Privacy-Preserving Fog Aggregation of Smart Grid Data Using Dynamic Differentially-Private Data Perturbation

FAWAZ KSERAWI†, SAEED AL-MARRI††, QUTAIBAH MALLUHI‡

†Computer Science and Engineering Department, College of Engineering, Qatar University, Doha, 2713 Qatar
‡Corresponding author: Fawaz Kserawi (e-mail: fawaz@qu.edu.qa).

This publication was made possible in part by grants NPRP12C-33905-SP-66 and GSRA7-1-0517-20065 from the Qatar National Research Fund (a member of Qatar Foundation). The findings achieved herein are solely the responsibility of the authors.

ABSTRACT The edge of the smart grid has a massive number of power and resource-constrained interconnected devices. Mainly, smart meters report power consumption data from consumer homes, industrial buildings, and other connected infrastructures. Multiple approaches were proposed in the literature to preserve consumers’ privacy by altering the data via additive noise, masking, or other data obfuscation techniques. A significant body of work in the literature employs differential privacy methods with constraining predefined parameters to achieve the optimal trade-off between privacy and utility of the data. However, billing accuracy can be degraded by using such additive noise techniques. We propose a differentially-private model that perturbs data by adding noise obtained from a virtual chargeable battery, while maintaining billing accuracy. Our model utilizes fog-computing data aggregation with lightweight cryptographic primitives to ensure the authenticity and confidentiality of data generated by low-end devices. We describe our differentially-private model with flexible constraints and a dynamic window algorithm to maintain the privacy-budget loss in infinitely generated time-series data. Our experimental results show a possible decrease in data perturbation error by 51.7% and 61.2% for smart meters and fog aggregators perturbed data, respectively, compared to the commonly used Gaussian mechanism.

INDEX TERMS Advanced Metering Infrastructure, Differential Privacy, Electrical Grid, Internet of Things, Information Privacy, Smart Grid, Smart Meter

I. INTRODUCTION

SMART grids (SG) are implemented to fulfill the goals of energy efficiency, optimal energy distribution, seamless integration with renewable energy resources, and a stable supply of energy. The deployment of smart grids can be a requirement in some nations to meet sustainability, and green economy goals [1]. A core component that enables SG is the Advanced Metering Infrastructure (AMI) subsystem, which allows for both-way connectivity between energy service providers and their consumers. This two-way connection facilitates real-time operations of the grid in load-balancing and optimizing workloads of the power grid. That, in turn, converts to more efficiency and cost savings thanks to fine-grained metrics collection by the AMI system. The AMI subsystem contains a distributed fleet of smart meter (SM) devices deployed at houses, local businesses, and industries. SMs periodically send power consumption metrics to the utility service providers or intermediate collection points. The availability of such data can allow the service provider to offer better dynamic pricing to consumers, optimize power delivery by predicting consumption patterns, and enhance power transmission planning. Therefore, data aggregation is a core task for every SG implementing a modern AMI.

Effective SG data aggregation requires mega-scale deployments of low-end devices (e.g. smart meters) that are interconnected. With a lack of an effective security model in a smart grid environment, the SG will be exposed to severe threats, some of which are presented in [2]–[6]. For example, a dishonest data aggregator could process the aggregated SM metrics to infer private information. Leaked information can include the number of residents in a given household and their availability. Additionally, the adversary can conclude the types of devices and appliances used within that particular household. The leaked data can be shared or sold without...
consumers’ consent, exposing and violating their privacy. Furthermore, the collected data can be exploited in many other ways [7]. For example, a user load profile can reveal various private information such as time of sleep or time of a specific appliance usage during the day [8] as indicated by Fig. 1.

**II. PROBLEM FORMULATION**
While frequent reporting of granular SM metrics to energy suppliers is beneficial, it can introduce major privacy problems. For example, adversaries can infer accurate high-resolution readings on power consumption by eavesdropping on the communication data link, violating consumer privacy. On the other hand, keeping the data hidden makes it unusable and defies the point of deploying smart meters. Therefore, it is crucial to find a solution that retains the utility of the data but preserves users’ privacy. Such solutions need to protect the SG data on all levels, from consumers to data collection points and energy providers. The computational cost of these solutions needs to be considered as the SG is made of low-end devices with constrained computation and communication abilities. Bandwidth is another essential key to consider as many low-end smart meter devices typically report their consumption data to a centralized point at a high rate.

Load balancing the power load is a common way of perturbing data and adding noise to it as seen in [2], [10]–[16]. Charging and discharging a battery to add noisy data to the power load is a straightforward concept. Negative noise is added to the power load whenever the battery is physically fed with power (electrically charged). However, the opposite is true when a positive noise is later added to the power load whenever the battery is physically drained of energy (discharged). A hardware component is responsible for charging and discharging the physical battery. The physical chargeable battery approach is costly and limited in the sense that the capacity of the battery limits the amount of possible data perturbation. Different methods in the literature use a low amount of noise with differential privacy (DP) techniques to maintain the utility of data [11]. However, differential privacy approaches fall short when it comes to avoiding data loss, such as inability to conduct accurate consumer billing, losing other critical data resolution, or failure to reconstruct data [17].

This article introduces a very lightweight, secure, and non-expensive aggregation method that preserves consumers’ privacy and the safety of transferred data. We employ lightweight cryptographic techniques to encrypt data, authenticate its sources, and ensure its protection in the presence of an eavesdropper. Moreover, we utilize noise generation from a virtual battery model as a cost-effective alternative to physical batteries, which preserves high-resolution time-series SM data privacy. Our approach protects against potential adversarial data aggregators or untrusted service providers while maintaining data utility, along with proper consumer billing. A technique based on fog-computing data aggregation is discussed, along with a novel dynamic window algorithm for differential privacy. This work is based on a graduate thesis work by the first author [18]. The preliminary idea and concept of virtual battery first appeared in our previous conference publication [19]. This paper expands on our prior work by introducing a novel differentially-private dynamic window algorithm. Additionally, we present new experimental results with a detailed discussion that evaluates and shows the improvements over the current traditional approaches in the literature. Our work ensures the privacy of SM devices while preserving the billing accuracy through fog-enabled data aggregation without trusting the intermediate fog nodes or the service provider.

The remainder of this paper is organized as follows: Section III establishes background knowledge needed to introduce our work in following sections. Section IV describes our proposed model. Section V discusses the main results of our experimental evaluation. Section VI describes related work found in the literature. Section VII concludes this paper with remarks on future work.

**III. BACKGROUND**

### A. FOG AGGREGATION
Typically, a large number of power- and resource-constrained IoT devices are deployed on the SG. These IoT devices cannot process the generated data themselves due to their limited capabilities and therefore need to delegate the processing to other resources. Moreover, the data generated by IoT devices must be transmitted using the consumers’ network or a separate wireless network, to the backend server. A significant geographical distance between the IoT device and the service provider means more power is required to transmit data between them. Therefore, there is a need for a power-efficient method to transmit data. In addition, the large number of these devices can pose a networking challenge by introducing a communication bottleneck at the centralized receiving service provider server. To overcome these challenges, we use a fog-computing architecture to meet the requirements of SG deployments [11]. The fog nodes compute and aggregate data...
from many SMs, where the aggregated result is further sent to the backend server. In our scenario the SG service providers manage the backend server. The advantage of applying fog computing is aggregating the data on fog nodes instead of sending all SMs traffic to the service provider, causing network congestion. In addition, fog nodes can perform some processing tasks on data, reducing processing required on backend servers and protecting data from service providers. This architecture performs better by offloading some work from service providers’ servers, preventing potential computing and networking bottlenecks. Such limitations can be introduced at the backend server, which handles requests from a massive number of IoT devices.

Fog-computing-enabled smart grid architectures usually employ the fog-computing layer as an intermediate layer to save cost, improve scalability and reduce the complexity of the system [20]. Figure 2 depicts a common fog-computing architecture. The advantages of such architecture are discussed in [21] and [11]. This paper adopts intermediate fog nodes, or data aggregators, in the middle layer of the aggregation architecture. In our architecture, fog nodes are an untrusted component of the system and are not allowed to access the private data produced by smart meters. Therefore, the smart meter data is obfuscated using noise that achieves differential privacy. The main objective of a data aggregator is to compute statistical data for an individual smart meter or a group of smart meters during a given period. By doing this, granular data is kept hidden from the utility provider. Additionally, the communication cost is reduced as aggregators perform the tasks of data collection and computing statistics rather than sending data directly to service providers. This paper interchangeably refers to fog nodes as data aggregators and IoT nodes as smart meters.

