Privacy-preserving and Accountable Billing in Peer-to-Peer Energy Trading Markets with Homomorphic Encryption and Blockchain

Kamil Erdayandi[a], Lucas C. Cordeiro[a] and Mustafa A. Mustafa[a,b]

[a]Department of Computer Science, The University of Manchester, Oxford Road, Manchester, M13 9PL, United Kingdom
[b]COSIC, KU Leuven, Leuven, 3001, Belgium

ARTICLE INFO

Keywords:
Billing
Privacy
Accountability
Peer-to-peer Energy Market
Homomorphic Encryption
Blockchain

ABSTRACT

This paper proposes a novel Privacy-preserving and Accountable Billing (PA-Bill) protocol for peer-to-peer energy trading markets. It addresses the challenges of discrepancies between committed and delivered energy volumes, ensuring accurate billing, privacy, and accountability. PA-Bill employs a universal cost-splitting mechanism to enhance fairness and prevent indirect privacy leakage. The protocol leverages homomorphic encryption to protect user data and uses blockchain technology to maintain accountability through an immutable and transparent distributed ledger. Additionally, it includes a dispute resolution mechanism to rectify erroneous bill calculations and identify responsible parties, thus ensuring non-repudiation. Our experimental and theoretical evaluations demonstrates that PA-Bill effectively supports large communities of up to 2000 households, offering a computationally efficient, privacy-preserving, and accountable billing solution in a semi-decentralised manner.

1. Introduction

1.1. Motivation and Background

Peer-to-Peer (P2P) energy trading is a promising solution to help us achieve our Net-Zero goals, as it enables users to obtain clean energy at more reasonable prices than the prices offered by traditional suppliers, making it accessible to a wider society [1]. It facilitates direct energy exchange between households that harness Renewable Energy Sources (RES) [2]. This approach empowers individuals to become active participants in the energy system [3], allowing RES owners to optimise their profits and reduce their bills through trading with other users [4]. The emergence of P2P markets could play a significant role in advancing the adoption of renewable energy, leading to improvements in grid reliability and efficiency, and driving reductions in greenhouse gas emissions [5]. These markets could be instrumental in fostering a climate-resilient environment by promoting the integration of RES [6], thereby benefiting RES owners and contributing to a more sustainable energy future [7].

Critical activities within P2P markets encompass bid generation, market clearance, and billing and settlement mechanisms, with the latter responsible for managing payments and tracking transactions [8]. They also contribute to the creation of trading platforms that encourage renewable energy owners to participate in P2P markets, thereby promoting the use of environmentally friendly energy [9]. However, challenges related to privacy, accountability, and variations between actual and committed energy could hinder the widespread adoption of advanced billing mechanisms in P2P energy trading markets.

This work was supported in part by the EPSRC through the projects EnnCore EP/T026995/1 and SCorCH EP/V000497/1 and by the Flemish Government through the FWO SBO project SNIPPET S007619. K.E is funded by The Ministry of National Education, Republic of Turkey.

kamil.erdayandi@manchester.ac.uk (K. Erdayandi);
lucas.cordeiro@manchester.ac.uk (L.C. Cordeiro);
mustafa.mustafa@manchester.ac.uk (M.A. Mustafa)

While data is essential for any market solution [10], privacy risks are a significant challenge in P2P markets due to the intensive collection, processing, and exchange of real-time data by smart grid equipment [8]. This data-centric environment poses threats to sensitive user information [11], potentially exposing energy consumption habits and lifestyle trends [12]. Correlating participants' energy offers and bids with their consumption data further escalates privacy risks [13]. For instance, revealing a prosumer's energy trading activity could disclose their presence at home [14]. Deeper insights into consumption data, like identifying appliance usage or usage patterns, enable behavioural deductions [15]. The exposure of such data in P2P markets heightens the risk of sensitive information disclosure [16].

The unprotected status of trading information in P2P markets significantly jeopardises participant privacy [17], violating protective measures mandated by regulations such as the General Data Protection Regulation (GDPR) [18] and California Consumer Privacy Act (CCPA) [19]. Secure processing and exchange of data are essential to prevent unauthorised access [20]. Therefore, deploying privacy-preserving mechanisms could be beneficial for providing confidence in trading platforms within smart grids, potentially promoting broader acceptance [21].

In addition, in P2P markets, participants typically submit bids before the actual trading periods, requiring them to predict the volume of energy they will need to trade based on historical data and estimated consumption, which is not always accurate [22]. Sometimes, due to prediction errors or other factors like intermittent RES output, participants commit to trading certain volumes of energy but then fail to fulfil these commitments [23]. These deviations can lead to disruptions in grid stability and increase grid balancing costs, potentially passed on to users [23]. Therefore, to minimise these costs, it is important for billing models in P2P markets to incentivise users to engage with P2P markets to handle deviations [24].

Nomenclature

Abbreviations

CCPA	California Consumer Privacy Act
CPA	Chosen-Plaintext Attack
DLT	Distributed Ledger/Blockchain
EC	Erroneous Calculation
ET	Execution Time
GDPR	General Data Protection Regulation
HE	Homomorphic Encryption
HEM	Home Energy Manage System
P2P	Peer-to-Peer
PA-Bill	Privacy-preserving and Accountable Billing
PoS	Proof-of-Stake
PoW	Proof-of-Work
RES	Renewable Energy Sources
SC	Settlement Cycle
SM	Smart Meter
TP	Trading Platform

Notations/Symbols

$TotRev_P^{Dev}$	Total Revenue from prosumers' excess energy due to deviations
π_{FiT}	Feed-in-Tariff price
π_{P2P}	P2P price
π_{RT}	Retail price
$\mathbb{Z}$	Set of integers
$\mathbb{Z}^*_{n^2}$	Set of integers relatively prime to n^2
$\mathbb{Z}^*_n$	Set of integers relatively prime to n
$\{.\}_{\mathcal{E}}$	Data homomorphically encrypted with PK_{sup}
c_i, p_j, u_k	The i-th consumer, j-th prosumer, and k-th user
$lcm(.)$	Least common multiple function
N^{BP}	Number of billing periods
N^{SC}	Number of settlement cycles
N_C, N_P	Number of consumers, prosumers
N_U, N_{MU}	Number of users, matched users
$Stat$	Array of user statements
$Stat^{Tot}$	Array of accumulated statements
V^{P2P}	P2P market's traded electricity volume array
V^{Real}	Real electricity consumption array
Bal_{sup}	Balances of the supplier
Dev^{Tot}	Total deviations of the users
H(.)	Hash Function
inDev	Array of individual user deviations
$KGen_{pe}(k)$	Paillier key generation method for user k
PK_{sup}, SK_{sup}	Public and Private (Secret) key pair of Supplier

Units

KB	kilobytes
ms	milliseconds
s	seconds

Lastly, it is important that any disagreements stemming from incorrect billing be resolved responsibly, ensuring that no party can evade their responsibility [25]. This entails guaranteeing the accuracy and auditability of transaction calculations, with privacy mechanisms aimed at *"accountability"*. In other words, these mechanisms should not only protect data privacy but also enable a clear audit trail, allowing stakeholders to track, verify, and authenticate transactions confidently [26]. By meticulously designing such measures, energy trading platforms can strike a balance, protecting user privacy while ensuring the integrity and verifiability of data [27], ensuring that every transaction remains auditable [28], and ultimately *"accountable"*.

In summary, for an effective billing mechanism incentivising users with RES to participate in P2P markets, designed solutions need to prioritise user privacy preservation, manage user deviations, and ensure accountability.

1.2. Relevant Literature

Within P2P energy trading, two crucial phases are market clearance and billing & settlement [29]. Since privacy-preserving market clearance mechanisms have been widely studied, utilising techniques such as game theory for market clearance and homomorphic encryption for privacy protection (as in [4]), as well , auction theory for market clearance, and multi-party computation for privacy preservation (as in [30]), this paper focuses on the billing phase.

Next, we review the state-of-the-art billing mechanisms used in P2P energy markets based on three key aspects. These aspects are privacy, handling user deviations, and accountability. We have selected them based on their critical importance to the functioning of P2P energy markets as mentioned in the Motivation and Background section. Each aspect addresses a specific challenge within these markets:

> **Privacy** protects user data and fosters trust, which is essential for market adoption.
>
> **Handling User Deviations** ensures fairness in user billing.
>
> **Accountability** ensures trust in the billing process, reducing the risk of disputes and providing accuracy and auditability of transaction calculations.

Madhusudan et al. [24] propose various billing models for P2P energy markets which account for deviations in energy volumes from the users' bids and incorporate individual, social, or universal cost-sharing mechanisms to ensure cost-effectiveness for both consumers and prosumers. Nonetheless, they do not explore user privacy. Furthermore, the

proposed models have only been tested on a limited-scale community.

In the study by Abidin et al. [31], a privacy-preserving auction-based market clearance mechanism employing multi-party computation ensures privacy in energy trading within smart grids. Moreover, they incorporate a privacy-friendly billing mechanism that allows suppliers to compute their customers' monthly bills without revealing any sensitive data. Alabdulatif et al. [32] introduce a novel approach for privacy-preserving cloud-based billing in sensor-enabled smart grid infrastructure. Their method employs lightweight homomorphic encryption to perform billing calculations over encrypted data in the cloud, ensuring user privacy. Despite these advancements, neither Abidin et al. [31] nor Alabdulatif et al. [32] address the issue of discrepancies between committed and actual energy consumption by users.

Among the blockchain solutions, Singh et al. [33] propose a method that uses homomorphic schemes to protect the confidentiality of user data while enabling efficient data analysis. However, they do not explore billing mechanisms effectively. Gür et al. [34] propose a system based on blockchain technology and IoT devices to facilitate billing. To ensure data confidentiality, the system employs session keys and stores the encrypted data on the blockchain. However, this is still vulnerable to breaches as unauthorised parties can gain access to these keys, enabling them to access sensitive data.

A privacy-preserving billing protocol that incorporates an individual cost-sharing mechanism has been proposed in [12]. Homomorphic encryption has been used to provide user privacy. Their solution also includes financial penalties to encourage accurate energy consumption predictions. However, to incentivise the users, the billing mechanism could also explore social or universal cost-splitting mechanisms rather than only investigating individual cost-sharing mechanisms. Moreover, establishing accountability for such discrepancies without compromising sensitive information remains a challenge, indicating a lack of an accountable mechanism in the proposed system.

