IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 23, 2022, accepted March 22, 2022, date of publication April 13, 2022, date of current version April 28, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3167015

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

FAWAZ KSERAWI, SAEED AL-MARRI, AND QUTAIBAH MALLUHI, (Member, IEEE)
Department of Computer Science and Engineering, College of Engineering, Qatar University, Doha, Qatar

Corresponding author: Fawaz Kserawi (fawaz@qu.edu.qa)

This work was supported in part by the Qatar National Research Fund (a member of Qatar Foundation) under Grant NPRP12C-33905-SP-66 and Grant GSRA7-1-0517-20065.

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 the privacy of consumers 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-computing data aggregators perturbed data, respectively, compared to the commonly used Gaussian mechanism.

INDEX TERMS Advanced metering infrastructure, differential privacy, electrical grid, the 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

The associate editor coordinating the review of this manuscript and approving it for publication was Pedro R. M. Inácio.

particular household. The leaked data can be shared or sold without the consent of consumers, 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.

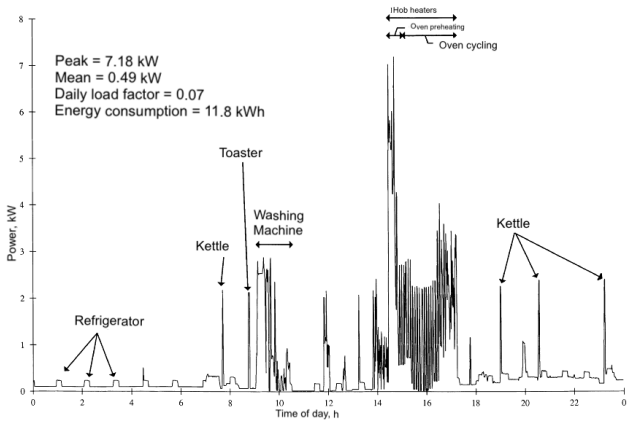

FIGURE 1. Household electricity demand profile [9].

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 the privacy of users. 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 [10] and [2], [11]–[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 the privacy of consumers and the safety of transferred data. We employ light-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 home network of the consumer 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 servers owned by service providers, 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.

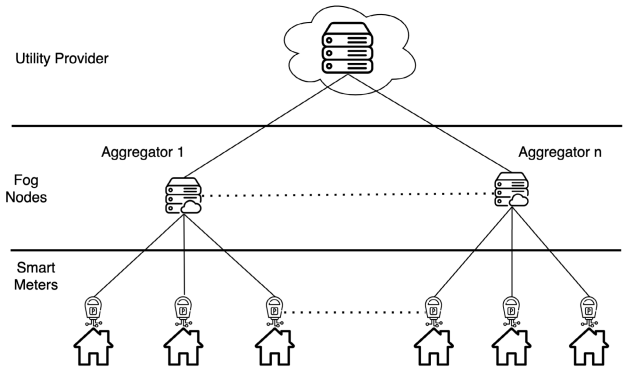

FIGURE 2. Hierarchical layering based on fog-computing in the smart grid.

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]. Fig. 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 contributions from a single user 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 the privacy of contributors in the dataset. 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 datasets $X, X' \in X_n$ that are different in one entry. We say that M is ε-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 dataset is small enough, the query result cannot infer the data of a single individual. Here, the difference between X and X' is the data belonging to one entity in the dataset. Therefore, we can get one dataset from another by either adding, removing, or changing the data of this entity.

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 ε 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 ε, where a smaller ε produces better privacy but less accuracy and vise versa. It is a challenge in smart metering to balance ε 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 ε-differentially private and $F : Y \rightarrow Z$ be an arbitrary randomized mapping. Then $F\ o\ M = F(M(X))$ is ε-differentially private.
2) Composition: Suppose $M = (M_1, \ldots, M_k)$ 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 ε should be to maintain privacy since differential privacy is usually added to static data by a trusted curator [25]. In time-series and growing datasets, it is unfeasible to apply differential privacy with a constant ε as the data is continuously growing. One crucial property to solve this is the composition of differential privacy. In a composition of T independent queries, the privacy parameters ε, δ must be accumulated.

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:

$$w_j = \sum_{t=i}^{i'} X(t_i) \tag{1}$$

For example, applying differential privacy with parameters ε_1, ε_2 and ε_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 (1) shows that the values of ε will increase over time. We found that many solutions in the literature ignore the deterioration of ε 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 or secure enclaves [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.

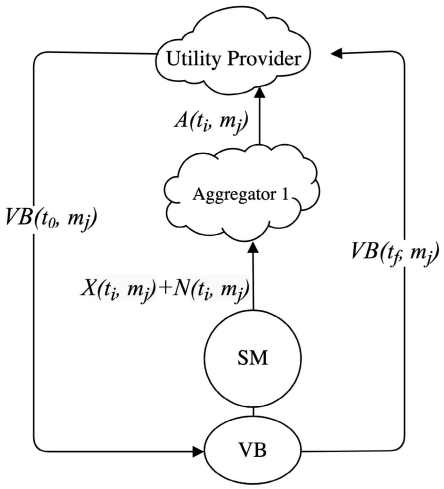

FIGURE 3. The architectural design based on the virtual battery concept from an individual SM device.

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. Fig. 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:

TABLE 1. VB system symbols.

Symbol	Meaning
m_j	Long time period for example a month at iteration j
t_i	Time period of smart meter reading at iteration i
(t_i, m_j)	Time t_i of period m_j
t_0	Initial time in a period m_j
t_f	Last time value in period m_j
f	Final iteration of time t in a period m_j
$X_n(t)$	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
$VB_n(t)$	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
$N_a(t)$	Noise added at time t to power load of an aggregator a
$A_n(t)$	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

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 SM update to utility provider of the first period m_0: $VB(t_0, m_0) = 0$.
2) After a short period t_1, the SM perturbs its original data by adding some noise N to it and then forwards this noisy data to the data aggregation node. Then, the SM device subtracts the noisy data from the VB value to preserve the accuracy of billing as shown in equation (2):

$$VB(t_1, m_j) = VB(t_0, m_j) - N(t_1, m_j) \tag{2}$$

3) When reaching the end of time period m_j, the SM sends $VB(t_i, m_j)$ to the utility provider:

$$VB(t_f, m_j) = \sum_{i=0}^{f} VB(t_i, m_j) - \sum_{i=0}^{f} N(t_i, m_j) \tag{3}$$

