Gridchain: an investigation of privacy for the future local distribution grid

Citation for the original published paper (version of record):

Picazo-Sanchez, P., Almgren, M. (2023). Gridchain: an investigation of privacy for the future local distribution grid. International Journal of Information Security, 22(1): 29-46.
http://dx.doi.org/10.1007/s10207-022-00622-6

International Journal of Information Security
https://doi.org/10.1007/s10207-022-00622-6

REGULAR CONTRIBUTION

Gridchain: an investigation of privacy for the future local distribution grid

Pablo Picazo-Sanchez[1] · Magnus Almgren[1]

Abstract

As part of building the smart grid, there is a massive deployment of so-called smart meters that aggregate information and communicate with the back-end office, apart from measuring properties of the local network. Detailed measurements and communication of, e.g., consumption allows for remote billing, but also in finding problems in the distribution of power and overall to provide data to be used to plan future upgrades of the network. From a security perspective, a massive deployment of such Internet of Things (IoT) components increases the risk that some may be compromised or that collected data are used for privacy-sensitive inference of the consumption of households. In this paper, we investigate the privacy concerns regarding detailed readings of smart meters for billing purposes. We present Gridchain, a solution where households can opt-in to hide their consumption patterns and thus make Non-Intrusive Load Monitoring (NILM) more challenging. Households form groups where they can trade *real consumption* among themselves to achieve *reported consumption* that would be resistant to NILM. Gridchain is built on a publish/subscribe model and uses a permissioned blockchain to record any trades, meaning that dishonest households can be discovered and punished if they steal from other households in the group or the electricity company in the end. We implement and release a proof of concept of Gridchain and use public datasets to allow reproducibility. Our results show that even if an attacker has access to the reported electricity consumption of any member of a Gridchain group, this reported consumption is significantly far from the actual consumption to allow for a detailed fingerprint of the household activities.

Keywords Privacy · Smart grid · NILM

1 Introduction

We are witnessing a transformation of the existing electricity grid to a new type of infrastructure. This new model is usually referred to as the "Smart Grid" and includes changes related to telecommunications, automation, distributed technologies, and IoT devices. The goals of the changes include resilience to system failures, providing self-configuration to the stakeholders, and a better understanding and control in near real time of the energy flow in the system.

Changes happen from the production through the transmission down to the distribution network. One example is the massive deployment of smart meters for electricity and gas. Over 200 million smart meters for electricity have been planned to be deployed by the end of 2020 [23]. In terms of adoption, this would translate into almost 72% of European consumers having a smart meter for electricity and 40% for gas. Other reports give similar estimates regarding the magnitude of this deployment [25].

Focusing on electrical smart meters, they are capable of supporting different services such as remote billing—by recording the consumption based on different cost models, like total consumption and peak consumption, and a so-called low-voltage Supervisory Control And Data Acquisition (SCADA) in that they—in real time—can monitor properties of the grid, like voltage control and power flow. The information that can be aggregated and refined based on electrical consumption is invaluable. For instance, grid suppliers might use this information in real time to trace consumption patterns in some areas and adjust the electricity production accordingly.

However, this information might also be misused. For instance, an attacker, by having access to the electricity consumption, can execute a set of Non-Intrusive Load Moni-

✉ Pablo Picazo-Sanchez
pablop@chalmers.se

[1] Chalmers University of Technology, Gothenburg, Sweden

Published online: 11 October 2022

toring (NILM) algorithms to infer when a household turns on the heater(s), has breakfast, leaves late for work, or even the number and the age of the people living at home [49,58]. As a consequence, consumer skepticism and concerns related to privacy have slowed down the adoption of smart meters [36].

To mitigate these privacy issues, the European Commission proposed the use of cryptographic protocols over aggregated data as a promising tool for securing data managed by Smart Grids [21]. Concretely, smart meters can form groups, and an aggregated consumption value is periodically sent to the energy supplier. Researchers have proposed solutions based on homomorphic encryption [42], privacy preservation schemes [43], secure multiparty computation [59], elliptic curves [45] or secret sharing functions [17] among others. Despite being promising, the real deployment into smart meters is still challenging due to the high computationally demanding operations of these schemes [6] and inherent risks with the aggregated data [19]. New security protocols that protect the data of the users are needed [3].

This paper investigates privacy concerns regarding detailed readings for billing purposes. Others have previously suggested that a local supply (battery) can be used to anonymize the usage patterns so that less information about the household can be gained from the energy traces [33,68]. This supply can either be used directly to hide consumption patterns or as an external source to disturb the signal in a differential-private way. Even though some households may have large-scale batteries in the future in the form of electric cars, today, there is no such deployment making these schemes exciting but challenging to use in practice.

We present Gridchain, a solution where households can opt-in to hide their consumption patterns and thus make Non-Intrusive Load Monitoring (NILM) more challenging. Households form groups where they can trade *real consumption* among themselves to achieve a *reported consumption* that would be resistant to NILM. In comparison to other proposals based on the modification of the consumption [31,69], Gridchain guarantees that both the real consumption and the reported one always are the same when required, for instance, at billing time.

Gridchain is built on a publish/subscribe model and uses a permissioned blockchain to record any trades, meaning that dishonest households can be discovered and punished if they steal from other households in the group or the electricity company in the end. No costly hardware or batteries are needed but just minor revisions of the software of the smart meters. However, any battery for some members would benefit the group.

Contributions Section 2 sets the basis for a better understanding of the rest of the paper whereas our contributions can be summarized as follows:

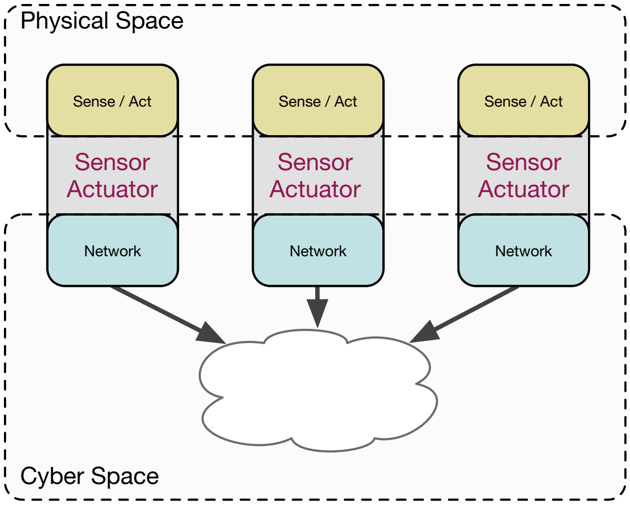


Fig. 1 Classical architecture of Cyber Physical Systems

- We present Gridchain, a privacy compliant real-time collaborative solution for smart meters (see Sect. 3).
- We provide a proof of concept of Gridchain and simulate multiple smart meters using a public dataset of electricity consumption (see Sect. 4).
- We evaluate the privacy acquired by the proof of concept and show how the reported electricity consumption significantly differs from the real one, thus increasing the privacy of the users (see Sect. 5).

In Sect. 6, we discuss the limitation of our work, describe the future work as well as some open issues we identified during our research. Related work can be seen in Sect. 7 while this paper ends with some conclusions in Sect. 8.

2 Background

In this section, we shortly define some well-known terminology and concepts used in the literature that facilitates reading this paper.

Consumers and Smart Meters Consumers and customers are the final elements in the electricity network, that is, the ones that use the electricity. Smart meters are usually placed at the consumer location and are part of Cyber Physical Systems (CPSs) networks (see Figure 1). They are small and constrained devices in charge of measuring the electricity consumption with a predefined frequency and sending such data to other parties in the network. Smart meters decrease metering errors, help in debt management, identify frauds and reduce the gap between peak demand and the available power at any given time [22]. Some of them can even interact with the environment.

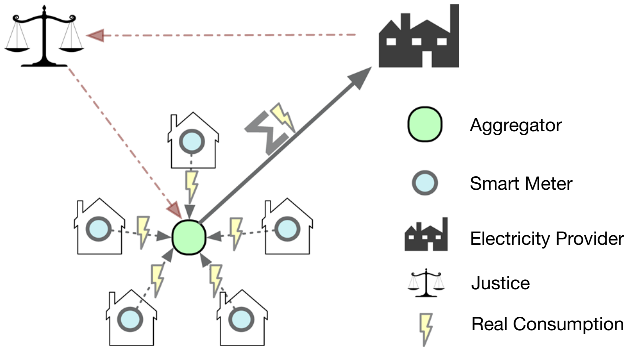


Fig. 2 Billing example in a electrical smart metering scenario

Grid Supplier The grid supplier is in charge of supplying electricity to the network. It controls all the electricity in the network, i.e., its distribution and the needed infrastructure. The information measured by the smart meters might be crucial to provide load balancing and adapt the network to peaks of electricity consumption in real time.

Aggregator The aggregator is usually a device provided by the grid supplier. It essentially receives the measured electricity consumption of the smart meters. It generates the data used for the billing companies (electricity providers) and the grid suppliers to provide other services such as load balancing.

Electricity Provider The electricity provider, also called electricity producer, is usually a company different from the grid supplier in charge of selling the electricity to the final consumers. We included in Figure 2 a simple example of how electricity providers compute the electricity billings for the final customers.

Communication It has been shown that a middleware layer can improve the efficiency, reliability, and security of Smart Grids [5]. There are two conditions that the middleware has to satisfy: communication with service providers and cooperation between the smart houses. The message-oriented middlewares, like the publish/ subscribe protocol [20], were identified as a promising mechanism to achieve the aforementioned requirements [5].

