ETradeChain: Blockchain-Based Energy Trading in Local Energy Market (LEM) Using Modified Double Auction Protocol

Umang Rajendra Barbhaya, Lokendra Vishwakarma, *Graduate Student Member, IEEE*, and Debasis Das, *Senior Member, IEEE*

***Abstract*—The smart grid's local energy market (LEM) enables each renewable-energy-powered residential unit to profit from trading energy with others. However, energy trading in LEMs is witnessing many cybersecurity challenges, such as transaction integrity and user authentication. Among these, energy trading and price computing using auctions have generally been accepted. However, state-of-the-art auction schemes are centralized and unfair, meaning that prosumers are not equally benefited. We proposed ETradeChain, a platform for energy trading based on blockchain technology. The trading in ETradeChain happens with the help of a modified double auction scheme to make it fully decentralized and fair for all the members of LEM, along with information secrecy. We have developed a pseudo coin called Pcoins (Power Coins) based on the energy generated by the prosumer for energy trading in LEM. The ETradeChain uses a double auction process with Pcoin as a stake to reach a consensus on the energy transaction. Furthermore, ETradeChain employs blockchain technology to demonstrate the viability of real-time peer-to-peer (P2P) trading for practical purposes. We have set up a Testbed for the experiments using Raspberry Pi 4 model B IoT devices. The experiment results show that the ETradeChain minimized the consensus delay up to 90% with 60% high throughput. It also achieved 80% low computational overhead and 70-80% low storage and communication overhead.**

***Index Terms*—Modified double auction, blockchain protocol, secure energy distribution, prosumer, energy trading.**

I. Introduction

In recent years, the surge in energy demand has intensified the strain on power generation, primarily sourced from polluting non-renewable resources. Renewable energy offers a cleaner alternative to meet this demand. The smart grid, an advanced energy distribution technology, offers superior control and efficiency over traditional systems, allowing energy companies to adjust distribution and pricing in real-time. Enabled with Internet technology, the smart grid ensures energy availability anytime, anywhere [1]. However, its centralization poses risks of single-point failures and data tampering. The integration of the Internet also introduces challenges like user and data security, with potential threats of data theft during transit. Hence, ensuring information security is paramount [2]. Decentralization is emerging as a solution in smart grid technology, facilitating decentralized energy management to cater to the escalating energy demand.

The local energy market (LEM) is one such system that is decentralized and that comprises the consumer and prosumer as well as the smart grid. The consumers become prosumers by trading the generated surplus to neighbors or to the main central grid. However, the prevention of tampering or loss of information has to be ensured while managing the energy trading transactions in LEM. Blockchain technology is the potential solution to address the issues of LEM [3], [4]. However, existing blockchain protocols are not suitable for energy trading in LEM due to the issues associated with corresponding consensus algorithms.

The smart grid system for LEM requires a customized blockchain system because existing protocols are unsuitable due to their high computational mining demands, significant message overhead, and transaction confirmation delays. Current cryptocurrencies like Bitcoin and Ether can not directly link energy trading with energy generation for transparent trading. Moreover, these systems are vulnerable to blockchain-specific attacks like double-spending, forking, and non-repudiation. Their performance is also subpar, marked by consensus delays, low throughput, and high computational, storage, and communication costs.

The proposed ETradeChain is a next-gen smart grid technology that resolves issues in current methods. It's a concurrent blockchain-based energy trading system ideal for decentralized LEM. This concurrent approach offers better efficiency and higher throughput than traditional blockchain for smart grids. In this permissionless network, energy is traded for smart homes using "Pcoins" (Power Coins), which represent energy generation. Prosumers can exchange generated energy for Pcoins and others can buy these Pcoins to get energy. An auction mechanism is essential for fair energy trading and allocation. An auction mechanism is a great way to manage and allocate energy.

Auctions have been widely researched for trading in areas like cloud virtual machines [5], [6], edge computing resources [5], [7], Smart Grids [8], [9], [10], [11]. Most

Manuscript received 11 February 2023; revised 27 May 2023 and 26 July 2023; accepted 8 August 2023. Date of publication 21 August 2023; date of current version 15 February 2024. This work was supported in part by the GITA-DST Project under Grant GITA/DST/TWN/P-92/2021, and in part by the Indian Institute of Technology Jodhpur, India. The editor coordinating the review of this article was J. Xu. *(Corresponding author: Debasis Das.)*

The authors are with the Department of Computer Science and Engineering, Indian Institute of Technology Jodhpur, Jodhpur 342030, India (e-mail: barbhaya.1@iitj.ac.in; vishwakarma.3@iitj.ac.in; debasis@iitj.ac.in).