**B. DIFFERENTIAL PRIVACY**

A trial for privacy in SGs was presented in [22] where the possibilities of data leakage were tested in various scenarios with or without a trusted party. First, perturbed data is submitted by the service provider and analyzed for a possible information leak. Then, if the probability of finding the actual data is higher than random guessing, privacy is considered breached. However, when accessing a larger dataset, initial conditions can vary for the adversary, rendering this approach to be inaccurate [12]. Many methods for data obfuscation are present in the literature; however, the current state of the art mainly uses implementations of differential privacy.

Differential privacy as introduced in [23], is a mechanism that presents dataset semantics and patterns while preserving the privacy of any individual data point in that dataset. In differential privacy, if any user’s contributions to the dataset were insignificant enough, then the result from a query to that dataset would not leak a significant amount of information about that individual user. Consequently, the overall result of a dataset query should not change when the data record of any individual user is altered (added, removed, or modified), thus assuring dataset contributors’ privacy. DP is a powerful concept for providing privacy guarantees with intriguing properties, namely post-processing, closure, and composition. In the context of smart grids, DP can obfuscate aggregated SM readings to ensure privacy while maintaining the benefits of data analysis.

1) Definition of Differential Privacy

Assuming a mechanism $M: X_n \rightarrow Y$. For any two neighbouring data-sets $X, X' \in X_n$ that are different in one entry. We say that $M$ is $\varepsilon$-differentially private if, for all neighboring $X, X'$, and all $T \subseteq Y$, we have: $Pr[M(X) \in T] \leq e^\varepsilon Pr[M(X') \in T]$, where $M$ is a randomization mechanism and can be an algorithm that introduces additive noise to the original data $X$. In the literature related to differential privacy, the word mechanism is often used; however, both terms “mechanism” and “algorithm” are used interchangeably.

In the literature, usually, Laplacian or Gaussian noise is introduced. Several algorithms and differential privacy properties are discussed in [24]. The previous definition states that if the effect of making an arbitrary single replacement in the data-set is small enough, the query result cannot infer a single individual’s data. Here, the difference between $X$ and $X'$ is the data belonging to one entity in the data-set. Therefore, we can get one data set from another by either adding, removing or changing this entity’s data.

There are algorithms that, by perturbing data, can turn query results into differentially private ones. In a smart grid, we consider a single query equivalent to one aggregate. In differential privacy, a parameter $\varepsilon$ shows the privacy strength and is referred to as the privacy budget. The perturbation applied to the data in differential privacy is inversely proportional to $\varepsilon$, where a smaller $\varepsilon$ produces better privacy but less accuracy and vise versa. It is a challenge in smart metering to balance $\varepsilon$ value to tweak the amount of added noise that does not violate privacy yet preserves data utility. A trade-off between privacy and accuracy is presented in [22].
2) Properties of Differential Privacy
Differential privacy offers several properties that make it modular and trivial to implement.

1) Post-Processing: A valuable property of differential privacy is that once the data is privatized with differential privacy, the privacy will not be breached if the data is not used again.

Let $M : X^n \rightarrow Y$ be $\varepsilon$-differentially private and $F : Y \rightarrow Z$ be an arbitrary randomized mapping. Then $F \circ M = F(M(X))$ is $\varepsilon$-differentially private.

2) Composition: Suppose $M = (M_1, ..., M_k)$ is a sequence of algorithms, where $M_i$ is $(\varepsilon_i, \delta_i)$-differentially private and the algorithms $M_i$’s are potentially chosen sequentially and adaptively. Then $M$ is $(\sum_{i=1}^{k} \varepsilon_i, \sum_{i=1}^{k} \delta_i)$-differentially private.

It is not always clear what the value of $\varepsilon$ should be to maintain privacy since differential privacy is usually added to static data by a trusted curator [25]. In time-series and growing data-sets, it is unfeasible to apply differential privacy to static data by a trusted curator [25]. In time-series and growing data-sets, it is unfeasible to apply differential privacy to static data by a trusted curator [25]. In time-series and growing data-sets, it is unfeasible to apply differential privacy to static data by a trusted curator [25]. In time-series and growing data-sets, it is unfeasible to apply differential privacy to static data by a trusted curator [25].

For example, at each time iteration, $t_i$ when applying differential privacy on the power load at a window of time $w_j$ and each power load $X(t)$ is summarized for a window; for the first period of 10 minutes $t = 0$ to $t = 10$ the $w_0$ value is $w_0 = \sum_{t=0}^{10} X(t)$. Therefore the window $w_j$ between $t = i$ and $t = i'$ values can be calculated by:

$$w_j = \sum_{i=i}^{i'} X(t_i)$$

For example, applying differential privacy with parameters $\varepsilon_1$, $\varepsilon_2$ and $\varepsilon_3$ on time windows $w_1$, $w_2$ and $w_3$ respectively; the composition property states that the overall privacy of all three windows is: $\varepsilon_{total} = \varepsilon_1 + \varepsilon_2 + \varepsilon_3$. The previous equation shows that the values of $\varepsilon$ will increase over time. We found that many solutions in the literature ignore the deterioration of $\varepsilon$ over time, and we will present our solution for this in later sections.

Sensitivity is another parameter to consider when implementing differential privacy. Sensitivity captures the influence by which a single data entry can affect the mechanism and therefore change the perturbation level needed to hide all data. Thus, the sensitivity of a function bounds the perturbation level we must introduce to preserve privacy.

In smart metering data, which is a time-series data, the maximum global value is unknown, making it challenging to measure sensitivity because future unknown readings with the highest values can break differential privacy.

IV. PROPOSED MODEL
Our proposed virtual battery (VB) system model assumes that the SM devices are resistant to physical attacks such as physically tampering with the device to change the values it produces. Some approaches in the literature aid in protecting against physical as well as software adversaries known as trusted execution environments [26]. We propose a fog-computing hierarchical architecture that ensures accurate billing and provides data perturbation via additive noise received from the VB. Adopting this VB approach offers the advantage of using more power loads without enduring the costs of having a large physical battery. Our proposed VB system utilizes a distributed data aggregation architecture based on fog computing, producing perturbation noise on each SM and data aggregator. We consider data aggregators and utility providers in our model to not be trusted, hence our use of differential privacy techniques on SMs and data aggregators.

FIGURE 3. The architectural design based on the virtual battery concept from an individual SM device

A. A VIRTUAL VALUE AS A BATTERY
Essentially, our VB is a shared value known to the fog-computing data collecting node, the power utility provider, and the smart meter device. Figure 3 depicts the architecture for establishing this value.

Table 1 contains a list of symbols used in this paper. We use $VB(t, m)$ to denote the value of the VB at a point in time $t$ of a period $m$ where $m$ starts at $t_0 = 0$ and ends with $t_f$. Therefore, at the start of period $m_j$, the initial value of VB is $VB(t_0, m_j)$. VB value should be $VB(t_f, m)$ after the entire period $m$ has passed. For the first update of the SM at the first period $m_0$, the value of $VB(t_0, m_0)$ is zero; the SM then transmits this value to the utility provider at the end of $m_0$ for the next period $m_1$. Below, we describe the steps of perturbing the original data with additive noise using a VB on an individual SM:

1) At the beginning of a relatively long period $m_j$, for example a month, the SM sends $VB(t_0, m_j)$ value to the utility provider for acknowledgment. For the first
TABLE 1. VB system symbols

| Symbol | Meaning |
|--------|---------|
| \( n \) | Time period of smart meter reading at iteration \( i \) |
| \( t \) | Time period of smart meter \( m \) |
| \( t_0 \) | Initial time in a period \( m \) |
| \( t_f \) | Last time value in period \( m \) |
| \( X_n(t) \) | Power load reading of smart meter \( n \) at time \( t \) |
| \( L_n(t) \) | Encrypted power load reading plus noise of smart meter \( n \) at time \( t \) |
| \( V_B_n(t) \) | Virtual Battery value of smart meter \( n \) at time \( t \) |
| \( N_n(t) \) | Noise added at time \( t \) to power load of smart meter \( n \) |
| \( A_n(t) \) | Coarse-grained aggregated value of smart meter \( n \) load plus noise |
| \( TC_n(t) \) | Total consumption for the smart meter \( n \) |
| \( U(t) \) | Total consumption of all smart meters at time \( t \) |

For dynamic billing, it is possible to use multiple virtual batteries; for example, when billing is different between daytime and nighttime, we can use \( V_B_{day} \), \( V_B_{night} \). Noise added at daytime is added to \( V_B_{day} \), and nighttime noise is added to \( V_B_{night} \). Both values are sent from the smart meter to the utility provider.