Aggregation Schemes One of the most promising ideas to provide privacy is to create clusters of smart meters so that only the aggregated consumption of the cluster is reported to the energy supplier instead of individual measurements [11]. In more detail, the way it works is the following: smart meters measure the electricity consumption and, with a given temporal granularity, they send the sensed consumption to the aggregator. When received, the aggregator computes the sum of the consumption from all the smart meters in the network and then reports such consumption to the electricity provider.

Blockchain A blockchain is a distributed ledger, in the form of an append-only data structure, replicated by multiple nodes [52]. Due to the blockchain's transparent and immutable smart contracts, multiple users can establish trustful relationships without complex algorithms [12]. Since its first appearance in 2008, blockchain has successfully been applied to different scenarios to fight against selfishness nodes in decentralized networks [1,41], healthcare environments [48], smart contracts [14], finances [24,52] and IoT [15] just to mention a few areas.

2.1 Non-Intrusive Load Monitoring

When the smart meters send the measured electricity consumption to the aggregator, the aggregator aggregates the consumption and reports it to the electricity provider, the grid supplier, or any other party with enough privileges. Formally, the consumption at a given instant t can be seen as: $P(t) = p_0(t) + p_1(t) + \ldots + p_{i-1}(t) + p_i(t)$ where p_i is the individual consumption of a smart meter i.

Non-Intrusive Load Monitoring (NILM) algorithms try to decompose $P(t)$ to detect appliances of single smart meters. This is done by detecting changes in consumption and comparing them with the pattern of the consumption of well-known appliances. According to Hart [29], Zeifman and Roth [72], and Baranski and Voss [9], there are four main families of appliances based on their operational states:

1. Appliances with two states, ON/OFF, e.g., lamp and toaster;
2. Finite state machines, that is, appliances with multiple states such as washing machines, ovens, and tumble dryers;
3. Appliances that do not follow a pattern in their consumption such as power drills and dimmer lights; and
4. Appliances that remain connected during long periods of time such as TV and smoke detectors.

The fingerprinting or signature of the appliances is obtained by detecting significant changes in the electricity consumption, i.e., NILM algorithms detect peaks in the electricity consumption, and by matching them with the known signatures, they extract the appliances [2,74].

3 Gridchain: a real-time collaborative system for electricity networks

Gridchain is designed to be a privacy-compliant electricity solution where households can opt-in to hide their consumption patterns and thus make NILM more challenging by flattening the consumption curve [61], that is, avoid as many peaks as possible or changing them to avoid inferring attacks [49].

In order to ease the readability, next, we introduce and define some key concepts we use throughout the rest of the paper. Later, we introduce the architecture, the attacker model, and a general description of Gridchain.

3.1 Definitions

Threshold β One key aspect of our proposal revolves around estimating the energy consumption of the next time window. We overcome the challenges of providing such an estimate by allowing the smart meters to be as fine-grained as desired. That is, the smart meter can compute an aggregated function of the consumption of the last N days, being possible to distinguish between different days (e.g., holidays, weekends, and working days) or different moments within the day. By doing so, the smart meter can make a prediction based on previous patterns and incorporate new knowledge or patterns into estimating the next slot.

Trading In Gridchain, meters can trade the electricity consumption they measure. Thus, we define the terms *publish*, *consume*, *transactions*, and *debts* as:

(1) the act of sending (part of) the measured consumption by a smart meter (said to be the *publisher*);
(2) the act of taking electricity consumption of other smart meter(s) (said to be the *consumer*);
(3) the amount of electricity that both publishers and consumers agree on trading, and finally;
(4) the amount of electricity that the publisher *owes* the consumer

respectively.

Consumption We differentiate between *real* and *reported* consumption. Real consumption (kWh_{real}) is the actual electrical consumption that a smart meter measures. Reported consumption ($\text{kWh}_{\text{reported}}$) is the electricity consumption reported to the aggregator and later to the electricity provider. When the consumption is reported (e.g., to generate the invoice of the customer), a boundary condition is that the sum of both types of consumption should be equal:

$\sum \text{kWh}_{\text{real}} = \sum \text{kWh}_{\text{reported}}$.

Trust There must be a mechanism that prevents non-repudiation attacks [38] and provides data consistency for accountability and auditing when required. In other words, there must be a conflict resolution protocol; that is, if someone denies having participated in a specific transaction, one must be able to trace back the transactions and clarify any problem that could have appeared. In Gridchain, this is provided by the blockchain, as explained later.

3.2 Architecture

A graphical representation or Gridchain can be seen in Figure 3. In the following, we describe all entities that take part in our model.

Smart Meters. Smart meters are electronic devices in charge of measuring electricity consumption. In Gridchain, smart meters are grouped into clusters or private networks. For instance, we say that a network has six smart meters if there are six apartments/houses that all belong to the same cluster. Additionally, smart meters can communicate with each other within the same network to trade electricity.

Aggregator. In our model, the aggregator is similar to the traditional model. It computes the total consumption that the smart meters report and sends that information to other parties in the network, e.g., the electricity provider in the case of computing the bills. Aggregators are thus entirely agnostic for the electricity sharing algorithm that smart meters execute.

Justice. If a customer manually tampers with the smart meter and starts misbehaving, justice steps in. An example of misbehavior is when a smart meter only trades its own consumption but never accepts trades of other smart meters' consumption, thus breaking the boundary condition described above and owing money at the end of the billing period to other customers. In a real system, justice could take the role of representatives from the electricity company. However, it could also result in a smart meter being excluded from a cluster and not allowed to participate in future rounds of GridChain.

3.3 Attacker model

We consider two types of attackers: an honest-but-curious attacker [26] and active attackers. In the honest-but-curious model, attackers follow the protocol as expected. However, they try to extract as much information as possible from the protocol execution or the data. Contrarily, active attackers can actively alter the protocol and the data to exploit the system.

More specifically, we assume that the aggregator follows the honest-but-curious model, whereas the smart meters can be tampered with to behave selfishly. In our setting, that could include a smart meter configured to trade its own consumption but never accepts any trades with consumption from others (with the perceived cost).

In summary, all the possible attacks our model considers are:

(i) selfishness;
(ii) trust;
(iii) non-repudiation; and
(iv) passive attacks to extract sensitive information and profiling consumers through the electricity consumption by running NILM algorithms in the aggregator device.

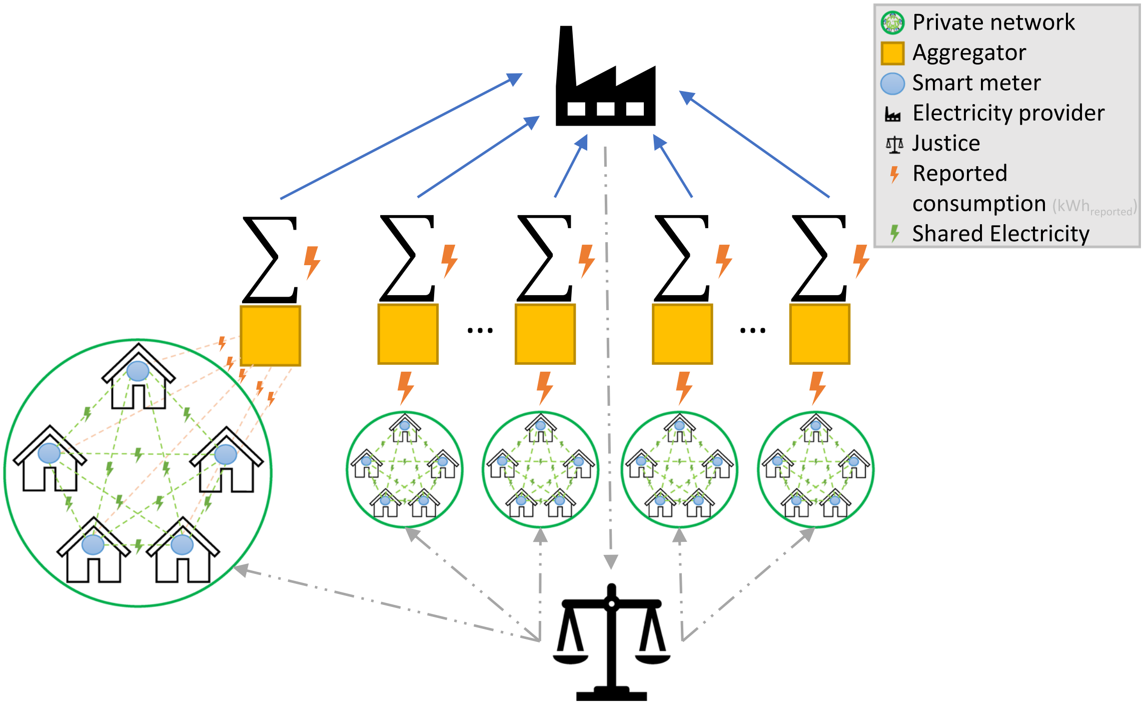


Fig. 3 Gridchain architecture overview

3.4 Protocol description

In Gridchain the goal is to remove one of the most important features of the consumption patterns that NILM algorithms use to identify appliances: peaks [2,11,47]. The consequence is that when meters report the consumption ($\mathrm{kWh}_{\mathrm{reported}}$), the aggregators, even though they might execute NILM algorithms, will not be able to know the real consumption of the consumers.

We can differentiate three main parts in Gridchain:

(1) the *setup* of the system;
(2) *trading*; and
(3) *billing*.

Note that billing does not necessarily need to be a periodic monthly executed task, but it can be executed anytime a service of an authorized party requires it, e.g., electricity provider, grid supplier and justice.

Setup The setup phase is the initial algorithm to cluster smart meters into private networks. The smart meters can act as both publishers and consumers in order to share and consume electricity to/from other smart meters in the network, respectively. In addition to the clustering, this stage is when a mechanism to keep track of the transactions to avoid non-repudiation attacks should be deployed.