It is important to note that the utility provider is unaware of the fine grained values $VB(t_i, m_j)$ and $N(t_i, m_j)$ and only receives the end value $VB(t_f, m_j)$ from the SM.
4) The aggregator receives periodic perturbed readings $(X(t_i, m_j) + N(t_i, m_j))$ from the SM and calculates $A(t_f, m_j)$ using equation (4). Later, the value $A(t_f, m_j)$ is sent to the utility provider:

$$A(t_f, m_j) = \sum_{i=0}^{f} (X(t_i, m_j) + N(t_i, m_j)) \tag{4}$$

It is worth mentioning that the aggregator calculates $A(t_f, m_j)$ without knowing the values of $X(t_i, m_j)$ or $N(t_i, m_j)$ individually, however it only receives their sum $(X(t_i, m_j) + N(t_i, m_j))$. By subtracting m_j, which is the VB power consumption of the period m_j, the utility provider can calculate $VB(t_0, m_j) - VB(t_f, m_j)$ to get the total consumption TC:

$$TC(t_f, m_j) = A(t_f, m_j) - (VB(t_0, m_j) - VB(t_f, m_j)) \tag{5}$$

5) After a reasonably long period, a month for example, the final VB value is exchanged. The exchange of VB value guarantees billing accuracy since the added noise was subtracted from the VB value. The reason behind monthly updates of VB value exchanges is to prevent the utility provider from inferring further information, such as individually added noise, from the fine-grained VB values. Furthermore, since subtracting individual noise from the VB value happens only at the SM device, the utility provider is unaware of distinct noise values added to the VB. Section IV-D discusses the aspects of security in terms of communicating the value of VB in more detail.

For dynamic billing, it is possible to use multiple virtual batteries; for example, when billing is different between day-time and night-time, we can use VB_{day}, VB_{night}. Noise added at daytime is added to VB_{day}, and night-time noise is added to VB_{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 bill for consumers, 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 $VB(t_i, m_j)$ will be the previous virtual battery value $VB(t_{i-1}, m_j)$ minus the absolute value of P_{peak} subtracted from $X(t_{i-1}, m_j)$ as shown in equation (6).

$$VB(t_i, m_j) = VB(t_{i-1}, m_j) - |X(t_{i-1}, m_j) - P_{peak}| \tag{6}$$

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 ε 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].

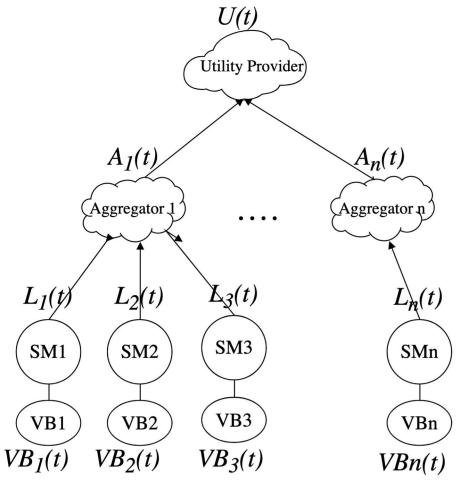

FIGURE 4. Distributed fog aggregation with virtual batteries.

1) AGGREGATION OF SMART METER DATA

Fog nodes reduce the cost of computation and communication as they offload work from servers of service providers. Coarse-grained statistics of each SM are sent to the aggregators. Furthermore, neighborhood power loads from multiple SM devices can be aggregated on fog nodes, and then results are forwarded to the service provider. Apart from the setup phase, our system assumes uni-directional communications only from SM devices to aggregators and from aggregators to the utility provider.

Our proposed architecture has some assumptions in place where we consider aggregators and the utility provider as adversaries and therefore are not allowed to read private information about SG consumers. We assume SM devices are resistant to physical tampering. SMs locally store their encryption keys safely and apply differential privacy on power consumption readings in a defined period we call a window before transmitting the readings to aggregators. This window can be a value between an already established parameters w_{min}, w_{max}, which will be discussed in a later section IV-C3. Our VB model uses fog nodes as aggregation points we call aggregators. A set of SMs directly communicate with multiple data aggregation points that gather and accumulate data to forward them to a specific utility provider. Each SM device adds perturbation noise to its original data with noise generated from a VB. The perturbed data is encrypted and forwarded to the nearest aggregator. 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 $VB(t_i, m_j)$ value would include all added noise from $N(t_0, m_j)$ to $N(t_f, m_j)$. Therefore, the perturbed load can be computed with: $L(t_f, m_j) = \sum_{i=0}^{f}(X(t_i, m_j) + N(t_i, m_j))$. $VB(t_f, m_j)$ is sent from the SM to 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_n 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 shows the aggregation architecture and Table 1 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 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

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

3) During a window of time w, an aggregator A_n receives a number of perturbed and encrypted values of consumption $L_{sm_1}(t_i, m_j)$ from (7) and calculates $\sum_{i \in w} L_{sm_1}(t_i, m_j)$ during w from SM_1. Each value of $L_{sm_1}(t_i, m_j)$ 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, m_j) and finishing at the end of w with $(t_{i'}, m_j)$ we calculate and encrypt the consumption of sm_1 by:

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

For example the first time window of 30 minutes in m_0 has values of $(t_i, m_0) = 0$ 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_1, 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_1, m_j)

$$U(t_1, m_j) = \sum_{i=1}^{n} Dec[A_i(t_1, m_j)]. \tag{10}$$

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):

$$\begin{aligned} TC_n(t_f, m_j) = {} & Dec[A_n(t_f, m_j)] \\ & - (Dec[VB_n(t_0, m_j)] \\ & - Dec[VB_n(t_f, m_j)]) \end{aligned} \tag{11}$$

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 w_{max} of the aggregators. 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. [11] 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_{sm_n} from equation 7 and receives $(\sum_{i \in w_{reg}} L_{sm_x}(t_i))$ during w_{reg} from each SM in the region; where x is the number of connected regional SMs. Each value of L_{sm_x} 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:

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

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:

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

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 σ standard deviation. Theorem A.1. IV-C1.a from [24] gives us the following definition:

Let f: $\mathbb{N}^{|x|} \rightarrow \mathbb{R}^d$ be a function of d-dimensions, and its ℓ_2 sensitivity is defined as $\triangle_2 f = max_{adjacent_{x,y}} ||f(x) - f(y)||_2$. The Gaussian mechanism that has the δ 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 $c^2 > 2ln(1.25/\delta)$ the Gaussian mechanism with parameter $\sigma \geq c\triangle_2 f/\varepsilon$ is (ε, δ)-differentially private.

