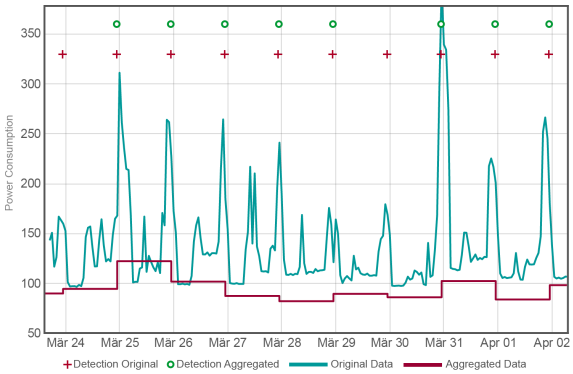

Fig. 2. Presence detection over an interval of 24 hours.

an interval is marked only by the algorithm using the original dataset, this is a false negative (FN). In case of an interval being marked only by the algorithm utilising the aggregated data, a false positive (FP) is issued. The accuracy is then computed as $\frac{TP+TN}{TP+TN+FP+FN}$.

Behavioural Detection: We implemented behaviour detection based on detecting devices being turned on. We utilised device specific power consumption signatures for the purpose of identification, for instance the TV requires between 40W and 70W while being powered on. We then matched the data with the signatures to detect occasions where this device is known to be on. From a privacy point of view, when observed over longer times this allows the derivation of behaviour patterns, e.g. reveals your favourite TV show. Fig. 3 illustrates the algorithm, the marked areas allow to easily identify the points in time when the TV has been switched on. After aggregating the data set, this method is no longer applicable, since there is no way to differentiate between distinctive power input anymore. In case of reduction of resolution this is different however,

since the power consumption as well as the sequence of events is sustained. The activation of devices can not be mapped to an absolute point in time though. To measure the privacy gain we compare the number of detected devices before and after the application of aggregation respectively perturbation. The accuracy is determined by calculating $\frac{number\_of\_dev\_detected_{after}}{number\_of\_dev\_detected_{before}}$. The resulting figure describes the percentage of devices which can still be detected in relation to the previously detectable devices.

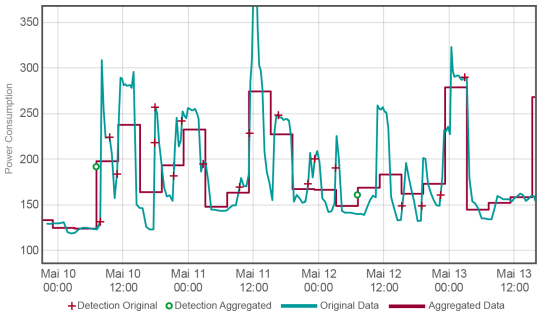

Fig. 3. Identification of SmartTV based on peak of certain height

To ensure the objectivity of our results, we also utilised an external peak detection algorithm based on Matlab, providing a well-established mathematical foundation. Thereby, a peak corresponds to a local maxima and has to be greater than its direct neighbours [3]. Semantically, a peak can be interpreted as some kind of activity. We applied the algorithm to the raw data set. Afterwards the same algorithm was executed on the data aggregated over 10 minute intervals with the harmonic mean. The results are illustrated in figure 4. Since every peak corresponds to activity, the reduction of 27 peaks to merely 2 indicates a clear privacy improvement. To estimate the privacy advantage, the formula $\frac{number\_of\_peaks_{after}}{number\_of\_peaks_{before}}$ gives the percentage of peaks in relation to the original number of peaks. Since perturbation introduces random noise, the number of peaks is increased instead of reduced. Thus $peaks_{before}$ are equal to correct peaks, while $peaks_{after}$ include numerous deceptive peaks. Consequently the quotient has to be turned around in case of perturbation, yielding the ratio of correct to incorrect peaks.

3.2 Compression Ratio

The amount of transmitted data is an important factor in the IoT. Our compression metric indicates the percentage by which the aggregation or perturbation is reducing the original data set and is calculated as $1 - \frac{number\_of\_entries_{after}}{number\_of\_entries_{before}}$.