Trading In Algorithms 1 and 2, we include the pseudocode of both publishers and consumers, respectively. Note these are a high level description of the algorithms where for simplicity, we did not include

(1) the source code needed to calculate the threshold (β);
(2) the source code that both publishers and consumers need in order to agree on the final amount of electricity to give/take;
(3) the performance improvements to solve the already ongoing transactions and debts; and,
(4) the billing procedure needed to generate the final invoice for the customer.

Algorithm 1 Sharing electricity's algorithm (Publishers)

```
$\beta$ = getElectricityPrediction(timeWindow)
**if** currentConsumption $>$ $\beta$ **then**
    toShare = currentConsumption - $\beta$
    neighboursElectricity = publishElectricity(toShare)
    **for** neighbourElectricity in neighboursElectricity **do**
        makeTransaction(neighbour, electricity)
    **end for**
**end if**
```

In more detail, the first step for both algorithms is to calculate a local threshold (β) used to determine whether this particular smart meter should act as a publisher or a consumer. This threshold is valid for some time and can be as fine-grained as desired. For instance, β can be valid for the time window considering red days, weekends, holidays,

Algorithm 2 Consuming electricity's algorithm (Consumers)

```
$\beta$ = getElectricityPrediction(timeWindow)
if currentConsumption $<$ $\beta$ then
    neighbour, electricity = consumeElectricity()
    electricity = min(electricity, ($\beta$ - currentConsumption))
    makeTransaction(neighbour, electricity)
end if
```

working days, and the past consumption time window. There are then three cases:

(1) the current electricity consumption measured by the smart meter is greater than β. In this case, the smart meter is willing to shed electricity consumption to other smart meters in the network (to reduce peaks), i.e., it becomes a publisher;
(2) the current value is lower than β. In this case, the smart meter becomes a consumer and looks for smart meters in the network to increase its apparent consumption (to avoid troughs); and
(3) β is equal to the current consumption. In this case, the smart meter neither publish nor consume electricity (as it is at an optimal level).

After that, the algorithms compute different values depending on whether the smart meter is a publisher or a consumer. In the case of publishers, they compute the *remaining* electricity, that is, the difference between the current consumption and β. The remaining electricity is published so that consumers can consume it. In the case of the consumers, they compute the amount of maximum electricity consumed (i.e., β - currentConsumption). Finally, when both consumers and publishers agree on sharing an amount of energy, they have to keep track of such agreements to avoid non-repudiation attacks.

Billing When the billing algorithm is fired, i.e., when the electricity provider wants to compute the customers' electricity consumption to generate the bill, it is entirely agnostic to the trading and the transactions carried out internally by the smart meters. The company takes the reported electricity, obtains the consumption of each smart meter, and generates the bill having that Gridchain always guarantees that $\sum$ $\text{kWh}_{\text{real}} = \sum \text{kWh}_{\text{reported}}$ at the end of the designed billing window.

Conflict resolution protocol

Assuming smart meters to be active attackers, a smart meter (consumer) may alter the consumption at any point in the protocol, either during transactions or by modifying the $\text{kWh}_{\text{reported}}$ to pay for less than used. However, this is easily detected as the electricity provider would know the electricity consumed by the cluster and compare this with the reported consumption. Gridchain implements a conflict resolution protocol with a verification mechanism for customers, electricity providers, and justice to check which smart meter is misbehaving by either spoofing the reported consumption or denying having participated in a transaction. In the current version, all transactions are stored in a blockchain with non-repudiation, as explained in Sect. 4. When required, authorized parties can access needed transactions and obtain the real consumption to check who is misbehaving and (if needed) to perform legal actions. However, note that such a resolution protocol only needs to be invoked when a party is malicious; as this is easily detected, it is expected to be needed very rarely as an attacker will realize any attempt of misreporting can be resolved among the parties.

4 Gridchain: a proof of concept

We deployed an instance of Gridchain in a computer with Intel(R) Core(TM) i7-4790 CPU @3.60GHz and 16Gb of RAM running Linux. We executed the same experiment 100 times independently to avoid bias. In the following, we present and discuss the implementation decisions we made in Gridchain, i.e., how to

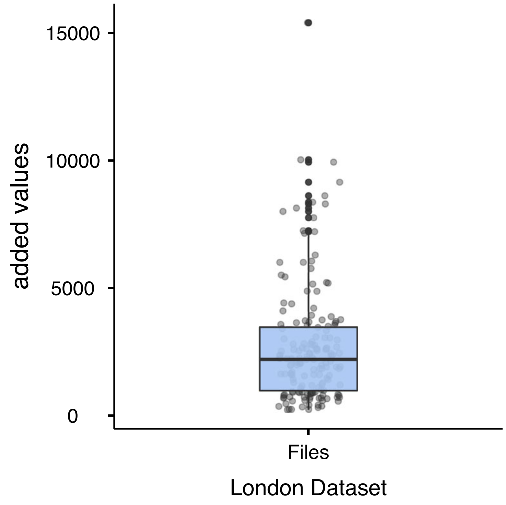
(i) simulate consumption;
(ii) cluster smart meters;
(iii) allow trading;
(iv) implement trust; and
(v) estimate the threshold β.

It is important to remark that these are implementation decisions; for other types of environments other choices may be more suitable within the design space as explained in Sect. 3.

4.1 Initial setup

Dataset We found that, in general, it is difficult to directly compare the results between different researchers in this area. We aimed to use a publicly available dataset to allow others to compare our results. Because the *Setup* protocol of Gridchain consists of grouping multiple meters into clusters, we required a dataset with enough data from different households. In particular, the London dataset contains energy consumption (in kWh) measured in periods of half an hour. It can be freely downloaded. [1] It is composed of 167 million rows of data which belong to approximately 5,567 different households. All the data were gathered between November 2011 and February 2014 in intervals of 30 minutes. We realized that in this dataset, even though it is pretty detailed, some households' consumption has some missing values.

[1] data.london.gov.uk/dataset.



(a) Boxplot of preprocessing

	added
N	168
Mean	2718
Median	2218
Minimum	238
Maximum	15392
25th percentile	991
50th percentile	2218
75th percentile	3477

(b) Descriptive stats of preprocessing

Fig. 4 Statistics of added values. Each point represents a file and the value is the number of added values in the preprocessing phase

To clean the dataset, we performed a preprocessing of all the files and automatically:

(1) deleted repeated values; and
(2) added missed values computing the standard average between the adjacent values, e.g., when the consumption corresponding to 10:00 is missing, but the values of 9:30 and 10:30 are there, we computed the average and added it to the file.

This process took us about 1 hour and a half, and we added on average 0.2% of values per file. We can see in Figure 4 that the added values per file have a certain spread.

To avoid manual manipulation as much as possible, we filtered out those files whose added values were larger than 1%, i.e., files that fall above the 25th percentile, with less than 991 added values per file. In total, we used for the experiments 42 files.

Neighbors clustering During the *Setup* we grouped smart meters into private networks so that they can share electricity consumption with others under the same domain. In order to allow other researchers to reproduce the experiments presented in this work and not be biased, we generated these clusters based on the digits of π. The first network comprises 31 neighbors, 41 the second, 59 the third, etc. In total, we picked the first 14 decimals of π to simulate 7 networks, i.e., 14, 15, 92, 65, 35, 89, and 79 meters per network in total.

Trading The publish/subscribe model [10,20] is a subset of message-oriented middleware and relies on an event notification service. It essentially provides an alternative to a traditional client-server architecture where the client directly communicates with the other party. In pub/sub model, subscribers register the interest in events—or pattern of events, by calling a *subscribe()* method. This subscription is secret, i.e., is not transmitted to other publishers or subscribers and remains in the event service. Publishers can generate events using a *publish()* operation, which asynchronously spreads the event to all the subscribers. The connection between publishers and subscribers is handled by a third component called *broker*, in charge of filtering and distributing all the messages to subscribers. We modeled smart meters that act as publishers, whereas consumers correspond to subscribers in the pub/sub model.

Concretely, we used RabbitMQ,[2] an open-source message broker and queuing server, for the middleware implementation. In other words, RabbitMQ is a broker that supports messaging protocols like Streaming Text Oriented Messaging Protocol (STOMP) [62], Advanced Messaging Queuing Protocol (AMQP) [54] and MQ Telemetry Transport (MQTT) [50]. Even though all the communications between devices within the same network are implemented using MQTT, any other message-oriented protocols listed above could have been used instead.

Trust and conflict resolutions In order to keep track of the transactions that publishers and consumers agree on, we deployed a blockchain per private network. By doing so, Gridchain guarantees that two smart meters agreeing on a transaction cannot deny having participated in it (non-repudiation [38]). Also, the shared electricity between smart meters is stored, providing data consistency for accountability and auditing when required (conflict resolution protocol in case a smart meter is not following the protocol). To be as generic as possible, we used a fundamental blockchain.[3] Furthermore, the implementation is suitable for constrained devices since the communication is carried out by HTTP

[2] https://www.rabbitmq.com.

[3] https://github.com/dvf/blockchain.

requests and it exposes an API to easily check all the transactions.

Threshold β For simplicity and to be as consistent as possible, we considered the consumption of the previous time slot in our proof of concept. We computed the median of the consumption for the threshold β. For instance, to simulate 1 day of electricity consumption, we took $\beta = 1$ day; to simulate 1 week, we set $\beta = 7$ days.

5 Experimentally measuring privacy

5.1 Background and methodology

One of the open issues that the field of experimental privacy has is the lack of formal methods to compare the achievements—in terms of privacy—of new proposals concerning others. In this paper, we adopt the same methodology as used in the past by other seminal work based on peak detection [11,47] and the edit distance [46] between these peaks.