Following the parameters of distributed differential privacy, we have:

$$\widehat{x}(t) = \sum_{i=1}^{n} (\widehat{x_i}(t)) = \sum_{i=1}^{n} (x_i(t) + r_i(t)) \tag{14}$$

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 σ satisfies the Theorem A.1. IV-C1.a.

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-C1.a, 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-C1.a in section IV-C1 we have the sensitivity S of our algorithm M is:

$$S(M) = \triangle_2 f = max_{adjacent_{x,y}} ||f(x) - f(y)||_2 \tag{15}$$

TABLE 2. Symbols for the differential privacy model.

Symbol	Meaning
w_{min}	Minimum window size
w_{max}	Maximum window size
P_{peak}	Peak power allowed in w_{max} with $err < err_{max}$
err_{max}	Maximum allowed error

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 dataset. 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_{min} and w_{max} which guarantees a differential privacy and limits the maximum power value at P_{peak}. Furthermore, the algorithm ensures that the error value of the perturbation does not exceed err_{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_{min}, w_{max}. Within the smart 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_{peak} applied on all readings inside w_{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_{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:

$$S(M) = \triangle_2 f = P_{peak}/w_{max} \tag{16}$$

b: STANDARD DEVIATION

In a one dimensional dataset, by applying Theorem A.1. IV-C1.a we can calculate the standard deviation σ of the Gaussian mechanism where $\varepsilon \in (0, 1)$ and $\delta \in (0, 1)$:

$$\sigma^2 = 2ln(1.25/\delta).(\triangle f)^2/\varepsilon^2 = 2ln(1.25/\delta).S(M)^2/\varepsilon^2 \tag{17}$$

c: PRIVACY BUDGET

From the differential privacy composition theorem, we know that for multiple algorithms applies on the same data the privacy budget parameters ε and δ for each algorithm are added up. For example, applying (ε, δ)-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 $\delta_1 = \delta/k$. 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_{max} or w_{max}.
2) w_{max} is the maximum number of power reading 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_{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_{max} value is reached. The P_{peak} value is used to achieve better performance in the algorithm since if adding the total consumption over w exceeds P_{peak} value, we know that w will break err_{max} without the need to generate noise. Here, it may also be possible for a single reading to exceed P_{peak} we threshold all such readings and add the trimmed values to the virtual battery value. Trimming single outliers that exceed P_{peak} guarantees the value of sensitivity.
4) err_{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:

$$MAE = \frac{\sum_{i=1}^{n} |X'_i - X_i|}{n}$$

w event ε-differential privacy was introduced in [30] and applied in [28] for ε-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. We assume a security policy is in place to set these parameters initially in the SMs enclosed enclave with permissions given to the service provider to only increase these values to ensure differential privacy. This will depend on the number of updates per unit of time and the required granularity of SM updates. For example, the initial w_{max} and err_{max} should not be too low, as this does not conform to the differential privacy rules. On the other hand, the initial value of err_{max} should not be too large as this will render the collected data unuseful for data analysis and a large w_{max} could delay data updates coming from the SM. After the initial setup, the values for w_{min}, w_{max} and err_{max} can be increased by the service provider, and this change does not affect differential privacy guarantees as larger error and window size means more perturbation and less consumed privacy budget.

Dynamic Window Differential Privacy Algorithm w-event differential privacy presents a solution for the infinite data stream; here, privacy is applied at sliding windows of size w in the smart grid. However, the fixed window size means unnecessarily sacrificing the privacy budget because of dynamic changes in the time-series power readings of the SMs. For example, the power load may change dramatically when using heavy appliances such as heaters; on the other hand, night power consumption is usually low. Consequently, we use a dynamically changing window size accompanied by specified bounds with differential privacy guarantees.

In w-event differential privacy, the perturbation of data in a single w leads to a consumption of a fixed privacy budget taken from the overall privacy budget. Increasing the window size will lead to a lower privacy budget loss but leads to a high perturbation and more significant error value. We present a dynamic window differential privacy algorithm that uses a dynamic window size w and limits the window size by P_{peak}, err_{max} and w_{max}. Our algorithm ensures that the error resulting from the perturbation err does not exceed a certain amount as the algorithm selects w in a way that bounds the resulting perturbed data with a maximum values of P_{peak} and err_{max}. At the same time, we bound w by w_{max} for consistent reporting to the aggregator and utility provider. As mentioned previously, at the initial run of the SM, our perturbation algorithm collects and increases the number of the included power readings w beginning at w_{min}, and perturbs the included data until $err = err_{max}$ or $w = w_{max}$. After that, the perturbed result is sent, and the algorithm is applied again on the next set of power readings, starting at w_{min} of the following period. Algorithm 1 describes the implementation of our algorithm.

Both SMs and aggregators use the same algorithm; however, the parameters' values will be different for aggregators. w_{min} and w_{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 ε 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

Algorithm 1 Dynamic Window Differential Privacy Algorithm

Input: $w_{min}, w_{max}, err_{max}$
current window $w = w_{min}$
Calculate P_{peak}
while $err \leq err_{max}$ **do**
  while $w \leq w_{max}$ **do**
    while $P(t) \leq P_{peak}$ **do**
      Calculate $P(t_i) = \sum_{i=0}^{i} X(t_i)$;
    end
    w++;
  end
  Calculate error: $err = P(t_i)/L(t_i)$;
  Apply perturbation on w: $L(t_i) = P(t_i) + \sum_{i=0}^{i} N(t_i)$;
end
Calculate new privacy budget parameters: $\varepsilon = \varepsilon - \varepsilon_i$, $\delta = \delta - \delta_i$;
Output perturbed value for $w_{i-1} : L(t_{i-1})$;

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

1) DIFFIE-HELLMAN KEY EXCHANGE

Adopting light cryptographic constructions is essential in IoT devices, such as SMs running in the smart grid. Our cryptographic approach requires two symmetric keys stored on the SM to secure connections to the utility provider and aggregator for sending the values of VB and load consumption, respectively. Symmetric keys are exchanged between SMs, aggregators, and the utility provider. 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 in our system. Typically, authentication is accomplished through an asymmetric-key cryptographic scheme in the related literature. However, we use our own simple and light Diffie–Hellman key exchange with challenge-response authentication method between any two parties, as depicted in Fig. 5. We adopt Diffie–Hellman key exchange with challenge-response authentication instead of public-key cryptography because of the inherent heavy computations of public-key cryptography and the dependency on trusted third party authority for key distribution.