Digital Object Identifier 10.1109/TGCN.2023.3307360

auction systems rely on a central authority to oversee the process, and some are computationally intensive, making them unsuitable for devices like smart meters. Many are also unfair and suboptimal. In contrast, ETradeChain offers a lightweight, fair, and decentralized auction mechanism. This auction is also used to select temporary leaders for consensus in block creation and verification. Specifically, our primary contributions are as follows:

1. New Cryptocurrency (Pcoin): We have introduced a novel energy cryptocurrency called Pcoin which stands for Power Coin. Unlike existing cryptocurrency such as Bitcoin, Ether, etc, Pcoins is associated with energy generated by any of the prosumers. Therefore, we have used Pcoins instead of an existing cryptocurrency. Further, the creation of Pcoins is completely dependent on the energy generated by the prosumers, unlike Bitcoin, Ether, and other cryptocurrencies, which are created based on high computing mining and reward. To create, transfer and consume the Pcoins, there are three types of transactions which are PcoinCreate, PcoinTransfer, and PcoinConsume.

2. Fair Energy Trading and Prevention from Overpricing: Along with the new energy cryptocurrency, we have addressed the issue of unfair mining and overpricing of energy using the modified double auction and Vickrey auction method. This combination helps the auction to increase its social welfare and also avoid overpaying. An optimal approach is proposed to allocate the consumer's bid to the prosumers. Before bid allocation, a novel sorting factor is added on the consumer end to avoid biased behavior. During Bid allocation, thresholding is applied to sell the Pcoins so that starvation is avoided on the prosumer's end. The calculation of the pricing depends on Vickrey's auction and also the mean of the bid prices. Also, the temporary leader selection is based on the prosumers who won the auction.

3. Performance and Security Analysis: We have also measured the system's performance based on parameters like consensus latency, throughput, computation overhead, energy consumption, and storage and communication overhead to show the effectiveness of this P2P energy trading system compared with state-of-the-art schemes. We compared the security analysis based on parameters like fork prevention, prevention of double-spending, and non-repudiation with state-of-the-art schemes.

A. Goal of ETradeChain

ETradeChain is a decentralized P2P network with N mining nodes. Each node uses a key pair to sign and verify blocks and transactions, eliminating the need for a central PKI. Transactions are cryptographically hashed using function $H(x)$ before block inclusion. At the time t, some nodes, $M(t) \in N$, are Byzantine and controlled by an adversary, while others are honest. Given a node's mining power as $Mining(K)$, the combined power of Byzantine nodes must be under 51% of the total N nodes' power. This is because ETradeChain included all such vulnerable nodes, which do selfish mining, under the control of the adversary.

$$\sum_{k \in M(t)} Mining(k) < 51\% \sum_{i \in N} Mining(i) : \forall K, i \qquad (1)$$

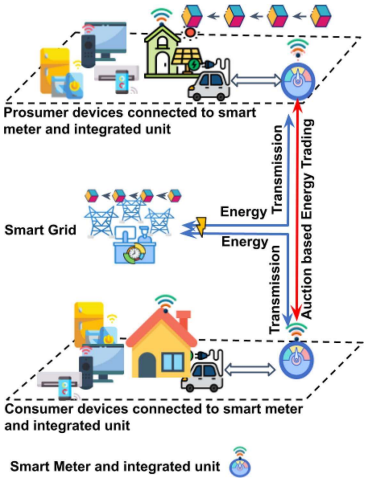


Fig. 1. ETradeChain system model.

B. System Model

ETradeChain's architecture comprises three components: Smart Grid (Auctioneer), prosumer (Seller+Buyer), and consumer (Buyer), depicted in Fig. 1. A LEM Testbed is set up using Raspberry Pi boards, acting as smart meters with integrated energy trading units. These meters, placed at consumer locations, track power consumption. The Smart Grid distributes power to consumers. Prosumers create a permissionless blockchain network, which securely logs user details and smart meter transactions in a ledger.

For each time slot $t \in T$, there's a consistent market-clearing price. Customers with full bid orders pay this price for the specified power amount. Blockchain facilitates both market operations and payments. Smart meters automatically measure and predict each customer's demand d_{it} and generation g_{it}. Based on this, any excess demand or supply is computed and credited to the customer's blockchain account, ensuring data privacy while aligning with consumption patterns. Accounts hold Pcoin balances and energy data. By combining a customer's energy surplus and credit balance, orders are tailored to their needs. This virtual trading doesn't affect actual electricity transfers. We use a supply and demand matching online mechanism, as well as price signals, to indicate the local scarcity of energy.