On the one hand, counting the number of peaks has been proven as one of the most significant features when NILM algorithms are executed [11]. To extract the number of peaks, we used the peak detection algorithm provided by Scipy[4] with the default parameters—to be as general as possible, i.e., when a value is higher than its neighbors then it is considered as a peak.

On the other hand, once we extracted the peaks, we calculated the edit distance between two lists of peaks. Formally, given two strings a, b of an alphabet $\sum$, the edit distance $d(a, b)$ is the minimum set of edit-operations needed to transforming a into b. The basic operations defined by Levenshtein [40] were deletions, insertions, and substitutions. In our work, we used an improved variant of the Levenshtein algorithm [60] as well as the Longest Common Subsequence (LCS) edit distances as metrics to measure the privacy of the users concerning the originally measured consumption. The difference between these two functions is that the former allows deletion, insertion, and substitution, whereas the LCS allows only insertion and deletion but not substitution.[5]

Let us provide a concrete example. We simulated Gridchain during one day for the seven networks and randomly picked one of the smart meters to show both the real and the reported consumption (see Figure 5). In such a figure, it is interesting to see that the more smart meters in the network, the more significant the difference between the two types of consumption; a somewhat expected behavior since there are more parties to share the electricity with.

Let us now focus on one of the networks in Figure 5. For simplicity, we used the one composed of 14 smart meters and, more concretely, the smart meter 10 (Fig. 5a). Let S and D be arrays containing the positions of peaks corresponding to kWh_{real} and $\text{kWh}_{\text{reported}}$, respectively.

$$\begin{cases} S = & [3, 5, 8, 14, 19, 21, 25, 31, 36, 39, 41, 44, 46] \\ D = & [5, 7, 10, 14, 19, 21, 23, 25, 28, 31, 33, 36, 39, \\ & 41, 44, 46] \end{cases}$$

In this example, the edit distance is 4, the number of matches is 11 ([5, 14, 19, 21, 25, 31, 36, 39, 41, 44, 46]) whereas the LCS is 5 (the string composed of [36, 39, 41, 44, 46] peaks). Finally, we have certified that $\sum \text{kWh}_{\text{real}} = \sum \text{kWh}_{\text{reported}}$ holds for all the networks at the end of the specified billing window, being equal to 0.154 kWh.

Contrarily to both the number of matches and the LCS , in the edit distance, the idea is that the more changes the attacker has to perform to transform the peaks of the reported consumption into the real ones, the better. In terms of privacy, the attacker will have more uncertainty in guessing where the real peaks of the electricity consumption would be.

For the rest of the results presented in this section, we always simulated the consumption of the seven networks during both 1 day and 1 week. We ran Gridchain 100 times independently and computed the mean of the results. Such simulations took us 2 and 10 days, respectively. One example of the results we obtained from our simulations can be seen in Table 1 where the numbers correspond to the simulation of a network of 14 smart meters during one day. Figure Figure 5 corresponds to one of these independent experiments.

5.2 Results: peaks

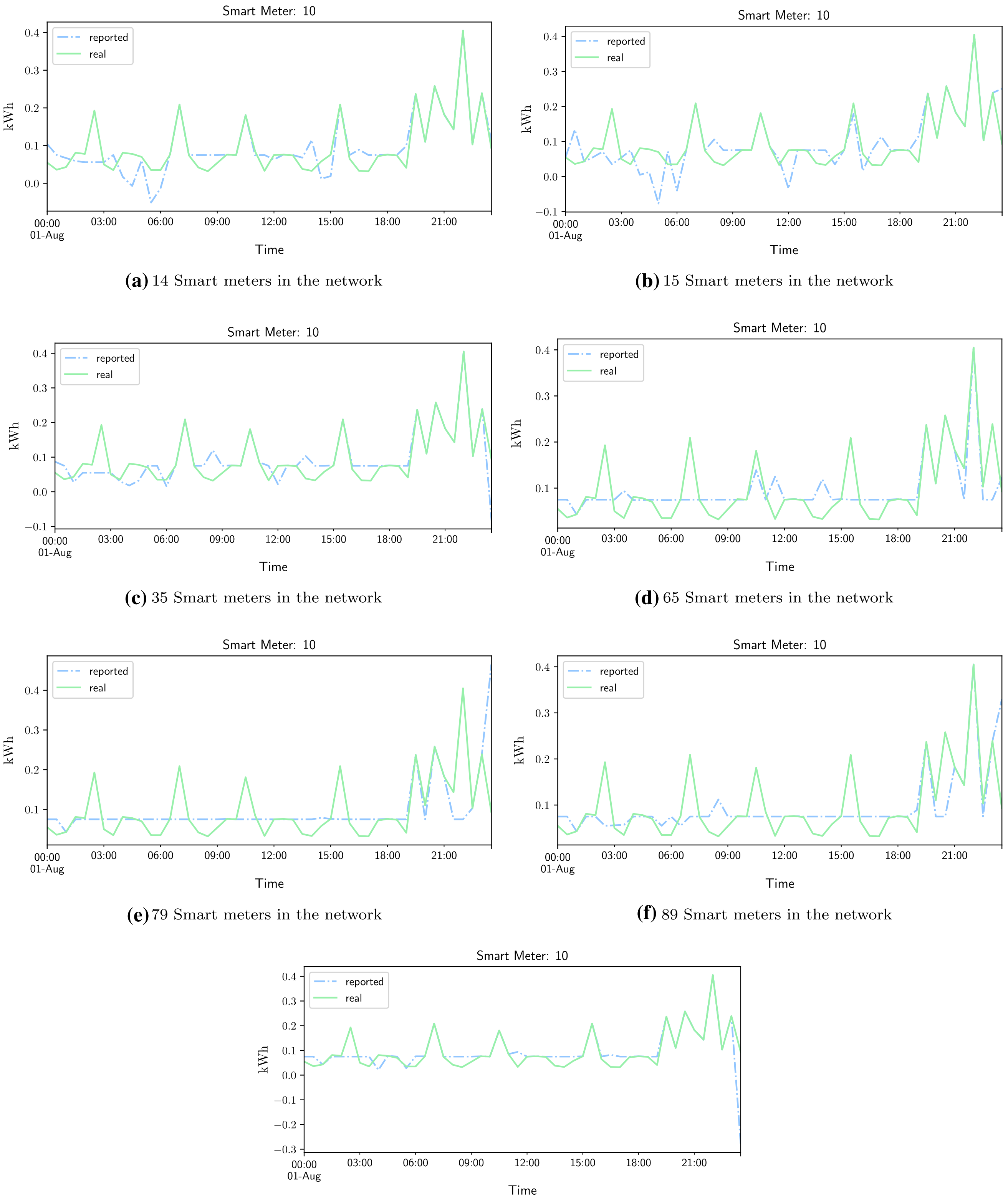
We measured the number of real peaks versus the number of reported peaks. We obtained that, in general, Gridchain reports over 20% fewer peaks for both 1-day and 1-week simulations than without our method. In Figure 6a, 6b we include the graphical results of this first experiment as well as the statistics in Table 2. It is also interesting to see that, after running Gridchain the reported peaks are less dispersed than the real peaks (see Std. Deviation), meaning that the peaks produced by Gridchain in the reported consumption are consistent, i.e., smart meters tend to produce the same number of peaks. We included more detailed plots in Figure 7 of each network's reported peaks in consideration.

5.3 Results: edit distance

As explained in the introduction of this section, we computed the edit distance between both peaks in real (kWh_{real}) and reported ($\text{kWh}_{\text{reported}}$) consumption (see Table 3). In addition

[4] https://docs.scipy.org.

[5] As the strings to be compared may have unequal lengths, the Hamming distance is not suitable for our purposes even though it is also part of the edit-distance family.



(a) 14 Smart meters in the network

(b) 15 Smart meters in the network

(c) 35 Smart meters in the network

(d) 65 Smart meters in the network

(e) 79 Smart meters in the network

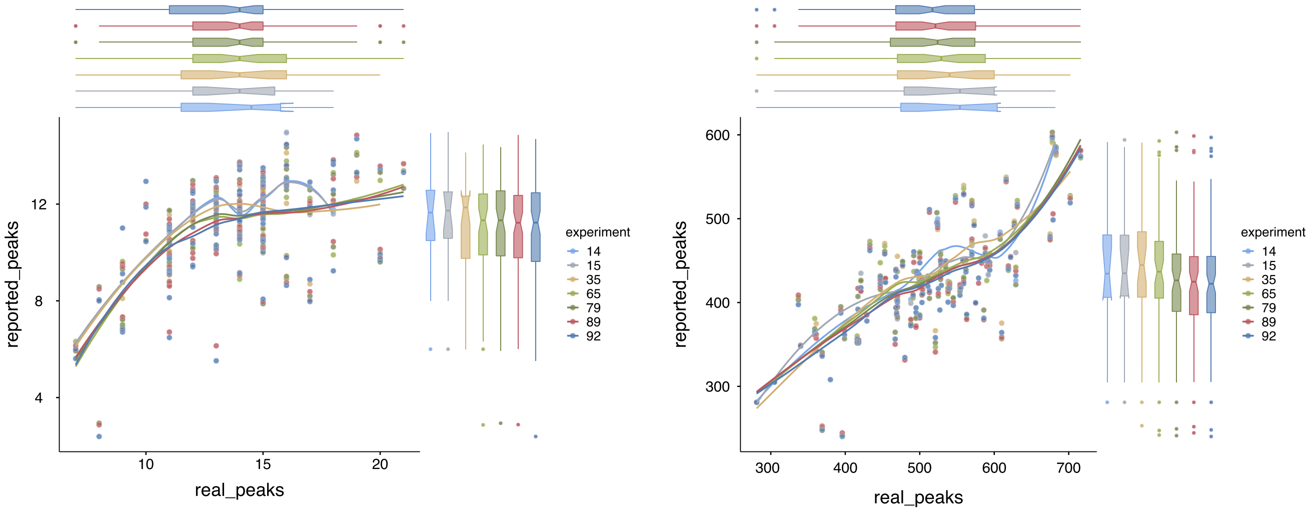
(f) 89 Smart meters in the network

(g) 92 Smart meters in the network

Fig. 5 Example of real and reported consumption after running Gridchain during one day and different number of smart meters per network