Alqahtani et al. [22] introduce a zone-based privacy-preserving billing protocol tailored for local energy markets. This protocol handles energy volume deviations of market participants from their bids by incorporating their locations on the grid to distribute the deviation costs. Employing multi-party computation, their billing model ensures privacy while maintaining accuracy and auditability. Similarly, Hutu et al. [26] propose a privacy-preserving billing and settlements protocol designed for local energy markets with imperfect bid-offer fulfilment. Their approach utilises homomorphically encrypted versions of users' data to provide privacy. However, neither Alqahtani et al. [22] nor Hutu et al. [26] integrate accountability mechanisms to guarantee the precision and auditability of transaction calculations, thereby ensuring the non-repudiation, integrity and verifiability of data, especially in scenarios involving erroneous calculations.

Table 1 provides a comparison of how the studies in relevant literature address key aspects such as privacy handling, user deviations, and accountability within P2P market billing systems. This comparison reveals that no other study[1] in the field of P2P market billing fully meets the three aspects: protecting user privacy, accommodating deviations in user consumption from their commitments, and maintaining strong system accountability. Neglecting any of these elements undermines the market trust, fairness and transparency. Furthermore, integrating these three features within a single platform poses considerable challenges.

Table 1
Billing Mechanisms in P2P Markets

Paper	Privacy	Handling User Dev.	Accountability
Madhusudan et al. [24]	✗	✓	✗
Abidin et al. [31]	✓	✗	✗
Alabdulatif et al. [32]	✓	✗	✗
Singh et al. [33]	✓	✗	✗
Gür et al. [34]	✓	✗	✗
Thandi et al. [12]	✓	✓	✗
Alqahtani et al. [22]	✓	✓	✗
Hutu et al. [26]	✓	✓	✗
This work	✓	✓	✓

1.3. Contributions

To address the limitations and challenges raised in the existing literature, we propose a novel Privacy-preserving and Accountable Billing (PA-Bill) protocol, which mitigates the challenges surrounding security, privacy, accountability, and user consumption variations prevalent in current studies. Specifically, the novel contributions of this work are:

- We design a novel privacy-preserving, fair and accountable billing protocol, named PA-Bill, for use in P2P energy trading markets. PA-Bill utilises a universal cost-splitting billing model that mitigates the risk of sensitive information leakage due to individual deviations. To preserve privacy, it employs homomorphic encryption in bill calculations. Moreover, PA-Bill utilises blockchain technology to integrate accountability mechanisms that address possible conflicts during the billing calculation process. To minimise privacy leakage, only the hashed version of the data is stored on the blockchain.
- We thoroughly evaluate the PA-Bill in terms of privacy and accountability. In addition, we implement and evaluate PA-Bill in terms of computations and communication overheads. We showcase the computational efficiency of PA-Bill for user sets of 1,000 and 2,000, with prosumer ratios of 25%, 50%, and 75%.

Unlike other solutions, PA-Bill integrates privacy protection, accountability, and accommodating user consumption variations into a single solution. To the best of our knowledge, no previous work has successfully implemented a billing model that simultaneously preserves privacy, ensures accountability, and effectively handles discrepancies between

[1] Except the preliminary version of this work [25].

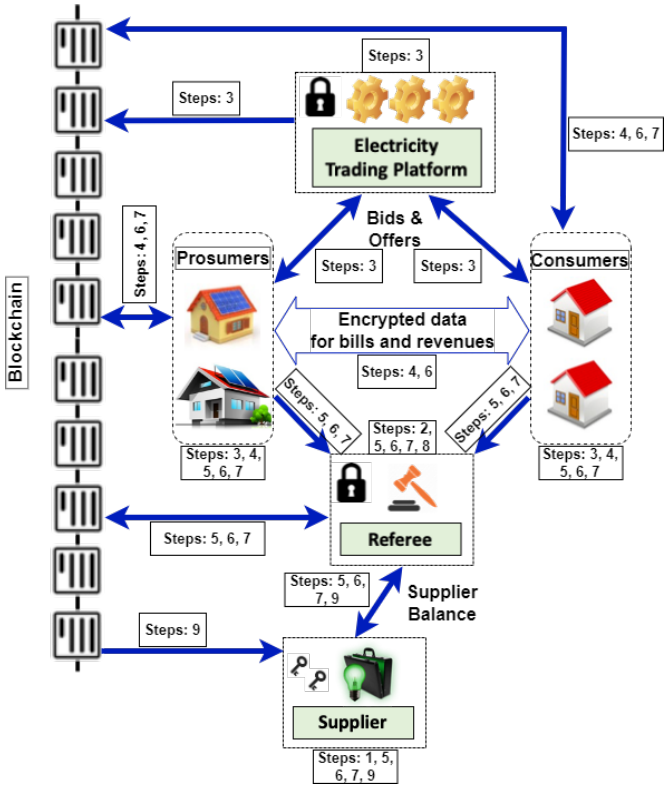

Figure 1: System model.

committed and delivered volume. Through PA-Bill, users can engage in energy trading with confidence, knowing that their privacy is protected, transactions are accurately accounted for, and deviations in consumption are appropriately handled effectively.

Note that this work extends our previous research [25] in (i) broadening the literature review, including recent studies, with a comparative table (Table 1), (ii) conducting comprehensive formal privacy and accountability analyses, (iii) conducting extensive evaluation analysis with a detailed theoretical computation overhead of PA-Bill (Tables 2 and 3), and (iv) conducting expanded simulations for different nummber of users and scenarios.

1.4. Organisation

The rest of the paper is structured as follows: Section 2 outlines the preliminaries. The proposed PA-Bill is presented in Section 3. The PA-Bill is evaluated in terms of privacy, accountability, computational complexity, communication overhead and performance respectively in Section 4. Finally, Section 5 concludes the paper.

2. Preliminaries

2.1. System Model

Our proposed billing protocol, illustrated in Fig. 1, involves prosumers, consumers, a Trading Platform (TP), a Distributed Ledger/Blockchain (DLT), a referee, and a supplier. Prosumers generate energy through renewables, consume the volume they require, and sell any surplus energy. Consumers solely consume energy. Households have Smart Meters (SMs) and Home Energy Manage Systems (HEMs) that measure electricity flows, provide real-time measurements, and facilitate P2P trading for the user. Prosumers and consumers can trade electricity through a P2P market using a TP. The TP is responsible for clearing the P2P market and calculating trading prices for the amount of energy sellers intend to sell. The supplier is responsible for providing energy at a higher price when it cannot be supplied through the P2P market, and for purchasing energy at a lower price when it remains unsold in the P2P market. If necessary, prosumers and consumers can buy or sell electricity from/to a supplier as a backup option. However, P2P trading is more beneficial than relying on the supplier due to pricing considerations [35]. Financial reconciliation occurs during Settlement Cycles (SCs) for users involved in trading. Within each SC, data regarding the actual electricity usage of households and their commitments to trade in the market are stored on DLT. The DLT is an immutable ledger which is used for accountability purposes. Households calculate their bills locally in a decentralised manner. If a dispute arises, the referee is responsible to resolve it by requesting data from households and retrieving it from DLT and identifying the responsible party.

2.2. Assumptions and Threat Model

This section outlines the assumptions and threat model employed in the study.

2.2.1. Assumptions

We assume that SMs are tamper-proof and sealed. Anyone, including their users, cannot tamper with them without being detected [36]. We also assume that the entities communicate over secure and authentic communication channels. Users act rationally by seeking the most cost-effective electricity to buy or sell [37]. Referees are assumed to be regulated in a manner consistent with the regulation of suppliers [38].

2.2.2. Threat Model

In our threat model, no entity has been chosen to be fully trusted with the access to sensitive data, so our threat model comprises untrustworthy and semi-honest entities.

Prosumers and consumers who may attempt to violate the protocol specifications and obtain sensitive data from other users are considered to be untrustworthy. Since billing is directly tied to their roles—prosumers seeking to maximise revenue and consumers aiming to minimise expenses—they have incentives to act maliciously, hence this is the reason why we select them to be untrustworthy parties.

Semi-honest entities, such as the TP, referee, and supplier, are usually regulated and expected to comply with protocol specifications. While they generally follow these protocols, they may still be curious to learn the sensitive data of users.

Such threat models are typical in billing mechanisms [39].

2.3. Design Requirements and Choices

The PA-Bill protocol should satisfy the following design requirements and choices.

- Privacy: The confidentiality of each user's energy trading volume, consumption, deviation amounts, and deviation sign must be provided.
- Accountability: Disputes arising from erroneous bill calculations must be addressed in an accountable way to prevent any party from denying responsibility.
- Fair deviation cost distribution: We choose to distribute the cost of P2P market deviations fairly among all participants. This approach encourages users to actively participate in P2P trading, fostering a more balanced and engaged market.

2.4. Building Blocks

In this study, we utilise the Homomorphic Encryption (HE) cryptosystem to safeguard user privacy by protecting sensitive information. HE is a cryptographic method that enables computations on encrypted data without requiring decryption. The resulting encrypted outputs yield the same results as if the operations were performed on unencrypted data, such that when the results of HE operations are decrypted, they are identical to those obtained without using HE [40]. This approach ensures the privacy of households by homomorphically encrypting sensitive information such as energy consumption data per SC. Billing calculations can be performed on this encrypted data, thereby preserving the confidentiality of the information.

HE can be divided into two subcategories: fully and partially HE. In fully HE, all arithmetic operations are supported, while in partially HE, only a limited number of operations are supported. Computational costs of fully HE is relatively high compared to partially HE [13]. To enhance computational efficiency, we opt for partially HE by employing the Paillier cryptosystem, which allows homomorphic addition and scalar multiplication directly on ciphertexts [41]. The Paillier cryptosystem we utilised is described in Section 2.4.1.

We use blockchain technology to provide accountability by ensuring that transactions are permanently recorded in a decentralised and immutable system with append-only storage. Transactions recorded on a blockchain cannot be altered by design, ensuring that they are accurate and trustworthy [33]. We give brief information about blockchain technology in Section 2.4.2.

2.4.1. Paillier Cryptosystem

The Paillier cryptosystem is structured around three core operations: key generation (KGen_{pe}), encryption (Enc_{pe}), and decryption (Dec_{pe}), extensively discussed in [41, 42].

- $\mathsf{KGen}_{pe}(k) \rightarrow (PK, SK)$: This key generation algorithm accepts a security parameter k and outputs a pair of keys, a public key PK and a secret key SK. It begins by selecting two large prime numbers p and q, calculates $n = p \cdot q$, and $\lambda = \mathrm{lcm}(p-1, q-1)$. A generator g within $\mathbb{Z}^*_{n^2}$ is then chosen, followed by the computation of $\mu = (L(g^\lambda \mod n^2))^{-1} \mod n$, where $L(x) = (x-1)/n$. The process yields $PK = (n, g)$ and $SK = (\lambda, \mu)$.
- $\mathsf{Enc}_{pe}(PK, m) \rightarrow c$: The encryption mechanism takes a plaintext message $m \in \mathbb{Z}$ along with the public key $PK = (n, g)$, producing a ciphertext c. It involves selecting a random value $r \in \mathbb{Z}^*_n$ and calculating $c = g^m \cdot r^n \mod n^2$.
- $\mathsf{Dec}_{pe}(SK, c) \rightarrow m$: Utilising the secret key $SK = (\lambda, \mu)$ and a ciphertext c, this decryption formula outputs the original message m. It does so by computing $m = L(c^\lambda \mod n^2) \cdot \mu \mod n$.