The secret is obtained by concatenating the Diffie-Hellman derived key with a fixed ID provided by the utility provider. The fixed ID is set in the secure enclave of the SM devices before their distribution. The utility provider shares a secret string with data aggregators instead, which is used similarly to the fixed ID of SM devices. This key exchange is performed only to authenticate the secret key to be used for encryption later on and never used to authenticate data. The two-way challenge-response authentication 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. This approach keeps our protocol light-weight, compute-efficient, and avoids third-party trust compared to other methods, such as public-key cryptography. As SMs in our model are running as a closed enclave, we assume that the stored keys are secured. However, if required by a security policy, or any other reason, we can repeat the key exchange process to generate new keys as desired.

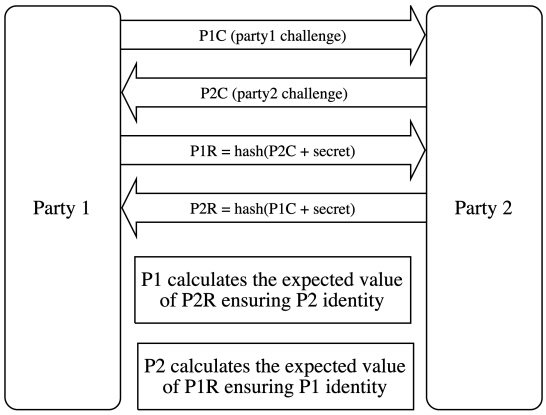

FIGURE 5. Initial two-way challenge-response authentication.

2) KEYED-HASH MESSAGE AUTHENTICATION CODE (HMAC)

HMAC is a variation of the Message Authentication Code (MAC) that uses secret keys, hash functions and provides data integrity and authenticity. HMAC is an improved version of MAC since MAC suffers from length-extension attack where an attacker can append data to the message without knowing the key. The implementation method and definitions are presented in [33] and [34]. A shared secret is used in HMAC implementation that does not require an involvement from a third party in key distribution. 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; in the first round, a hash is produced from the inner key with the already encrypted message.
3) In the second round, 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]:

$$HMAC(K, m) = H((K' \oplus opad) \parallel H((K' \oplus ipad) \parallel m))$$

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 zeros. Table 3 contains an explanation of the used symbols.

TABLE 3. Symbols for the HMAC model.

Symbol	Meaning
H	Cryptographic hash function
K	Secret key
K'	Block-sized key
$\parallel$	Concatenation
$\oplus$	XOR
$opad$	Block-sized outer padding
$ipad$	Block-sized inner padding

V. EXPERIMENTS AND RESULTS DISCUSSION

This section evaluates the performance accuracy of our model and compares it with the traditional Gaussian mechanism. We apply our model to two actual smart home datasets. Our dynamic window model is applied to the individual household electric power consumption dataset [36] which we call dataset A. Furthermore, we also apply our model on the UMass Smart* Dataset [37] which we refer to as dataset B. Dataset A contains electric power consumption measurements in a household with a one-minute sampling rate for almost four years starting at 2006. Dataset B contains an actual home power consumption for a year measured every minute during 2016. We apply our perturbation with the dynamic window differential privacy algorithm on SMs and then on aggregator nodes. For every window w, we aggregate the values from SMs and apply noise, losing an amount of privacy budget ε_w. Next, the aggregator receives the perturbed data and applies another perturbation; the noise added to achieve data perturbation is also added to the virtual battery of each smart meter. Error is calculated over the most recent time-series data values for each time window w. The values for w_{min}, w_{max}, err_{max} and P_{peak} are initially set for the smart meter and aggregator with different values. Naturally, the values of parameters w_{min}, w_{max} are selected to be larger for the smart meter than those used for the aggregator. Choosing a larger window size for SMs hides their data from the aggregator and consumes less privacy budget for the SM window because the SM processes more fine-grained data. On the other hand, err_{max}, P_{peak} values are chosen to be larger for the aggregator as aggregators process 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:

$$MRE = \frac{100}{T} \sum_{i=0}^{T} \left(\frac{\sum_{i=0}^{T} L(t_i) - \sum_{i=0}^{T} X(t_i)}{\sum_{i=0}^{T} X(t_i)} \right) \tag{18}$$

Fig. 6 and 7 show the aggregation of time-series for a single smart meter for dataset A and B, respectively. 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 ε per window size w with fixed w used in traditional Gaussian mechanism and dynamic w 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 w. The higher accuracy in our model is evident in the presented larger range of the traditional Gaussian mechanism signal and the number of outliers. We can notice in Fig. 6 and 7 that the values are trimmed at the highest values by P_{peak} and at the lowest values by minimum noise in our model. These values are trimmed and sent to the virtual battery and do not affect differential privacy because of the post-processing property of differential privacy. Similarly, Fig. 8 and 9 shows the aggregation of time-series for an aggregator for both datasets A and B, respectively. In our experiment for both datasets A and B, 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 dataset.

Table 4 shows a comparison between the MRE difference between our model and the traditional Gaussian mechanism for both datasets. Table 4 shows that the error rate in our model is much lower than the traditional Gaussian mechanism while maintaining the differential privacy guarantees. The error is low since the dynamic algorithm selected the perturbation window size w optimally as bounded by err_{max}, and we trimmed the values of power load above P_{peak} and lower than zero by adding the trimmed power load to the VB. The dynamic window size consumes less privacy budget as the added noise is applied to 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. It is important to note here that we could lower err_{max} to achieve better accuracy. Depending on the application and the required level of accepted error in the data, it is possible to increase the amount of err_{max} as desired to achieve better privacy. An initial value of err_{max}, which must be set in a way that guarantees differential privacy, is set on the SM running as a secure enclave. A permission is given to the utility provider to only increase err_{max} value, which will not break differential privacy but reduces the accuracy of the data.

Fig. 10, 11, 12, and 13 shows smart meter perturbation error MRE over various values of differential privacy budget ε for datasets A and B. Our experiments outcomes support the theoretical principles of differential privacy; lower values

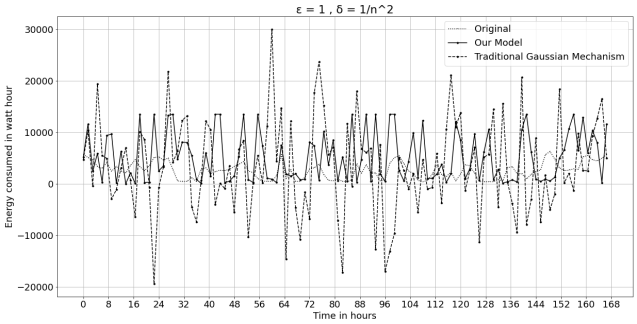

FIGURE 6. Smart meter aggregation for our model vs. traditional Gaussian model (dataset A).