Other privacy-preserving approaches that rely on physical batteries perturbation, such as [10] [13], can take advantage of our virtual system; the benefits can be in terms of cost and perturbation level. Cost is less since maintenance and initial battery costs are no longer needed. Perturbation is enhanced since the capacity is no longer tied to the actual capability of the physical battery. Additionally, our VB method is agnostic to the used resource and can be used with non-electric utilities, such as natural gas or water. Moreover, other models that rely on noise from power storage devices or consumer devices such as [14] can replace that reliance with noise from a VB system. Methods that adopt differential privacy [11] can utilize the Gaussian mechanism noise from the VB to achieve robust and correct consumer bill estimation. For instance, when a consumer suddenly demands a huge amount of power, causing a spike in demand, the amount of perturbation noise needed to obfuscate that spike is relatively large. Therefore, if this perturbation noise is not adequately addressed in calculating the consumer’s bill, it will lead to inaccurate billing. Thus, we can effectively guarantee the power-load demand from the consumer to be limited via thresholding the power load at a maximum value and adding the subtracted value to the VB. For example, if \( P_{peak} \) is exceeded at time \( (t_{i-1}, m_j) \) the value of \( V_B(t_i, m_j) \) will be the previous virtual battery value \( V_B(t_{i-1}, m_j) \) minus the absolute value of \( P_{peak} \) subtracted from \( X(t_{i-1}, m_j) \) as shown in equation (6):

\[
V_B(t_i, m_j) = V_B(t_{i-1}, m_j) - |X(t_{i-1}, m_j) - P_{peak}| \tag{6}
\]

Furthermore, it is possible to use excessive noise in any data perturbation method while adding its consumption to the VB to maintain billing correctness since VB value will be sent later for billing. Differential privacy can achieve an extra level of privacy by using the VB noise with a lower value of \( \varepsilon \) and, therefore, a better privacy budget loss. This ensures greater privacy guarantees at some cost of data utility. Our system adopts a distributed scheme of differential privacy.
B. DISTRIBUTED DATA COLLECTION SYSTEM

Our fog-computing distributed system is comprised of three layers, as shown in Fig. 4. The lowest layer is all SM devices, the data aggregator fog nodes are in the middle layer, and the service provider backend servers are at the highest. Multiple methods that can be utilized for the SM-to-aggregator communication are shown in [27].

The aggregator decrypts, accumulates values, encrypts the results, and finally delivers these results to the utility provider; further details regarding the cryptographic techniques used are discussed in Section IV-D. At last, the utility provider decrypts the aggregated results. Then, at recurring intervals, SMs forward encrypted VB data to the utility provider to be compared and accumulated to generate a correct power bill for consumers.

As an example, for a SM that updates its power load every minute, we add noise $N(t_i, m_j)$ to each reading at $(t_i, m_j)$ where $(t_i, m_j) - (t_{i-1}, m_j) = 1$ minute. At a monthly period $m_j$, from $t_0$ to $t_f$, the VB value would include all added noise from $N(t_i, m_j)$ to $N(t_f, m_j)$. Therefore, the perturbed load can be computed with: $L(t_f, m_j) = \sum_{i=0}^{f} (X(t_i, m_j) + N(t_i, m_j))$. VB data is sent from the SM to the utility provider at the end of $m_j$ to be accounted for in billing. The actual load can be computed using: $ActualLoad(t_f, m_j) = L(t_f, m_j) - VB(t_f, m_j)$ where actual load is equal to the noise subtracted from the perturbed load. To stop the utility provider from inferring other information by observing perturbed data and power consumption, we send VB values after a long period. Each SM device $SM_n$ contains a VB value with an initial value $VB_n(t_0, m_j)$. Initially, at the start of the first period $m_0$, the value $VB_n(t_0, m_0)$ is equal to zero. For the next period $m_1$, $VB(t_0, m_1)$ is already sent from the SM to the utility provider. It is worth mentioning that $VB(t_0, m_1)$ for the period $m_1$ is equal to $VB(t_f, m_0)$ at the end of period $m_0$: $VB(t_f, m_0) = VB(t_0, m_1)$.

Fig. 4 explains the used symbols. The following steps describe the complete aggregation process of a period $m$ where $Enc$ and $Dec$ are encryption and decryption methods discussed in Section IV-D.

1) Virtual batteries $VB_n$ for each smart meter device $SM_n$ with values $VB_n(t_0, m_j)$ are encrypted using methods discussed in Section IV-D and sent to the utility provider by the SMs. At the first update by SMs, or when SMs are first powered-on, $VB_n(t_0, m_0)$ is equal to zero.

2) The SM device $SM_n$ containing a VB value of $VB_n$ sends the initial value $VB_n(t_0, m_j)$ to the utility provider. $SM_n$ then performs data perturbation on its original raw data at time period $(t_1, m_j) : X_n(t_1, m_j)$ with noise $N_n(t_1, m_j)$. The added noise is taken from $VB_n(t_0, m_j)$ by setting the new value $VB_n(t_1, m_j)$ from (2). $SM_n$ then sends $L_n(t_1, m_j)$ to the data aggregator, which is calculated by

$$L_n(t_1, m_j) = Enc[X_n(t_1, m_j) + N_n(t_1, m_j)]. \quad (7)$$

Depending on the granularity of the data, the value of period $m_j$ is determined. During the period $m_j$, possibly a month, $VB_n(t_f, m_j)$ is calculated by adding all generated noise over this $m_j$ period to the VB.
$VB_n(t_f, m_j)$ value is then encrypted before transmission to the service provider to bill the corresponding $SM_n$ device 

$$VB_n(t_f, m_j) = Enc[VB_n(t_0, m_j) - \sum_{i=0}^{n} N_n(t_i, m_j)]. \quad (8)$$

3) During a window of time $w$, an aggregator $A_n$ receives a number of perturbed and encrypted values of consumption $L_{sm1}(t_i, m_j)$ from $n$ and calculates $\sum_{i\in w} L_{sm1}(t_i, m_j)$ during $w$ from $SM_1$. Each value of $\sum_{i\in w} L_{sm1}(t_i, m_j)$ at time $t_i$ is decrypted and the aggregation is done on the new coarse grained $w$. Aggregator adds its own noise to this load for a parallel differential privacy $N_a(t_i, m_j)$. With the aggregated value starting at $t_i$ and finishing at the end of $w$ with $(t_i', m_j)$ we calculate and encrypt the consumption of $sm_1$ by:

$$A_n(t_i', m_j) = Enc[\sum_{k=1}^{i'} Dec(L_{sm1}(t_k, m_j))] + N_a(t_i', m_j). \quad (9)$$

For example the first time window of 30 minutes in $m_0$ has values of $(t_i, m_0)$ to $(t_i', m_0) = 30$ we have:

$$A_n(t_i', m_0) = Enc[\sum_{k=0}^{30} Dec(L_{sm1}(t_k, m_0))] + N_a(30, m_0)$$

4) For time $(t_i, m_j)$, each aggregator sends an $n$ number of aggregated values that the utility provider then receives. The perturbed values of power load for all SMs are then decrypted and finally summarized by the utility provider for the period $(t_i, m_j)$

$$U(t_i, m_j) = \sum_{i=1}^{n} Dec[A_i(t_i, m_j)]. \quad (10)$$

5) The utility provider calculates $A_n(t_f, m_j)$ which is the power load plus noise for an individual SM $n$. The utility provider subtracts the consumed value of the VB from $A_n(t_f, m_j)$ to compute the total consumption $TC$ for the period $(t_f, m_j)$:

$$TC_n(t_f, m_j) = Dec[A_n(t_f, m_j) - (Dec[VB_n(t_0, m_j)] - Dec[VB_n(t_f, m_j)])] \quad (11)$$

2) Regional Aggregation

Regional aggregation is when the service provider requires an instant power consumption report of a neighborhood or a region. Regional aggregation is not related to billing, and it is mainly used for planning power distribution and optimizing power generation. However, applying differential privacy by aggregating SM readings to more coarse-grained data causes a delay in transmitting the final result as either the SM or the aggregator is waiting for the following values to be accounted for. Therefore, this section discusses the aggregation of SM devices connected to an aggregator summarizing regional data. Our proposed system utilizes an algorithm that uses a dynamic window size which is discussed in section [IV-C3].