Table 1 Statistical results after running Gridchain 100 times for a network composed of 14 smart meters during 1 day

Smart meter	Peaks (kWh_{real})	Peaks ($kWh_{reported}$)	Edit distance	Matches	LCS
1	7	6	1	6	6
2	16	14.95	4.4	12.56	5.22
3	17	11.62	9.59	7.47	3.41
4	15	11.44	6.1	8.94	5.35
5	18	11.68	9.91	8.14	3.16
6	14	12.4	4.8	9.4	5.35
7	11	10.37	5.65	6.46	3.51
8	8	8	0	8	8
9	13	12.63	5.32	8.63	5.81
10	13	12.71	5.41	8.66	5.32
11	18	11.78	9.91	8.12	3.56
12	15	10.2	7.14	7.99	2.94
13	15	13.38	5.64	9.8	5.39
14	11	10.86	5.13	6.52	3.93
MEAN	13.64	11.28	5.71	8.33	4.78



(a) 1-day of simulation

(b) 1-week of simulation

Fig. 6 Relation of real vs reported peaks after running Gridchain 1 day Figure 6a and 1 week Figure 6b

to that, we computed the number of peaks that match (see Table 4) as well as the longest match between peaks (see Table 5) of the consumption. The more changes between the different types of consumption, the more privacy is offered as it becomes more difficult for the attacker to run NILM.

In Table 3 we see that the more smart meters in the network, the larger the edit distance is. The reason is that the more parties to share the electricity with, the higher the privacy level is. In more detail, Gridchain increased by 6.313 ± 0.596 and 190.884 ± 9.038 the number of operations an attacker should perform to obtain the position of the real peaks for the 1 day and 1 week of simulation, respectively.

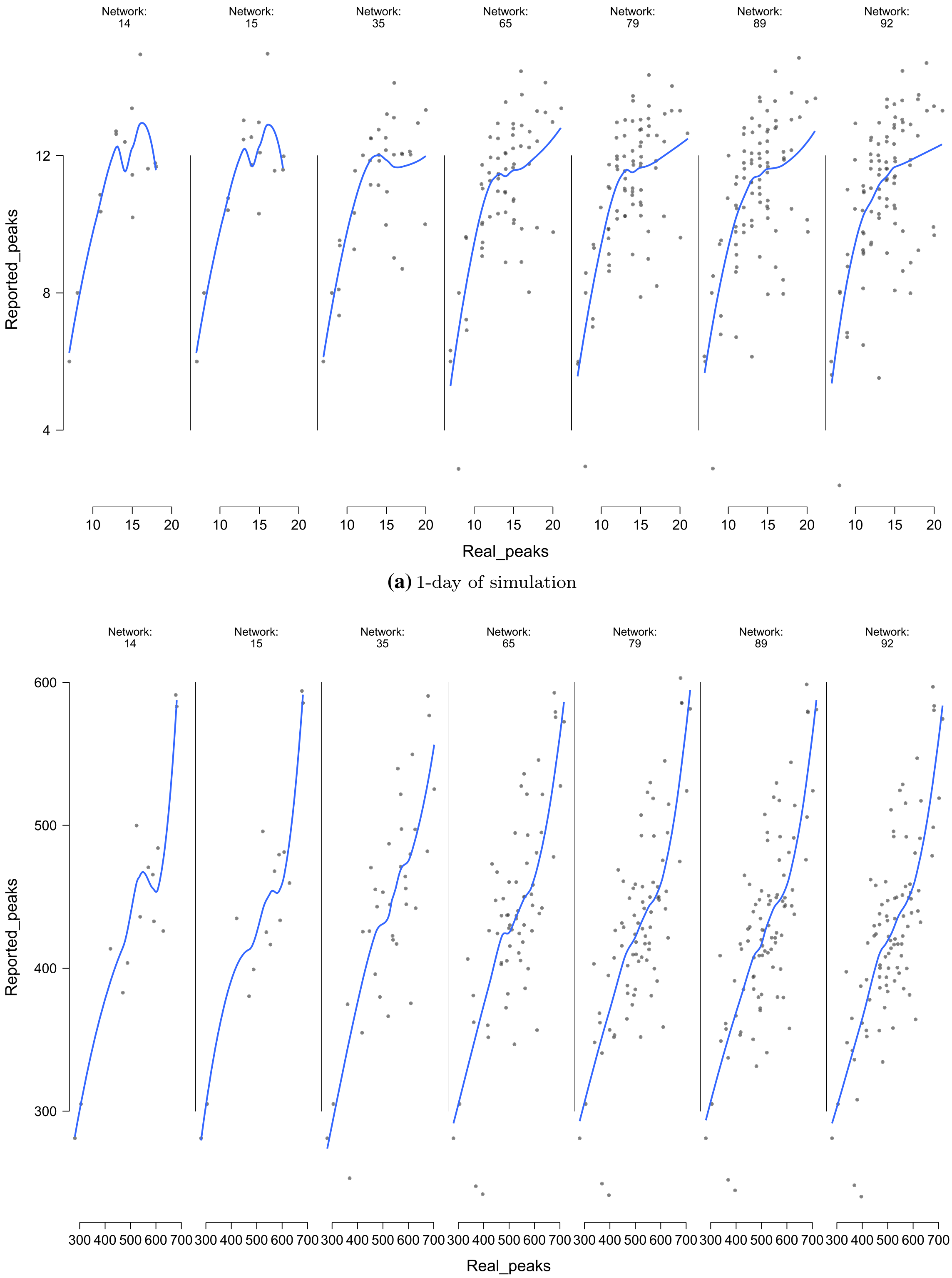
The improvement in the privacy of the customers is also supported by the number of perfect matches and the LCS. In Tables 4 and 5, we can see how the total number of the similar peaks and the LCS of both consumption (kWh_{real}) and ($kWh_{reported}$) is inversely correlated with the number of smart meters the network is composed of.

Regarding the perfect matches, in the 1-day simulation, we observe that the number of perfect matches goes from 61% of the 14 smart meters network to 52% of the network composed of 92 smart meters. Note that in the last case—network with 92 smart meters, almost half of the reported peaks in ($kWh_{reported}$) are different from those of the real consumption (kWh_{real}).

On the other hand, the results we got from the LCS suggest that even though the increase of the smart meters in the network seems not to have a great impact on the number

Table 2 Real peaks vs reported peaks

Network		Real Peaks							Reported Peaks						
		14	15	35	65	79	89	92	14	15	35	65	79	89	92
1 day	Mean	13.643	13.667	13.971	13.800	13.646	13.674	13.598	11.287	11.361	11.075	10.934	10.970	10.878	10.842
	Median	14.500	14.000	14.000	14.000	14.000	14.000	14.000	11.650	11.730	11.860	11.330	11.330	11.230	11.235
	Std. Deviation	3.411	3.288	3.417	3.203	3.059	3.074	3.089	2.225	2.148	1.911	2.117	2.063	2.150	2.220
1 week	Mean	526.714	528.533	529.543	525.123	516.329	516.719	514.815	441.088	442.616	442.520	436.727	428.559	427.338	424.746
	Median	554.000	554.000	540.000	529.000	524.000	521.000	517.000	434.385	434.890	444.740	436.780	426.460	424.770	422.355
	Std. Deviation	123.886	119.587	104.217	95.097	94.757	92.283	92.002	87.477	85.319	76.521	71.242	69.230	67.116	67.112



(a) 1-day of simulation

(b) 1-week of simulation

Fig. 7 Relation of real vs reported peaks after running Gridchain 1 day Figure 7a and 1 week Figure 7b

Table 3 Edit distance statistics for 1 day and 1 week of Gridchain simulation with 7 networks

Network		Edit distance						
		14	15	35	65	79	89	92
1 day	Mean	5.717	5.624	6.995	6.783	6.784	6.919	6.909
	Median	5.525	5.510	6.840	6.450	6.340	6.650	6.590
	Std. Deviation	2.913	2.763	2.638	2.190	2.107	2.208	2.242
1 week	Mean	181.846	183.758	200.640	203.630	198.361	200.169	199.922
	Median	193.330	210.110	202.950	205.820	196.990	199.110	198.920
	Std. Deviation	85.467	82.086	65.819	51.810	53.748	51.295	51.872

Table 4 Statistics of the number of peaks that match for 1 day and 1 week of Gridchain simulation with 7 networks

Network		Matches						
		14	15	35	65	79	89	92
1 day	Mean	8.335	8.443	7.338	7.353	7.249	7.154	7.116
	Median	8.130	8.400	7.570	7.530	7.340	6.980	7.155
	Std. Deviation	1.641	1.587	2.117	2.209	2.205	2.248	2.252
1 week	Mean	357.474	357.593	343.410	337.472	333.465	332.039	330.056
	Median	344.660	333.580	328.490	329.290	324.060	316.680	316.950
	Std. Deviation	71.309	73.452	69.878	64.685	64.760	62.912	61.400

Table 5 Statistics of the longest sequence of common peaks for 1 day and 1 week of Gridchain simulation with 7 networks

Network		Longest Common Subsequence (LCS)						
		14	15	35	65	79	89	92
1 day	Mean	4.782	4.885	4.412	4.677	4.551	4.605	4.593
	Median	5.270	5.220	4.400	4.600	4.480	4.500	4.555
	Std. Deviation	1.418	1.311	1.477	1.637	1.637	1.713	1.693
1 week	Mean	62.675	58.587	42.592	34.682	33.758	31.264	30.489
	Median	28.235	21.440	19.820	18.840	18.470	18.290	17.970
	Std. Deviation	89.898	87.894	63.106	48.660	45.746	42.961	42.042

of longest common sequences in the 1-day simulation, we can see the improvement when we simulated Gridchain for 1 week. Using the mean, the network composed of 14 smart meters got a LCS of 62.675 peaks, whereas, for the network composed of 92 smart meters, the longest sequence is 30.489 peaks. The consequence is that the information the attacker has access to, i.e., $\text{kWh}_{\text{reported}}$ differs much more than the original one.