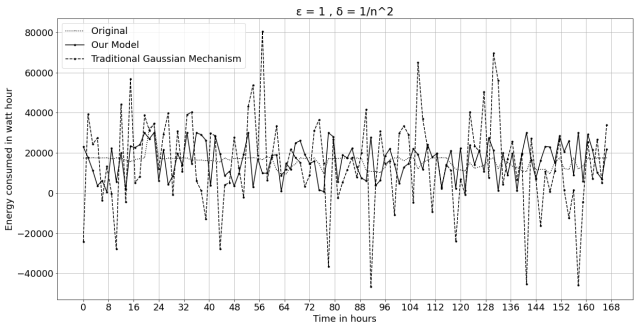

FIGURE 7. Smart meter aggregation for our model vs. traditional Gaussian model (dataset B).

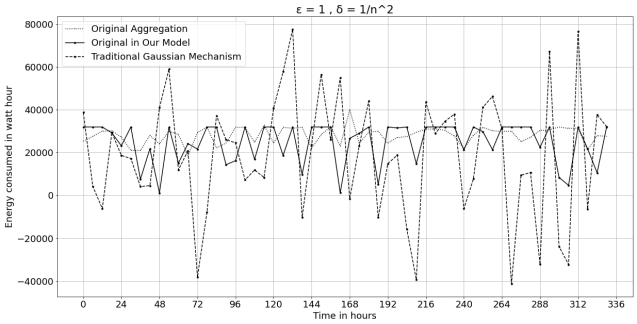

FIGURE 8. Aggregator aggregation for our model vs. traditional Gaussian model (dataset A).

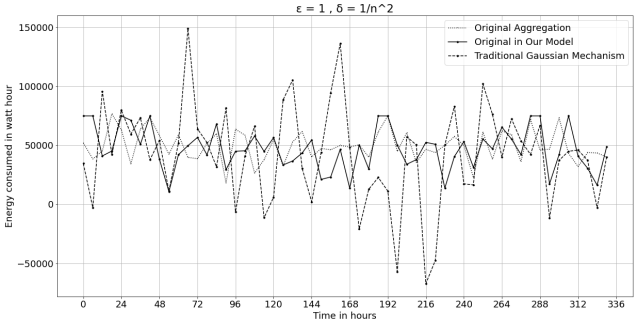

FIGURE 9. Aggregator aggregation for our model vs traditional Gaussian model (dataset B).

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

Layer	Dataset A		Dataset B	
	Our model	Gausssian	Our model	Gaussian
Smart meter	0.0899	0.1861	0.0023	0.0058
Aggregators	0.0312	0.0805	0.0063	0.0117

of privacy budget ε results in more noise and larger error values. Fig. 10 suggests that the error value does not exceed 3.5% for $\varepsilon = 3$ while MRE is 79% in Gaussian mechanism with the same ε value shown in 11. Furthermore, the MRE value reaches up to 0.075% for $\varepsilon = 3$ in Fig. 12 and for traditional Gaussian mechanism MRE is 0.8% for the same $\varepsilon = 3$ as shown in Fig. 13. Fig. 14 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 ε. In the used datasets, we have readings of the power load for every minute. A fixed window size will be, for example, 10 minutes. The applied ε on a fixed window for the monthly period will be the addition of all ε 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 discussed previously, larger ε 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 is reached first. Fig. 15 represents the loss in the privacy budget ε when applied to a monthly period. In Fig. 15, 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 ε overall.

VI. RELATED WORK

Many approaches were introduced in the literature to preserve privacy and securely aggregate smart grid data. These approaches can be categorized as cryptographic approaches [13], [38]–[41], approaches that rely on adding noise [2], [10], [12], [14]–[16], [42], or hybrid approaches [2], [11], [43], [44]. Homomorphic encryption methods are cryptographic approaches that enable data aggregators to compute

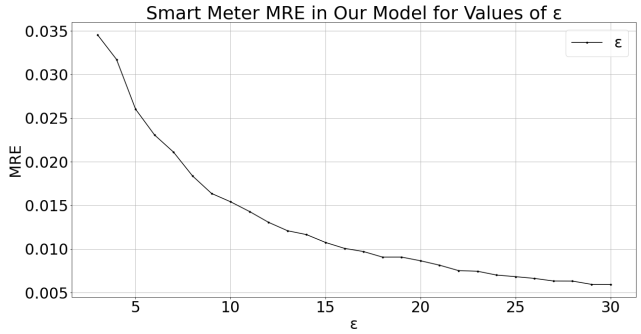

FIGURE 10. Variable ε values effect on MRE for our model (dataset A).

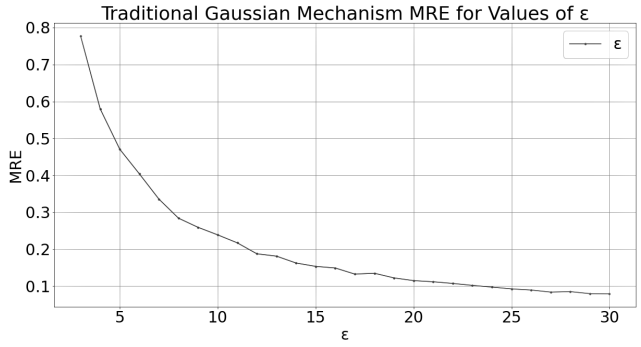

FIGURE 11. Variable ε values effect on MRE for traditional Gaussian mechanism (dataset A).

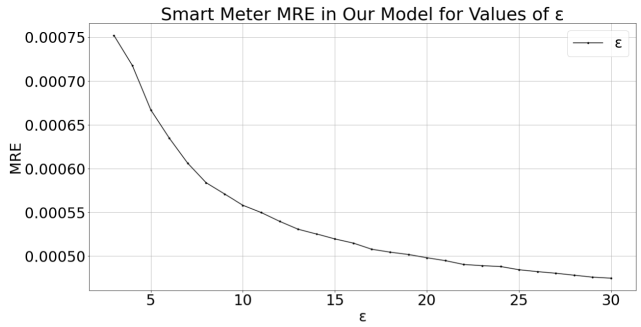

FIGURE 12. Variable ε values effect on MRE for our model (dataset B).

