

# A Lightweight Privacy-Preserving Smart Metering Billing Protocol with Dynamic Tariff Policy Adjustment

Farid Zaredar, and Morteza Amini

**Abstract**—The integration of information and communication technology (ICT) with traditional power grids has led to the emergence of smart grids. Advanced metering infrastructure (AMI) plays a crucial role in smart grids by facilitating two-way communication between smart meters and the utility provider. This bidirectional communication allows intelligent meters to report fine-grained consumption data at predefined intervals, enabling accurate billing, efficient grid monitoring and management, and rapid outage detection. However, the collection of detailed consumption data can inadvertently disclose consumers' daily activities, raising privacy concerns and potentially leading to privacy violations. To address these issues and preserve individuals' privacy, we propose a lightweight privacy-preserving smart metering protocol specifically designed to support real-time tariff billing service with dynamic policy adjustment. Our scheme employs an efficient data perturbation technique to obscure precise energy usage data from internal adversaries, including the intermediary gateways and the utility provider. Subsequently, we validate the efficiency and security of our protocol through comprehensive performance and privacy evaluations. We examined the computational, memory, and communication overhead of the proposed scheme. The execution time of our secure and privacy-aware billing system is approximately 3.94540 seconds for a complete year. Furthermore, we employed the Jensen-Shannon divergence as a privacy metric to demonstrate that our protocol can effectively safeguard users' privacy by increasing the noise scale.

**Index Terms**—Smart Meter Privacy, Secure and Private Dynamic Billing, Data Perturbation, Dynamic Tariff Policy Adjustment

## I. INTRODUCTION

THE continuous growth of smart cities, has led to significant changes in energy demand and supply patterns. Traditional electrical grids can no longer adequately support the evolving energy requirements of diverse sectors in modern urban environments. Consequently, the concept of smart grids has been introduced to address energy infrastructure needs, providing capabilities such as dynamic billing, real-time monitoring, efficient grid management, rapid self-healing, and accurate load forecasting [1].

Today, Smart grids are expanding rapidly across the globe. For instance, the United States invested \$6.4 billion on smart grid technology in 2018 and this investment is projected to grow to \$16.4 billion annually by 2026 [2]. The smart grid market in the United States witnesses significant growth and

F. Zaredar is with Department of Computer Engineering, Sharif University of Technology, Tehran, Iran (e-mail: farid.zaredar78@sharif.edu).

M. Amini is with Department of Computer Engineering, Sharif University of Technology, Tehran, Iran (e-mail: amini@sharif.edu).

is forecasted to reach \$64.34 billion by 2030 [3]. Similarly, According to the IMARC group [4], the smart grid market in the United Kingdom was valued at \$1.84 billion in 2023 and is predicted to reach \$8.46 billion by 2032, reflecting a compound annual growth rate (CAGR) of 16.30% during the period 2024 to 2032. The European Union (EU) has adopted a rigorous strategy for deploying smart metering systems, investing €47 billion in this sector by 2030 [5].

Smart grids are more reliable, efficient, and sustainable compared to traditional power grids [6]. Unlike the earlier technology, automated meter reading (AMR), which enabled only unidirectional communication [7], the advanced metering infrastructure (AMI) facilitates bidirectional communication between smart meters and the control center. This capability allows intelligent meters to transmit consumption data frequently (e.g., every 15 or 30 minutes) and at fine granularity to the meter data management system (MDMS). The MDMS collects and processes consumers' consumption readings to support various functional utilities, such as billing, operational, and value-added services [8], [9], [10]. Furthermore, these modern power grids can significantly reduce carbon emissions by integrating with renewable energy sources (RES) and utilizing clean energy sources, such as solar and wind power [11].

While smart grids offer numerous valuable services to both customers and utility providers, they also raise notable privacy concerns. For instance, the collection of fine-grained consumption data for billing, can unintentionally reveal users' daily routines, such as leaving for work (i.e., individual absence), cooking, watching TV, working with home computers or smart devices, and sleeping patterns, thereby compromising inhabitants' privacy. As a consequence, safeguarding customer privacy has become a critical issue that must be addressed.

In the literature, numerous privacy-preserving smart metering protocols have been proposed. These schemes preserve consumers' privacy while enabling essential utilities, including bill generation, load monitoring, and anomaly detection (e.g., outage diagnosis). Prior studies have employed various privacy-enhancing techniques, such as secure multi-party computation, zero-knowledge proofs, commitment schemes, homomorphic encryptions, and data masking. However, these approaches often introduce significant computational, memory, and communication overhead.

To mitigate such performance challenges, we propose a lightweight privacy-preserving smart metering protocol designed to enable real-time tariff billing with dynamic tariff

policy adjustments. Our protocol leverages an efficient data perturbation technique to obscure fine-grained consumption readings, preventing the disclosure of users' detailed energy usage data. Additionally, our scheme is adaptable to tariff policy changes adopted by the energy suppliers. To the best of our knowledge, this is the first privacy-preserving smart metering billing protocol that supports dynamic tariff policy adjustments. Since IoT devices such as smart meters have limited computational resources, our approach employs a resource-friendly data perturbation technique to introduce calibrated noise into consumption values. Billing calculations are performed using these noisy consumption values, which are reported at predefined intervals (e.g., every 15 minutes) to the control center. Furthermore, the utility provider can define the billing period and compute daily, weekly, or monthly bills based on its charging policy.

Our paper contribution is summarized as follows:

- We introduce an efficient lightweight data perturbation technique to safeguard customers' privacy while offering an accurate real-time tariff billing service.
- Our protocol allows utility providers to specify the billing period to charge customers based on predefined charging policies.
- In response to energy pricing adjustments (or tariff updates), the utility provider can recompute new bills for its customers dynamically.
- Our protocol is resource-friendly and incurs minimal computational overhead compared to existing protocols in the literature.

The rest of the paper is organized as follows. Section II reviews existing related work in the literature. Section III outlines the system model, threat model, design objectives, and key assumptions. Section IV details the proposed protocol. Section V presents the evaluation of the scheme. Finally, Section VI concludes the paper.

## II. RELATED WORK

Numerous privacy-preserving smart metering schemes designed to support billing utilities have been proposed in recent years. To compute users' bills, it is necessary to collect either coarse-grained or fine-grained consumption readings [12]. Although the attribution of meters' power usage data can lead to privacy risks, such as user behavior profiling or daily activity recognition, it remains an essential requirement to calculate a customer's monthly bill. To overcome these privacy challenges and deliver dynamic billing services, several schemes have been devised, which can be categorized as follows and are shown in Fig. 1.

- 1) Customer-Side Secure Billing
- 2) Third-Party Mediated Secure Billing
- 3) Utility-Centric Secure Billing

Since we outlined each class, let's delve deeper into each category and survey the existing schemes in the literature.

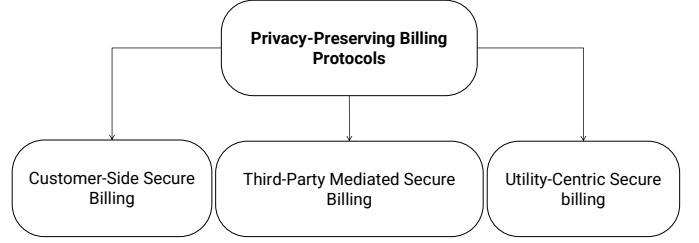


Fig. 1. Categorization of privacy-preserving smart metering billing protocols

### A. Customer-Side Secure Billing

In these approaches, the billing computation occurs on the customer's side (e.g., on a smart meter or a customer device) using trusted hardware or cryptographic techniques such as commitment schemes and zero-knowledge proofs. While these schemes empower customers with full control over their consumption data and eliminate the need to share fine-grained readings, they can introduce significant computational complexity, raise trust issues (i.e., utilities cannot always rely on customer premise bill calculations), and incur additional costs due to the utilization of trusted hardware.