The Paillier cryptosystem allows for certain arithmetic operations on encrypted data. Given data m encrypted with $\mathsf{Enc}_{pe}(PK, m)$, denoted as $\{m\}_{\mathcal{E}}$, which represents the data homomorphically encrypted with the public key PK, the supported operations over homomorphically encrypted data include addition of encrypted data and scalar multiplication.

2.4.2. Blockchain Technology

Blockchain technology provides a secure and transparent method for conducting transactions without the need for intermediaries [43]. Its cornerstone feature, immutability, guarantees the integrity of transaction records across the network. Through consensus mechanisms, blockchain achieves agreement among all nodes on the validity of transactions. Adaptations of blockchain technology, including public, private, and hybrid models, cater to diverse application needs [44].

Focusing on the Ethereum blockchain [45], our study examines its deployment of smart contracts [46]. These automated contracts execute agreements based on predefined conditions, minimising the need for manual oversight. Ethereum's transition from Proof-of-Work (PoW) to Proof-of-Stake (PoS) marks a significant advancement in energy efficiency, performance, and security [47]. As a public blockchain, Ethereum promotes transparency and trust, enabling open participation [48]. This aligns with the principles of transparency and inclusivity discussed by Andoni et al. [43]. Both the transparency and immutability features of Ethereum blockchain technology in PA-Bill ensure precise and auditable transaction calculations to provide *accountability*. This feature guarantees data integrity, non-repudiation, and verifiability, which is crucial for addressing errors in transaction computations.

3. Privacy-preserving and Accountable Billing (PA-Bill) Protocol

In this section, we propose a privacy-preserving and accountable billing protocol for P2P energy market where users' actual energy consumption may differ from the volumes they committed. It protects sensitive household information and enables system entities to verify accurate billing calculations. We first provide an overview of the protocol before providing specific technical details.

3.1. PA-Bill Overview

The process of the PA-Bill protocol is illustrated in Fig. 1, which includes interactions between the entities. The system utilises the supplier's public-private key pair for all

homomorphically encrypted calculations. A distinct set of HE keys, namely PK_{sup} and SK_{sup}, are generated for each billing month. Additionally, each month, the consumers and prosumers are paired together to perform accountable calculations.

In the energy trading model, users send homomorphically encrypted bid-offer data to the TP, which calculates the final P2P trading price and the encrypted energy trade for each user in the P2P market, as in [49].

During each SC, the P2P trading price is publicly released. The encrypted energy trade is shared with related paired users for future calculations, and its hash is stored on the DLT for future verification. SMs measure their users' actual imported/exported electricity and transmit their encrypted versions to relevant users. The hash of this encrypted version is also stored on the DLT.

After sending and storing related data for billing, the calculation of bills among prosumers and consumers is performed in the following steps in a privacy-preserving way: initially, individual deviations of users are calculated. Consumers calculate the individual deviations of prosumers and vice versa. Next, the total deviations of consumers and prosumers are calculated by six users selected from consumers and prosumers. Ultimately, users' statements (bills/revenues) are calculated.

To protect sensitive data such as energy consumed/traded and individual energy deviations of households, our work utilises the HE scheme to process data while preserving privacy. However, it is crucial to design the billing algorithm in such a way that it avoids any indirect leakage of private information despite the use of encryption. Traditional billing methods [12] have the potential to expose confidential information by using individual deviations between actual and committed energy volumes to determine the "conditions" in calculating bills. This enables inferences to be made about whether the actual electricity consumption volume is lower or higher than the committed data. To address this issue, we propose a privacy-preserving and accountable cost-splitting billing that uses total deviations of consumers and prosumers rather than individual deviations to determine billing conditions.

In the event of a dispute, the referee requests the necessary data from households and retrieves the hash of the previously stored data from DLT (to ensure the accuracy of the data requested from households) to settle the dispute. In this case, the referee corrects erroneous computations of the pair of customer and prosumer whose calculations do not match each other and identifies the responsible party in the pair. The responsible party is penalised, incentivising them to act truthfully, which would otherwise result in penalties. Besides, the referee can directly calculate the supplier's balance since the calculations do not involve any confidential information.

Finally, at the end of the month, final bills, revenues, and the balance of the supplier are released with the help of the referee and the private homomorphic key of the supplier.

3.2. Technical Details of PA-Bill

The PA-Bill comprises distinct phases, each with varying frequencies of occurrence. While each phase consists of its own steps, the entire PA-Bill process is comprised of nine main steps, labelled in Figure 1 to clarify the system dynamics. The first phase starts with pre-arrangements for billing calculations, which take place every billing period (typically monthly). During this phase, steps 1-2 are executed, involving the generation of HE key pairs and the pairing of consumers and prosumers for billing calculations. The second phase occurs at each SC, where essential data for billing calculations is exchanged in steps 3-4, followed by the actual billing computations in steps 5-8. The final phase occurs at the conclusion of each billing period, in step 9. Here, final bills, revenues, and supplier balances are transferred and announced, completing the PA-Bill process. Each step is detailed as follows:

Step 1: Generation of Keys

The supplier generates a public-private HE (Paillier) key pair: $KGen_{pe}(k) \rightarrow PK_{sup}, SK_{sup}$.

Step 2: Matching consumers and prosumers

The referee conducts a random matching process in which each consumer is paired with a list of prosumers and vice versa. The number of users in the lists may exceed one in cases where $N_C > N_P$ or $N_C < N_P$, while the lists contain only one user if $N_C = N_P$, where N_C and N_P denote the respective number of consumers and prosumers. The function $M(u_k)$ returns the list of users that have been matched to the user u_k.

Step 3: Transfer and Storage of P2P Traded Data

TP makes the P2P trading price, π_{P2P}, publicly available by storing it at DLT in plaintext. For each u_k, TP transmits homomorphically encrypted value of traded volume $\left\{V^{P2P}[u_k]\right\}_{\mathcal{E}}$ to user u_k and to users in $M(u_k)$. The privacy-preserving calculation of the encrypted traded value for each user u_k ($\left\{V^{P2P}[u_k]\right\}_{\mathcal{E}}$) can be performed with the transmission of bids-offers by users in a homomorphically encrypted format. However it is assumed the TP has already calculated $\left\{V^{P2P}[u_k]\right\}_{\mathcal{E}}$. Once this data has been transmitted to relevant parties, the TP also hashes the homomorphically encrypted traded volume of user u_k, i.e., $H(\left\{V^{P2P}[u_k]\right\}_{\mathcal{E}})$, and stores the result at the DLT.

Step 4: Collection, Transfer and Storage of SM Data

At the end of each SC, each SM measures the real volume of energy imported from (or exported to) the grid by their user, i.e., $V^{Real}[u_k]$, encrypts it with supplier public key, PK_{sup} and hashes it, i.e., $H(\left\{V^{Real}[u_k]\right\}_{\mathcal{E}})$. It then stores the hash value to DLT. The user SM also stores $\left\{V^{Real}[u_k]\right\}_{\mathcal{E}}$ itself as well as sends it to the users in $M(u_k)$.

Step 5: Calculation of Individual Deviations

In this step, each user u_k calculates the individual deviations ($inDev$) from the volume of energy they committed for themselves and their corresponding matched users in $M(u_k)$

Algorithm 1: Calculating Individual Deviations

Input: N_U
Output: $\{inDev\}_{\mathcal{E}}, \{inDev_M\}_{\mathcal{E}}$
for *each* u_k **do**
  $\{inDev[u_k]\}_{\mathcal{E}} \leftarrow \{V^{Real}[u_k]\}_{\mathcal{E}} - \{V^{P2P}[u_k]\}_{\mathcal{E}}$;
  for *each* m_l **in** $M(u_k)$ **do**
    $\{inDev_M[M(m_l)]\}_{\mathcal{E}} \leftarrow \{V^{Real}[M(m_l)]\}_{\mathcal{E}} - \{V^{P2P}[M(m_l)]\}_{\mathcal{E}}$;
  end
end

(see Alg. 1). To calculate $inDev$, each user u_k subtracts their committed volume from the volume measured by their SM for themselves (u_k) and the users m_l in $M(u_k)$. The calculations are carried out in a homomorphically encrypted format. The encrypted results $\{inDev\}_{\mathcal{E}}$ and $\{inDev_M\}_{\mathcal{E}}$ are sent to the referee.

After the referee receives the encrypted individual deviations from users, it checks whether the computations have been done correctly. For each user and its matched user, the referee receives four encrypted results. The user u_k provides its own encrypted result, $\{inDev[u_k]\}_{\mathcal{E}}$, as well as that of its matched user. For the matched consumer c_i and prosumer p_j, the referee checks if the calculated values for the specific users are the same. To accomplish this, the referee employs a method that subtracts the calculated values of specific users from each other within a pair. These values include the user's own calculated value and the value calculated by its counterpart, all conducted within a homomorphically encrypted format. Two subtractions are carried out for a pair. The result of these two subtractions is then sent to the supplier, who has the private key to perform homomorphic encryption operations. The supplier decrypts the result of subtraction and sends it back to the referee. The referee checks whether the received value from the supplier is zero or not. If it is zero, it considers the calculations to be accurate and proceeds to store the hash of the resulting computation of user u_k (not that of the matched user) in DLT to facilitate future verification. Otherwise (if the received result is not zero), the referee intervenes to correct any erroneous calculations and identify the responsible party. To do so, the referee requests $\{V^{Real}\}_{\mathcal{E}}$ and $\{V^{P2P}\}_{\mathcal{E}}$ from the users, checks their correctness by hashing and comparing them with the previously stored hashes in blockchain by TP and SMs. If the encrypted data received from the users is accurate, the referee recalculates the inDev in encrypted format for c_i and p_j, whose results were incorrect. Next, the referee follows the same process of subtracting the four previously calculated values from the one calculated by the referee and having the result decrypted by the supplier to compare the recalculated outcome with the values obtained from c_i and p_j. The referee then identifies the party accountable for the mismatch.