For example, in our model, if the SM power reading is low, resulting in a small error value, the window size $w$, which is adaptively set by the algorithm, might be selected to be repeatedly large. Therefore, if an aggregator utilizes our algorithm to summarize the power load of a region, then the large $w$ value may cause an undesirable delay in updating power readings to the service provider. Hence, regional aggregation readings are sent separately in case a large window size causes a large delay in updating regional power consumption. The regional aggregation is conducted over high-resolution data generated by SM devices. Here, the level of granularity is limited only by the max window size $w_{max}$ of the SM device, which in our system is relatively small compared to the aggregator’s $w_{max}$. Therefore, the delay for regional aggregation will be small since its value is limited by the SMs $w_{max}$ value. The aggregated summaries do not contain SM identifiable information. Moreover, neither cost nor VB values are sent, preventing the inferral of private data. Aggregators send only the total load consumption of the neighborhood for their connected SM devices to the utility provider. [I] describes a similar approach.

1) For a window specified for regional aggregation $w_{reg} = w_{sm_{max}}$ where $w_{sm_{max}}$ is the $w_{max}$ parameter set for SMs, an aggregator $A$ receives a number of perturbed and encrypted values of consumption for $n$ number of smart meters $L_{sm1}$ from equation[7] and receives $(\sum_{i\in w_{reg}} L_{sm1}(t_i))$ during $w_{reg}$ from each SM in the region; where $x$ is the number of connected regional SMs. Each value of $L_{sm1}$ is decrypted, and the aggregators perturbs the sum of all SMs readings during $w_{reg}$. With the aggregated value starting at $t_i$ and finishing at the end of $w_{reg}$ with $t_i'$, we calculate regional consumption of $A$ by:

$$A(t_i') = Enc[\sum_{x=1}^{i'} \sum_{k=1}^{n} Dec[L_{sm1}(t_k)]) + N_a(t_i')] \quad (12)$$

2) The perturbed regional aggregated value of $N$ aggregators are sent to the utility provider individually. Each aggregated value can be analysed for power distribution and prediction for the area connected to the specific aggregator. The utility provider decrypts and summarizes values from all regions power consumption:

$$U(t) = \sum_{i=1}^{N} Dec[A_i(t)] \quad (13)$$

C. DIFFERENTIAL PRIVACY
1) Gaussian Mechanism

Gaussian mechanism decomposition implements noise collected from all participants. Each participant produces little amounts of noise, and the privacy of the consumer is ensured if the summarized noise from all participants has a $\sigma$ standard deviation. Theorem A.1. [IV-C1a] from [24] gives us the following definition:

Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$ be a function of $d$-dimensions, and its $\ell_2$ sensitivity is defined as $\Delta_2 f = \max_{x,y \in \mathbb{R}^d} \|f(x) - f(y)\|_2$. The Gaussian mechanism that has the $\delta$ parameter adds noise scaled to $N(0, \sigma^2)$ to every $d$ components of the output.

a: Theorem A.1

Let $\varepsilon \in (0, 1)$ be arbitrary. For $e^2 > 2\ln(1.25/\delta)$ the Gaussian mechanism with parameter $\sigma \geq c\Delta_2 f/\varepsilon$ is $(\varepsilon, \delta)$-differentially private.

Following the parameters of distributed differential privacy, we have:

$$\hat{x}(t) = \sum_{i=1}^{n} (\hat{x}_i(t)) = \sum_{i=1}^{n} (x_i(t) + r_i(t)) \quad (14)$$

Where the number of user is $n$, $x_i$ is the data generated at time $t$ for user $i$, the additive noise is $r$, and $\sigma$ satisfies the Theorem A.1. [IV-C1a]

2) Differential Privacy in Distributed Systems

In a distributed setting, distributed differential privacy uses noise aggregated from several contributors. While approaches to preserve privacy based on differential privacy mechanisms are well discussed in the related field [23], our approach offers a dynamic window algorithm, and a data denoising capability via seamless deduction of the sum of added noise from the VB value. This enables correct billing for consumers while preserving privacy, improving over traditional methods. Looking closely to Theorem A.1 [IV-C1a] $r$ is the additive noise which gets deducted from VB value for each SM to preserve the value of overall added noise for a precise billing. Moreover, the utility provider can grant clients the option for absolute privacy with no benefit of data analysis by using high level of perturbation that does not affect billing. A completely different SM software can present data analysis and suggestions to the user since the SM device can get the value of the $VB(t)$ without relying on other parties, such as the aggregation points or utility provider for providing statistics. Although we show a single VB per smart meter device in our experiments, our proposed VB system supports many methods of data perturbation.

a: Sensitivity

From Theorem A.1. [IV-C1a] in section [IV-C1] we have the sensitivity $S$ of our algorithm $M$ is:

$$S(M) = \Delta_2 f = \max_{x,y} \|f(x) - f(y)\|_2 \quad (15)$$

Various articles in the literature propose frameworks that implement a point-wise sensitivity. However, we argue that this does not guarantee differential privacy, as the sensitivity calculation should account for the entire data-set. In a smart grid, not knowing all power readings in the entire dataset, as future SM readings are continuously updated from SMs, predicting the sensitivity is a non-trivial task. A sensible way is needed to determine the sensitivity in a private manner [12]. We employ an algorithm that uses several parameters to ensure bounds on future sensitivity. The algorithm uses a dynamic window bounded between $w_{\text{min}}$ and $w_{\text{max}}$ which guarantees a differential privacy and limits the maximum power value at $P_{\text{peak}}$. Furthermore, the algorithm ensures that the error value of the perturbation does not exceed $err_{\text{max}}$.

We explain the previously mentioned parameters and algorithm in section [IV-C3] and Table 2. The sensitivity can be calculated for the dynamic window $w$ bounded by $w_{\text{min}}$, $w_{\text{max}}$.

Within the soft grid that applies differential privacy on a dynamic and bound $w$ we know that the maximum power consumption for $w$ is the maximum power $P_{\text{peak}}$ applied on all readings inside $w_{\text{max}}$. On the other hand, the lowest power reading summarized for $w$ will be the lowest power reading possible, which is zero, perturbed by the least differentially private noise $N$ applied in $w_{\text{min}}$. By applying the previous statements, as we have bounded the future maximum consumption by selecting a dynamic window size, we can measure sensitivity by:

$$S(M) = \Delta_2 f = P_{\text{peak}}/w_{\text{max}} \quad (16)$$

b: Standard Deviation

In a one dimensional dataset, by applying Theorem A.1. [IV-C1a] we can calculate the standard deviation $\sigma$ of the Gaussian mechanism where $\varepsilon \in (0, 1)$ and $\delta \in (0, 1)$:

$$\sigma^2 = 2\ln(1.25/\delta) \cdot (\Delta f)^2/\varepsilon^2 = 2\ln(1.25/\delta) \cdot S(M)^2/\varepsilon^2 \quad (17)$$

c: Privacy Budget

From the differential privacy composition theorem, we know that for multiple algorithms applies on the same data the privacy budget parameters $\varepsilon$ and $\delta$ for each algorithm are added up. For example, applying $(\varepsilon, \delta)$-differential privacy on each window $(w_1, w_2, \ldots, w_k)$ in a space $k$ number of windows. We apply the following differentially-private parameters $(\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_k, \delta_1, \delta_2, \ldots, \delta_k)$. Therefore the overall privacy budget are: $\varepsilon = \sum_{i=1}^{k} \varepsilon_i$ and $\delta = \sum_{i=1}^{k} \delta_i$ meaning for the first window we lose from the privacy budget $\varepsilon_1 = \varepsilon/k$ and...
δ_l = δ/k. Therefore, it is important to increase the window size when possible in order to consume less privacy budget.

3) Dynamic Window Differential Privacy
In the SG, the data stream is considered infinite. Therefore, the privacy budget deteriorates over time, and many bounding algorithms exist in the literature to address this [28, 29]. To our knowledge, there is no algorithm in the literature that considers our chosen parameters. In this section we define the parameters used as shown in Table 2.

1) \( w_{min} \) is the minimum window size defined by the Utility Provider, which is the number of power readings in the time series of the fine-grained smart meter or aggregator readings. \( w_{min} \) should not be selected to be too small, as that would unnecessarily drain the privacy budget, and it should be larger for the aggregator. In our model, the perturbation algorithm will increase the number of the included power readings beginning at \( w_{min} \) until one of the two parameters is reached \( err_{\text{max}} \) or \( w_{\text{max}} \).