Jawurek et al. [13] proposed a privacy-preserving protocol based on a plug-in component (or privacy component) located between smart meters and energy suppliers. It proves the correctness of the bill calculation to the energy supplier using the Pedersen commitment scheme and generates consumers' monthly bills.

In another study, Molina-Markham et al. [14] introduced a privacy-enhancing scheme based on zero-knowledge proofs and anonymization. Intelligent meters are responsible for computing consumers' monthly bills and proving the correctness of the calculation through a challenge-response protocol.

Rial et al. [15] employ a zero-knowledge proof in conjunction with Groth integer commitments [16] to deliver secure billing services at the customers' premises. Their subsequent study [17] generalizes to multi-user, multi-meter settings, adopts polynomial commitments to improve efficiency, and supports complex tariffs.

Eccles et al. [18] presented two protocols for time-of-use billing using trusted, tamper-proof meters. The first protocol is straightforward and utilizes trusted hardware within the smart meter that computes the user's bill at each billing period. The second protocol utilizes the Pedersen commitment approach to prove the integrity and correctness of bills computed within smart meters.

Petrlic [19] incorporated TPMs in meters for secure billing and used remote attestation to confirm software integrity.

Romdhane et al. [20] leverage the elliptic-curve Pedersen commitments to offer a private and secure time-of-use billing service. Trusted meters transmit commitments, allowing utilities to verify monthly bills without accessing raw data.

Zhang et al. [21] designed a multi-channel system balancing privacy and functionality. Meters locally compute bills using zero-knowledge proofs to confirm accuracy without sharing fine-grained data.

### B. Third-Party Mediated Secure Billing

In this class, the billing calculation is performed by third-party intermediaries in a privacy-preserving manner. These intermediary nodes are typically assumed to be trusted or semi-trusted entities (i.e., honest but curious). While such solutions offer simplicity and support computational outsourcing (due to meter's limited computational resources), they introduce additional costs due to third-party deployment, raise privacy risks due to the single point of trust, and increase computational overhead due to use of homomorphic encryption techniques.

Bohli et al. [22] proposed a privacy model for smart metering that supports both billing and operational utilities. A trusted third party is employed in their protocol that collects metering data and performs a temporal aggregation operation on it. Afterward, the trusted third party only reports the total consumption per smart meter over a billing period (e.g., a month) to the energy supplier.

A privacy-preserving data aggregation scheme is introduced by Wang et al [23]. Meters encrypt multi-dimensional data; fog nodes aggregate and forward to arbitration centers, which decrypt partially for billing.

Xu et al. [24] introduced a privacy-preserving smart metering framework adaptable to varying trust scenarios using homomorphic encryption and secure MPC, managing fine-grained data through trusted entities or collaborative computations. This scheme can provide different services such as billing.

### C. Utility-Centric Secure Billing

In this category, the utility provider securely computes and generates customers' bills based on a predefined billing period. In these approaches, the utility provider manages the entire billing process, allowing for seamless integration. Additionally, by avoiding third parties, they reduce additional costs and single points of failure. In contrast, these schemes increase customers' privacy concerns due to data sharing. Consequently, it is important to guarantee customer privacy by obscuring fine-grained meter readings while offering dynamic billing services.

Efthymiou et al. [25] proposed a protocol using distinct identifiers for billing and operational purposes, obscuring fine-grained data.

Ababne et al. [26] proposed a private and secure billing protocol based on Shamir secret sharing. In their scheme consumption values are blinded and then distributed among peers. Peers reconstruct the aggregate blinded reading, and a leader smart meter collects, verifies, and encrypts the reconstructed data before storing it on a distributed file system (i.e., IPFS). Lastly, Utilities reconstruct aggregated data without detailed user insights and compute monthly bills.

Li et al. [27] introduced a fine-grained privacy-preserving aggregation scheme for outsourced smart metering data. In their approach, smart meters periodically report encrypted fine-grained consumption data to locally deployed fog servers. These fog servers aggregate the encrypted data according to rules defined by an outsourcing service provider, who also sets dynamic electricity pricing. Although fog servers and the

service provider are considered semi-trusted, the power utility is fully trusted. Aggregated encrypted data is forwarded by fog servers to the service provider, who later obtains decryption keys from the utility to calculate interval-specific electricity prices. These computed prices allow the utility to determine partial bills per interval, and subsequently, total monthly bills for each customer.

The PPETD framework [28], proposed by Nabil et al., employs secret sharing to preserve user privacy. More specifically, meters report metering data periodically after masking the data by utilizing a secret sharing technique. This masking approach ensures that when the system operator aggregates readings for various purposes such as billing and load monitoring, the individual masks cancel out, thereby providing accurate total consumption.

Abidin et al. [29] developed two protocols; one enabled local energy trading, and the other masked meter readings for aggregated billing, allowing utilities to compute consumers' bills securely.

### D. Related Work Conclusion

In this section, we have examined various privacy-preserving smart metering billing protocols. We have classified these schemes into three main categories and outlined the advantages and disadvantages of each category. The first category mostly employs zero-knowledge proofs and commitment schemes to prove the correctness of bill generation. The second category mostly relies on trusted or semi-trusted third parties to compute consumers' monthly bills. The third category leverages numerous privacy-enhancing techniques, such as secret sharing schemes, homomorphic encryption, data masking, and temporal-based aggregation schemes to safeguard customers' privacy while offering accurate billing.

## III. PROBLEM STATEMENT AND DESIGN GOALS

The collection of attributable fine-grained consumption readings is essential to provide numerous utilities, such as dynamic billing, load forecasting, grid monitoring and management, and a wide range of value-added services for both customers and utility providers. In fact, higher data precision and granularity lead to more accurate bills, enhanced energy consumption predictions, and near real-time grid monitoring. Although fine-grained consumption data can help to offer various utilities, such data can raise significant privacy concerns. As we mentioned earlier by analyzing energy readings, energy suppliers can infer private information, including customer presence or absence, consumer daily activities (e.g., watching TV, having meals, and using personal computers), and even identifying home appliance types.

Previous studies have proposed various privacy-enhancing approaches that introduce significant computational and algorithmic complexity, thereby demanding computational resources from both smart meters and control centers. To tackle these privacy and resource-constrained concerns, we propose a lightweight privacy-preserving smart metering protocol for real-time tariff billing with dynamic policy adjustment. While our protocol preserves user privacy, it maintains data utility

to offer dynamic and accurate billing by using real-time tariffs. Before describing our protocol, it is crucial to examine our system model, threat model, underlying assumptions, and design goals.

#### A. System Model

Our system model defines a smart grid architecture comprising three main entities: (1) smart meters, (2) local aggregators, and (3) the utility provider. In the scheme's network model, each entity communicates with another entity using different communication technologies. For example, smart devices and home appliances connect to the smart meter via a home area network (HAN). Then each smart meter connects to its local aggregator through a neighborhood area network (NAN). Finally, local aggregators connect to the utility provider via a wide area network (WAN). In the following, we explain each entity and its role in the proposed scheme in more detail.

- 1) **Smart Meters:** Smart electricity meters report fine-grained consumption readings at predefined intervals (e.g., every 15 minutes) to the utility provider via their local intermediary gateways, referred to as aggregators. Smart meters are computationally bounded and have low computational resources. It is important to notice that smart meters are capable of performing primitive cryptographic operations to deliver consumption values over TLS to the control center.
- 2) **Local Aggregators:** Aggregators are considered as intermediary nodes that relay meter readings to the utility provider. These local nodes have sufficient computing resources and also can be utilized as local gateways to handle outsourced computations. In our scheme, they only act as relay nodes.
- 3) **Utility Provider:** The utility provider is responsible for providing various services to customers, including dynamic billing, grid and demand-supply management, and load forecasting. In our protocol, the utility provider offers a secure and private billing service to the customer.

The smart grid architecture and network model are shown in Fig. 2.