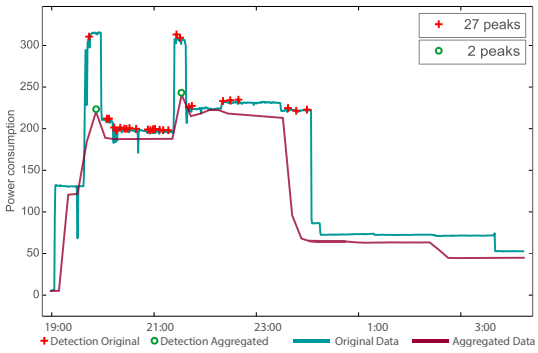

Fig. 4. Peak detection on original and aggregated data

3.3 Energy Consumption Accuracy

The data set contained timestamped ticks. So we transformed them into a different representation, by calculating: $total_{trans}(kWh) = \frac{W\_t_{now}}{1000} \cdot \frac{(t_{now}(ms) - t_{prior}(ms))}{3.600.000}$. Given the total consumption in kWh, we set the accuracy function to the difference between the original and the processed data: $accuracy = 1 - \frac{|total_{after} - total_{before}|}{total_{after}}$.

3.4 Computational Overhead

Computation time is the average over ten runs on an Intel(R) Xeon(R) CPU 5110 @ 1.60GHz single core system.

4 Evaluation

Comparing different parameters for aggregation and perturbation we check if the resulting data still allows deductions. In other words, we check if "data minimisation" [2] can take place.

Aggregation: We aggregated using the harmonic mean over different time intervals ranging from 10 minutes to 1 hours. As Fig. 5 shows, long time intervals results in far less data. Obviously, it reduces the amount of private information, but still as the 4h and 24h presence detection shows, it remains usable data, e.g., for statistical predictions in the smart grid.

Reduction: As Fig. 6 shows that the dataset with a reduced temporal resolution, i.e. 1 minute and 8 hours had no impact on the empirical accuracy. Although 8 hours are double the interval of presence detection, only a marginal impact on the presence detection is observed. It remains to be seen if this due to peculiarities of this household.

Aggregation over Time - Feature Extraction Accuracy (%)		
Presence Detection - 4 hours		97,3 %
Presence Detection - 24 hours		92,5 %
Turning-On of devices - 4 hours		10,4 %
Turning-On of devices - 8 hours		5,4 %
Peak detection - 10 minutes		7,4 %
Peak detection - 1 hour		6,2 %
Interval	**1 minute**	**8 hours**
Accuracy (%)	99,9 %	99,2 %
Compression (%)	67,7 %	99,7 %
Comp. Overhead (s)	3.4 sec	0.8 sec

Fig. 5. Results for aggregation over different time intervals

Reduction of Resolution: Feature Extraction Accuracy (%)		
Presence Detection - 4 hours		74,2 %
Presence Detection - 24 hours		78,5 %
Turning-On of devices - 4 hours		15,9 %
Turning-On of devices - 8 hours		10,8 %
Peak detection - 10 minutes		100 %
Peak detection - 1 hour		100 %
Interval	**1 minute**	**8 hours**
Accuracy (%)	100 %	99,9 %
Compression (%)	0,0 %	0,0 %
Comp. Overhead (s)	14.3 sec	8.1 sec

Fig. 6. Results for reduction of resolution of time

Perturbation: From the standpoint of privacy protection the notion of differential privacy seems to be promising [4]. We just kept it much simpler, knowing that we loose on privacy [8]: First, we take the average determined by the `AVG` function of MySQL, standard deviation determined by the `STD` function of MySQl. Second, we calculate the new value by adding the noise to the previous value. We utilise a uniform or a gauss distribution, calculated in python as follows:

$$new\_val = old\_val + rand.uniform(\tfrac{avg}{2}, (avg + \tfrac{avg}{2}))$$
$$new\_val = old\_val + rand.gauss(avg, std\_dev)$$

Fig. 7 again shows that large and generic detections, even if simplistic, can hardly be disturbed by noisy data. Which again means, that simple noise is to be tolerated for some applications and hence "the data collected [..] should be strictly necessary for the specific purpose previously determined by the data controller (the ″data minimisation″ principle)" [2]. However, simple noise does not add to a statistically provable consumer privacy [8].