Step 6: Calculation of Total Deviations

To calculate total demand and supply deviations, the referee selects three consumers and three prosumers. A size of three selections for consumers and prosumers provides redundancy. By choosing more than one consumer and a prosumer representative, the system ensures that no single entity can manipulate the results. If only one consumer and one prosumer were selected, the risk of collusion or errors affecting the overall result would be higher. In contrast, selecting a much larger number of representatives for consumers and prosumers would increase the computational and communication overhead. This would slow down the process and require more resources. Therefore, selecting three representatives from both consumers and prosumers strikes a balance between enhancing redundancy to reduce errors and collusion and minimising the use of computational and communication resources.

Each consumer c_i sends their respective $\{inDev[c_i]\}_{\mathcal{E}}$ to the selected prosumers and vice versa. Selected prosumers and consumers verify the received encrypted deviations by hashing and comparing them with stored hashes in DLT. Then, selected prosumers sum up $\{inDev[c_i]\}_{\mathcal{E}}$ for each c_i to calculate $\{Dev_C^{Tot}\}_{\mathcal{E}}$ (eq. 1) and selected consumers do the same for each p_i, (eq. 2).

$$\{Dev_C^{Tot}\}_{\mathcal{E}} \leftarrow \sum_{i=0}^{N_C-1} \{inDev_C[c_i]\}_{\mathcal{E}} \tag{1}$$

$$\{Dev_P^{Tot}\}_{\mathcal{E}} \leftarrow \sum_{j=0}^{N_C-1} \{inDev_P[p_j]\}_{\mathcal{E}} \tag{2}$$

After calculating $\{Dev_C^{Tot}\}_{\mathcal{E}}$ and $\{Dev_P^{Tot}\}_{\mathcal{E}}$, selected prosumers and consumers send them to a referee for verification. If the results match, the referee sends them to the supplier. The supplier then decrypts the results and makes them publicly available by storing Dev_C^{Tot} and Dev_P^{Tot} into DLT. If the results do not match, the referee corrects any erroneous calculations and identifies the responsible party. This is done by recalculating (eq. 1) and (eq. 2) in an encrypted format and subtracting the recalculated values from those calculated by selected users. In total, six subtractions are conducted. The results of the subtractions are decyrpted by the supplier.

Step 7: Calculation of Bills and Rewards

We present our proposed privacy-preserving and accountable universal cost-splitting billing model that employs total deviations instead of individual deviations to establish billing conditions. The proposed billing model is presented in Alg. 2. The algorithm takes as input $N_U, \{V^{P2P}\}_{\mathcal{E}}, Dev_C^{Tot}, Dev_P^{Tot}, \pi_{P2P}, \pi_{RT}, \pi_{FiT}$ and calculates the bills/revenues of consumers/prosumers. The algorithm outputs Statements $Stat[u_k]$, $Stat_M[u_k]$ for user u_k and its matched users in $M(u_k)$, respectively. $Stat[u_k]$ indicates the bill of u_k when u_k is a consumer and it stands for the revenue of u_k if u_k is a prosumer. We have devised universal formulas such as $Stat[u_k]$, which is applicable to both consumers and prosumers. The algorithm works in three modes based on the difference

Algorithm 2: Calculating Bills and Revenues

Input: $N_U, \{V^{P2P}\}_{\mathcal{E}}, Dev_C^{Tot}, Dev_P^{Tot}, \pi_{P2P}, \pi_{RT}, \pi_{FiT}$
Output: $\{Stat\}_{\mathcal{E}}, \{Stat_M\}_{\mathcal{E}}, \{Stat^{Tot}\}_{\mathcal{E}}, \{Stat_M^{Tot}\}_{\mathcal{E}}$

1 **for** *each* u_k **do**
2 **if** $Dev_P^{Tot} = Dev_C^{Tot}$ **then**
3 $\{Stat[u_k]\}_{\mathcal{E}} \leftarrow \{V^{P2P}[u_k]\}_{\mathcal{E}} \cdot \pi_{P2P} + \{inDev[u_k]\}_{\mathcal{E}} \cdot \pi_{P2P}$
4 **for** *each* m_l **in** $M(u_k)$ **do**
5 $\{Stat[m_l]\}_{\mathcal{E}} \leftarrow \{V^{P2P}[m_l]\}_{\mathcal{E}} \cdot \pi_{P2P} + \{inDev[m_l]\}_{\mathcal{E}} \cdot \pi_{P2P}$
6 **end**
7 **end**
8 **if** $Dev_P^{Tot} < Dev_C^{Tot}$ **then**
9 $\{Stat[u_k]\}_{\mathcal{E}} \leftarrow \{V^{P2P}[u_k]\}_{\mathcal{E}} \cdot \pi_{P2P} + \{inDev[u_k]\}_{\mathcal{E}} \cdot \pi_{RT}$
10 **for** *each* m_l **in** $M(u_k)$ **do**
11 $\{Stat[m_l]\}_{\mathcal{E}} \leftarrow \{V^{P2P}[m_l]\}_{\mathcal{E}} \cdot \pi_{P2P} + \{inDev[m_l]\}_{\mathcal{E}} \cdot \pi_{RT}$
12 **end**
13 **end**
14 **if** $Dev_P^{Tot} > Dev_C^{Tot} c$ **then**
15 **if** u_k *is a consumer* **then**
16 $\{Stat[u_k]\}_{\mathcal{E}} \leftarrow \{V^{P2P}[u_k]\}_{\mathcal{E}} \cdot \pi_{P2P} + \{inDev[u_k]\}_{\mathcal{E}} \cdot \pi_{P2P}$
17 **for** *each* m_l **in** $M(u_k)$ **do**
18 $\{Stat[m_l]\}_{\mathcal{E}} \leftarrow \{V^{P2P}[m_l]\}_{\mathcal{E}} \cdot \pi_{P2P} + \{inDev[m_l]\}_{\mathcal{E}} / Dev_P^{Tot} \cdot TotRev_P^{Dev}$
19 **end**
20 **else**
21 $\{Stat[u_k]\}_{\mathcal{E}} \leftarrow \{V^{P2P}[u_k]\}_{\mathcal{E}} \cdot \pi_{P2P} + \{inDev[u_k]\}_{\mathcal{E}} / Dev_P^{Tot} \cdot TotRev_P^{Dev}$
22 **for** *each* m_l **in** $M(u_k)$ **do**
23 $\{Stat[m_l]\}_{\mathcal{E}} \leftarrow \{V^{P2P}[m_l]\}_{\mathcal{E}} \cdot \pi_{P2P} + \{inDev[m_l]\}_{\mathcal{E}} \cdot \pi_{P2P}$
24 **end**
25 **end**
26 **end**
27 $\{Stat^{Tot}[u_k]\}_{\mathcal{E}} \leftarrow \{Stat^{Tot}[u_k]\}_{\mathcal{E}} + \{Stat[u_k]\}_{\mathcal{E}}$
28 **for** *each* m_l **in** $M(u_k)$ **do**
29 $\{Stat_M^{Tot}[m_l]\}_{\mathcal{E}} \leftarrow \{Stat^{Tot}[m_l]\}_{\mathcal{E}} + \{Stat[m_l]\}_{\mathcal{E}}$
30 **end**
31 **end**

between total deviations of consumers and prosumers and proceeds as follows.

If $Dev_P^{Tot} = Dev_C^{Tot}$, prosumers have generated enough electricity to meet the demand of customers, resulting in a balanced P2P market. In this case, individuals can purchase the required energy from other households and sell their excess energy to other households at π_{P2P} in addition to their commitments in the P2P market rather than relying on suppliers. Energy sharing between households to compensate for deviations is advantageous for both consumers and prosumers, as they can exchange energy at a price of π_{P2P}, which is higher than π_{FiT} and lower than π_{RT}, compared to relying on suppliers to buy electricity at π_{RT} and sell electricity at π_{FiT}. The statements for each user u_k and for paired users in $M(u_k)$ are calculated between ln. 3-6 in the algorithm.

If $Dev_P^{Tot} < Dev_C^{Tot}$, there is a shortage of electricity in the P2P market as prosumers have not generated enough electricity to meet customer demand. If there is a shortage of electricity that cannot be compensated by other users, the only option is to purchase it from the supplier at π_{RT}. Users with a shortage of electricity can buy it at this price, while households with a surplus can sell it at π_{RT} instead of selling it to the supplier for π_{FiT}, which is advantageous for prosumers. In accordance with this, the statements for each user u_k and for paired users in $M(u_k)$ are calculated between ln. 9-11 in the algorithm.

If $Dev_P^{Tot} > Dev_C^{Tot}$, there is excess electricity in the P2P market as prosumers have generated more electricity than is needed to meet customer demand.

In this scenario, consumers experiencing an electricity shortage due to deviation can purchase the corresponding amount of energy from prosumers' excess energy (from their positive deviations) at the P2P market price (π_{P2P}). Thus, the total revenue for prosumers from selling their corresponding amount of excess electricity to consumers (experiencing an electricity shortage due to deviation) is given by $Dev_C^{Tot} \times \pi_{P2P}$. However, the remaining total excess electricity of prosumers ($Dev_P^{Tot} - Dev_C^{Tot}$) needs to be injected into the grid at the FiT price (π_{FiT}). The revenue from this transaction is given by $(Dev_P^{Tot} - Dev_C^{Tot}) \times \pi_{FiT}$.

As a result, the total revenue generated by prosumers from the excess energy due to deviation ($TotRev_P^{Dev}$) is calculated by summing these two components as:

$$TotRev_P^{Dev} = Dev_C^{Tot} \times \pi_{P2P} + (Dev_P^{Tot} - Dev_C^{Tot}) \times \pi_{FiT} \quad (3)$$

This revenue ($TotRev_P^{Dev}$) is then distributed among the prosumers in proportion to the excess energy they provided (in proportion to $inDev_P[u_k]/Dev_P^{Tot}$). Specifically, each prosumer's share of the total revenue is proportional to their contribution to the total excess energy. Additionally, consumers purchase their needed energy due to deviations at the π_{P2P}, so the calculations for individual deviations are made at this price.

In accordance with this, Alg. 2 calculates statements for each user u_k and for paired users in $M(u_k)$ between ln. 16-19, if u_k is a consumer. Otherwise, the statements are calculated between ln. 21-24.

At the end of the algorithm, statements are accumulated on $stat^{Tot}$ in encrypted format for u_k and user in $M(u_k)$ assuming that $stat^{Tot}$ was set to zero before the first SC.

After each pair calculates their statements bilaterally, they send the results to the referee for verification. If the results do not match, the referee intervenes to correct any erroneous calculations and identify the responsible party. This is done by running Alg. 2 for the unmatched pairs after requesting and verifying the required data for computation via DLT.