2) \( w_{max} \) is the maximum number of power readings points allowed by the utility provider. This value is required and set by the utility provider to limit the windows of periodic updates for data statistics. For example, at night, where there could be zero power consumption, our algorithm may increase \( w \) and not be bounded by the error for an extended amount of time. It is important to consider that the larger the value of \( w_{max} \), the less privacy budget loss; therefore, this value should consider privacy vs. utility provider updates requirements.

3) \( P_{\text{peak}} \) is the peak power load, which is calculated by applying noise on increasing similar power values up to maximum allowed power in \( w_{max} \) until \( err_{\text{max}} \) value is reached. The \( P_{\text{peak}} \) value is used to achieve better performance in the algorithm since if adding the total consumption over \( w \) exceeds \( P_{\text{peak}} \) value, we know that \( w \) will break \( err_{\text{max}} \) without the need to generate noise. Here, it may also be possible for a single reading to exceed \( P_{\text{peak}} \) we threshold all such readings and add the trimmed values to the virtual battery value. Trimming single outliers that exceed \( P_{\text{peak}} \) guarantees the value of sensitivity.

4) \( err_{\text{max}} \) is the maximum error allowed for perturbation by the utility provider. The value of \( err \) for each \( w \) is calculated by the mean absolute error:

\[
MAE = \frac{\sum_{i=1}^{n} |X_i - \bar{X}|}{n}
\]

\( w \) event \( \varepsilon \)-differential privacy was introduced in [30] and applied in [28] for \( \varepsilon \)-differential privacy to protect event sequence occurring in a window of \( w \) time. We expand on \( w \)-event differential privacy for time series data in the smart grid with minimum privacy budget loss.

```
Algorithm 1: Dynamic Window Differential Privacy Algorithm

Input: \( w_{\text{min}}, w_{\text{max}}, err_{\text{max}} \)

for current window \( w = w_{\text{min}} \)

Calculate \( P_{\text{peak}} \)

while \( err \leq err_{\text{max}} \) do

while \( w \leq w_{\text{max}} \) do

if \( P(t) \leq P_{\text{peak}} \) do

Calculate \( P(t_i) = \frac{1}{L} \sum_{i=0}^{L} X(t_i) \)

end

\( w++ \)

end

Calculate error: \( err = P(t_i)/L(t_i) \)

Apply perturbation on \( w \):

\( L(t_i) = P(t_i) + \frac{1}{L} \sum_{i=0}^{L} N(t_i) \)

end

Calculate new privacy budget parameters: \( \varepsilon = \varepsilon - \varepsilon_t, \delta = \delta - \delta_t \)

Output perturbed value for \( w_{i-1} : L(t_{i-1}) \)
```

Both SMs and aggregators use the same algorithm; however, the parameters’ values will be different for aggregators. \( w_{\text{min}} \) and \( w_{\text{max}} \) values will be larger for the aggregator, as we expect the aggregators to collect more coarse-grained data. Smart meters perturb their data only to protect it from the aggregator, since we consider aggregators to be a possible adversary. It is reasonable to assign larger values of \( \varepsilon \) the privacy budget for the smart meters, since it is less likely for the aggregator to be an adversary. Additionally, the data
will be perturbed again by the aggregator before forwarding it to the utility provider. Another benefit of increasing window size is to lower network communication costs, since both SMs and aggregators summarize the windows’ power load into a single value before sending it.

D. CRYPTOGRAPHIC METHODS

a: Diffie Hellman Key Exchange

Adopting compute-light cryptographic constructions is essential in IoT environments such as smart meters in the smart grid. Our approach requires two symmetric keys to secure connections to the utility provider and fog-aggregator to send the values of VB and load consumption, respectively. Symmetric keys are exchanged between fog aggregation nodes and every connected smart meter device and upstream utility provider. However, all keys for aggregation and smart meter nodes must be present at the utility provider servers. We utilize Diffie–Hellman key exchange \[31\] to perform any key exchanges between parties. In addition, we adopt AES \[32\] as an encryption scheme, although any symmetric algorithm can be used instead if needed in our system. Typically, authentication is accomplished with an asymmetric-key cryptographic scheme. However, we use our own simpler and more light-weight challenge-response between two parties depicted in Fig. 5 due to the inherent heavy computations of public-key schemes and the dependency on trusted authority which is a third party for key distribution.

![FIGURE 5. Initial two-way challenge-response authentication](image)

The final secret is obtained by concatenating the Diffie Hellman derived key with a utility-provider-supplied fixed ID. The utility provider sets the utility-provider-supplied fixed ID before SM device distribution. The utility provider shares a secret string for data aggregators instead, similar to the fixed ID of SM devices. This key exchange is performed once to authenticate the secret key to be used for encryption later on and never used to authenticate data. This key bi-directional challenge and response key exchange establishes protection against man-in-the-middle attacks; Whereby an attacker can impersonate both sides of the communication channel, compromising confidentiality and integrity. The utility provider, aggregators, and SM devices store their secrets locally, and they no longer perform any further key exchanges. This approach keeps our protocol light-weight, compute-efficient, and avoids third-party trust compared to other methods such as public-key cryptography.

b: Keyed-Hash Message Authentication Code (HMAC)

HMAC is a variation of the Message Authentication Code (MAC) that uses secret keys and hash functions and provides data integrity and authenticity. HMAC is an improved version of MAC since MAC is known to suffer from length-extension attack where an attacker can append data to the message without knowing the key. The implementation methodology and definitions are presented in \[33\] and \[34\]. A shared secret is used in HMAC implementation that does not require a third party’s involvement in key distribution, as is the case with public-key cryptography. Following up on the previous section, we assume that the key is already exchanged between parties:

1) The key is used to acquire two separate keys referred to as the inner and outer keys.
2) Two hash rounds are applied; the first round produces a hash from the inner key with the already encrypted message.
3) The resulting hash is hashed again, with the outer key producing the final HMAC code.

Usually, HMAC applies iterative hashing functions such as SHA-256 or SHA-512 over multiple fixed-sized blocks; for example, SHA-256 works on blocks of 512-bit. Since we cannot guarantee our communication block size, we truncate our data blocks to the proper size. The encrypted message is then sent alongside the HMAC code to the other party. The other party will then hash the message again, and the computed hashes should match the received hash authenticating the message. The definition of HMAC from \[33\] and \[35\].

\[
\text{HMAC}(K, m) = H((K' \oplus \text{ipad}) \parallel H((K' \oplus \text{opad}) \parallel m))
\]

When \(K\) is larger than block size then \(K' = H(K)\) otherwise \(K' = K\). \(K'\) is a block-sized key obtained from the secret key \(K\) either by padding with zeroes up to the block size or by hashing down to \(\leq\) block size and later padding with zeroes. Table 3 contains an explanation of the used symbols.

| Symbol | Meaning |
|--------|---------|
| \(H\) | Cryptographic hash function |
| \(K\) | Secret key |
| \(K'\) | Block-sized key |
| \(\parallel\) | Concatenation |
| \(\oplus\) | XOR |
| \(\text{ipad}\) | Block-sized outer padding |
| \(\text{opad}\) | Block-sized inner padding |

V. EXPERIMENTS AND RESULTS DISCUSSION

This section evaluates our model’s performance accuracy and compares it with the traditional Gaussian mechanism. We ap-
ply our model to an actual smart home dataset. Our dynamic window model is applied to the individual household electric power consumption dataset \cite{dataset}. The dataset contains electric power consumption measurements in a household with a one-minute sampling rate for almost four years. We apply our perturbation with the dynamic window differential privacy algorithm on smart meters and then on aggregator nodes. For every window \( w \), we aggregate the values from the smart meters and apply noise, losing an amount of privacy budget \( \varepsilon_w \). Next, the aggregator receives the perturbed data and applies another perturbation level; the noise added to achieve data perturbation is added to the virtual battery of each smart meter. Error is calculated over the most recent time series values for each time window \( w \). The values for \( w_{\text{min}}, w_{\text{max}}, \sigma_{\text{max}} \) and \( P_{\text{peak}} \) are set for the smart meter and aggregator. Naturally, the values of parameters \( w_{\text{min}}, w_{\text{max}} \) are selected to be larger for the smart meter than those used for the aggregator. Choosing a larger window size for smart meters hides the smart meter’s data from the aggregator and consumes less privacy budget for the smart meter windows as the smart meter processes more fine-grained data. On the other hand, \( \sigma_{\text{max}}, P_{\text{peak}} \) values are chosen to be larger for the aggregator as aggregators aggregate larger values for smaller window sizes. The accuracy is visualized and presented by the Mean Relative Error (MRE) for a time period \( T \) in percentage as defined below:

\[
MRE = \frac{100}{T} \sum_{i=0}^{T} \left( \frac{T}{i=0} \sum_{i=0}^{T} \frac{L(t_i) - \sum_{i=0}^{T} X(t_i)}{X(t_i)} \right)
\]