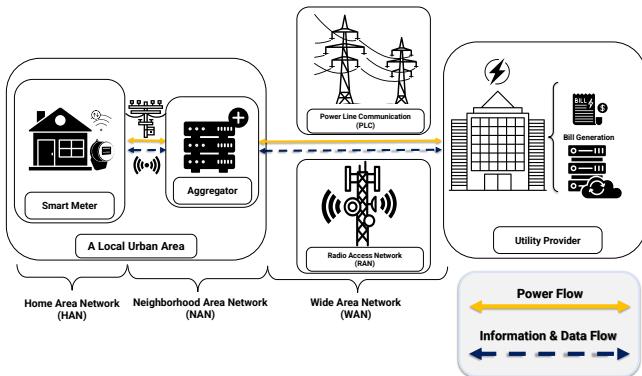


Fig. 2. The smart grid's architecture integrates smart meters, aggregators, and the utility provider

#### B. Threat Model and Assumptions

In the proposed scheme, we specify distinct levels of trust for each main entity based on the protocol's system model. Consequently, we can define our scheme's threat model as follows.

- **Smart Meters:** Smart meters are fully trusted and tamper-resistant. These intelligent power meters are equipped with a tamper-proof protection mechanism. This security mechanism prevents customers from various malicious activities, such as reverse engineering and physical hardware attacks (i.e., invasive, semi-invasive, and non-invasive attacks). Upon any intrusion attempts, the tamper-proof mechanism triggers an alarm and informs the energy supplier. Consequently, the latest state of the meter is logged and reported to the utility provider. Subsequently, the meter erases its security credentials (e.g., encryption and decryption keys), and the meter's status changes to non-functional. Ultimately, the power utility agent investigates the incident and initiates appropriate measures, such as issuing penalties to the customer.
- **Aggregators:** Local aggregators are assumed to be honest but curious, which means they adhere to the protocol but may attempt to infer private information out of curiosity. Local aggregators are managed and configured by the utility provider which can result in collusion attacks.
- **Utility Provider:** The utility provider is considered as a semi-trusted entity similar to local intermediary gateways discussed earlier. This entity follows the protocol but may analyze consumers' energy readings to extract useful usage patterns and information for its own advantage.

Our scheme's threat model is shown in Fig. 3.

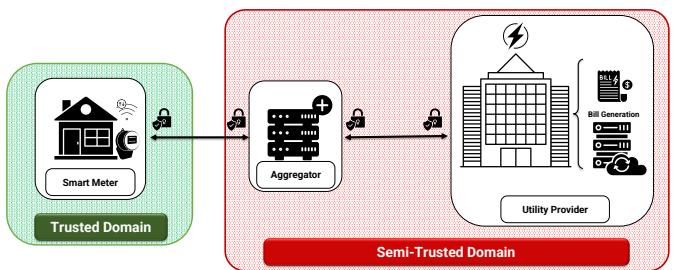


Fig. 3. Our scheme's threat model

Our protocol also establishes multiple assumptions relevant to the AMI infrastructure communication and its components, which are discussed as follows.

- a) **Secure and Fortified Bidirectional Channels:** All bidirectional links between AMI components are assumed to be secured using TLS, thereby preventing adversaries from launching both passive and active attacks.
- b) **Privacy-Aware Pre-Authentication:** Prior to protocol initiation, AMI components use a privacy-aware authentication scheme to register themselves within the AMI network.
- c) **Robust Public-Key Dissemination:** Public keys are distributed securely among meters, aggregators, and the utility provider.

provider — often through mechanisms like public key certificates — ensuring their integrity throughout the process.

### C. Design Goals

In our lightweight privacy-preserving smart metering billing protocol, we aim to achieve multiple goals as follows:

- I. **Data Privacy:** The main goal is to preserve meter data privacy by incorporating lightweight privacy-preserving techniques such as data perturbation and data minimization. In the proposed scheme, a resource-friendly noise application mechanism is utilized to perturb consumption readings. As a consequence, only noisy consumption values are reported, thereby safeguarding consumers' privacy.
- II. **Data Utility:** While our scheme preserves the privacy of attributable fine-grained metering data, it also maintains data utility through the zero-sum noise property, introduced by the noise application mechanism.
- III. **Lightweight Dynamic Accurate Billing:** Our scheme supports the real-time pricing (RTP) policy, which is a time-based dynamic pricing approach. This ensures that specified real-time tariffs are delivered to smart meters and informs customers about electricity costs at each interval based on grid status. By introducing a real-time tariff billing protocol, we guarantee more accurate bill generation compared to traditional linear billing policies.
- IV. **Dynamic Tariff Policy Adjustment:** Upon any sudden tariff policy changes, the utility provider can recompute the users' bills in a secure and private manner.

## IV. PROPOSED SCHEME

As we discussed earlier, attributable fine-grained consumption values are necessary to generate accurate dynamic bills for customers. Although fine-grained readings are required for bill generation and accounting, such time-series data can reveal consumers' daily habits and behaviors throughout the day. To overcome these privacy challenges and provide a precise dynamic billing system based on real-time tariffs, we propose a lightweight, privacy-preserving smart metering protocol for real-time tariff billing with dynamic tariff policy adjustment. In this section, we present an overview of our protocol to offer a clear and concise introduction, followed by a formal and detailed explanation.

### A. Protocol Overview

Our privacy-aware billing protocol is designed under two distinct scenarios. In the first scenario, tariff information is delivered at each interval to smart meters and remains unchanged. This is a typical scenario where the utility provider maintains the adopted tariff policies. However, in the second scenario, the utility provider may change tariff policies after the billing period. For example, it may reduce electricity costs for certain intervals due to governmental or regulatory mandates. In this situation, the utility provider can easily recompute the bills and notify customers about the newly adopted tariff policies and newly generated bills. We first

examine the initial scenario, which consists of multiple steps, outlined below:

- 1) The utility provider initiates the protocol by specifying different tariffs for each urban or rural district within a city and transmits the defined real-time tariffs to local aggregators at each interval (e.g., every 15 minutes).
- 2) Each local aggregator sends tariffs to the smart meters within its network to inform customers about the electricity costs corresponding to their local urban or rural area at that interval.
- 3) Upon receiving the tariff information, each smart meter informs the customer about the electricity price at that interval via an in-home display. In the case of peak hours, this can help customers reduce their energy consumption and shift non-essential energy-consuming tasks to off-peak hours, thereby optimizing usage and saving money.
- 4) Afterward, each smart meter generates a random value using a secure PRNG (i.e., ThreeFry) that follows a normal distribution with a mean of zero and a standard deviation of  $\sigma$ . These random values are retained for a specific duration in anticipation of potential tariff policy adjustments.
- 5) It then perturbs the consumption reading of that interval by adding the original consumption data to the generated random value to generate a noisy consumption value. Subsequently, each meter sends the noisy consumption reading to the utility provider.
- 6) Simultaneously, each meter multiplies the random value by the tariff received for that period and stores the result.
- 7) The process continues through successive intervals until the one immediately preceding the bill generation period. In the last interval, the smart meter performs a summation operation over the stored multiplied results. Then, each meter computes the additive inverse of the resulting sum and divides it by the last received tariff to generate a new noise value for the last consumption data.
- 8) It then perturbs the last power usage data and sends the final noisy reading to the utility provider.
- 9) The utility provider only collects the noisy consumption values over a billing period (e.g., a week, two weeks, or a month). As we mentioned earlier, our noise mechanism adheres to a zero-sum property that leads to accurate dynamic bills for each customer. As a result, the utility provider can generate bills using noisy consumption values.
- 10) The utility provider multiplies each noisy consumption value by the corresponding tariff and generates a vector of noisy partial bills.
- 11) Subsequently, it performs a summation over the noisy partial bills and computes the final bills for each customer for that specified billing period, leveraging the zero-sum property of the applied noise mechanism to ensure accuracy.
- 12) Finally, the utility provider forwards the final computed bills to each customer through local intermediary gateways and issues the charges.