Perturbation: Feature Extraction Accuracy (%)		
Presence Detection - 4 hours		88,1 %
Presence Detection - 24 hours		99,5 %
Turning-On of devices - Gauss		2,5 %
Turning-On of devices - Uniform		1,6 %
Peak detection - Gauss		13,8 %
Peak detection - Uniform		23 %
Distribution:	**Gaussian**	**Uniform**
Accuracy (%)	22,1 %	21,2 %
Compression (%)	0,0 %	0,0 %
Comp. Overhead (s)	8.7 sec	8.6 sec

Fig. 7. Results for perturbation

5 Conclusion

This case study gives an overview of the differences in aggregation, resolution reduction and perturbation of real-life energy consumption data. We gathered the data from a family household[2]. Additionally, as we automatically obtained the uptime of certain IP-enabled appliances, e.g., SmartTV, and because the inhabitants kept diaries and we conducted interviews, we have a good ground truth to identify which actions correlate to consumption data. The rising quality of the gathered data which increases the sensitivity of the recorded data to be privacy invasive. For this case-study we devised relatively simple threshold driven machine-learning algorithms to extract features about the behaviour from the energy consumption data. Even with the compression property of aggregation or the noise introduced by perturbation the presence detection still works quite accurately ($> 74\,\%$). It is worthwhile to note, that although simple presence detection is still feasible on the processed data set, more detailed inferences requiring higher temporal or energy-level details are clearly aggravated.

The loss of data quality that is occurring with all methods can always be seen as a privacy gain, e.g. we showed that presence detection within an interval of 4 hours is still achievable with far lower data quality. Even more interesting, if you consider our simple extraction algorithms. As a result we conclude, that for each IoT application's well defined purpose — and purpose must be defined to operate within the EU's legal boundaries — you must carefully validate if you could not offer the same service with less data.

6 Future Work

As future work, we want to correlate our results with applications of differential privacy [4]. Secondly we want to follow the idea that if perturbation parameters are known, you can use them to de-noise perturbed data. The distribution function along with the respective parameters can be understood as key, while the

[2] raw data available on request via `ict-rerum.eu`

generated noise serves as quasi-encryption. Hence, if you store the parameters locally, as a sort of a secret, you can store the noisy data at a third party server. This removes the need to store long histories of data locally. As the data is noisy it shall be less privacy-invasive, of course it remains to be seen what noise to add. If there is the need to view values without noise, the locally known perturbation parameters can be used to de-noise data for authorised local viewers.

Acknowledgment

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 609094.

References

1. *TIME Magazine: The Smarter Home - 39 page special report*, volume 184. Jul 7 2014.
2. ARTICLE 29 DATA PROTECTION WORKING PARTY. Opinion 8/2014 on the on Recent Developments on the Internet of Things, Sept. 2014.
3. Marcos Duarte. Bmc - detection of peaks in data @ONLINE, 2013.
4. Cynthia Dwork. Differential privacy. In *ICALP (2)*, pages 1–12, 2006.
5. EU. Regulation 45/2001 of the European Parliament and of the Council of 18 December 2000 on the protection of individuals with regard to the processing of personal data by the Community institutions and bodies and on the free movement of such data. *Official Journal*, L 8/1, Jan. 2001.
6. Elgar Fleisch and Friedemann Mattern. *Das Internet der Dinge*. Springer, 2005.
7. Marek Jawurek. *Privacy in Smart Grids*. PhD thesis, Friedrich-Alexander-University Erlangen-Nuernberg, 2013.
8. H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar. On the privacy preserving properties of random data perturbation techniques. In *Proc. of IEEE ICDM 2003*, pages 99–106. IEEE, 2003.
9. L. Liu, B. Thuraisingham, M. Kantarcioglu, and L. Khan. An adaptable perturbation model of privacy preserving data mining. Technical Report UTDCS-04-06, Univ. of Texas at Dallas, Jan. 2006.
10. Henrich C. Pöhls and Markus Karwe. Redactable signatures to control the maximum noise for differential privacy in the smart grid. In *Proc. of SmartGridSec'14*, volume 8448 of *LNCS*. Springer, 2014.