Fig. 6 shows the aggregation of time series for a single smart meter. The original data is shown in dotted line, the stripped line shows traditional Gaussian mechanism, and the solid line is used to represent our model. The perturbation is applied by consuming values of \( \varepsilon \) per window size \( w \) with fixed \( w \) used in traditional Gaussian mechanism and dynamic \( w \) size in our model. We can see from the results that we were able to control the error value and achieve a specific error value while consuming less privacy budget since we are using a dynamic window size. The accuracy in our model is evident by the larger range of the traditional Gaussian mechanism signal or the number of outliers. Note that the values are trimmed at the highest values by \( P_{\text{peak}} \) and at lowest value by minimum noise in our model. These values are trimmed and sent to the virtual battery and do not affect differential privacy due to the differential privacy’s post-processing property. Similarly, Fig. 7 shows the aggregation of time series for the aggregator. In both results, the values of differential privacy parameters per \( w \) are set as \( \varepsilon = 1 \) and \( \delta = 1/n^2 \) where \( n \) is the number of readings in the data set.

Table 4 shows a comparison between the MRE difference between using our model and the traditional Gaussian mechanism. Note that the error rate in our model is much lower than the traditional Gaussian mechanism while we kept the differential privacy guarantees. The error is low since we choose the perturbation window size optimally with the \( \sigma_{\text{max}} \) and trim the power between \( P_{\text{peak}} \) and noise applied on zero values. The window size also improved upon the loss of privacy budget as the added noise is applied on a larger window size when possible. We argue that this is better for data analysis and that the only drawback is using more coarse-grained data dynamically. It is important to note here that we could specify \( \sigma_{\text{max}} \) to achieve better accuracy; however, depending on the application and the required level of accepted error in the data, it is possible to increase the amount of \( \sigma_{\text{max}} \) as desired. An initial value of \( \sigma_{\text{max}} \) can be set on the smart meter running as a closed enclave with the ability only to increase this value given to the utility provider.

![Figure 6](Image 298x599 to 539x722)  
**FIGURE 6.** Smart meter aggregation for our model vs traditional Gaussian model

![Figure 7](Image 448x747 to 539x770)  
**FIGURE 7.** Aggregator aggregation for our model vs traditional Gaussian model

**TABLE 4.** MRE comparison between our model and traditional Gaussian mechanism

| Layer          | Our model | Gaussian Mechanism |
|----------------|-----------|--------------------|
| Smart Meter    | 0.0899    | 0.1861             |
| Aggregator     | 0.0312    | 0.0805             |

Fig. 8 shows smart meter perturbation error MRE over various values of differential privacy budget \( \varepsilon \). Our test outcomes support differential privacy’s theoretical principles; Lower values of privacy budget \( \varepsilon \) results in more noise and larger error values. Furthermore, we can see that the error value does not exceed 3.1% for \( \varepsilon = 3 \) while MRE is 79% in Gaussian mechanism with the same \( \varepsilon \) value.

Fig. 9 represents the effect of increasing the window size \( w \) on the MRE applied over a monthly period. We can see...
that increasing the window size reduces the value of MRE. However, there are some inconsistencies in the graph; for example, at $w = 210$, this is due to the randomness introduced when applying noise from a Gaussian distribution. We can see that after a certain amount of window size, $w = 190$, increasing the window size does not decrease the MRE value by much; this is because the MRE is limited by the value of $err_{max}$. $err_{max}$ value is used on the window size $w$ taken from the overall consumption; therefore, the overall MRE will reduce to a certain level. However, even after reaching this value of $w$, the increase of $w$ still benefits our model in reducing the consumed privacy budget $\varepsilon$. In the used dataset, we have readings of the power load for every minute. A fixed window size will be, for example, 10 minutes. The applied $\varepsilon$ on a fixed window for the monthly period will be the addition of all $\varepsilon$ applied on each window by using the total composition property of differential privacy. Therefore, it is trivial that using a larger window size will consume less privacy budget overall. As we discussed previously, larger $\varepsilon$ yields less privacy; therefore, it is beneficial to reduce the privacy budget loss. Our model used differential privacy with additive noise from the Gaussian distribution on a dynamic window size $w$. In our model, the window size $w$ is set to $w_{min}$ and increased until we reach $P_{peak}$, $err_{max}$ or $w_{max}$ whichever comes first. Fig. 10 represents the loss in the privacy budget $\varepsilon$ when applied to the overall monthly period. In [10] the value of $w = w_{max} - w_{min}$. It is important to note that the actual value of $w$ will vary between $w_{min}$ and $w_{max}$ as mentioned before. Nonetheless, increasing $w_{max}$, and subsequently $w$, will consume less privacy budget $\varepsilon$ for the overall data.

VI. RELATED WORK

Many approaches were introduced in the literature to preserve privacy and securely aggregate data. These can be categorized as cryptographic approaches [13], [37]–[40], approaches that rely on adding noise [2], [10], [12], [14]–[16], [41], or hybrid approaches [2], [11], [42], [43]. Homomorphic encryption methods are cryptographic approaches that enable data aggregators to compute arithmetic functions directly over the user ciphertext. Public-key cryptography is typically used to provide authenticity in systems. A careful consideration must be made before adopting any cryptographic approach because of the high computation costs these methods incur.

A. PRESERVING PRIVACY USING CRYPTOGRAPHY

Lossless data aggregation via task scheduling to aggregate encrypted perturbed data using a Decisional Diffie-Hellman scheme is introduced in [17]. While this approach is scalable and compute-efficient, both utility provider and aggregation points are trusted. A fog-computing with stream cipher cryptography based on layered architecture along with asymmetric-key scheme is given in [11]. Since this work uses...
homomorphic primitives and asymmetric-key cryptography, it requires high computational cost and third-party trust dependency. The approach in [44] adopts lightweight Elliptic Curve cryptography (ECC) to achieve authenticity between parties. Although it presents promising performance, a trusted party is needed for keys and secret management which introduces maintenance cost as well as single point of failure. The work in [43] describes a system based on distributed ledger technology and fog-computing technology to achieve privacy and security of the SG. They describe a group signatures and covert-channel authorization technique to validate users. Being blockchain-based solution, they implemented a smart contracts system to realize a security strategy that runs on these smart contracts across the blockchain network. Their system provides authenticity and anonymity via the blockchain inherent features.

B. PRESERVING PRIVACY USING A CHARGEABLE BATTERY

By using rechargeable power storage devices, one can mask and perturb actual load data on demand. Although the effect on billing may not be significant for more extended billing periods, changing consumer power load with a considerable noise can comprise data utility. Moreover, these approaches need extra monetary costs to obtain a suitable battery device and works exclusively with electric power as a utility disregarding other utilities such as water and natural gas. Consumer power load signature is masked in [10] via routing the power load utilizing a chargeable battery with an algorithm for mixing power routes. While this approach allowed for adjustable privacy moderation via an algorithm, relying on physical batteries endures extra costs and limits the maximum amount of possible data perturbation depending on the capacity of the physical battery. The work in [13] improved on the model introduced in [10] with 26% better information hiding achieved by introducing a better algorithm and applying perturbation policies. However, the approach still falls short in allowing some consumer data loss in data analytics. Differential privacy with limited capacity chargeable battery to minimize cost is used in [16]. Although this approach presented improvements over traditional differential privacy mechanisms, maintenance and setup costs were not taken into account.

C. PRESERVING PRIVACY BY ADDITIVE NOISE

Adding noise can effectively mask power load readings, but it is not billing friendly as it forces the billing values to be explicitly sent separately. Sending billing values may work for long-term billing, but it fails when billing happens on daily basis for different times of day and dynamic pricing. Moreover, adding too much noise can be counter productive as it will make the data inaccurate, which affects its utility. It is crucial to keep perturbation levels low as to not render the original data useless. A framework to guarantee the utility of data and simultaneously preserve privacy via methods based on derivations from Markov modeling is shown in [14]. This work improves rechargeable battery-based methods while maintaining data utility and analytics. However, such a framework needs to use batteries which causes it to share similar problems that other battery-based approaches face.