In the second scenario, due to a tariff policy adjustment,

the utility provider must recalculate the bills using a new tariff vector, in which some tariff elements are modified while others remain unchanged. This scenario is composed of several steps, discussed below.

- 1) At the end of the billing period, when the last noisy consumption value is reported, the utility provider is required to change the adopted tariff policy (e.g., due to regulatory mandates).
- 2) In this situation, the utility provider forwards a new tariff vector, which consists of tariff information for different intervals, to smart meters through local aggregators for urban or rural areas.
- 3) Each smart meter receives the vector and multiplies each stored random value by the corresponding new tariff, then performs a summation over the resulting values. It then computes the additive inverse of the resulting sum, which is divided by the last new tariff, to generate a new noise value.
- 4) It perturbs only the final consumption data using the calculated noise value and transmits the final noisy consumption value to the utility provider.
- 5) The utility provider applies the tariffs to the noisy consumption values and aggregates the noisy partial bills. Due to the zero-sum property of the data perturbation mechanism, the noise values cancel out, and a new final bill is generated for each customer.
- 6) Finally, the utility provider sends the new final bill to customers through local intermediary gateways and issues the corresponding charges.

According to our protocol, we enable energy suppliers to dynamically generate consumers' accurate bills using the RTP policy in a secure and privacy-preserving way. Now we delve deeper into each of the two scenarios and explain the secure billing scheme in detail.

#### B. Protocol Details

We now explain each privacy-preserving billing scenario formally. The symbols utilized in the formal specification of the protocol are presented in Table I.

TABLE I  
BILLING SERVICE COMPONENTS

Symbols	Description	Formal Representation
$M$	Number of registered and active smart meters in the network	$M \in \mathbb{N}$
$L$	Number of intervals during a billing period	$L \in \mathbb{N}$
$\lambda$	Number of days in a billing period	$\lambda \in \mathbb{N}$
$t_i$	The $i$ -th time interval	$T = \langle t_1, t_2, \dots, t_i, \dots, t_L \rangle$
$c_{t_i}$	The consumption data at $i$ -th time interval	$C = \langle c_{t_1}, c_{t_2}, \dots, c_{t_i}, \dots, c_{t_L} \rangle$
$trf_{t_i}$	The tariff information/electricity cost at $i$ -th time interval	$TRF = \langle trf_{t_1}, trf_{t_2}, \dots, trf_{t_i}, \dots, trf_{t_L} \rangle$
$s_{t_i}$	The random value generated using a secure PRNG by a smart meter at $i$ -th time interval	$S = \langle s_{t_1}, s_{t_2}, \dots, s_{t_i}, \dots, s_{t_L} \rangle$
$nc_{t_i}$	The noisy consumption data at $i$ -th time interval	$NC = \langle nc_{t_1}, nc_{t_2}, \dots, nc_{t_i}, \dots, nc_{t_L} \rangle$
$nb_{t_i}$	The noisy partial bill at $i$ -th time interval	$NB = \langle nb_{t_1}, nb_{t_2}, \dots, nb_{t_i}, \dots, nb_{t_L} \rangle$
$FB_{id_j}$	The issued final bill for a smart meter with an ID of $id_j$	$FB = \{FB_{id_1}, FB_{id_2}, \dots, FB_{id_j}, \dots, FB_{id_M}\}$

In our protocol, smart meters report perturbed values every 15 minutes to the utility provider. Based on the length of re-

porting intervals  $|t_i|$  (i.e. 15-minute intervals) and the number of days in a billing period  $\lambda$ , the number of reports/intervals during a billing period is computed as follows:

$$L = \left( \left\lceil \frac{24 \cdot 60}{|t_i|} \right\rceil \right) \cdot \lambda \quad (1)$$

For instance, a smart meter configured to transmit data every 15 minutes is expected to send a total of 96 reports per day.

The energy supplier specifies tariffs for the electricity unit (e.g., \$0.05 kWh) at each interval to offer real-time tariffs based on grid status, each area's history, and regional characteristics (e.g., rural, urban, and industrial regions are charged based on different tariff policies). In fact, these provisioned real-time tariffs are broadcast to smart meters in each area through local aggregators. This process is shown in Fig. 4.

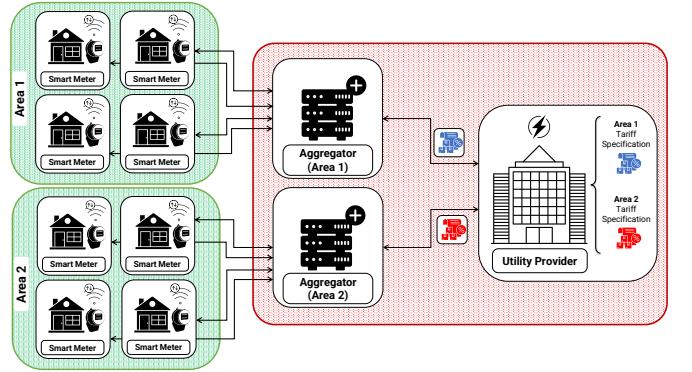


Fig. 4. Tariff dissemination based on grid status, area's history, and regional characteristics.

As shown in Fig. 4, the utility provider sends tariffs to each area at each interval and informs customers about the energy cost in near real-time. Customers can reduce their usage, thereby reducing their weekly or monthly bills.

Upon receiving the tariff, the smart meter generates a random value using a secure PRNG with a secure initial parameter  $seed$ , that follows a Gaussian distribution with a zero mean and a standard deviation of  $\sigma$ .

$$s_{t_i} \leftarrow RandGen(seed, \sigma), \quad 1 \leq i \leq L - 1 \quad (2)$$

Then the smart meter perturbs the recorded consumption reading  $c_{t_i}$  as follows:

$$nc_{t_i} \leftarrow c_{t_i} + s_{t_i}, \quad 1 \leq i \leq L - 1 \quad (3)$$

Afterward, it sends noisy consumption values at each interval through a secure channel to the utility provider via its local intermediary gateway. Simultaneously, the smart meter multiplies the received tariff at each interval by the corresponding generated random value and stores it in its storage.

$$Store(s_{t_i} \cdot trf_{t_i}), \quad 1 \leq i \leq L - 1 \quad (4)$$

This process continues until the time interval  $L - 1$ .

As we previously discussed, our data perturbation mechanisms take advantage of zero-sum property. Upon using this noise cancellation feature, the utility provider can generate

accurate dynamic bills for each customer using the noisy consumption readings reported during the billing period. This zero-sum property is defined as follows:

$$\sum_{i=1}^L (s_{t_i} \cdot trf_{t_i}) = 0 \quad (5)$$

To achieve the zero-sum property, defined in Relation 5, the smart meter introduces a controlled noise value using the stored values and the last tariff to perturb the last consumption data at  $L$ -th time interval as follows.

$$s_{t_L} = -\frac{\sum_{i=1}^{L-1} (s_{t_i} \cdot trf_{t_i})}{trf_{t_L}} \quad (6)$$

Now, the smart meter applies the last generated noise to the last power usage data and sends it to the utility provider.

$$nc_{t_L} \leftarrow c_{t_L} + s_{t_L} \quad (7)$$

Upon receiving the last noisy consumption data, the utility provider starts to compute the user's bill by multiplying perturbed consumption values by their corresponding tariff as follows:

$$nb_{t_i} \leftarrow nc_{t_i} \cdot trf_{t_i}, \quad 1 \leq i \leq L \\ NB = \langle nb_{t_1}, nb_{t_2}, \dots, nb_{t_L} \rangle \quad (8)$$

Subsequently, the utility provider performs a summation operation over the noisy partial bills to generate the final electricity bill for the customer.

$$FB_{id_j} = \sum_{i=1}^L nb_{t_i} \quad (9)$$

The following relation formally illustrates the noise cancellation process step by step.