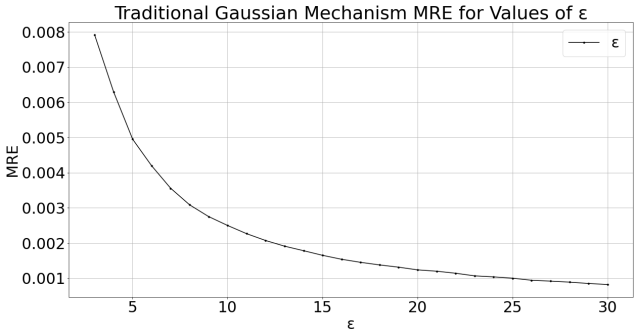

FIGURE 13. Variable ε values effect on MRE for traditional Gaussian mechanism (dataset B).

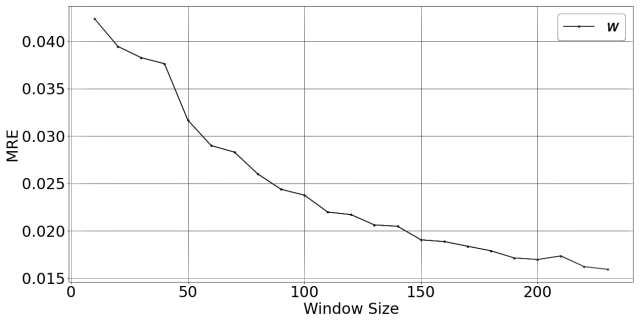

FIGURE 14. Variable window size effect on MRE in our model.

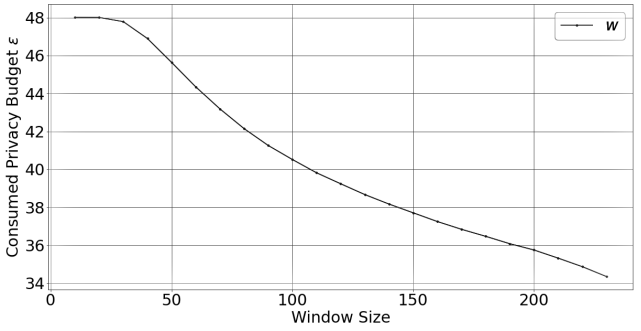

FIGURE 15. Variable window size effect on consumed privacy budget ε in our model.

arithmetic functions directly over the ciphertext. Public-key cryptography is typically used to for authentication in these systems. A careful consideration must be made before adopting any cryptographic approach because of the high computational 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 the utility provider and aggregators are trusted. A fog-computing with stream cipher cryptography based on layered architecture along with asymmetric-key scheme is given in [11]. However, as their work uses homomorphic primitives and asymmetric-key cryptography, it requires high computational power and third-party trust dependency. The approach in [45] 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 and a single point of failure. The work in [46] describes a system based on distributed ledger 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 a 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 facilitated by the blockchain.

B. PRESERVING PRIVACY USING A CHARGEABLE BATTERY

By using rechargeable power storage devices, the actual load data can be masked and perturbed 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 compromise data utility. Moreover, these approaches need extra monetary costs for deploying battery devices and work 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 presenting an optimized algorithm and applying perturbation policies. However, the approach allows for some loss in consumer data analytics. Differential privacy with limited-capacity batteries for cost minimization is used in [16]. Although their results showed improvements over traditional differential privacy mechanisms, maintenance and setup costs were the main limitation.

C. PRESERVING PRIVACY BY ADDITIVE NOISE

Adding noise can effectively mask power load readings, however, it is not billing friendly as it forces the billing data to be sent separately. Sending billing values may work for long-term billing, but it fails when billing is needed for different times of the day, as in dynamic pricing. Moreover, adding too much noise can be counterproductive, as it will make the data inaccurate, affecting its utility. It is crucial to keep perturbation levels low 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 uses noise from batteries or constantly powered devices, which causes it to share similar problems that other battery-based approaches encounter and possibly leak constantly powered devices data.

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 algorithms that perturb datasets in a way that prevents an adversary from obtaining specific information related to a particular user in the dataset. It uses ε 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 adopted to generate noise for time-series data. A common application of differential privacy is where service providers are trusted by the SG consumers to apply differential privacy mechanisms to the original data prior to sharing it for data analysis. If those aggregators and service providers are not trusted, a distributed differential privacy approach [47] 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 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 [48]. The dataset was inspected after injecting it with both white and colored noises. Their ε-privacy model adopts a Gaussian noise perturbation mechanism and evaluates privacy levels. This ε-privacy value is adjusted concerning the injected noise over the dataset. Their approach sufficiently addresses their considered adversarial model. Authors in [12] implemented differential privacy over actual data, using differential privacy on large datasets of SM 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 SMs is required to achieve a useful utility after applying differential privacy. Such implementation does not consider the deterioration of privacy budget ε. Improving on the work in [12], an advanced algorithm for protecting peak power values for renewable energy sources is introduced in [29]. Power load is perturbed using a differentially-private real-time load monitoring (DPLM) algorithm with Laplacian noise. Point-wise privacy from [12] is used to apply ε-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 noise is generated in relation to (ε, δ)-differential privacy [24] at the SMs, and encryption is used based on one-time-pad and asymmetric-key schemes for data perturbation and authentication, respectively. Although their approach preserves both

privacy and utility of the data while being efficient in terms of power and bandwidth costs, a trusted third-party 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 dynamic w over time-series data 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. Moreover, machine learning models were utilized to get better accuracy of data aggregation, cluster IoT devices and inject perturbation noise. Although the grouping provides improved privacy, it does not benefit billing use-cases unless billing data is sent independently.

VII. CONCLUSION

This paper presented a model that guarantees consumer information privacy in smart power grids based on a non-physical (virtual) battery. The proposed differential privacy model offers privacy and retains data utility. Deducting the additive noise from a VB guarantees accurate consumer billing, regardless of how aggressively the data was perturbed. Because our model employs a VB, it is applicable in areas other than the electrical power grid, such as 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 setup parameters which can control privacy levels and keep the amount of error under control using a dynamic window algorithm. Our presented algorithm uses a dynamic window size of the SM power consumption readings to maintain the privacy budget. Our system offers enhancements over traditional methods by allowing an adjustable error, offering lower values compared to the traditional Gaussian mechanism, consuming less privacy budget, and enabling accurate consumer billing. 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 window of time 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 window differential privacy algorithm to ensure accurate and private updates to the service provider. Several components of our proposed system can be integrated with other perturbation mechanisms.