5.4 Summary

Given our experimental methodology, we can see the numbers achieved by Gridchain after 1 week of simulation (the fourth row of Tables 2 to 5). It detected 526.714 peaks on mean (after running 100 simulations), whereas only 441.088 were reported. Gridchain changed the position of 181.846 peaks, directly affecting both the LCS (62.675 peaks on mean) and the perfect matches of the peaks (357.474 on mean). In particular, we can see how, in general, the smart meters achieve significant levels of privacy by altering the position of the peaks in the reported consumption ($\text{kWh}_{\text{reported}}$). Thus, if an attacker has access to the reported consumption $\text{kWh}_{\text{reported}}$ it would be hard for her to reconstruct the real consumption from it.

6 Discussion

This section offers discussion and points to future work.

The attacker model

Let us return to the attacks our model considers and summarize how they have been dealt with in Gridchain.

i) selfishness;
ii) trust;
iii) non-repudiation; and
iv) passive attacks to extract sensitive information and profiling consumers through the electricity consumption by running NILM algorithms.

The smart meters can have been tampered with and become active attackers. One goal would be to just trade its own consumption but never accept any trades with consumption from others (with the perceived cost). This would break the boundary condition listed earlier that the sum of both types of consumption should be equal at the end of the current billing window: $\sum \text{kWh}_{\text{real}} = \sum \text{kWh}_{\text{reported}}$. This would result in some other smart meters having to pay for more electricity than actually used (easily detected). The owners can, in this case, invoke the conflict resolution protocol and ask the electricity provider to become involved. All the transactions are stored in the blockchain and cannot be tampered with. Thus, the electricity provider can trace back the transactions during a window to determine who owes electricity (money) to whom, and any misbehaving node can be punished (terminated as customers).

A smart meter can also report less consumption than used, meaning the overall reported consumption over the cluster would be less than the actual consumption. Also, this condition would be easily detected, and the electricity provider can then take actions based on the recorded transactions in the blockchain.

The key to the system is thus the non-repudiation offered by the blockchain and the trust that any misbehaving node can be discovered (conflict resolution discussed in Sect. 4.1). This also implies that it is unlikely that any conflict resolution needs to be invoked as it would be resolved, and the attacker would be found and punished.

Our attacker model also considers the privacy risks of the consumption patterns and that an honest-but-curious aggregator can use these patterns to infer sensitive information from customers using NILM. As shown Sect. 5, we showed experimentally that with Gridchain it becomes more difficult to run NILM successfully. Assuming that no more info is provided to the aggregators, they will never be sure whether the combination of the peaks they generate is or is not the correct one. Also, note that other smart meters cannot run the NILM algorithms, as they only see the respective transactions and not any of the full consumption patterns.

Negative consumption In Gridchain, the reported consumption $\text{kWh}_{\text{reported}}$ is equal to the kWh_{real} when the billing is computed (e.g., Figures 5a,5b and 5g). However, during the *trading* period, there might be some negative consumption values reflected in some cases in the $\text{kWh}_{\text{reported}}$ data. This is, of course, not possible since there is no negative electricity. The explanation is that Gridchain tries to resolve all the ongoing transactions with optimization, and the negative values in consumption do not correspond to the kWh_{real} nor the $\text{kWh}_{\text{reported}}$ but to the agreements that smart meters carry out among each other. This is not a problem from a privacy point of view since the goal is to modify the electricity consumption to hide the real peaks so that NILM algorithms cannot extract useful information.

Billing In Gridchain, we modeled the billing task as a periodic function that, when executed, all the smart meters send the reported consumption to the aggregator. Then, it summarizes all the data and forwards it to the electricity provider to compute the billing. Note that the aggregator is not aware of the transactions nor the real consumption of the smart meters as it is not part of the blockchain network. Assuming the aggregator to be honest-but-curious, i.e., it does not alter the data it receives, any information it might extract will be useless. Despite $\sum\text{kWh}_{\text{real}}=\sum\text{kWh}_{\text{reported}}$, the curve and, more importantly, the peaks are not the same.

The most common model for billing the electricity companies' energy consumption is usually calculated monthly. Therefore, our system considers the monthly billing scenario; however, we are not limited to it. Our system is flexible enough to configure the billing algorithm as fine-grained as desired, being even possible to resume the consumption daily. With this, we deal with scenarios where users move from one house to another, being able to resume their debts and not leaving them for the next tenant (in case it exists) [53].

The importance of β In Gridchain there is one crucial parameter, the threshold β. This threshold determines whether a smart meter acts as a publisher or consumer or does not participate in the trading algorithm.

In our experiments, we used different time frames and combinations of previous consumption patterns, i.e., we tested with mean, median, and mode. We are, of course, aware that the best value for β is by computing the mean of all the meters, i.e., this will produce the same flat curve in all the reported consumption. This is, however, only possible either by using a trusted party that computes such value (like the aggregator) or by using more elaborated cryptographic techniques like homomorphic encryption, where all the meters can agree on a value without revealing their consumption to others [51]. For our proof of concept, we decided to keep it as simple and the most lightweight as possible.

Limitations and future work

The blockchain we used for our proof of concept was not encrypted. For a real deployment of our proposed solution, such sensitive information should be protected and allow authorized parties to access a set of data when required. Attribute Based Encryption (ABE) is a public encryption scheme that has been proved to have great performance in other IoT systems with high-constrained devices [71]. The idea behind it is that smart meters encrypt the data of the transactions using the public attributes of the smart meters involved in the sharing agreement plus the public attributes of other parties like electricity providers, the grid suppliers, and the courts. Only those parties with the corresponding public attributes can decrypt the data and access the transaction stored in the blockchain. As a consequence, the attacker model we use can be expanded, and even if passive attackers have access to the messages shared in the blockchain network, they will get nothing but confirmation that the algorithm is working and the electricity is being shared among the participants. In addition, ABE allows consumers to add other services apart from billing, and authorize other parties to have access to the electricity consumption to, for instance, implement a health care monitoring system for older people.

Gridchain defines a threshold β which indicates whether a smart meter should consume or send electricity from/to others. We modeled such a threshold by using a statistical average of previous consumption frames; however, we strongly think advanced techniques based on machine learning will help predict such a parameter. Not only information about previous consumption but learning from customer habits, including more complex information from external sources like the weather forecast or processing information from sensors placed around the house, will increase the privacy of the customers and the whole network.

7 Related work

There are different ways a smart meter could be attacked, and the consequences would depend on the attacker's intent where Ashgar et al. give a good summary of different approaches with their advantages and disadvantages [6]. Costache et al. give an example of how the control of smart meters can cause more significant disruptions on the grid [16]. Tabrizi and Pattabiraman describe how to make smart meters (and other IoT devices) more secure from attacks [63]. Such work could make the fraudulent modifications of meters to steal electricity, as happened in Puerto Rico [37], more difficult. Our work complements such efforts by looking at the information measured and communicated from the smart meter from a privacy perspective, not its actual security mechanisms.

The privacy risk of smart meters was early documented, for example, Molina-Markham et al. [49]. In NILM, one measures the aggregated load for a household to identify single appliances and their usage. If the consumption is measured in enough detail, it may even reveal the TV channel watched in the household [27]. Lately, new techniques utilizing advances in deep neural networks have also been applied to NILM [18].

This has led to several investigations of what type of information is collected and how the privacy-implications can be reduced. These are summarized in Table 6 and further discussed below with comparisons to Gridchain. Asghar et al.[6] and Gross et al.[28] in their systematic reviews, have taken a broader look and considered more use cases than just billing, which is the main focus here.

Tudor et al. [66] investigate privacy-enhancing techniques such as data granularity, retention time, and pseudonyms to understand better the trade-offs between the utility of the data and its privacy (likelihood of identifying individual customers). They present a theoretical framework and empirically investigate how changing the properties of the available data would change an attacker's success. Also, differential privacy has been suggested to protect customers [67], where Tudor et al. design and evaluate a prototype based on a streaming framework to scale to very large data. By adding noise, there is a risk that the usefulness of the data collected will decrease. However, as demonstrated in [65], this may not always be the case for some typical applications such as Short-Term Load Forecasting (STLF). The accuracy of a forecast based on differentially protected data may be very similar to that of a forecast using the original data. However, one disadvantage with the above approaches is that for billing, privacy will cost. Changing the granularity of the collected data as in [66] or using differential privacy as in [67], mean that a customer is not charged for her actual consumption. It should be noted that limiting the consequences to the utility of noise-adding techniques is an ongoing research challenge (see, e.g., [35]). In this work, we focus on having correct billing but still share data that may be useful for grid operation while making NILM analysis more challenging. Our work complements the above investigations in that such privacy-enhancing techniques can also be added to our scenario.