As a result, the large-scale power grid can be accessed. A forecast of consumption and production at the customer level is used to match the LEM's demand and supply restrictions. To keep things simple, we only include electricity trading and ignore bundled orders (like electricity and heat). The LEM's result is calculated as the total cost of electricity for all customers. In other words, this is the total amount of electricity x_{ijt} sold from one customer i to another customer j at a price c_{ijt} at time t. The minimum cost of the energy sold from prosumer i to prosumer/consumer j is represented in Equation (2). Equation (3) ensures that every prosumer satisfies their demand d_{it} and sells the surplus generation g_{it}. Equation (4) and (5) ensure that the smart grid balances supply and demand within the LEM. The hard constraints for this are the equations from (2) to (5). The decision variable x_{ijt} has to be positive because only a positive amount of electricity

can be traded.

$$min \sum_{i\in I, j\in J} c_{ijt} * x_{ijt} \tag{2}$$

$$g_{it} - d_{it} - \sum_{j\in I}\left(x_{ijt} - x_{jit}\right) = 0, \forall i \in I \tag{3}$$

$$\sum\left(g_{it} - d_{it} \leq \sum_{i\in I} X_{igt}\right) \tag{4}$$

$$-\left(\sum_{i\in I}(g_{it} - d_{it})\right) \leq \sum_{i\in I} x_{git} \tag{5}$$

The rest of the paper is organized as follows. In Section II, we have discussed the literature survey in detail. The problem definition of energy trading is illustrated in Section III. In Section IV, we have defined our proposed scheme in detail. The performance analysis is discussed in Section V. Section VI is all about the analysis of the auction method and the security aspect of ETradeChain. Section VII is the conclusion of the proposed scheme and its future direction.

II. Related Work

The related work is categorized into three areas: security in the smart grid, blockchain-based energy trading in the smart grid, and blockchain-based auction strategies for energy trading. These techniques closely relate to our proposed scheme. We've also highlighted the shortcomings of current methods. The classification of the related work helps to understand issues in existing schemes.

A. Security in Smart Grid

A detailed review of blockchain for smart cities is presented in [1], covering smart grid aspects like smart metering, energy trading, and global projects like WePower, Sunchain, and SunContract. Li et al. [12] introduced a system prioritizing consumer information privacy using software guard extensions. Xue et al. [13] designed a real-time pricing model for smart grids emphasizing fault tolerance and customer resilience. Guan et al. [14] suggested an authentication method for data aggregation. ECC-based security was discussed in [15], [16]. However, these state-of-the-art methods, being centralized, are prone to single-point failures and often struggle with accuracy.

To address the centralization issue in smart grids, many researchers have integrated blockchain. Liang et al. [17] enhanced the smart grid's defense against cyberattacks using a blockchain-based security framework. However, it's not fit for lightweight devices like smart meters due to its high processing time, computational cost, and consensus delay. Jindal et al. [18] introduced a blockchain-based smart grid scheme called "guardian," but its PoW consensus algorithm has a prolonged confirmation time, making it unsuitable for smart grid use. Wang et al. [19] presented anonymous authentication with key management, but its cryptographic approach is computationally intensive for devices like smart meters. While these methods address the smart grid's centralization, their consensus delays are too long, and throughput is too low for optimal smart grid operations.

To advance energy trading in the smart grid, Ning et al. [20] presented a fault inspection and secure transmission scheme using blockchain. Named BEST, it ensures secure transmission and employs an enhanced market matching algorithm for fault detection. However, it faces scalability and message overhead challenges, affecting performance. Karumba et al. [21] introduced HARB, a hypergraph-based framework for energy trading using consortium blockchain. HARB's three-layered network architecture addresses scalability and distributed trust issues in energy trading communities.

The above schemes solve the centralization problem of the smart grid. However, the consensus delay of these schemes is high for the smart grid environment with low throughput. In contrast, ETradeChain first modified the consensus algorithm and the miner nodes' organization. This modification helps us achieve optimized results such as low consensus delay, high throughput, and less computational, storage, and communication cost.

B. Blockchain-Based Energy Trading in Smart Grid

Blockchain-based energy trading in the smart grid is an emerging trend. Several researchers have introduced techniques for this purpose. Wang et al. [22] suggested using smart contracts and delegated consensus mechanisms for smart grid trading, emphasizing fairness and a reputation system for P2P trading. However, its consensus algorithm, similar to PoW, is computationally intensive. AlSkaif et al. [23] introduced a peer-to-peer energy trading strategy for residential systems, determining bilateral trading preferences based on surplus power matching and trader distance. Ping et al. [24] presented a privacy-preserved Optimize Energy Trading (OET) scheme using blockchain. To account for dishonesty and ensure accurate trading results, they proposed a privacy-preserving byzantine fault tolerance coordination algorithm.