For future work, as smart meters are IoT devices that are limited in power usage and computing ability, a first area of improvement in this proposal is to investigate more lightweight cryptographic protocols that consume less energy and computing power. Secondly, in our proposal, we presented a dynamic window differential privacy algorithm with a window w that dynamically change between w_{min} and w_{max} as inputs for our algorithm. While the service provider can increase these values for less frequent data updates and more privacy, a possible future study is to research an algorithm that sets these parameters dynamically to further optimize our algorithm. 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 for solving this is the use of temporally discounted differential privacy for evolving datasets presented in [49].

ACKNOWLEDGMENT

The findings achieved herein are solely the responsibility of the authors. The Open Access funding provided by the Qatar National Library.

REFERENCES

[1] Z. Fan, G. Kalogridis, C. Efthymiou, M. Sooriyabandara, M. Serizawa, and J. McGeehan, "The new frontier of communications research: Smart grid and smart metering," in *Proc. 1st Int. Conf. Energy-Efficient Comput. Netw.*, 2010, pp. 115–118.

[2] G. Ács and C. Castelluccia, "I have a DREAM! (Differentially private smart metering)," in *13th Int. Conf. Inf. Hiding (IH)*. Prague, Czech Republic: Springer, May 2011, pp. 118–132.

[3] K. Kimani, V. Oduol, and K. Langat, "Cyber security challenges for IoT-based smart grid networks," *Int. J. Crit. Infrastruct. Protection*, vol. 25, pp. 36–49, Jun. 2019.

[4] J. Khazaei and M. H. Amini, "Protection of large-scale smart grids against false data injection cyberattacks leading to blackouts," *Int. J. Crit. Infrastruct. Protection*, vol. 35, Dec. 2021, Art. no. 100457.

[5] A. Hansen, J. Staggs, and S. Shenoi, "Security analysis of an advanced metering infrastructure," *Int. J. Crit. Infrastruct. Protection*, vol. 18, pp. 3–19, Sep. 2017.

[6] C. Alcaraz and S. Zeadally, "Critical infrastructure protection: Requirements and challenges for the 21st century," *Int. J. Crit. Infrastruct. Protection*, vol. 8, pp. 53–66, Jan. 2015.

[7] B. J. Murrill, E. C. Liu, and R. M. Thompson, "Smart meter data: Privacy and cybersecurity," Congressional Res. Service, Washington, DC, USA, Tech. Rep. R42338, Feb. 2012.

[8] E. L. Quinn, "Privacy and the new energy infrastructure," *SSRN Electron. J.*, p. 43, Feb. 2009.

[9] G. Wood and M. Newborough, "Dynamic energy-consumption indicators for domestic appliances: Environment, behaviour and design," *Energy Buildings*, vol. 35, no. 8, pp. 821–841, Sep. 2003.

[10] G. Kalogridis, C. Efthymiou, S. Z. Denic, T. A. Lewis, and R. Cepeda, "Privacy for smart meters: Towards undetectable appliance load signatures," in *Proc. 1st IEEE Int. Conf. Smart Grid Commun.*, Oct. 2010, pp. 232–237.

[11] L. Lyu, K. Nandakumar, B. Rubinstein, J. Jin, J. Bedo, and M. Palaniswami, "PPFA: Privacy preserving fog-enabled aggregation in smart grid," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3733–3744, Aug. 2018.

[12] G. Eibl and D. Engel, "Differential privacy for real smart metering data," *Comput. Sci., Res. Develop.*, vol. 32, nos. 1–2, pp. 173–182, Mar. 2017.

[13] D. Varodayan and A. Khisti, "Smart meter privacy using a rechargeable battery: Minimizing the rate of information leakage," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2011, pp. 1932–1935.

[14] L. Sankar, S. R. Rajagopalan, S. Mohajer, and H. V. Poor, "Smart meter privacy: A theoretical framework," *IEEE Trans. Smart Grid*, vol. 4, no. 2, pp. 837–846, Jun. 2013.

[15] M. Savi, C. Rottondi, and G. Verticale, "Evaluation of the precision-privacy tradeoff of data perturbation for smart metering," *IEEE Trans. Smart Grid*, vol. 6, no. 5, pp. 2409–2416, Sep. 2015.

[16] Z. Zhang, W. Cao, Z. Qin, L. Zhu, Z. Yu, and K. Ren, "When privacy meets economics: Enabling differentially-private battery-supported meter reporting in smart grid," in *Proc. IEEE/ACM 25th Int. Symp. Quality Service (IWQoS)*, Jun. 2017, pp. 1–9.