Homomorphic encryption and secure multiparty computation protocols offer intriguing properties; from having been used for other use cases, researchers have also suggested protocols for billing (notably [70]) but these methods tends to have relatively high computing complexity and communication overhead as argued in [6] and [39]. In Gridchain a blockchain is used, but recent research has shown that to be very feasible even for less capable microcontrollers than found in smart meters [56,57]. Others have also suggested blockchains for other use cases in the smart grid [4,13,73].

Our work is inspired by the work of Kalogridis et al. [32–34]. They introduce the notion of *reconciled privacy* where a local battery is used to hide peaks of the consumer. The method is proving, in theory, to be quite effective but expensive in practice as each consumer needs to invest in a large battery. Even though batteries or other storage facilities may at some point have a large adoption among house owners (electric vehicles), it is far from the case today, and, likely, most apartment dwellers will not be able to use similar schemes. Other privacy-schemes using a battery can be found in [8,47,64,68].

In this paper, we designed a system where consumers, together as a group, can hide their consumption pattern by replacing the battery (used in the above approaches) with borrowing / paying back apparent consumption among the

Table 6 A summary of privacy-enhancing techniques for billing data

Type	Examples	Challenges
Data granularity	[66]	Course-grained granularity and randomly
Differential privacy	[30,44,65,67]	Added noise may interfere with accurate billing
Secure multiparty computation	[59]	Relatively high computing complexity and overhead
Homomorphic encryption	[42,70]	
Local energy storage facilities	[7,8,32–34,47,55,64,68]	Expensive and not widely available

group members. Each customer is billed only for her actual consumption, there is no need for costly batteries, and NILM is made more difficult than without having the system. Worth pointing out is that McLaughlin et al. [47] use a similar evaluation framework in that they also measured their success in how they managed to hide events based on NILM analysis.

In most of the proposals analyzed in this paper, either authors did not release the datasets they used for testing their solutions, the methodology and algorithms they proposed are unavailable, or a combination of both. Either way, we could not compare our proposal to others. Motivated by the difficulties we found in reproducing the experiments of other proposals in this field, we not only used a public database (https://data.london.gov.uk/dataset) and released all the source code to facilitate future research on the field (https://github.com/Pica4x6/GridChain.), but also detailed how we preprocessed the dataset with missing values and the implementation decisions we tool.

Summarizing Gridchain with other approaches listed in Table 6, there are not any significant requirements for changes to the infrastructure; e.g., there is no need for large batteries in the household as previous literature requires. However, any such addition to some group members would benefit the group. Gridchain requires communication in the publisher/subscriber method, but no more than would be required for other related work such as homomorphic encryption or multi-party computations. Comparatively speaking, the latter two methods also have computational challenges for microcontrollers. Gridchain only uses a simple blockchain, in this work, chosen to be permissioned and with a relatively low computational burden suitable for a microcontroller.

Finally, after running Gridchain we confirmed what other papers concluded when trying to hide the electricity consumption from attackers: the number of smart meters in the network affects the final privacy of the whole network [11].

8 Conclusions

In this paper, we investigated the privacy concerns regarding detailed readings for billing purposes from smart meters or similar devices. Others have previously suggested that a local supply (battery) can be used to anonymize the usage patterns so that less information about the household can be gained from the energy traces [68]. This supply can either be used directly to hide consumption patterns or as an external source to disturb the signal from a differential-private way. Even though some households may have large-scale batteries in the future as electric cars, today, there is no such deployment making these schemes exciting but challenging to use in practice.

We presented Gridchain, a method where households can opt-in to hide their consumption patterns. Households form groups where they can trade *real consumption* among themselves to achieve *reported consumption* that would be resistant to NILM. Gridchain is built on a publish/subscribe model and uses a permissioned blockchain to record any trades, meaning that dishonest households can be discovered and punished if they steal from other households in the group or the electricity company in the end. Opting into Gridchain would be as simple as changing the software of the smart meter. Gridchain does not need large batteries in the household as previous literature requires. However, any such addition to some group members would benefit the group. Last, we implemented and released a proof of concept of Gridchain and used public datasets to allow reproducibility.

Funding This work was partially supported by the Swedish Foundation for Strategic Research (SSF), the Swedish Research Council (VR), the Swedish Civil Contingencies Agency (MSB) through the projects “RICS2” and “RIOT”, the Vinnova-funded project “KIDSAM”, Knut and Alice Wallenberg Foundation, and the European Community’s Horizon 2020 Framework Programme through the “UNITED-GRID” project under grant agreement 773717.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Abbas, S., Javaid, N.: Blockchain based vehicular trust management and less dense area optimization. In: FIT, pp. 250–2505 (2019)
2. Aiad, M., Lee, P.H.: Non-intrusive load disaggregation with adaptive estimations of devices main power effects and two-way interactions. Energy Build. **130**, 131–139 (2016)
3. Al-Turjman, F., Abujubbeh, M.: IoT-enabled smart grid via SM: An overview. Future Generation Computer Systems **96**, 579–590 (2019). https://doi.org/10.1016/j.future.2019.02.012http://www.sciencedirect.com/science/article/pii/S0167739X1831759X
4. Alam, A., Islam, M.T., Ferdous, A.: Towards blockchain-based electricity trading system and cyber resilient microgrids. In: 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE), pp. 1–5 (2019). https://doi.org/10.1109/ECACE.2019.8679442
5. Albano, M., Ferreira, L.L., Pinho, L.M., Alkhawaja, A.R.: Message-oriented middleware for smart grids. Comput. Stand. Interfaces **38**, 133–143 (2015)
6. Asghar, M.R., Dán, G., Miorandi, D., Chlamtac, I.: Smart meter data privacy: A survey. IEEE Commun. Surv. Tutor. **19**(4), 2820–2835 (2017)
7. Avula, R.R., Oechtering, T.J.: Privacy-enhancing appliance filtering for smart meters. In: ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 9042–9046 (2022). https://doi.org/10.1109/ICASSP43922.2022.9746644
8. Backes, M., Meiser, S.: Differentially private smart metering with battery recharging. In: DPM, pp. 194–212 (2014)
9. Baranski, M., Voss J: Nonintrusive appliance load monitoring based on an optical sensor. In: PTC, vol. 4, p. 8 (2003)
10. Birman, K., Joseph, T.: Exploiting virtual synchrony in distributed systems. SIGOPS Oper. Syst. Rev. **21**(5), 123–138 (1987)
11. Buescher, N., Boukoros, S., Bauregger, S., Katzenbeisser, S.: Two is not enough: privacy assessment of aggregation schemes in smart metering. Proc. Priv. Enhanc. Technol. **2017**(4), 198–214 (2017)
12. Cai, W., Wang, Z., Ernst, J.B., Hong, Z., Feng, C., Leung, V.C.M.: Decentralized applications: the blockchain-empowered software system. IEEE Access **6**, 53019–53033 (2018)
13. Chen, G., He, M., Gao, J., Liu, C., Yin, Y., Li, Q.: Blockchain-based cyber security and advanced distribution in smart grid. In: 2021 IEEE 4th International Conference on Electronics Technology (ICET), pp. 1077–1080 (2021). https://doi.org/10.1109/ICET51757.2021.9451130
14. Christidis, K., Devetsikiotis, M.: Blockchains and smart contracts for the internet of things. IEEE Access **4**, 2292–2303 (2016)
15. Conoscenti, M., Vetrò, A., De Martin, J.C.: Blockchain for the internet of things: A systematic literature review. In: AICCSA, pp. 1–6. IEEE (2016)
16. Costache, M., Tudor, V., Almgren, M., Papatriantafilou, M., Saunders, C.: Remote control of smart meters: Friend or foe? In: EC2ND, pp. 49–56. IEEE (2011)
17. Danezis, G., Fournet, C., Kohlweiss, M., Zanella-Béguelin, S.: Smart meter aggregation via secret-sharing. In: SEGS, pp. 75–80 (2013)
18. Devlin, M.A., Hayes, B.P.: Non-intrusive load monitoring and classification of activities of daily living using residential smart meter data. IEEE Trans. Consum. Electron. **65**(3), 339–348 (2019). https://doi.org/10.1109/TCE.2019.2918922
19. Dietrich, A., Leibenger, D., Sorge, C.: On the lack of anonymity of anonymized smart meter data: An empiric study. In: 2020 IEEE 45th Conference on Local Computer Networks (LCN), pp. 405–408 (2020). https://doi.org/10.1109/LCN48667.2020.9314798
20. Eugster, P.T., Felber, P.A., Guerraoui, R., Kermarrec, A.M.: The many faces of publish/subscribe. ACM Comput. Surv. **35**(2), 114–131 (2003)
21. European Commission: Cybersecurity strategy of the European Union: An open, safe and secure cyberspace. https://eur-lex.europa.eu/legal-content/EN/TXT/?qid=1553779410177&uri=CELEX:52013JC0001 (2020)
22. European Commission: Essential regulatory requirements and recommendations for data handling, data safety, and consumer protection. https://ec.europa.eu/energy/sites/ener/files/documents/Recommendations%20regulatory%20requirements%20v1_0_clean%20%282%29.pdf (2020)
23. European Commission: Smart grids and meters. https://ec.europa.eu/energy/topics/markets-and-consumers/smart-grids-and-meters/overview_en (2020)
24. Eyal, I., Gencer, A.E., Sirer, E.G., Renesse, R.V.: Bitcoin-NG: A scalable blockchain protocol. In: NSDI, pp. 45–59 (2016)
25. Global Market Insights: Smart electric meter market size. https://www.gminsights.com/industry-analysis/smart-electric-meter-market (2020)
26. Goldreich, O.: Foundations of cryptography: volume 2, basic applications. Cambridge university press (2009)
27. Greveler, U., Glösekötterz, P., Justusy, B., Loehr, D.: Multimedia content identification through smart meter power usage profiles. In: IKE, p. 1 (2012)
28. Gross, J., Breitenbach, J., Granson, W., Japs, D., Reci, A., Koengeter, A., Buettner, R.: A systematic literature review of data privacy solutions for smart meter technologies. In: 2021 IEEE International Conference on Big Data (Big Data), pp. 3305–3310 (2021). https://doi.org/10.1109/BigData52589.2021.9671814
29. Hart, G.W.: Nonintrusive appliance load monitoring. Proc. IEEE **80**(12), 1870–1891 (1992)
30. Hassan, M.U., Rehmani, M.H., Chen, J.: Differential privacy techniques for cyber physical systems: A survey. IEEE Commun. Surv. Tutor. **22**(1), 746–789 (2020). https://doi.org/10.1109/COMST.2019.2944748
31. Hong, Y., Liu, W.M., Wang, L.: Privacy preserving smart meter streaming against information leakage of appliance status. IEEE Trans. Inf. Forensics Secur. **12**(9), 2227–2241 (2017)
32. Kalogridis, G., Cepeda, R., Denic, S.Z., Lewis, T., Efthymiou, C.: ElecPrivacy: evaluating the privacy protection of electricity management algorithms. IEEE Trans. Smart Grid **2**(4), 750–758 (2011)
33. Kalogridis, G., Dave, S.: PeHEMS: Privacy enabled HEMS and load balancing prototype. In: SmartGridComm, pp. 486–491 (2012)
34. Kalogridis, G., Efthymiou, C., Denic, S.Z., Lewis, T.A., Cepeda, R.: Privacy for smart meters: Towards undetectable appliance load signatures. In: SmartGridComm, pp. 232–237 (2010)
35. Khwaja, A.S., Erkucuk, S., Anpalagan, A., Venkatesh, B.: Evaluation of noise distributions for additive and multiplicative smart meter data obfuscation. IEEE Access **10**, 27717–27735 (2022). https://doi.org/10.1109/ACCESS.2022.3157390
36. King, N.J., Jessen, P.W.: For privacy's sake: Consumer "opt out" for smart meters. Computer Law & Security Review **30**(5), 530–539 (2014)
37. Krebbs on Security: FBI: smart meter hacks likely to spread. Internet (2009). https://krebsonsecurity.com/2012/04/fbi-smart-meter-hacks-likely-to-spread/comment-page-1/