1) Differential Privacy

With its introduction in [23] and [24], differential privacy is a technique where collected data is obfuscated in a privacy-preserving manner while maintaining the actual original data properties and semantics for analytical purposes. Differential privacy employs an algorithm that inserts noise into a dataset in such a way that disallows an adversary from learning or obtaining specific information related to a particular user in the dataset. It uses $\varepsilon$ as a privacy parameter that specifies the level of privacy. Additive noise can be generated via multiple algorithms. Typically, the Gaussian mechanism and Laplace distribution are commonly adopted to generate noise for time-series data. Historically, service providers or fog aggregation nodes are trusted by the SG consumers to honestly apply differential privacy mechanisms to the original unperturbed consumer generated dataset prior to sharing any consumer data statistics. If those trusted aggregators and service providers are no longer trusted, a distributed differential privacy approach [46] can be adopted. Distributed differential privacy utilizes shares of random noise values obtained from many participants in the distributed system.

An investigation was conducted in [15] to explore the trade-off between data privacy and data-utility in a relatively large dataset. This investigation was based on computing the probability of having a successful attack on the perturbed data from [47]. The dataset was inspected after injecting it with both white and colored noises. Their $\varepsilon$-privacy model adopts a Gaussian noise perturbation mechanism to evaluate privacy levels. This $\varepsilon$-Privacy value is adjusted concerning the injected noise over the dataset. Their approach sufficiently addresses their considered adversarial model.

Eibl et al. [12] implemented differential privacy over actual smart meter data, USING differential privacy on large data sets of smart meters readings. Point-wise privacy is used with a privacy budget of $\varepsilon = 1$ spent for each aggregation period with a total budget calculated from the composition property of differential privacy. Additionally, the perturbed data is smoothed to increase utility while keeping it differentially private exploiting the post-processing property of differential privacy. For the usability of data statistics, it was found that a large amount of data from thousands of smart meters is required to achieve a useful utility after applying differential privacy. Such implementation does not consider the deterioration of privacy budget $\varepsilon$. Improving on the work in [12], an advanced algorithm for protecting peak power values for renewable energy resources is introduced by Hassan et al. [29]. Power load is made using a differentially-private real-time load monitoring (DPLM) algorithm with Laplacian noise. Point-wise privacy from [12] is used to apply $\varepsilon$-differential privacy for small periods. Furthermore,
the DPLM algorithm limits peak power values by trimming them from the current reading and adding excessive energy for the next iteration period. A promising error rate of 1.5% was achieved for specific peak values. However, similar to the previous model, the point-wise differential privacy does not account for the deterioration of the privacy budget due to the composition properties.

2) Differential Privacy with Fog Aggregation

In [11], a privacy-preserving fog aggregation method is introduced using differential privacy in a distributed deployment in the smart grid. Data aggregators in this distributed system receive metrics from SM devices and then forward their aggregates to the utility provider, where a distributed Gaussian differential privacy deployment is used. The Gaussian additive noise is generated in relation to (ε, δ)-differential privacy [24] at the smart meter nodes, and encryption used is based on one-time-pad and asymmetric-key schemes for noisy data and authentication, respectively. Although their approach preserves both privacy and utility of the data while being advantageous in terms of power and bandwidth costs, a third-party trusted authority is required to distribute keys.

The work in [28] presented a solution that aggregates data of IoT deployments. It utilizes an adaptive w-event DP by performing DP on changing w points in time over time-series in an edge-computing aggregation system. They employed a stream data aggregation that maintains privacy with an adaptive time window size (w) which is based on a quality of privacy metric that they proposed as well. Moreover, machine learning models were utilized to get better accuracy of data aggregation, cluster IoT devices and inject perturbation noise. Although the grouping does provide improved privacy, it does not benefit billing use-cases unless power consumption cost data is sent explicitly.

VII. CONCLUSION

This paper presented a model to guarantee consumer information privacy in a power grid based on a non-physical (virtual) battery. The proposed differential privacy model offers privacy and retains data utility. The proposed system is agnostic to the perturbation mechanism. Deducting the additive noise from the VB guarantees accurate consumer billing regardless of how aggressively the data was perturbed. Because our model is virtual, it is applicable in areas other than the electrical power grid, including natural gas and water utilities. The proposed system avoids reliance on a trusted third party for key distributions to achieve authentication and utilizes light cryptographic schemes for confidentiality. Data aggregation based on edge-computing architecture is utilized, where intermediate compute nodes aggregate SM generated readings for a less granular data aggregation. Our data obfuscation technique is based on employing differential privacy with the Gaussian mechanism over infinite time series data. We describe model setup parameters which can control privacy levels and keep the amount of error under control using a dynamic window algorithm. The novel algorithm uses a varying window size of the SM power consumption readings to maintain the privacy budget. Our system offers multiple enhancements over traditional methods that do not use our dynamic approach. More specifically, our system is adjusting the error with lower values compared to the traditional Gaussian mechanism, consuming less privacy budget, and enabling correct billing of customers without loss of accuracy. Ultimately, our findings are that by utilizing some extra parameters, we can control the level of error. Furthermore, we observed that by setting the size of SM readings time window to be high, we obtained a lower MRE value and a lower privacy budget consumption. However, having a static time window size is affecting the error rate, and therefore, we used a maximum limit value with our dynamic time window size algorithm to ensure accurate and private updates to the service provider.

Another potential future study can explore a different approach to solve the same problem by pushing all analytics and calculations to the edge IoT devices rather than collecting and aggregating data in a potentially privacy-violating centralized manner at the utility provider or aggregator premises. Considering that approach, methods of confidential computing [26] may be worth adopting and exploring. Finally, estimating privacy budget loss over a long period is very important since it is one of the most challenging tasks to achieve in a differentially-private model. A possible direction to solving this is the use of temporally discounted differential privacy for evolving data sets presented in [48].

REFERENCES

[1] Zhong Fan, Georgios Kalogridis, Costas Efthymiou, Mahesh Soodiyabandara, Mutsumu Serizawa, and Joe McGehee. The new frontier of communications research: smart grid and smart metering. In Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking, pages 115–118, 2010.
[2] Gergely Ács and Claude Castelluccia. I have a dream/(differentially private smart metering). In International Workshop on Information Hiding, pages 118–132. Springer, 2011.
[3] Kenneth Kimani, Vitalice Oduol, and Kibet Langat. Cyber security challenges for iot-based smart grid networks. International Journal of Critical Infrastructure Protection, 25:36–49, 2019.
[4] Javad Khazaee and M. Hadi Amini. Protection of large-scale smart grids against false data injection cyberattacks leading to blackouts. International Journal of Critical Infrastructure Protection, 35:100457, 2021.
[5] Aaron Hansen, Jason Staggs, and Sujeet Shenoi. Security analysis of an advanced metering infrastructure. International Journal of Critical Infrastructure Protection, 18:3–19, 2017.
[6] Cristina Alcaraz and Sherali Zeadally. Critical infrastructure protection: Requirements and challenges for the 21st century. International Journal of Critical Infrastructure Protection, 8:53–66, 2015.
[7] Brandon J. Murrill, Edward C. Liu, and II Thompson Richard M. Smart meter data: Privacy and cybersecurity, 2.2012.
[8] Elias Leake Quinn. Privacy and the new energy infrastructure. SSRN Electronic Journal, 2 2009.
[9] G Wood and M Newborough. Dynamic energy-consumption indicators for domestic appliances: environment, behaviour and design. Energy and Buildings, 35:821–841, 2003.
[10] Georgios Kalogridis, Costas Efthymiou, Stojan Z Denic, Tim A Lewis, and Rafael Cepeda. Privacy for smart meters: Towards undetectable appliance load signatures. In 2010 First IEEE International Conference on Smart Grid Communications, pages 232–237. IEEE, 2010.
[11] Lingjuan Lyu, Karthik Nandakumar, Ben Rubinstein, Jiong Jin, Justin Bedo, and Marinuthu Palaniswami. Pfpa: Privacy preserving fog-enabled
aggregation in smart grid. IEEE Transactions on Industrial Informatics, 14(8):3733–3744, 2018.

[12] Günther Eibl and Dominik Engel. Differential privacy for real smart metering data. Computer Science-Research and Development, 32(1-2):173–182, 2017.