Step 8: Calculating the Balance of the Supplier

The referee calculates the supplier's balance using only public information and does so in a non-encrypted format. In the case where $Dev_P^{Tot} = Dev_C^{Tot}$, Bal_{sup} is set to zero ($Bal_{sup} \leftarrow 0$) since there is no excess or shortage of electricity in the P2P market to compensate from the supplier. If

($Dev_P^{Tot} > Dev_C^{Tot}$), there is excess energy in P2P market and the supplier purchases it at FiT price π_{FiT}, resulting in a negative balance for the supplier to pay. Bal_{sup} is calculated as the negative product of the total excess energy ($Dev_P^{Tot} - Dev_C^{Tot}$) and π_{FiT}, i.e.

$$Bal_{sup} \leftarrow -(Dev_P^{Tot} - Dev_C^{Tot}) \cdot \pi_{FiT} \tag{4}$$

If ($Dev_P^{Tot} < Dev_C^{Tot}$), there is a shortage of energy in the P2P market that needs to be compensated by the supplier at retail price π_{RT}. Bal_{sup} is calculated as the product of supplied energy ($Dev_P^{Tot} - Dev_C^{Tot}$) and π_{RT}, i.e.

$$Bal_{sup} \leftarrow (Dev_C^{Tot} - Dev_P^{Tot}) \cdot \pi_{RT}. \tag{5}$$

At each SC, the resulting Bal_{sup} is accumulated to the total supplier balance except when the SC is equal to zero where Bal_{sup}^{Tot} is set to Bal_{sup}.

Step 9: Transfer and Announcement of Bills, Revenues and Supplier Balance

The final accumulated monthly statements of households are not protected from the supplier, as payments must be made for each billing period; the users send encrypted accumulated statements consisting of bills and revenues to the supplier (Note: this transmission is omitted from Figure 1 for clarity). The supplier verifies the correctness of accumulated statement values by hashing them and comparing them with the hashed data in DLT. In addition, the referee sends the supplier balance to the supplier.

The unique HE keys generated to calculate this month's statements are discarded once the accumulated statements are obtained, as these keys serve no further purpose beyond that point. New unique HE keys for the next month are calculated in Step 1.

4. Evaluation

In this section, we demonstrate that PA-Bill provides privacy, accountability, and the ability to accommodate user consumption variations in an effective way. PA-Bill effectively addresses these critical aspects while maintaining a level of computational efficiency. We prove our claims through both theoretical analysis and experiments.

4.1. Privacy and Accountability Analysis

In this subsection, we conduct an analysis of Privacy and Accountability respectively.

4.1.1. Privacy Analysis

The protocol addresses privacy concerns by leveraging HE to encrypt sensitive user data, including energy consumption volumes (V^{Real}) and P2P transaction volumes (V^{P2P}). This encryption ensures that sensitive information remains confidential throughout the billing computation processes.

Energy volume data, specifically V^{Real} and V^{P2P}, are encrypted using the public key, denoted as PK, before being transmitted to any external entities. As the data sender retains exclusive access to the corresponding private key, symbolised as SK, recipients of the encrypted data are unable to determine the specifics of the energy volume data.

Utilising a Chosen-Plaintext Attack (CPA) model, we demonstrate that energy volume data remains confidential.

Proof: Consider PK as the public key utilised for encrypting a user's energy volume data, with SK being the respective private key. An adversary, denoted as $\mathcal{A}$, attempts to get information from the homomorphically encrypted energy volume data $\mathsf{Enc}_{pe}(PK, Volume)$.

1. $\mathcal{A}$ selects a volume value $volume_1$ and acquires its encryption $\mathsf{Enc}_{pe}(PK, volume_1)$. 2. Subsequently, $\mathcal{A}$ faces a challenge involving the encryption of either $volume_1$ or an alternative volume $volume_2$, selected by a challenger. $\mathcal{A}$ is tasked with identifying which volume data underwent encryption.

The Paillier cryptosystem, known for its probabilistic encryption capabilities, ensures that the encryption of identical plaintexts multiple times results in distinct ciphertexts of uniform length. This attribute significantly bolsters the security of the encryption scheme by masking the frequency of specific plaintext values within a dataset, thus hindering cryptanalytic efforts. This principle is extensively detailed in the work of Paillier [41]. Consequently, $\mathcal{A}$ is unable to differentiate $\mathsf{Enc}_{pe}(PK, volume_1)$ from $\mathsf{Enc}_{pe}(PK, volume_2)$, affording no substantial advantage over random guessing. Therefore, the confidentiality of the energy volume data is securely maintained.

In addition, the PA-Bill protocol employs a cost-splitting mechanism that utilises the total deviations of users rather than individual deviations to calculate billing modes. This method avoids indirect privacy leakage of individual deviations. Given an aggregated deviation, Dev_P^{Tot} or Dev_C^{Tot}, an adversary $\mathcal{A}$ cannot deduce the deviation of an individual user, $inDev$, without specific information about all other users, which is practically unattainable. This ensures individual deviations remain confidential, as $inDev$ alone does not provide enough information to reverse-engineer any single $inDev$.

4.1.2. Accountability Analysis

The integration of Blockchain technology within the billing framework ensures the immutability and transparency of essential data records. By creating a secure and unalterable ledger of data hashes necessary for billing computations, the system guarantees both the verification of critical data and the accountability of all involved parties throughout the billing cycle.

Immutability: Once a transaction has been recorded in a block and added to the blockchain, it becomes infeasible to alter. If an adversary, denoted as $\mathcal{A}$, attempts to modify the transaction data in a block, this action would invalidate the hash of the block and all subsequent blocks due to the cryptographic linking. This feature ensures that all essential data recorded on the blockchain remains unchanged post-confirmation.

Transparency and Verification: Given the decentralised nature of the blockchain, all network participants have access to the distributed ledger and can independently verify the hashes of transaction data. This transparency enables any party to validate the integrity of the data used in billing computations, thus fostering trust among all stakeholders.

Accountability Through Blockchain: The immutable record of data hashes on the blockchain ensures that every transaction and action taken by the parties during the billing process is permanently recorded and verifiable. This accountability mechanism holds all parties responsible for their actions, as any attempt to falsify or manipulate data would be detectable and traceable by utilising the permanently stored data on the blockchain.

4.2. Computational Complexity

The time complexity of the method is mainly determined by the input parameters of Alg. 1 and Alg. 2, which include the number of users (N_U). The time required to perform the algorithm grows depending on the input size. Specifically, the nested double loops in Alg. 1 and Alg. 2 lead to quadratic time complexity of $\mathcal{O}(n^2)$ for cases where in cases where $N_C > N_P$ or $N_C < N_P$, the time complexity is reduced to $\mathcal{O}(n)$ with a single iteration in the inner loop when $N_C = N_P$ where each user has only one matched user. The time complexity of the calculations in eq. 1 and eq. 2 is $\mathcal{O}(n)$, where n depends on the inputs N_C and N_P, respectively.

4.3. Communication Overhead

In this study, the data exchanged via communication channels is categorised into three types: homomorphically encrypted, hashed, or plaintext – denoted as "$\{data\}_{\mathcal{E}}$", "$\{data\}_H$", and "data" respectively. The number of bits exchanged in each step of PA-Bills for each transfer direction is theoretically detailed through variables in Table 2 and Table 3 per each settlement cycle and billing period, respectively. The notation |d| denotes the length of "d" in a number of bits where "d" represents a data type. The term "EC" represents cases where there is an Erroneous Calculation requiring referee intervention to identify the responsible party.

Data transfers predominantly occur after the initial phase of PA-Bill (steps 1-2), during which pre-arrangements for billing calculations are established. The primary data exchanges start during the second phase of the PA-Bill, specifically from step 3 onward. Therefore, our focus in this part is directed towards steps occurring from step 3 onward. The communication overhead for steps 3-7, occurring every settlement cycle, is detailed in Table 2, while the communication overhead for step 9, occurring every billing period, is outlined in Table 3.

In step 3, for each SC, TP sends the trading price, denoted as π_{P2P}, to the DLT. Additionally, TP sends the hashed homomorphically encrypted traded volume of each user u_k, represented as $H(\{V^{P2P}[u_k]\}_{\mathcal{E}})$, to DLT. Moreover, TP transfers the homomorphically encrypted value of the traded volume, $\{V^{P2P}[u_k]\}_{\mathcal{E}}$, to both user u_k and users in $M(u_k)$, where $M(u_k)$ represents a set of matched users. It is assumed that TP has already calculated π_{P2P} and $\{V^{P2P}[u_k]\}_{\mathcal{E}}$ for each user u_k via a clearance mechanism, as described in prior work such as [4].

In step 4, for each SC, users send SM measurement data to their matched users in a homomorphically encrypted format. Additionally, they hash this data and send it to the DLT.

In step 5, for SC, each user in a matched couple sends their own and their matched user's individual deviations to the referee after calculating these deviations. Each user within a pair transmits two pieces of homomorphically encrypted data to the referee, resulting in a total of four pieces of data for the pair. The referee subtracts these data for the specific users within the pair in a homomorphically encrypted format and then sends the resulting subtraction for each user in the pair. The supplier returns to the referee with the results of the subtractions for each user in the pair. If there are no errors in these calculations, the final result of each individual calculation is sent to the DLT.

If there is an error in the calculations, the referee requests the values of $\{V^{Real}\}_{\mathcal{E}}$ and $\{V^{P2P}\}_{\mathcal{E}}$ from each user, along with their hashed values from the DLT. After receiving these values, the referee conducts its own calculations and verifies each calculation made by the users (two for each user in a pair) through subtraction. The referee then sends the results of these subtractions to the supplier, who returns the decrypted results of subtraction for verification.

In step 6, for each SC, all users send their individual deviations to the selected three users (from all consumers to selected three prosumers and vice versa) in a homomorphically encrypted format. Also, selected users receive the hashed version of homomorphically encrypted individual deviations from DLT for the purpose of verification. Six selected users from consumers and prosumers, send the total deviation results to the referee. The referee subtracts the results of selected consumers (and prosumers) from each other and sends the results of the six subtractions to the supplier, who decrypts and returns the results. If the results match, the final total deviations for both consumers and prosumers are sent to the DLT and made publicly available. Otherwise, in the erroneous case, the referee does not request the individual deviations in homomorphically encrypted format because the referee has the necessary information from the previous stage for the current SC to recalculate the total deviations and perform subtractions for verification. However, for verification purposes, the differences between recalculated values and the original values for six selected users are sent to the supplier, which decrypts and returns them back to the referee.