38. Kremer, S., Markowitch, O., Zhou, J.: An intensive survey of fair non-repudiation protocols. Comput. Commun. **25**(17), 1606–1621 (2002)
39. Kumar, P., Lin, Y., Bai, G., Paverd, A., Dong, J.S., Martin, A.: Smart grid metering networks: A survey on security, privacy and open research issues. IEEE Commun. Surv. Tutor. **21**(3), 2886–2927 (2019). https://doi.org/10.1109/COMST.2019.2899354
40. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. Soviet physics doklady **10**(8), 707–710 (1966)
41. Li, C., Fu, Y., Yu, F.R., Luan, T.H., Zhang, Y.: Vehicle position correction: A vehicular blockchain networks-based GPS error sharing framework. IEEE Transactions on Intelligent Transportation Systems pp. 1–15 (2020)
42. Li, F., Luo, B., Liu, P.: Secure information aggregation for smart grids using homomorphic encryption. In: SmartGridComm, pp. 327–332 (2010)
43. Li, F., Luo, B., Liu, P.: Secure and privacy-preserving information aggregation for smart grids. Int. J. Secure. Network. **6**(1), 28–39 (2011)
44. Liu, T., Li, S., Sun, W., Wu, S., Li, K., Huang, F.: Power data publishing method based on differential privacy considering spatiotemporal correlation. In: 2021 IEEE Sustainable Power and Energy Conference (iSPEC), pp. 4197–4202 (2021). https://doi.org/10.1109/iSPEC53008.2021.9735899
45. Mahmood, K., Chaudhry, S.A., Naqvi, H., Kumari, S., Li, X., Sangaiah, A.K.: An elliptic curve cryptography based lightweight authentication scheme for smart grid communication. Future Generation Computer Systems **81**, 557–565 (2018). https://doi.org/10.1016/j.future.2017.05.002http://www.sciencedirect.com/science/article/pii/S0167739X17309263
46. Marzal, A., Vidal, E.: Computation of normalized edit distance and applications. IEEE Trans. Pattern Anal. Mach. Intell. **15**(9), 926–932 (1993)
47. McLaughlin, S., McDaniel, P., Aiello, W.: Protecting consumer privacy from electric load monitoring. In: CCS, pp. 87–98 (2011)
48. Mettler, M.: Blockchain technology in healthcare: The revolution starts here. In: Healthcom, pp. 1–3 (2016)
49. Molina-Markham, A., Shenoy, P., Fu, K., Cecchet, E., Irwin, D.: Private memoirs of a smart meter. In: BuildSys, pp. 61–66 (2010)
50. MQTT.org: MQ telemetry transport. http://mqtt.org (2020)
51. Naehrig, M., Lauter, K., Vaikuntanathan, V.: Can homomorphic encryption be practical? In: CCSW, pp. 113–124 (2011)
52. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system. Tech. rep, Manubot (2019)
53. National Association of Citizens Advice Bureaux: Moving home - dealing with your energy supply. https://www.citizensadvice.org.uk/consumer/energy/energy-supply/moving-home-your-energy-supply/moving-home-dealing-with-your-energy-supply/ (2019)
54. OASIS: Advanced messaging queuing protocol. https://www.amqp.org (2020)
55. Pham, C.T., Månsson, D.: A study on realistic energy storage systems for the privacy of smart meter readings of residential users. IEEE Access **7**, 150262–150270 (2019). https://doi.org/10.1109/ACCESS.2019.2946027
56. Profentzas, C., Almgren, M., Landsiedel, O.: IoTLogBlock: Recording off-line transactions of low-power IoT devices using a blockchain. In: 2019 IEEE 44th Conference on Local Computer Networks (LCN), pp. 414–421 (2019). https://doi.org/10.1109/LCN44214.2019.8990728
57. Profentzas, C., Almgren, M., Landsiedel, O.: TinyEVM: Off-chain smart contracts on low-power IoT devices. In: 2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS), pp. 507–518 (2020). https://doi.org/10.1109/ICDCS47774.2020.00025
58. Quinn, E.L.: Smart metering and privacy: Existing laws and competing policies. Available at SSRN (2009)
59. Rahman, M.A., Manshaei, M.H., Al-Shaer, E., Shehab, M.: Secure and private data aggregation for energy consumption scheduling in smart grids. IEEE Trans. Dependable Secure Comput. **14**(2), 221–234 (2017)
60. Ristad, E.S., Yianilos, P.N.: Learning string-edit distance. IEEE Trans. Pattern Anal. Mach. Intell. **20**(5), 522–532 (1998)
61. Ruano, A., Hernandez, A., Ureña, J., Ruano, M., Garcia, J.: NILM techniques for intelligent home energy management and ambient assisted living: A review. Energies **12**(11) (2019)
62. STOMP: The simple text oriented messaging protocol. http://stomp.github.io (2020)
63. Tabrizi, F.M., Pattabiraman, K.: Design-level and code-level security analysis of IoT devices. ACM Trans. Embed. Comput. Syst. **18**(3) (2019)
64. Tan, O., Gunduz, D., Poor, H.V.: Increasing smart meter privacy through energy harvesting and storage devices. IEEE J. Sel. Areas Commun. **31**(7), 1331–1341 (2013)
65. Tudor, V., Almgren, M., Papatriantafilou, M.: Employing private data in AMI applications: Short term load forecasting using differentially private aggregated data. In: SmartWorld, pp. 404–413 (2016)
66. Tudor, V., Almgren, M., Papatriantafilou, M.: The influence of dataset characteristics on privacy preserving methods in the advanced metering infrastructure. Comput. Secur. **76**, 178–196 (2018)
67. Tudor, V., Gulisano, V., Almgren, M., Papatriantafilou, M.: BES: Differentially private event aggregation for large-scale IoT-based systems. Future Generation Computer Systems (2018)
68. Varodayan, D., Khisti, A.: Smart meter privacy using a rechargeable battery: Minimizing the rate of information leakage. In: ICASSP, pp. 1932–1935 (2011)
69. Wang, S., Cui, L., Que, J., Choi, D., Jiang, X., Cheng, S., Xie, L.: A randomized response model for privacy preserving smart metering. IEEE Trans. Smart Grid **3**(3), 1317–1324 (2012)
70. Wang, X.F., Mu, Y., Chen, R.M.: An efficient privacy-preserving aggregation and billing protocol for smart grid. Security and Communication Networks **9**(17), 4536–4547 (2016). https://doi.org/10.1002/sec.1645https://onlinelibrary.wiley.com/doi/abs/10.1002/sec.1645
71. Yao, X., Chen, Z., Tian, Y.: A lightweight attribute-based encryption scheme for the internet of things. Futur. Gener. Comput. Syst. **49**, 104–112 (2015)
72. Zeifman, M., Roth, K.: Nonintrusive appliance load monitoring: review and outlook. IEEE Trans. Consum. Electron. **57**(1), 76–84 (2011)
73. Zhuang, P., Zamir, T., Liang, H.: Blockchain for cybersecurity in smart grid: a comprehensive survey. IEEE Trans. Industr. Inf. **17**(1), 3–19 (2021). https://doi.org/10.1109/TII.2020.2998479
74. Zoha, A., Gluhak, A., Imran, M., Rajasegarar, S.: Non-intrusive load monitoring approaches for disaggregated energy sensing: a survey. Sensors **12**(12), 16838–16866 (2012)