[13] David Varadayan and Ashish Khisti. Smart meter privacy using a rechargeable battery: Minimizing the rate of information leakage. In 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 1932–1935. IEEE, 2011.

[14] Lalitha Sankar, S Raj Rajagopalan, Soheil Mohajer, and H Vincent Poor. Smart meter privacy: A theoretical framework. IEEE Transactions on Smart Grid, 4(2):837–846, 2012.

[15] Marco Savi, Cristina Rottondi, and Giacomo Verticale. Evaluation of the precision-recall tradeoff of data perturbation for smart metering. IEEE Transactions on Smart Grid, 6(5):2409–2416, 2015.

[16] Zijian Zhang, Wenqiang Cao, Zhan Qin, Liehuang Zhu, Zhengtao Yu, and Kui Ren. When privacy meets economics: ENabling differentially-private battery-supported meter reporting in smart grid. In 2017 IEEE/ACM 25th International Symposium on Quality of Service (IWQoS), pages 1–9. IEEE, 2017.

[17] Ulas Baran Baloglu and Yakup Demir. Lightweight privacy-preserving data aggregation scheme for smart grid metering infrastructure protection. International Journal of Critical Infrastructure Protection, 22:16–24, 2018.

[18] Fawaz Kserawi. Privacy-preserving data aggregation in smart power grid systems. IEEE Trans. Smart Grid, 6:2010–2016.

[19] Fawaz Kserawi and Qutaibah M. Malluhi. Privacy preservation of aggregated data using virtual battery in the smart grid. In 2020 IEEE 6th International Conference on Dependability in Sensor, Cloud and Big Data Systems and Applications (DependSys), pages 106–111, 2020.

[20] Rabindra K Barik, Sathish Kumar Gudey, Guji Girdhar Reddy, Meenakshi Pant, Harishchandra Dubey, Kunal Manokdiya, and Vinay Kumar. Fog-grid: Leveraging fog computing for enhanced smart grid network. In 2017 14th IEEE India Council International Conference (INDICON), pages 1–6. IEEE, 2017.

[21] Jia-Nan Liu, Jian Weng, Anjia Yang, Yizhao Chen, and Xiaodong Lin. Enabling efficient and privacy-preserving aggregation communication and function query for fog computing-based smart grid. IEEE Transactions on Smart Grid, 11(1):247–257, 2019.

[22] Jens-Matthias Bohli, Christoph Sorge, and Osman Ugus. A privacy model for smart metering. In 2010 IEEE International Conference on Communications Workshops, pages 1–5. IEEE, 2010.

[23] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In Theory of cryptography conference, pages 265–284. Springer, 2006.

[24] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. Foundations and Trends in Theoretical Computer Science, 9(3-4):211–407, 2014.

[25] Chris Clifton and Tamir Tassa. On syntactic anonymity and differential privacy. In 2013 IEEE 29th International Conference on Data Engineering Workshops (ICDEW), pages 88–93. IEEE, 2013.

[26] Victor Costan and Srinivas Devadas. Intel sgx explained. IACR Cryptol. ePrint Arch., 2016(66):1–118, 2016.

[27] Lingjiang Lyu, Jong Jin, Sutharshan Rajasegarar, Xuani He, and Marimuthu Palaniwami. Fog-empowered anomaly detection in IoT using hyperellipsoidal clustering. IEEE Internet of Things Journal, 4(5):1174–1187, 2017.

[28] Yan Huo, Chengtao Yong, and Yanfei Lu. Re-adp: real-time data aggregation with adaptive-event differential privacy for fog computing. Wireless Communications and Mobile Computing, 2018, 2018.

[29] Muneeb Ull Hassan, Masbahur Husain Rehmani, Ramamohanarao Kota-giri, Jiekiu Zhang, and Jinjun Chen. Differential privacy for renewable energy resources based smart metering. Journal of Parallel and Distributed Computing, 131:69–80, 2019.

[30] Georgios Kellaris, Stavros Papadopoulos, Xiaokui Xiao, and Dimitris Papadias. Differentially private event sequences over infinite streams. Proceedings of the VLDB Endowment, 7(12):1155–1166, 2014.

[31] Whitfield Diffie and Martin Hellman. New directions in cryptography. IEEE transactions on Information Theory, 22(6):644–654, 1976.

[32] NIST-FIPS Standard. Announcing the advanced encryption standard (aes). Federal Information Processing Standards Publication, 197(1):51–3-3, 2001.

[33] Hugo Krawczyk, Mihir Bellare, and Ran Canetti. Hmac: Keyed-hashing for message authentication, 1997.

[34] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Keying hash functions for message authentication. In Annual international cryptology conference, pages 1–15. Springer, 1996.

[35] Wikipedia contributors. Hmac — Wikipedia, the free encyclopedia, 2021. [Online; accessed 31-October-2021].

[36] Georges Hebrail and Alice Berard. Individual household electric power consumption data set, 2012.

[37] Fengjun Li, Bo Luo, and Peng Liu. Secure information aggregation for smart grids using homomorphic encryption. In 2010 first IEEE international conference on smart grid communications, pages 327–332. IEEE, 2010.

[38] Klaus Kursawe, George Danezis, and Markulf Kohlweiss. Privacy-friendly aggregation for the smart-grid. In International Symposium on Privacy Enhancing Technologies Symposium, pages 175–191. Springer, 2011.

[39] Rongxing Lu, Xiaohui Liang, Xu Li, Xiaodong Lin, and Xuejin Shen. Eppa: An efficient and privacy-preserving aggregation scheme for secure smart grid communications. IEEE Transactions on Parallel and Distributed Systems, 23(9):1621–1631, 2012.

[40] T-H Hubert Chan, Elaine Shi, and Dawn Song. Privacy-preserving stream aggregation with fault tolerance. In International Conference on Financial Cryptography and Data Security, pages 200–214. Springer, 2012.

[41] Neeraj Kumar Singh and Vasundhara Mahajan. End-user privacy protection scheme from cyber intrusion in smart grid advanced metering infrastructure. International Journal of Critical Infrastructure Protection, 3(1):1004, 2021.

[42] Vihbor Rastogi and Suman Nath. Differentially private aggregation of distributed time-series with transformation and encryption. In Proceedings of the 2010 ACM SIGMOD International Conference on Management of data, pages 735–746, 2010.

[43] Zekierya Erkin and Gene Tsudik. Private computation of spatial and temporal power consumption with smart meters. In International Conference on Applied Cryptography and Network Security, pages 561–577. Springer, 2012.

[44] Keke Gai, Yulu Wu, Liehuang Zhu, Lei Xu, and Yan Zhang. Permissioned blockchain and edge computing empowered privacy-preserving smart grid networks. IEEE Internet of Things Journal, 6(5):7992–8004, 2019.

[45] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pages 486–503. Springer, 2006.

[46] Sean Barker, Aditya Mishra, David Irwin, Emmanuel Cencetti, Prashant Shenoy, Jeannie Albrecht, et al. Smart*: An open data set and tools for enabling research in sustainable homes. SustKDD, August, 111(112):108, 2012.

[47] Hadi Farokhi. Temporally discounted differential privacy for evolving datasets on an infinite horizon. In 2020 ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPS), pages 1–8. IEEE, 2020.
SAEED AL-MARRI is a doctoral student in Computer Science at Qatar University (QU), working on Privacy Enhancing Technologies (PET) in the smart grid. He graduated from Hamad Bin Khalifa University (HBKU) in 2019 with a MSc. in Cybersecurity. He was awarded in 2020 by Qatar National Research Fund (QNRF) as part of the Graduate Sponsorship Research Award (GSRA) program to conduct doctoral research on PET in the smart grid. His interests span distributed and confidential computing domains, with special interest in Internet of Things (IoT) security and privacy.

QUTAIBAH MALLUHI is a Professor at the Department of Computer Science and Engineering at Qatar University (QU). He was the Head of the Department between 2006 and 2012 and the director of the KINDI Center for Computing Research at QU between 2012 and 2016. He served as a professor at Jackson State University. Dr. Malluhi was the co-founder and CTO of Data Reliability Inc. He was a consultant for several telecommunication companies where he built networks, designed distributed applications and developed telecommunication management software. Prof. Malluhi has received the QU research award, the JSU Technology Transfer award, the JSU faculty Excellence award, and several best paper awards. He has received MS and PhD degrees in computer science from the University of Louisiana, Lafayette, and BS and MS degrees in Computer Engineering from King Fahd University or Petroleum and Minerals, Saudi Arabia.

* * *

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/