Christidis et al. [25] introduced a framework for creating a blockchain-based local energy market, highlighting that the blockchain type significantly impacts energy distribution and trading performance. However, this framework faces scalability issues. In [26], an energy trading system leveraging blockchain and based on the minimum cut and maximum flow theory was presented. Trading occurs between producers and consumers, scheduled according to distributed energy sources. Foti et al. [27] introduced a consensus for optimal power flow (OPF) in energy trading. They proposed the proof of OPF (PoOPF) consensus algorithm specifically for smart grid energy trading. PoOPF maintains a constant transaction count per block. Privacy in LEM trading is a crucial concern addressed in [28]. The study presents an energy system for IoT-enabled smart homes using a modified PBFT consensus algorithm.

Guan et al. [29] introduced a privacy-preserving energy trading scheme using blockchain and attribute-based encryption. They also presented a credibility-based equity-proof consensus algorithm to boost efficiency. In [30], a trusted energy trading framework using blockchain and optimization was proposed

to exclude dishonest participants, illustrating its effectiveness in preventing market failures due to such participants. Chen et al. [31] put forth a distributed and robust security-constrained economic dispatch (SCED) algorithm based on blockchain for energy distribution. This SCED algorithm remains effective against malicious and selfish actors and is efficient without a central coordinating entity.

Yang et al. [32] introduced a consortium blockchain trading mechanism for P2P energy trading based on a proof-of-stake protocol. They employed smart contracts and launched a cryptocurrency named "elecoin" for the P2P market. Similarly, Cao et al. [33] presented a solution addressing credit and security concerns in energy trading, utilizing elecoins to optimize the utility of energy traders. Their approach models the elecoins-purchasing dilemma as a hierarchical Stackelberg game. A miner reward mechanism was suggested in [34] for energy trading. This paper recommends a consortium blockchain system using a Proof-of-Stake (PoS) consensus algorithm to ensure secure, transparent energy transactions and reduce trading costs. They also introduced the elecoin cryptocurrency for P2P trading within microgrids. Guo et al. [35] proposed an architecture for distributed energy trading on a byzantine-based blockchain, leveraging the PBFT consensus algorithm and smart contracts. However, the PBFT-based model is deemed inefficient due to high message overhead. To address these challenges and ensure efficient, secure energy trading, we introduce ETradeChain.

C. Auction Strategies in the Blockchain

This section discusses auction strategies for decentralized blockchain energy trading. Comprehensive surveys on auction models integrated with blockchain technology are found in [36], [37]. Auction models are pivotal across industries for ensuring fair trading, a cornerstone for business expansion. The Smart Grid is poised to become a significant energy trading platform. Hence, selecting an appropriate auction model is vital for fostering fairness and profitability in energy trading. Lin et al. [38] introduced a blockchain-based fair payment system optimized for cloud computing. However, it falls short in offering optimal deals to both buyers and sellers due to the absence of an auction mechanism. Addressing this, Hassan et al. [39] presented a Virkrey-Clarke-Groves (VCG) auction for Smart Grid trading. A limitation here is its focus solely on maximizing the buyer's social welfare, neglecting the seller's utility. Ha et al. [8] also employed a VCG auction for energy market transactions, utilizing a brokerage system as the auctioneer.

Being centralized, the brokerage system has the problem of a single point of failure, and it can also be a potentially bad actor. Singla et al. [9] also used the Virkrey-Clarke-Groves Auction for efficient energy trading along with providing the fair optimal pricing algorithm that does not discriminate and increases social welfare. But it also has a centralized auctioneer, which can have a single point of failure problem. Ma et al. [5] used a blockchain-based distributed Feedback-based Combinatorial Multiunit Double Auction (FC-MDA) mechanism for trading the resources of the cloud.

While Doan et al. [10] introduced a real-time P2P energy trading system utilizing a double auction-based game-theoretic method, incorporating Stackelberg's non-cooperative game between the auctioneer and prosumers. However, it doesn't ensure maximum revenue for service providers. In [40], bid-rigging strategies are explored where mobile blockchain apps vie for edge computing resources via auctions, but this approach lacks fairness for both buyers and sellers. Chowdhury et al. [41] crafted a wholesale energy market simulation to assess PDA bidding strategies, but the energy resource allocation strategy wasn't detailed.