$$\begin{aligned} FB_{id_j} &= \sum_{i=1}^L nb_{t_i} \\ &= \sum_{i=1}^L (nc_{t_i} \cdot trf_{t_i}) \\ &= \sum_{i=1}^L ((c_{t_i} + s_{t_i}) \cdot trf_{t_i}) \\ &= \sum_{i=1}^L ((c_{t_i} \cdot trf_{t_i}) + (s_{t_i} \cdot trf_{t_i})) \\ &= \sum_{i=1}^L (c_{t_i} \cdot trf_{t_i}) + \sum_{i=1}^L (s_{t_i} \cdot trf_{t_i}) \\ &= \sum_{i=1}^L (c_{t_i} \cdot trf_{t_i}) + \sum_{i=1}^{L-1} (s_{t_i} \cdot trf_{t_i}) + (s_{t_L} \cdot trf_{t_L}) \\ &= \sum_{i=1}^L (c_{t_i} \cdot trf_{t_i}) + \sum_{i=1}^{L-1} (s_{t_i} \cdot trf_{t_i}) - \sum_{i=1}^{L-1} (s_{t_i} \cdot trf_{t_i}) \\ &= \sum_{i=1}^L (c_{t_i} \cdot trf_{t_i}) \end{aligned} \quad (10)$$

The protocol flow in the first scenario is depicted in Fig. 5.

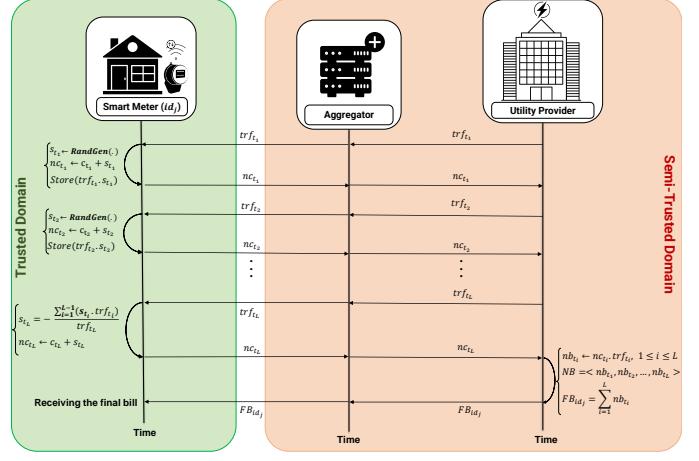


Fig. 5. Our proposed privacy-preserving billing protocol flow

In the second scenario, the utility provider is required to recompute the meter's final bill due to governmental or regulation mandates after the billing period. In such scenarios, the utility provider provisions a new tariff vector  $TRF_{new} = \langle trf'_{t_1}, trf'_{t_2}, \dots, trf'_{t_L} \rangle$  and broadcasts it to the meters within the network.

Since the utility provider must charge customers using the new tariff, each smart meter must recompute a new noise value to perturb the last consumption data. To achieve this, the meter first loads the generated random values during the billing period and multiplies each by the corresponding new tariff  $trf'_{t_i}$  and performs a summation over them. Then, the additive inverse of the resulting value divided by the last new tariff  $trf'_{t_L}$  is computed.

$$s'_{t_L} = -\frac{\sum_{i=1}^{L-1} (s_{t_i} \cdot trf'_{t_i})}{trf'_{t_L}} \quad (11)$$

Now the meter perturbs the last consumption data and forwards it to the utility provider.

$$nc'_{t_L} = c_{t_L} + s'_{t_L} \quad (12)$$

Since the new final perturbed reading enables the utility provider to recompute the customer's final bill, it only replaces the  $nc_{t_L}$  with  $nc'_{t_L}$  (i.e., the utility provider retains and reuses the previously reported  $N - 1$  noisy readings). The new final bill is then recomputed as follows.

$$\begin{aligned} NB &= \langle nc'_{t_1} \cdot trf'_{t_1}, nc'_{t_2} \cdot trf'_{t_2}, \dots, nc'_{t_L} \cdot trf'_{t_L} \rangle \\ FB'_{id_j} &= \sum_{i=1}^L nb'_{t_i} \end{aligned} \quad (13)$$

The second scenario is depicted in Fig. 6.

## V. EVALUATION

We evaluate our scheme from two distinct aspects: performance and privacy. To be more precise, we initiate the scheme's performance evaluation by leveraging three different

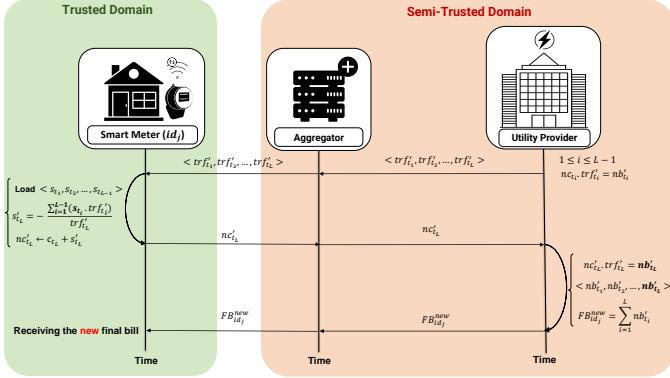


Fig. 6. The Secure billing protocol flow in the case of a tariff policy adjustment

metrics, including computational, communication, and memory overhead. Each performance metric is examined through both analytical evaluation and experimental assessment.

For the assessment of the scheme's privacy, we employed the Jensen-Shannon divergence metric. Jensen-Shannon (JS) divergence [30] is considered as a privacy metric that measures the divergence between two probability distributions such as the original and perturbed data probability distributions.

#### A. Performance Evaluation

As we previously discussed, our performance evaluation consists of evaluating the computational, memory, and communication overhead. We begin our assessment by establishing an in-depth analytical evaluation, which is then followed by an experimental validation (i.e., protocol execution time, memory consumption approximation, and minimum required bandwidth based on the size of delivered packets).

In our study, we implemented our scheme on two distinct computing platforms. To emulate the smart meter, we utilized the Quick Emulator (QEMU) to recreate the hardware environment of an Orange Pi One PC, featuring a 1.2GHz 32-bit ARM Cortex-A7 processor and 1GB of RAM. An Armbian Ubuntu-based operating system, which is specifically designed for Internet of Things (IoT) applications, runs on this visualized setup. To represent the aggregator and the utility provider, the second platform was equipped with a 1.8GHz Intel Core i7 processor (8 cores) and 32GB of RAM. To investigate the protocol's performance under constrained memory conditions, we intentionally restricted the RAM on this system to 1GB. It is also important to note that this machine ran on a Debian 12 operating system.

The protocol was developed in Python, which relies on robust and secure cryptographic libraries to enhance its functionality. We employed the secrets library [31] to generate initial secrets (seeds) and the randomgen library [32] to generate pseudorandom numbers, which were used to perturb consumption readings via the Threecryp pseudorandom number generator. Additionally, the NumPy library was used to perform mathematical computations on multi-dimensional arrays efficiently [33].

We based our analysis on a dataset from Slovakia's AMI [34], which includes energy consumption records from 1,000

anonymized residential smart meters. It records both active and reactive energy every 15 minutes over the course of a year. The dataset is complete, with no missing values or issues related to sparsity.

1) *Computational Overhead*: To start our analysis on the scheme's computational overhead, we present a notation table (see Table II), which lists formal symbols used in the analytical evaluation.