In step 7, for each SC, each user retrieves the publicly available data of Dev_C^{Tot}, Dev_P^{Tot}, and π_{P2P} from the DLT. They then calculate their own statement and accumulated statement as well as the statements and accumulated statements of their matched users. Following this, each user within the pair sends their statement and accumulated statement values, along with those of their matches, to the referee in a homomorphically encrypted format. The referee verifies the correctness of the calculations by sending the subtraction of statements and accumulated statements for each matched couple to the supplier, which decrypts the results and sends

Table 2
Communication Overhead per Settlement Cycle

Step	Direction	Number of Bits
Transfer and Storage of P2P Traded Data (Step 3)	TP → DLT TP → Users	$\|data\| + N_U \times \|\{data\}_H\|$ $(N_U + N_{MU}) \times \|\{data\}_{\mathcal{E}}\|$
Collection, Transfer and Storage of SM Data (Step 4)	Users → Matched users Users → DLT	$N_{MU} \times \|\{data\}_{\mathcal{E}}\|$ $N_U \times \|\{data\}_H\|$
Calculation of Individual Deviations (Step 5)	Users → Referee Referee → Supplier Supplier → Referee Referee → DLT	$2 \times N_{MU} \times \|\{data\}_{\mathcal{E}}\|$ $N_{MU} \times \|\{data\}_{\mathcal{E}}\|$ $N_{MU} \times \|data\|$ $N_U \times \|\{data\}_H\|$
Calculation of Individual Deviations – Erroneous Case (Step 5 – EC)	Users → Referee DLT → Referee Referee → Supplier Supplier → Referee	$2 \times N_U \times \|\{data\}_{\mathcal{E}}\|$ $2 \times N_U \times \|\{data\}_H\|$ $2 \times N_{MU} \times \|\{data\}_{\mathcal{E}}\|$ $2 \times N_{MU} \times \|data\|$
Calculation of Total Deviations (Step 6)	Users → Selected Users DLT → Selected Users Selected Users → Referee Referee → Supplier Supplier → Referee Referee → DLT	$3 \times N_U \times \|\{data\}_{\mathcal{E}}\|$ $3 \times N_U \times \|\{data\}_H\|$ $6 \times \|\{data\}_{\mathcal{E}}\|$ $6 \times \|\{data\}_{\mathcal{E}}\|$ $6 \times \|data\|$ $2 \times \|data\|$
Calculation of Total Deviations – Erroneous Case (Step 6 – EC)	Referee → Supplier Supplier → Referee	$6 \times \|\{data\}_{\mathcal{E}}\|$ $6 \times \|data\|$
Calculation of Bills and Rewards (Step 7)	DLT → Users Users → Referee Referee → Supplier Supplier → Referee Referee → DLT	$3 \times N_U \times \|data\|$ $4 \times N_{MU} \times \|\{data\}_{\mathcal{E}}\|$ $2 \times N_{MU} \times \|\{data\}_{\mathcal{E}}\|$ $2 \times N_{MU} \times \|data\|$ $2 \times N_U \times \|\{data\}_H\|$
Calculation of Bills and Rewards – Erroneous Case (Step 7 – EC)	Users → Referee DLT → Referee Referee → Supplier Supplier → Referee	$N_U \times \|\{data\}_{\mathcal{E}}\|$ $N_U \times \|\{data\}_H\| + \|data\|$ $4 \times N_{MU} \times \|\{data\}_{\mathcal{E}}\|$ $4 \times N_{MU} \times \|data\|$

Table 3
Communication Overhead per Billing Period

Step	Direction	Number of Bits
Transfer and Announcement of Bills, Revenues and Supplier Balance (Step 9)	Users → Supplier DLT → Supplier Referee → Supplier	$N_U \times \|\{data\}_{\mathcal{E}}\|$ $N_U \times \|\{data\}_H\|$ $\|data\|$

them back to the referee. Based on the results of the subtraction, if the calculated statement and accumulated statement values match, the hashed versions of these values are sent and recorded in the DLT.

Otherwise, if the calculated values do not match where EC occurred, the referee recalculates the incorrect statement calculations to find the responsible party for erroneous calculations. Firstly, it requests and receives the required data for re-calculation. Individual and total deviation data for the current SC is available by the referee, so it only requests $\{V^{P2P}\}_{\mathcal{E}}$ values from the users. In addition, the referee requests the hashed version of $\{V^{P2P}\}_{\mathcal{E}}$ from DLT to verify the correctness of the data. It also receives publicly available π_{P2P} from DLT. After performing the calculations to determine the responsible party, the referee subtracts the recalculated values of statements and accumulated statements from those calculated by the matched users. The subtraction results are then sent to the supplier for decryption, after which the supplier sends them back to the referee.

In step 8, no data transfer occurs; instead, the referee performs calculations using the data received in previous steps.

In step 9, for each billing period, users send their encrypted accumulated statements to the supplier. The supplier

then requests the hashed versions of these encrypted accumulated statements from the DLT for verification. The referee sends the supplier balance to the supplier.

Overall, this subsection details the data transferred for each step where data transfer occurs. When the direction of transfer is towards the DLT (i.e., from an entity to DLT), as shown in Tables 2 and 3, the data is permanently stored in the DLT. Additionally, the 'Number of Bits' column in these tables represents the amount of energy stored in the DLT when the transfer direction is towards the DLT.

4.4. Experimental Results

We evaluate the performance of PA-Bill by running simulations on a PC with an Intel Core i5-8350U CPU @ 1.7GHz CPU and 8GB of RAM to demonstrate its efficiency. We utilise the SHA3-256 algorithm for hashing and the Paillier cryptosystem for homomorphic encryption. These operations were implemented using the Python libraries hashlib[2] and phe[3].

We utilised an Ethereum test network to prototype our blockchain platform. For deployment and testing, we used Ganache[4], which provided us with a personal Ethereum blockchain. Ganache allows us to create a local blockchain, making it ideal for development and testing purposes. The smart contracts were written in Solidity[5], a statically typed programming language specifically designed for developing smart contracts on Ethereum. To compile these smart contracts, we employed Remix[6], a powerful online integrated development environment that offers a suite of tools for writing, testing, and debugging Solidity contracts.

To interact with the Ethereum blockchain, we used the Python Web3.py library[7]. Web3.py is a comprehensive collection of libraries that enables developers to communicate with the Ethereum blockchain via the JSON-RPC protocol. This library allowed us to read and write data to the blockchain and interact with smart contracts. By leveraging Web3.py, we could perform various blockchain operations such as checking, deploying contracts, and calling smart contract functions within our Python scripts.

Our development workflow involved several key steps: First, we wrote and compiled our smart contracts in Solidity using the Remix IDE. Next, we deployed these contracts to our local Ganache blockchain for testing. During this phase, we used Web3.py to connect to the Ganache blockchain, perform operations, and interact with the deployed contracts. This setup provided a robust and flexible environment for developing and testing our blockchain application, ensuring that our smart contracts behaved as expected before considering deployment to a live network.

While the DLT functions as a transparent and immutable bulletin board, providing accountability, we did not conduct a separate performance assessment of the blockchain platform itself. This is because the primary focus of our experiments was not on the blockchain platform, as we utilised existing tools to prototype the blockchain platform. In addition, our previous work [49] can be deployed as an electricity trading platform, so we do not reevaluate it in this context either. Instead, our primary focus lies in evaluating the performance of the privacy and accountable billing model.

4.4.1. Runtime Simulations

In this part, we showcase that our solution provides a computationally efficient privacy-preserving billing mechanism. To enhance system performance, various studies have deployed hardware accelerators like FPGAs to enhance performance [50] and energy efficiency [51] and explored embedded systems [52]. In contrast, PA-Bill employs a semi-decentralised approach that allows users to independently and concurrently process their calculations, leveraging parallelisation to improve computational efficiency and increase throughput. In such parallel systems, the slowest process often termed the "bottleneck", can limit overall performance. In PA-Bill, this bottleneck dictates the execution time by being the slowest component in the calculation process, thereby influencing the time taken to complete each step and transmit values to subsequent stages.

The efficacy of our billing model was tested with simulations on user sets of 1,000 and 2,000, with prosumer ratios of 25%, 50%, and 75%. Tables 4 and 5 present the execution times for the "Individual Deviations," "Total Deviations," and "Bills and Rewards" steps for 1,000 and 2,000 users, respectively. The "Users" column represents the execution times of both consumers and prosumers in specific steps, while the "RC-W-Case" (Referee Check Worst Case) column shows the execution time for the referee in the worst-case scenario, where all calculations are erroneous and must be recalculated by the referee to identify the responsible party.

The execution time (ET) simulations indicate that users' ET is lower than their RC-W-Case ET, as users perform their calculations independently in parallel. The slowest user, referred to as the 'bottleneck', determines the overall ET for users. In contrast, RC-W-Case requires the referee to handle all calculations in the worst-case scenario, which increases ET. The "Individual Deviations" step has a higher ET than "Bills and Rewards" due to the intensive subtraction operations required to compute deviations using homomorphically encrypted data. Naturally, the "Total Deviations" step incurs the highest ET because selected users must calculate the total deviations for all prosumers and vice versa.

In scenarios with uneven prosumer distributions (e.g., 25% and 75%), users' ET are higher compared to the even 50% ratio. This is because in uneven distributions, some users have more than one matched user, increasing their computational load as they perform additional calculations for their matched users.

When comparing Tables 4 and 5, representing 1,000 and 2,000 users, respectively, we observe similar ET for users in the "Individual Deviations" and "Bills and Rewards" steps across all user ratios. This is because users perform their

[2] https://docs.python.org/3/library/hashlib.html
[3] https://pypi.org/project/phe/
[4] https://www.trufflesuite.com/ganache
[5] https://solidity.readthedocs.io/en/v0.8.7/
[6] https://remix.ethereum.org/
[7] https://web3py.readthedocs.io/en/stable/

Table 4
Execution Time Results for 1,000 Users per Settlement Cycle.

Ratio of Prosumers ⇒	25%		50%		75%	
Calculation Step ⇓	Users	RC-W-Case	Users	RC-W-Case	Users	RC-W-Case
Individual Deviations	21.07 ms	13.40 s	11.50 ms	9.06 s	21.05 ms	13.38 s
Total Deviations	29.34 ms	90.93 ms	28.18 ms	89 ms	34.94 ms	80.28 ms
Bills and Rewards	2.86 ms	9.43 s	1.56 ms	6.40 s	2.71 ms	9.49 s

RC-W-Case: Referee Check Worst Case

Table 5
Execution Time Results for 2,000 Users per Settlement Cycle.