[17] U. B. Baloglu and Y. Demir, "Lightweight privacy-preserving data aggregation scheme for smart grid metering infrastructure protection," *Int. J. Crit. Infrastruct. Protection*, vol. 22, pp. 16–24, Sep. 2018.
[18] F. Kserawi, "Privacy-preserving data aggregation in smart power grid systems," M.S. thesis, Qatar Univ., Doha, Qatar, Jun. 2021. [Online]. Available: http://hdl.handle.net/10576/21579
[19] F. Kserawi and Q. M. Malluhi, "Privacy preservation of aggregated data using virtual battery in the smart grid," in *Proc. IEEE 6th Int. Conf. Dependability Sensor, Cloud Big Data Syst. Appl. (DependSys)*, Dec. 2020, pp. 106–111.
[20] R. K. Barik, S. K. Gudey, G. G. Reddy, M. Pant, H. Dubey, K. Mankodiya, and V. Kumar, "FogGrid: Leveraging fog computing for enhanced smart grid network," in *Proc. 14th IEEE India Council Int. Conf. (INDICON)*, Dec. 2017, pp. 1–6.
[21] J.-N. Liu, J. Weng, A. Yang, Y. Chen, and X. Lin, "Enabling efficient and privacy-preserving aggregation communication and function query for fog computing-based smart grid," *IEEE Trans. Smart Grid*, vol. 11, no. 1, pp. 247–257, Jan. 2020.
[22] J.-M. Bohli, C. Sorge, and O. Ugus, "A privacy model for smart metering," in *Proc. IEEE Int. Conf. Commun. Workshops*, May 2010, pp. 1–5.
[23] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. 3rd Theory Cryptogr. Conf. (TCC)*. New York, NY, USA: Springer, Mar. 2006, pp. 265–284.
[24] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, nos. 3–4, pp. 211–407, 2014.
[25] C. Clifton and T. Tassa, "On syntactic anonymity and differential privacy," in *Proc. IEEE 29th Int. Conf. Data Eng. Workshops (ICDEW)*, Apr. 2013, pp. 88–93.
[26] V. Costan and S. Devadas, "Intel SGX explained," *IACR Cryptol. ePrint Arch.*, vol. 2016, no. 86, pp. 1–118, 2016.
[27] L. Lyu, J. Jin, S. Rajasegarar, X. He, and M. Palaniswami, "Fog-empowered anomaly detection in IoT using hyperellipsoidal clustering," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1174–1184, Oct. 2017.
[28] Y. Huo, C. Yong, and Y. Lu, "Re-ADP: Real-time data aggregation with adaptive ω-event differential privacy for fog computing," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–13, Jul. 2018.
[29] M. U. Hassan, M. H. Rehmani, R. Kotagiri, J. Zhang, and J. Chen, "Differential privacy for renewable energy resources based smart metering," *J. Parallel Distrib. Comput.*, vol. 131, pp. 69–80, Sep. 2019.
[30] G. Kellaris, S. Papadopoulos, X. Xiao, and D. Papadias, "Differentially private event sequences over infinite streams," *Proc. VLDB Endowment*, vol. 7, no. 12, pp. 1155–1166, 2014.
[31] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theory*, vol. IT-22, no. 6, pp. 644–654, Nov. 1976.
[32] *Announcing the Advanced Encryption Standard (AES)*, NIST-FIPS Standard 197, Federal Information Processing Standards Publication, 2001, pp. 1–51.
[33] H. Krawczyk, M. Bellare, and R. Canetti, "HMAC: Keyed-hashing for message authentication," Netw. Work. Group, Tech. Rep. RFC 2104, 1997.
[34] M. Bellare, R. Canetti, and H. Krawczyk, "Keying hash functions for message authentication," in *Proc. 16th Annu. Int. Cryptol. Conf.* Santa Barbara, CA, USA: Springer, Aug. 1996, pp. 1–15.
[35] M. Bellare, R. Canetti, and H. Krawczyk, "Message authentication using hash functions: The HMAC construction," *RSA Lab. CryptoBytes*, vol. 2, no. 1, pp. 12–15, 1996.
[36] G. Hebrail and A. Berard, "Individual household electric power consumption data set," Univ. California, Irvine, CA, USA, Tech. Rep. 2012-08-30, 2012.
[37] S. Barker, "UMass smart* dataset—2017 release," Manning College Inf. Comput. Sci., Univ. Massachusetts Amherst, Amherst, MA, USA, Tech. Rep. 2017 Release, 2017.
[38] F. Li, B. Luo, and P. Liu, "Secure information aggregation for smart grids using homomorphic encryption," in *Proc. 1st IEEE Int. Conf. Smart Grid Commun.*, Oct. 2010, pp. 327–332.
[39] K. Kursawe, G. Danezis, and M. Kohlweiss, "Privacy-friendly aggregation for the smart-grid," in *Proc. 11th Int. Symp. PETS*. Waterloo, ON, Canada: Springer, Jul. 2011, pp. 175–191.
[40] R. Lu, X. Liang, X. Li, X. Lin, and X. Shen, "EPPA: An efficient and privacy-preserving aggregation scheme for secure smart grid communications," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 9, pp. 1621–1631, Sep. 2012.
[41] T.-H. H. Chan, E. Shi, and D. Song, "Privacy-preserving stream aggregation with fault tolerance," in *Proc. 16th Int. Conf. FC*. Kralendijk, Bonaire: Springer, Mar. 2012, pp. 200–214.
[42] N. K. Singh and V. Mahajan, "End-user privacy protection scheme from cyber intrusion in smart grid advanced metering infrastructure," *Int. J. Crit. Infrastruct. Protection*, vol. 34, Sep. 2021, Art. no. 100410.
[43] V. Rastogi and S. Nath, "Differentially private aggregation of distributed time-series with transformation and encryption," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Jun. 2010, pp. 735–746.
[44] Z. Erkin and G. Tsudik, "Private computation of spatial and temporal power consumption with smart meters," in *Proc. 10th Int. Conf. ACNS*. Singapore: Springer, Jun. 2012, pp. 561–577.
[45] K. Mahmood, S. A. Chaudhry, H. Naqvi, S. Kumari, X. Li, and A. K. Sangaiah, "An elliptic curve cryptography based lightweight authentication scheme for smart grid communication," *Future Generat. Comput. Syst.*, vol. 81, pp. 557–565, Apr. 2018.
[46] K. Gai, Y. Wu, L. Zhu, L. Xu, and Y. Zhang, "Permissioned blockchain and edge computing empowered privacy-preserving smart grid networks," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7992–8004, Oct. 2019.
[47] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *Proc. 24th Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* St. Petersburg, Russia: Springer, May/Jun. 2006, pp. 486–503.
[48] S. Barker, A. Mishra, D. Irwin, E. Cecchet, P. Shenoy, and J. Albrecht, "Smart*: An open data set and tools for enabling research in sustainable homes," in *Proc. SustKDD*, Aug. 2012, vol. 111, no. 112, p. 108.
[49] F. Farokhi, "Temporally discounted differential privacy for evolving datasets on an infinite horizon," in *Proc. ACM/IEEE 11th Int. Conf. Cyber-Phys. Syst. (ICCPS)*, Apr. 2020, pp. 1–8.

FAWAZ KSERAWI received the bachelor's degree in computer engineering from Al-Ahliyya Amman University and the M.Sc. degree in computing from the College of Engineering, Qatar University (QU), in 2021. He is currently working as a Senior Application Developer with the IT Department, QU. His research work is centered on smart grid and IoT devices' data privacy. His research interests include machine learning, data privacy, and security.

SAEED AL-MARRI received the M.Sc. degree in cybersecurity from Hamad Bin Khalifa University (HBKU), in 2019. He is currently pursuing the Ph.D. degree in computer science with Qatar University (QU), working on privacy enhancing technologies (PET) in the smart grid. In 2020, he was awarded by the 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 research interests include distributed and confidential computing domains, with special interest in the Internet of Things (IoT) security and privacy.

QUTAIBAH MALLUHI (Member, IEEE) received the B.S. and M.S. degrees in computer engineering from the King Fahd University of Petroleum & Minerals, Saudi Arabia, and the M.S. and Ph.D. degrees in computer science from the University of Louisiana at Lafayette. He is currently a Professor with the Department of Computer Science and Engineering, Qatar University (QU). He was the Head of the Department (2006–2012) and the Director of the KINDI Center for Computing Research (2012–2016) at QU. He served as a Professor at Jackson State University. He 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. He has received the QU Research Award, the JSU Technology Transfer Award, the JSU Faculty Excellence Award, and several best paper awards.