TABLE II  
NOTATION TABLE FOR COMPUTATIONAL OVERHEAD

Symbol	Description
$t_{PRNG}$	Time of generating a random value using a secure PRNG
$t_{arthm}$	Time of performing basic arithmetic operations to perturb data

According to the proposed scheme, each meter only performs three key operations, including random value generation, data perturbation, and generating the final noise value that results in noise cancellation.  $t_{sm}$  represents the smart meter's execution time, which is computed as follows where  $L$  is the number of intervals during the billing period.

$$t_{sm} = (L - 1)t_{PRNG} + (2L + 1)t_{arthm} \quad (14)$$

As we discussed earlier, aggregators act as intermediary nodes that only relay messages between smart meters and the utility provider. The utility provider performs two core operations, including computing noisy partial bills and adding them together to generate the final bill for the customer. The analytical computational overhead for the utility provider is as follows.

$$t_{up} = (2L)t_{arthm} \quad (15)$$

Finally, the total protocol computational overhead is computed as follows:

$$t_{bill} = t_{sm} + t_{up} \quad (16)$$

Furthermore, we performed an experiment to assess the performance and efficiency of the proposed solution. Table III indicates the protocol execution time for each entity, including the smart meter and the utility provider. As indicated, the execution time for one month, the first three months, the first six months, the first nine months, and the entire year is computed.

TABLE III  
PROTOCOL EXECUTION TIME (IN SECONDS) FOR EACH ENTITY IN THE SECURE DYNAMIC BILLING SERVICE

Period (Month)	Smart Meter Time ( $t_{sm}$ )	Utility Provider Execution Time ( $t_{up}$ )	Protocol Total Execution Time ( $t_{bill} = t_{sm} + t_{up}$ )
First Month	0.3720071145	0.0002259426	0.372233057
First 3 Months	0.9709989568	0.0006595637	0.97165852
First 6 Months	2.0005692140	0.0014254007	2.001994615
First 9 Months	2.8847361503	0.0020511732	2.886787324
Yearly	3.9425683663	0.0028328361	3.945401202

2) *Memory Overhead*: For the memory overhead, we also provide a notation table (see Table IV), which lists the symbols utilized in our analysis.

To calculate the memory overhead at customer premise, smart meters should retain the generated random values, measured readings, noisy consumption values, tariff information,

TABLE IV  
NOTATION TABLE FOR MEMORY OVERHEAD

Symbol	Description
$S_{rand}$	Size of the random value generated by a secure PRNG at each time interval $t_i$
$S_c$	Size of the consumption value measured by the smart meter at each time interval $t_i$
$S_{nc}$	Size of the noisy consumption value generated by the smart meter at each time interval $t_i$
$S_{trf}$	Size of the tariff information at each time interval $t_i$
$S_{s\cdot trf}$	Size of the resulting product at each time interval $t_i$
$S_{nb}$	Size of the noisy partial bill generated by the utility provider at each time interval $t_i$
$S_{FB}$	Size of other entities' public key

and the resulting products in their memory during the protocol execution. As a result, the memory overhead for the smart meters is computed as follows.

$$S_{sm} = L(S_{rand} + S_c + S_{nc} + S_{trf}) + (L - 1)S_{s\cdot trf} \quad (17)$$

The utility provider retains the noisy consumption values, noisy partial bills, and the final bill for each customer. Consequently, the memory overhead for the utility provider is as follows.

$$S_{up} = L(S_{nc} + S_{nb} + S_{trf}) + S_{FB} \quad (18)$$

Now, the memory overhead for the entire protocol is shown as follows.

$$S_{bill} = S_{sm} + S_{up} \quad (19)$$

In our experiment, the total memory consumed by the entire protocol for a billing period (i.e., a month) is approximately 5.91484MB, which is far less than 1GB of memory dedicated for each entity.

3) *Communication Overhead*: To compute the communication overhead for each link, we provide a table (see Table V), that defines the symbols utilized in the analytical evaluation.

TABLE V  
NOTATION TABLE FOR COMMUNICATION OVERHEAD

Symbol	Description
$P_{trf}$	The data size for the tariffs, which are delivered to smart meters at each time interval $t_i$
$P_{\langle trf'_1, \dots, trf'_L \rangle}$	The data size for the new tariff vector, upon new tariff policy adjustments
$P_{nc}$	The data size of the noisy consumption value at each time interval $t_i$
$P_{FB}$	The data size of the generated customer's final bill after the billing period

The minimum bandwidth required to deliver packets during protocol execution with acceptable latency on each link is estimated based on the maximum packet size exchanged between smart meters and the utility provider.

$$P_{bill} = \max(P_{trf}, P_{\langle trf'_1, \dots, trf'_L \rangle}, P_{nc}, P_{FB}) \quad (20)$$

In this work, we evaluated the communication overhead in both Neighborhood Area Networks (NAN) and Wide Area

Networks (WAN) by considering the link layer technologies used in each. In the proposed architecture, smart meters transmit their consumption data to a local aggregator via 6LoWPAN and IEEE 802.15.4g (Wi-SUN)—the latter of which is standardized for NAN connectivity and facilitates links between smart meters and intermediary gateways [35]. Once the aggregator receives these perturbed measurements, it forwards them to the utility provider over a 4G-LTE network. More specifically, the data is initially transmitted to the 4G-LTE Radio Access Network (RAN), often referred to as the Evolved Node B (eNB), then routed to the Packet Data Network Gateway (PGW), and finally delivered to the utility provider.

Table VI summarizes the communication overhead for each network segment. Each segment employs its own network protocol stack; as a result, the estimated overhead varies across different segments. As shown in Table VI, the available bandwidth for the Wi-SUN technology is about 250 kbps and for the 4G-LTE and the Ethernet is 1000 kbps. IEEE 802.15.4g provides a bandwidth ranging from 40 kbps up to 1000 kbps [35]. But in the standard case, it provides a bandwidth of 250 kbps. Although we recommend 802.11ah for NAN technology due to its 78 Mbps bandwidth capacity and 1000 m coverage [36], its adoption among utility providers remains limited compared to 802.15.4g—primarily due to the high costs associated with deploying new technologies.

### B. Privacy Evaluation

As we mentioned earlier, we utilized the JS divergence as a metric to evaluate the privacy introduced by the protocol. The JS divergence is a fundamental concept in information theory and statistics used to measure the difference/similarity between two distinct probability distributions. More specifically, this privacy metric measures how much the noisy consumption data distribution differs from the original consumption data distribution. The JS divergence is based on Kullback–Leibler (KL) divergence, but introduces two key improvements. First, it is symmetric: either distribution can serve as the baseline, and the result remains the same (i.e.,  $D_{JS}(Q||P) = D_{JS}(P||Q)$ , where  $P$  and  $Q$  are distinct distributions). Second, JS divergence is bounded between 0 and 1 [37] (i.e.,  $0 \leq D_{JS}(Q||P) \leq 1$ ), enabling a more interpretable assessment of privacy. In contrast, the KL divergence is asymmetric (i.e.,  $D_{KL}(Q||P) \neq D_{KL}(P||Q)$  where  $P$  and  $Q$  are distinct distributions) and unbounded (i.e.,  $0 \leq D_{KL}(Q||P) \leq \infty$ ) [38], which does not support assessing the privacy degree in the proposed solution. The JS metric is considered superior to the KL divergence due to its two key advantages. Higher JS values (closer to 1) indicate stronger privacy, as the noisy data reveals minimal information about the original data. Conversely, lower JS values (closer to 0) indicate weaker privacy, implying that more of the original data has been exposed and leaked.