Ratio of Prosumers ⇒	25%		50%		75%	
Calculation Step ⇓	Users	RC-W-Case	Users	RC-W-Case	Users	RC-W-Case
Individual Deviations	21.64 ms	25.12 s	10.93 ms	21.45 s	20.22 ms	17.87 s
Total Deviations	69.05 ms	107.87 ms	58.25 ms	106.69 ms	48.18 ms	104.80 ms
Bills and Rewards	3.17 ms	17.85 s	2.05 ms	15.26 s	1.57 ms	12.73 s

RC-W-Case: Referee Check Worst Case

calculations independently and in parallel. However, the "Total Deviations" step shows a higher ET with 2,000 users, as the selected users must compute deviations for a larger user base. For the RC-W-Case, the ET increases for 2,000 users compared to 1,000, as the referee must handle all erroneous calculations, and the increased number of users amplifies the ET accordingly.

Despite the observed variations and similarities across different user counts and scenarios, our execution time simulations yielded results within a matter of *seconds* in the worst-case scenarios. This highlights the computational efficiency of our PA-Bill billing protocol.

4.4.2. Communication Overhead Simulations

In this part, we present the data transfer analysis simulations across various steps of the PA-Bill process, spanning from 100 users to 500. Our setup maintains homomorphically encrypted data at a length of 4096 bits (denoted as $|\{data\}_{\mathcal{E}}| = 4096$ bits), hashed data at 256 bits ($|data_H| = 256$ bits), and plaintext data at 32 bits. In our simulations, we've ensured that N_C is equivalent to N_P. Figure 2 illustrates the aggregate data transfer in kilobytes (KB) for steps 3-7, depicting data transmission during each settlement cycle. Concurrently, Figure 3 shows the total data transfer in KB for step 9, where data transfer occurs in each billing period.

From Figure 2, we observe a linear increase in data transfer corresponding to the number of users. Notably, Step 7 and Step 7-EC are the most data-intensive stages. Step 7 is pivotal for computing final statements, necessitating significant data for referee verification. Conversely, steps like 3 and 4 entail minimal data exchange since they solely involve data transfer without computational tasks, hence eliminating the need for additional verification data. In Step 5 and Step-EC, only data related to individual deviations are requested, contributing to moderate communication requirements. Step 6 exhibits lower communication as the referee verifies calculations for only six users. Remarkably, data transfer in Step 6-EC is the lowest, nearly zero, as the referee possesses the required information from the preceding steps to identify responsible parties. Turning to Figure 3, we observe a modest, linear increase in data transfer for each billing period, aligning with the growing user count.

4.5. Limitations of This Work

In the PA-Bill, the expenses related to electricity distribution and transmission are disregarded. This approach works well in cases where the energy trade occurs within small, geographically compact communities. In such scenarios, the close proximity within small communities results in minimal distribution and transmission costs, and the likelihood of congestion, along with its associated management expenses, can be considered negligible. However, when electricity is exchanged between households linked by a more complex network, the associated network costs—encompassing distribution, transmission and congestion management— could become a major concern. Although Paudel et al. [53] suggest a clearance mechanism that considers network costs, they do not address privacy issues or explore billing mechanisms. On the other hand, the PA-Bill focuses on privacy within billing mechanisms but does not account for network costs, which is the limitation of this work. Consequently, there is a promising opportunity for future research to develop billing mechanisms that preserve privacy while also factoring in network costs.

5. Conclusion

In this work, we proposed PA-Bill, a privacy-preserving and accountable billing protocol that addresses privacy, and accountability issues in P2P markets at the billing and settlements stage. PA-Bill utilises a universal cost-splitting billing model, local semi-decentralised calculation, and Homomorphic Encryption for privacy protection. Blockchain technology is deployed for accountability mechanisms that resolve

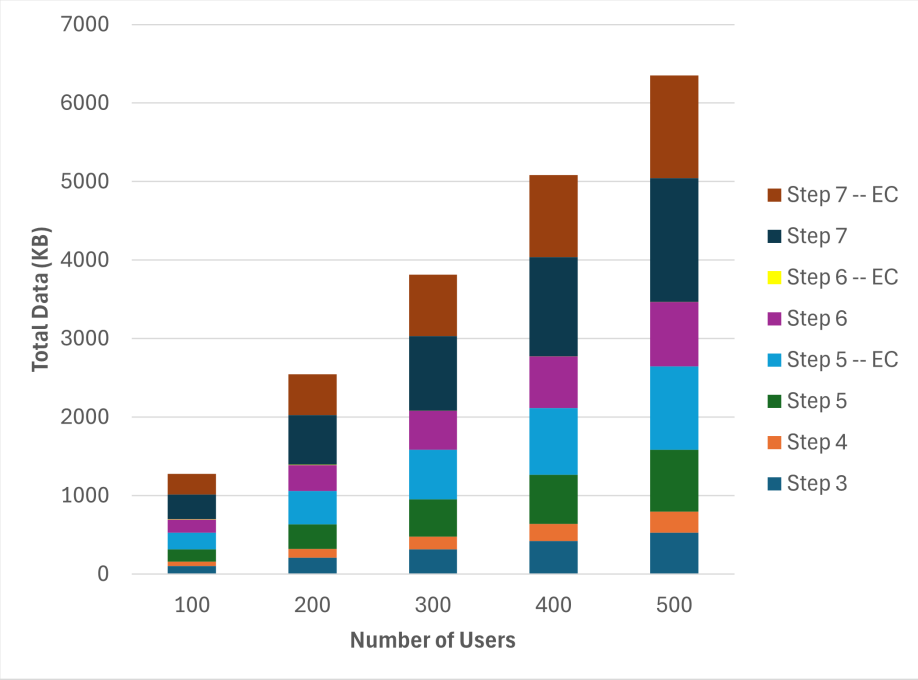

Figure 2: Communication Overhead per Settlement Cycle

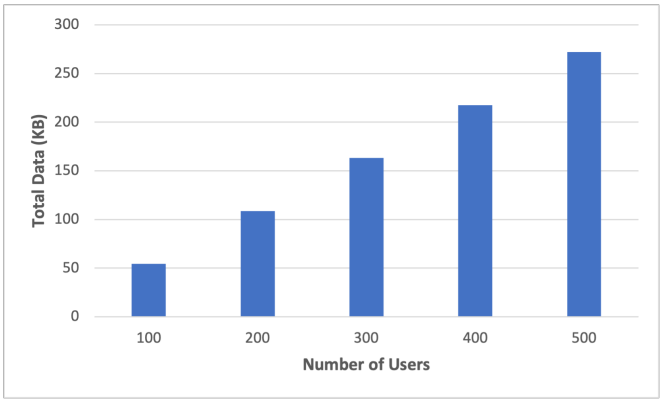

Figure 3: Communication Overhead per Billing Period

conflicts during billing calculation. PA-Bill is evaluated in a communities of up to 2000 households.

In our future work, we plan to investigate network constraints.

Acknowledgements

We express our gratitude to Talgar Bayan for enriching discussions on Blockchain technology and to Dr. Bernardo Magri for invaluable recommendations about this work.

References

[1] E. A. Soto, L. B. Bosman, E. Wollega, W. D. Leon-Salas, Peer-to-peer energy trading: A review of the literature, Applied Energy (2021).

[2] W. Tushar, T. K. Saha, C. Yuen, D. Smith, H. V. Poor, Peer-to-peer trading in electricity networks: an overview, IEEE Transactions on Smart Grid (2020).

[3] M. Gržanić, T. Capuder, N. Zhang, W. Huang, Prosumers as active market participants: A systematic review of evolution of opportunities, models and challenges, Renewable and Sustainable Energy Reviews 154 (2022) 111859.

[4] K. Erdayandi, M. A. Mustafa, Pp-lem: Efficient and privacy-preserving clearance mechanism for local energy markets, Sustainable Energy, Grids and Networks (2024) 101477.

[5] F. Zach, F. Kretschmer, G. Stoeglehner, Integrating energy demand and local renewable energy sources in smart urban development zones: New options for climate-friendly resilient urban planning, Energies 12 (2019) 3672.

[6] X. Wu, X. Wang, W. Zhang, Y. Huang, Lem for ders and flexible loads, IET Generation, Transmission & Distribution 13 (2019) 3556–3563.

[7] R. Andrade, J. Vitorino, S. Wannous, E. Maia, I. Praça, Lemmas: a secured and trusted local energy market simulation system, in: 2022 18th International Conference on the European Energy Market (EEM), IEEE, 2022, pp. 1–5.

[8] E. Alqahtani, M. A. Mustafa, Privacy-preserving local energy markets: A systematic literature review, Available at SSRN 4483407 (2023).

[9] J. Abdella, K. Shuaib, Peer to peer distributed energy trading in smart grids: A survey, Energies 11 (2018) 1560.

[10] S. Galperti, A. Levkun, J. Perego, The Value of Data Records, The Review of Economic Studies 91 (2023) 1007–1038.

[11] G. Kalogridis, M. Sooriyabandara, Z. Fan, M. A. Mustafa, Toward unified security and privacy protection for smart meter networks, IEEE Systems Journal 8 (2013) 641–654.

[12] R. Thandi, M. A. Mustafa, Privacy-enhancing settlements protocol in peer-to-peer energy trading markets, in: 2022 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), IEEE, 2022, pp. 1–5.

[13] K. Erdayandi, L. C. Cordeiro, M. A. Mustafa, Towards privacy preserving local energy markets, in: Competitive Advantage in the Digital Economy (CADE 2022): Resilience, Sustainability, Responsibility, and Identity, 2022, pp. 1–8.

[14] M. Jawurek, F. Kerschbaum, G. Danezis, Sok: Privacy technologies for smart grids–a survey of options, Microsoft Res., Cambridge, UK 1 (2012) 1–16.

[15] E. L. Quinn, Privacy and the new energy infrastructure, Available at SSRN 1370731 (2009).

[16] M. Montakhabi, A. Madhusudan, S. Van Der Graaf, A. Abidin, P. Ballon, M. A. Mustafa, Sharing economy in future peer-to-peer electricity trading markets: Security and privacy analysis, in: Proc. of Workshop on Decentralized IoT Systems, 2020, pp. 1–6.

[17] M. A. Mustafa, S. Cleemput, A. Abidin, A local electricity trading market: Security analysis, in: 2016 IEEE PES innovative smart grid technologies conference Europe (ISGT-Europe), IEEE, 2016, pp. 1–6.

[18] Regulation (eu) 2016 - general data protection regulation, 2016. URL: https://eur-lex.europa.eu/eli/reg/2016/679/oj, (Accessed on 14/12/2023).

[19] California consumer privacy act (ccpa), 2018. URL: https://cppa.ca.gov/regulations/pdf/cppa_act.pdf, (Accessed on 28/08/2024).

[20] J. Wang, F. Long, B. Jin, D. Dai, F. Liu, A privacy preserving energy trading platform based on smart contract, in: Proceedings of the 2022 5th International Conference on Algorithms, Computing and Artificial Intelligence, 2022, pp. 1–7.

[21] Z. Li, H. Xu, F. Zhai, B. Zhao, M. Xu, Z. Guo, A privacy-preserving, two-party, secure computation mechanism for consensus-based peer-to-peer energy trading in the smart grid, Sensors 22 (2022) 9020.

[22] E. Alqahtani, M. A. Mustafa, Zone-based privacy-preserving billing for local energy market based on multiparty computation, in: 2023 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), 2023, pp. 1–7. doi:10.1109/SmartGridComm57358.2023.10333925.

[23] V. Dudjak, D. Neves, T. Alskaif, S. Khadem, A. Pena-Bello, P. Saggese, B. Bowler, M. Andoni, M. Bertolini, Y. Zhou, et al., Impact of local energy markets integration in power systems layer: A comprehensive review, Applied Energy 301 (2021) 117434.

[24] A. Madhusudan, F. Zobiri, M. A. Mustafa, Billing models for peer-to-peer electricity trading markets with imperfect bid-offer fulfillment, in: 2022 IEEE Int. Smart Cities Conf. (ISC2), 2022, pp. 1–7.

[25] K. Erdayandi, L. C. Cordeiro, M. A. Mustafa, A privacy-preserving and accountable billing protocol for peer-to-peer energy trading markets, in: 2023 International Conference on Smart Energy Systems and Technologies (SEST), IEEE, 2023, pp. 1–6.

[26] A. Hutu, M. A. Mustafa, Privacy preserving billing in local energy markets with imperfect bid-offer fulfillment, in: 2023 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), 2023, pp. 1–9. doi:10.1109/SmartGridComm57358.2023.10333894.

[27] D. Mitrea, T. Cioara, I. Anghel, Privacy-preserving computation for peer-to-peer energy trading on a public blockchain, Sensors 23 (2023) 4640.

[28] M. Jokumsen, T. P. Pedersen, M. S. Daugaard, D. Tschudi, M. W. Madsen, T. Wisbech, Verifiable proofs for the energy supply chain: small proofs brings you a long way, Energy Informatics 6 (2023) 28.

[29] A. Abidin, R. Callaerts, G. Deconinck, S. Van Der Graaf, A. Madhusudan, M. Montakhabi, M. A. Mustafa, S. Nikova, D. Orlando, J. Schroers, et al., Poster: Snippet—secure and privacy-friendly peer-to-peer electricity trading, in: Proceedings of the Network and Distributed System Security Symposium (NDSS 2020), San Diego, CA, USA, 2020, pp. 23–26.

[30] A. Abidin, A. Aly, S. Cleemput, M. A. Mustafa, An mpc-based privacy-preserving protocol for a local electricity trading market, in: International Conference on Cryptology and Network Security, Springer, 2016, pp. 615–625.

[31] A. Abidin, A. Aly, S. Cleemput, M. A. Mustafa, Secure and privacy-friendly local electricity trading and billing in smart grid, arXiv preprint arXiv:1801.08354 (2018).

[32] A. Alabdulatif, H. Kumarage, I. Khalil, M. Atiquzzaman, X. Yi, Privacy-preserving cloud-based billing with lightweight homomorphic encryption for sensor-enabled smart grid infrastructure, IET Wireless Sensor Systems 7 (2017) 182–190.

[33] P. Singh, M. Masud, M. S. Hossain, A. Kaur, Blockchain and homomorphic encryption-based privacy-preserving data aggregation model in smart grid, Computers & Electrical Engineering 93 (2021) 107209.

[34] A. Ö. Gür, Ş. Öksüzer, E. Karaarslan, Blockchain based metering and billing system proposal with privacy protection for the electric network, in: Istanbul smart grids and cities congress and fair (ICSG), IEEE, 2019, pp. 204–208.

[35] H. Lv, Q. Wu, H. Ren, S. Shi, Evaluation of the economics and resilience of p2p electricity trading, in: 2024 9th Asia Conference on Power and Electrical Engineering (ACPEE), IEEE, 2024, pp. 1698–1702.

[36] M. A. Mustafa, N. Zhang, G. Kalogridis, Z. Fan, Dep2sa: A decentralized efficient privacy-preserving and selective aggregation scheme in advanced metering infrastructure, IEEE Access 3 (2015) 2828–2846.

[37] A. Abidin, A. Aly, S. Cleemput, M. A. Mustafa, Towards a local electricity trading market based on secure multiparty computation (2016).

[38] A. REGULATIONS, The electricity safety, quality and continuity regulations 2002, Crown Copyr (2002).

[39] M. A. Mustafa, N. Zhang, G. Kalogridis, Z. Fan, Roaming electric vehicle charging and billing: An anonymous multi-user protocol, in: 2014 IEEE International Conference on Smart Grid Communications (SmartGridComm), IEEE, 2014, pp. 939–945.

[40] X. Yi, R. Paulet, E. Bertino, X. Yi, R. Paulet, E. Bertino, Homomorphic encryption, Springer, 2014.

[41] P. Paillier, Public-key cryptosystems based on composite degree residuosity classes, in: International conference on the theory and applications of cryptographic techniques, Springer, 1999, pp. 223–238.

[42] K. Jastaniah, N. Zhang, M. A. Mustafa, Efficient privacy-friendly and flexible iot data aggregation with user-centric access control, arXiv preprint arXiv:2203.00465 (2022).

[43] M. Andoni, V. Robu, D. Flynn, S. Abram, D. Geach, D. Jenkins, P. McCallum, A. Peacock, Blockchain technology in the energy sector: A systematic review of challenges and opportunities, Renewable and Sustainable Energy Reviews 100 (2019) 143–174.

[44] T. Bayan, R. Banach, Exploring the privacy concerns in permissionless blockchain networks and potential solutions, arXiv preprint arXiv:2305.01038 (2023).

[45] Ethereum foundation. ethereum blockchain app platform,, 2024. URL: https://www.ethereum.org/, (Accessed on 17/03/2024).

[46] S. S. Kushwaha, S. Joshi, D. Singh, M. Kaur, H.-N. Lee, Ethereum smart contract analysis tools: A systematic review, Ieee Access 10 (2022) 57037–57062.

[47] Y. Chen, L. Yang, Y. Fan, L. Zhang, L. Tian, Study on energy efficiency and carbon neutral path of ethereum blockchain: from pow to pos, in: Ninth International Conference on Energy Materials and Electrical Engineering (ICEMEE 2023), volume 12979, SPIE, 2024, pp. 889–894.

[48] M. Farnaghi, A. Mansourian, Blockchain, an enabling technology for transparent and accountable decentralized public participatory gis, Cities 105 (2020) 102850.

[49] K. Erdayandi, A. Paudel, L. Cordeiro, M. A. Mustafa, Privacy-friendly peer-to-peer energy trading: A game theoretical approach, in: 2022 IEEE Power & Energy Society General Meeting (PESGM), IEEE, 2022, pp. 1–5.

[50] S. Liang, S. Lu, J. Lin, Z. Wang, Low-latency hardware accelerator for improved engle-granger cointegration in pairs trading, IEEE Transactions on Circuits and Systems I: Regular Papers 68 (2021) 2911–2924.

[51] Y. Aksehir, K. Erdayandi, T. Z. Ozcan, I. Hamzaoglu, A low energy adaptive motion estimation hardware for h. 264 multiview video coding, Journal of Real-Time Image Processing 15 (2018) 3–12.

[52] G. A. Malazgirt, B. Kiyan, D. Candas, K. Erdayandi, A. Yurdakul, Exploring embedded symmetric multiprocessing with various on-chip architectures, in: 2015 IEEE 13th International Conference on Embedded and Ubiquitous Computing, IEEE, 2015, pp. 1–8.

[53] A. Paudel, L. Sampath, J. Yang, H. B. Gooi, Peer-to-peer energy trading in smart grid considering power losses and network fees, IEEE

Transactions on Smart Grid 11 (2020) 4727–4737.

Kamil Erdayandi earned his PhD in Computer Science from The University of Manchester. He holds a BSc degree in Electrical & Electronics Engineering and a dual degree in Computer Science, as well as an MSc in Electronics. His current research focuses on several key areas, including Data Privacy, Applied Cryptography, Game Theory, Homomorphic Encryption, Blockchain, Local Energy Markets and Smart Grids.

Lucas C. Cordeiro received a B.Sc. degree in electrical engineering and an M.Sc. degree in computer engineering from the Federal University of Amazonas (Brazil) in 2005 and 2007, respectively. He received a Ph.D. in computer science from the University of Southampton (UK) in 2011. He is a Reader in the Department of Computer Science at the University of Manchester (UK), where he leads the Systems and Software Security (S3) Research Group. Dr. Cordeiro is also the Arm Centre of Excellence Director at UoM. In addition, he is affiliated with the Trusted Digital Systems Cluster at the Centre for Digital Trust and Society, the Formal Methods Group at the University of Manchester, UK, and the Post-Graduate Programs in Electrical Engineering (PPGEE) and Informatics (PPGI) at the Federal University of Amazonas, Brazil. Before joining the University of Manchester, he worked as a post-doctoral researcher at the University of Oxford and as a research engineer at Diffblue. In addition, Dr. Cordeiro worked for five years as a software engineer at Siemens / BenQ Mobile and CTPIM / NXP semiconductors. His work focuses on software model checking, automated testing, program synthesis, software security, and embedded and cyber-physical systems. He has co-authored more than 150 peer-reviewed publications in the most prestigious venues. He has received various international awards, including the Most Influential Paper Award at ASE'23, the Distinguished Paper Award at ICSE'11, and 46 awards from the international competitions on software verification (SV-COMP) and testing (Test-Comp) 2012-2024. He has a proven track record of securing research funding from EPSRC, Intel, Motorola, Samsung, Nokia Institute of Technology, CNPq, FAPEAM, British Council, and Royal Society (career total over USD 13M).

Mustafa A. Mustafa is a Senior Lecturer (Associate Professor) in the Department of Computer Science at The University of Manchester, where he leads the Trusted Digital Systems Cluster – part of the university-wide Centre for Digital Trust and Society. He received the B.Sc. degree in communications from the Technical University of Varna, Varna, Bulgaria, in 2007, the M.Sc. degree in communications and signal processing from Newcastle University, Newcastle upon Tyne, U.K., in 2010, and the Ph.D. degree in computer science from The University of Manchester, Manchester, U.K., in 2015. He then was a post-doctoral research fellow with the imec-COSIC research group, Department of Electrical Engineering (ESAT), KU Leuven, Belgium. From July 2018 till June 2023, he was a Dame Kathleen Ollerenshaw Research Fellow in the Department of Computer Science at The University of Manchester.