Let  $P$  and  $Q$  be two distinct probability distributions. The KL divergence between  $P$  and  $Q$  is defined as follows:

$$D_{KL}(Q||P) = \sum_{x \in X} Q(x) \log_2 \left( \frac{Q(x)}{P(x)} \right) \quad (21)$$

TABLE VI  
PACKET SIZES AND PER-METER TRANSMISSION TIMES (20 SMs, 12.5 KBPS PER METER IN NAN, 50 KBPS PER METER IN WAN)

Payload	NAN		WAN					
	SM $\leftrightarrow$ AGG IEEE 802.15.4g (Wi-SUN)		AGG $\leftrightarrow$ eNB 4G-LTE (PDCP-LTE)		eNB $\leftrightarrow$ PGW IEEE 802.3 (Ethernet)		PGW $\leftrightarrow$ UP IEEE 802.3 (Ethernet)	
	Packet Size	Time	Packet Size	Time	Packet Size	Time	Packet Size	Time
$P_{trf} = 4\text{ B}$	29 B	0.01856 s	58 B	0.00928 s	126 B	0.02016 s	70 B	0.01120 s
$P_{\langle trf'_1, \dots, trf'_k \rangle} = 11520\text{ B}$ (# of tariffs = 2880)	14,333 B	9.17312 s	11,960 B	1.91360 s	12,504 B	2.00064 s	12,056 B	1.92896 s
$P_{nc} = 4\text{ B}$	29 B	0.01856 s	58 B	0.00928 s	126 B	0.02016 s	70 B	0.01120 s
$P_{FB} = 4\text{ B}$	29 B	0.01856 s	58 B	0.00928 s	126 B	0.02016 s	70 B	0.01120 s
Available BW	250 kbps		1000 kbps		1000 kbps		1000 kbps	
BW per SM	12.5 kbps		50 kbps		50 kbps		50 kbps	

Using KL divergence, we can compute the JS divergence as follows:

$$D_{JS}(Q||P) = \frac{1}{2} (D_{KL}(Q(x)||M(x)) + D_{KL}(P(x)||M(x))) \quad (22)$$

Where  $M$  is defined as follows:

$$M = \frac{1}{2}(Q(x) + P(x)) \quad (23)$$

In our protocol, the semi-trusted entities (i.e., intermediary gateways and the utility provider) may attempt to use noisy consumption values to infer users' private information. However, the use of a data perturbation mechanism reduces the risk of private information deduction based on the adopted noise scale. As a consequence, we computed the Jensen–Shannon divergence between the distributions of noisy consumption data and original consumption data. Our experimental results are presented in Table VII and Fig. 7. A higher noise scale results in preserving customers' privacy and lowering information disclosure and privacy violations. In contrast, lower JS values result in information leakage. In our solution, only perturbed consumption values are sent to semi-trusted entities (i.e., intermediary gateways and the utility provider). By adding higher noise values, we can minimize the utility provider's ability to infer customers' life patterns, thereby preserving consumers' privacy against semi-trusted entities. Additionally, this scheme can also resist collusion attacks by semi-trusted entities. As previously mentioned, only noisy readings are reported, and any collusion attempts are useless and bring no advantage to aggregators or the utility provider.

Based on the conducted analyses, increasing the noise scale causes the JS divergence value to approach 1, denoting stronger privacy. Conversely, decreasing the noise scale causes the JS divergence value to approach zero, signaling weaker privacy. This characteristic is illustrated in Fig. 8.

## VI. CONCLUSION

In this paper, we proposed a lightweight, privacy-preserving smart metering billing protocol based on real-time tariffs with

TABLE VII  
JENSEN-SHANNON DIVERGENCE BASED ON APPLIED NOISE LEVEL

Applied Noise Level (Standard Deviation Scale)	Jensen-Shannon Divergence Value
$\frac{\sigma}{9}$	0.62088
$\frac{\sigma}{6}$	0.81997
$\frac{\sigma}{3}$	0.92851
$\sigma$	0.96510
$\sigma \cdot 3$	0.97628
$\sigma \cdot 6$	0.98630
$\sigma \cdot 9$	0.98866

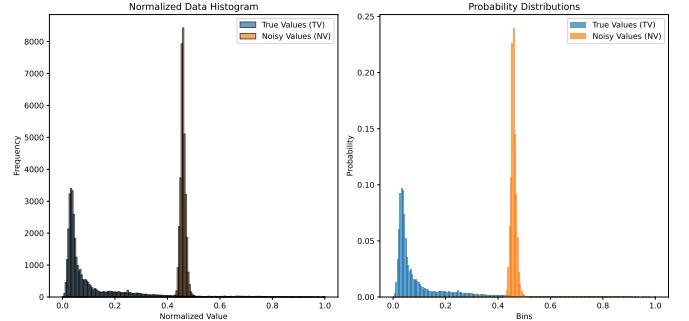


Fig. 7. The divergence between original (i.e., actual consumption readings) and perturbed (i.e., noisy consumption readings) distributions

dynamic tariff policy adjustment. Smart meters report their fine-grained consumption values to the utility provider for various services, such as accurate dynamic billing. Subsequently, the energy supplier generates weekly or monthly bills for customers based on the policies adopted by the energy company.

While such fine-grained readings are essential for billing, operational, and value-added services, they can also reveal sensitive private information, including users' presence or absence, daily routines, and behavioral patterns. To address these privacy concerns, we introduced a secure and privacy-aware billing system that supports real-time pricing policies

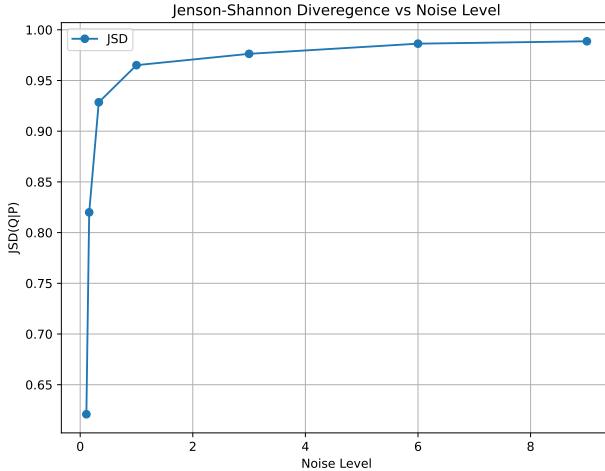


Fig. 8. The impact of noise levels on the JS divergence metric is significant; a higher noise level results in a dramatic increase in JS value, indicating enhanced customer privacy preservation.

for more accurate billing. Our scheme leverages a data perturbation mechanism with a zero-sum property and is designed to operate under two distinct scenarios.

In the first scenario, smart meters report noisy consumption values to the utility provider at each time interval. The utility provider then applies the corresponding tariff to each noisy consumption value to compute partial noisy bills. Finally, it performs a summation over all partial noisy bills to generate the customer's final bill.

In the second scenario, the utility provider must recompute customers' bills due to newly mandated government or regulatory tariff changes. To facilitate this, a new tariff vector is sent to the smart meters via intermediary gateways. Each smart meter then uses the data perturbation mechanism to perturb the last consumption value and sends the updated noisy reading to the utility provider. The provider, having stored the previous  $N - 1$  noisy values, applies the updated tariffs and computes the new final bill by summing the corresponding noisy partial bills.

Furthermore, we conduct a comprehensive evaluation of both performance and privacy. We introduce both analytical and experimental assessment to evaluate our scheme in terms of computational, memory and communication overhead. According to our experimental outcomes, the protocol's execution time for a complete year is approximately 3.94540 seconds which represents the scheme's efficiency while safeguarding privacy.

Lastly, we assessed the privacy guarantees using the Jensen–Shannon divergence metric. Our findings indicate that increasing the noise scale results in stronger privacy preservation. Specifically, the distribution of perturbed consumption data increasingly diverges from that of the original data, thereby minimizing information leakage and safeguarding user privacy.

## REFERENCES

- [1] H. Souri, A. Dhraief, S. Tili, K. Drira, and A. Belghith, "Smart metering privacy-preserving techniques in a nutshell," *Procedia Computer Science*, vol. 32, pp. 1087–1094, 2014.
- [2] U.S. Department of Energy, "2020 smart grid system report," United States Department of Energy, Tech. Rep., May 2022. [Online]. Available: [https://www.energy.gov/sites/default/files/2022-05/2020%20Smart%20Grid%20System%20Report\\_0.pdf](https://www.energy.gov/sites/default/files/2022-05/2020%20Smart%20Grid%20System%20Report_0.pdf)
- [3] Markets and Data, "United states smart grid market assessment, opportunities, and forecast, 2016-2030," 2025, [Accessed: March 8, 2025]. [Online]. Available: <https://www.marketsanddata.com/industry-reports/united-states-smart-grid-market>
- [4] IMARC Group, "Uk smart grid market: Industry trends, share, size, growth, opportunity and forecast," 2025, [Accessed: March 8, 2025]. [Online]. Available: <https://www.imarcgroup.com/uk-smart-grid-market>
- [5] European Commission, "Smart grids and meters," 2025, [Accessed: March 8, 2025]. [Online]. Available: [https://energy.ec.europa.eu/topics/markets-and-consumers/smart-grids-and-meters\\_en](https://energy.ec.europa.eu/topics/markets-and-consumers/smart-grids-and-meters_en)
- [6] P. Kumar, Y. Lin, G. Bai, A. Paverd, J. S. Dong, and A. Martin, "Smart grid metering networks: A survey on security, privacy and open research issues," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2886–2927, 2019.
- [7] F. Siddiqui, S. Zeadally, C. Alcaraz, and S. Galvao, "Smart grid privacy: Issues and solutions," in *2012 21st International Conference on Computer Communications and Networks (ICCCN)*. IEEE, 2012, pp. 1–5.
- [8] M. R. Asghar, G. Dán, D. Miorandi, and I. Chlamtac, "Smart meter data privacy: A survey," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2820–2835, 2017.
- [9] V. Kayalvizhy and A. Banumathi, "A survey on cyber security attacks and countermeasures in smart grid metering network," in *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*. IEEE, 2021, pp. 160–165.
- [10] A. A. Ansari and D. Giribabu, "State of art and comprehensive study on smart meter networking," *Flexible Electronics for Electric Vehicles: Select Proceedings of FlexEV—2021*, pp. 377–393, 2022.
- [11] J. Kua, M. B. Hossain, I. Natgunanathan, and Y. Xiang, "Privacy preservation in smart meters: current status, challenges and future directions," *Sensors*, vol. 23, no. 7, p. 3697, 2023.
- [12] S. Sultan, "Privacy-preserving metering in smart grid for billing, operational metering, and incentive-based schemes: A survey," *Computers & Security*, vol. 84, pp. 148–165, 2019.
- [13] M. Jawurek, M. Johns, and F. Kerschbaum, "Plug-in privacy for smart metering billing," in *International Symposium on Privacy Enhancing Technologies Symposium*. Springer, 2011, pp. 192–210.
- [14] A. Molina-Markham, P. Shenoy, K. Fu, E. Cecchet, and D. Irwin, "Private memoirs of a smart meter," in *Proceedings of the 2nd ACM workshop on embedded sensing systems for energy-efficiency in building*, 2010, pp. 61–66.
- [15] A. Rial and G. Danezis, "Privacy-preserving smart metering," in *Proceedings of the 10th annual ACM workshop on Privacy in the electronic society*, 2011, pp. 49–60.
- [16] J. Groth, "Non-interactive zero-knowledge arguments for voting," in *Applied Cryptography and Network Security: Third International Conference, ACNS 2005, New York, NY, USA, June 7–10, 2005. Proceedings* 3. Springer, 2005, pp. 467–482.
- [17] A. Rial, G. Danezis, and M. Kohlweiss, "Privacy-preserving smart metering revisited," *International Journal of Information Security*, vol. 17, pp. 1–31, 2018.
- [18] T. Eccles and B. Halak, "Performance analysis of secure and private billing protocols for smart metering," *Cryptography*, vol. 1, no. 3, p. 20, 2017.
- [19] R. Petric, "A privacy-preserving concept for smart grids," *Sicherheit in vernetzen Systemen*, vol. 18, pp. B1–B14, 2010.
- [20] R. Ben Romdhane, H. Hammami, M. Hamdi, and T.-H. Kim, "An efficient and privacy-preserving billing protocol for smart metering," in *International Conference on Advanced Information Networking and Applications*. Springer, 2021, pp. 691–702.
- [21] X.-Y. Zhang, S. Kuenzel, J.-R. Córdoba-Pachón, and C. Watkins, "Privacy-functionality trade-off: A privacy-preserving multi-channel smart metering system," *Energy*, vol. 13, no. 12, p. 3221, 2020.
- [22] J.-M. Bohli, C. Sorge, and O. Ugus, "A privacy model for smart metering," in *2010 IEEE International Conference on Communications Workshops*. IEEE, 2010, pp. 1–5.

- [23] H. Wang, Y. Gong, Y. Ding, S. Tang, and Y. Wang, “Privacy-preserving data aggregation with dynamic billing in fog-based smart grid,” *Applied Sciences*, vol. 13, no. 2, p. 748, 2023.
- [24] W. Xu, J. Sun, R. Cardell-Oliver, A. Mian, and J. B. Hong, “A privacy-preserving framework using homomorphic encryption for smart metering systems,” *Sensors*, vol. 23, no. 10, p. 4746, 2023.
- [25] C. Efthymiou and G. Kalogridis, “Smart grid privacy via anonymization of smart metering data,” in *2010 first IEEE international conference on smart grid communications*. IEEE, 2010, pp. 238–243.
- [26] M. Ababneh, K. Kolachala, and R. Vishwanathan, “Private and secure smart meter billing,” in *Proceedings of the 8th ACM on Cyber-Physical System Security Workshop*, 2022, pp. 15–25.
- [27] H. Li, X. Li, and Q. Cheng, “A fine-grained privacy protection data aggregation scheme for outsourcing smart grid,” *Frontiers of Computer Science*, vol. 17, no. 3, p. 173806, 2023.
- [28] M. Nabil, M. Ismail, M. M. Mahmoud, W. Alasmary, and E. Serpedin, “Ppetd: Privacy-preserving electricity theft detection scheme with load monitoring and billing for ami networks,” *Ieee Access*, vol. 7, pp. 96 334–96 348, 2019.
- [29] A. Abidin, A. Aly, S. Cleemput, and M. A. Mustafa, “Secure and privacy-friendly local electricity trading and billing in smart grid,” *arXiv preprint arXiv:1801.08354*, 2018.
- [30] Wikipedia contributors, “Jensen–shannon divergence,” 2025, accessed: April 4, 2025]. [Online]. Available: [https://en.wikipedia.org/wiki/Jensen%E2%80%93Shannon\\_divergence](https://en.wikipedia.org/wiki/Jensen%E2%80%93Shannon_divergence)
- [31] Python Package Index, “python-secrets,” 2025, accessed: April 4, 2025]. [Online]. Available: <https://pypi.org/project/python-secrets/>
- [32] ———, “randomgen,” 2025, accessed: April 4, 2025]. [Online]. Available: <https://pypi.org/project/randomgen/>
- [33] ———, “numpy,” 2025, accessed: April 4, 2025]. [Online]. Available: <https://pypi.org/project/numpy/>
- [34] M. Cenky, J. Bendík, B. Cintula, P. Janiga, Ž. Eleschová, and A. Beláň, “Dataset of 15-minute values of active and reactive power consumption of 1000 households during single year,” *Data in Brief*, vol. 50, p. 109588, 2023.
- [35] K.-H. Chang and B. Mason, “The ieee 802.15. 4g standard for smart metering utility networks,” in *2012 IEEE Third international conference on smart grid communications (SmartGridComm)*. IEEE, 2012, pp. 476–480.
- [36] N. Ahmed, H. Rahman, and M. I. Hussain, “A comparison of 802.11 ah and 802.15. 4 for iot,” *Ict Express*, vol. 2, no. 3, pp. 100–102, 2016.
- [37] J. Lin, “Divergence measures based on the shannon entropy,” *IEEE Transactions on Information theory*, vol. 37, no. 1, pp. 145–151, 1991.
- [38] I. Wagner and D. Eckhoff, “Technical privacy metrics: a systematic survey,” *ACM Computing Surveys (Csur)*, vol. 51, no. 3, pp. 1–38, 2018.