Nguyen and Thai [42] discusses an Ethereum blockchain-based framework for iterative double auction that's trustless and decentralized. It boasts low latency and fee-less on-chain transactions. Chen et al. [6] introduce a fair cloud auction design using blockchain, achieving financial fairness through a timed commitment scheme with deposits. However, this design exposes other parties' bids, leading to potential bid-rigging. Generally, blockchain and double auction systems risk exposing sensitive buyer and seller information. In contrast, our auction approach is budget-balanced, fair to all participants, lightweight, and decentralized in bid allocation, relying on the entire peer network.

III. Problem Definition of Energy Trading

Let us define and formulate our problem. We consider that we have N prosumers and M consumers. Let P define the set of prosumers and C defines the set of consumers such that $P = \{1, 2, 3, \ldots, N\}$ and $C = \{1, 2, 3, \ldots, M\}$. Any prosumer, while bidding, will have two factors $(bidP_j,\ s_j)$. $bidP_j$ denotes the selling price of one Pcoin as a bid raised by j^{th} prosumer, and s_j denotes the total number of Pcoins j^{th} prosumer wants to sell. Similarly, consumers also have two factors $(bidC_i,\ d_i)$. $bidC_i$ denotes the bidding price of one Pcoin raised by i^{th} consumer and d_i denotes the total Pcoins demand of i^{th} consumer. The utility of i^{th} consumer will be $u(C_i) = d_i \cdot bidC_i - d_{si} \cdot avg(C_i)$ where the $avg(C_i)$ denotes the average amount i^{th} consumer has to pay for one Pcoin to the seller and d_{si} denotes the total Pcoins received by i^{th} consumer. Similarly, the utility of j^{th} prosumer is equal to $u(P_j) = s_{sj} \cdot act(P_j) - s_j \cdot bidP_j$ where $act(P_j)$ denotes the actual amount received by the prosumer for the Pcoins and s_{sj} denotes the total Pcoins received by the j^{th} prosumer. Therefore the total Utility of the auction can be defined as follows

$$U = \sum_{j=1}^{N} u(P_j) + \sum_{i=1}^{M} u(C_i) \tag{6}$$

We are considering the auction process to be balanced, i.e., the total amount received by the prosumer should be equal to the total amount given by the seller at the end of the auction process. Therefore $\sum_{j=1}^{N} s_{sj} \cdot act(P_j) = \sum_{i=1}^{M} d_{si} \cdot avg(C_i)$

$$U = \sum_{j=1}^{N} s_j \cdot bidP_j - \sum_{i=1}^{M} d_i \cdot bidC_i \tag{7}$$

Therefore, we can define our problem as follows.

TABLE I
SYMBOL TABLE

Symbol	Description	Symbol	Description
$H()$	SHA256 Hash function	Bl	Block
BH	Block Hash	PB	Previous Block
PBH	Hash of Previous block	TP	Transaction Pool
Mrt	Merkle Root	Tr_i	i^{th} Transaction
TL	Temporary leader	M_{ID}	Smart meter unique ID
T_i	Current Timestamp	SK_{pub}	Public Key of smart meter

Maximise $U = \sum_{j=1}^{N} s_j \cdot bidP_j - \sum_{i=1}^{M} d_i \cdot bidC_i$
Subject to:

$$0 \leq s_{sj} \leq s_j, \forall j \in N \tag{8}$$

$$0 \leq d_{si} \leq d_i, \forall i \in M \tag{9}$$

$$\sum_i d_{si} = \sum_j s_{sj}, \forall i \in M, \forall j \in N \tag{10}$$

$$d_i, s_j > 0, \forall i \in M, \forall j \in N \tag{11}$$

According to Equation (8), the total number of Pcoins sold by the prosumer should be less than or equal to the total number of Pcoins available with the prosumer to sell. Equation (9) tells that the total number of Pcoins received by the consumer should be less than or equal to the Pcoins requested. Equation (10) defines the total Pcoins sold by the prosumer should be equal to the total Pcoins received by the consumer. Further, Equation (11) explains the total Pcoins bid by both prosumer and consumer should be greater than 0.

IV. ETradeChain: Energy Trading in LEM With Blockchain

This section explains the protocol in detail. Various mathematical symbols being used in algorithms can be found in Table I. ETradeChain is a blockchain-based Smart Grid system for processing transactions. Here we have considered three types of transactions, PcoinCreate, PcoinTransfer, and PcoinConsume. The detail of these transactions is given in the subsequent section. All the energy trading happens with the use of a modified double auction scheme. The winners of the auction are selected as temporary leaders. These leaders create the blockchain network for the consensus on block creation. In this network, one super leader is elected. The super leader selection is based on the stake deposited by the leader nodes. The Pcoins are used as a stake. The leader node that deposited the maximum stake will have the highest probability of being selected as the super leader node. The super leader node is responsible for block creation and transaction verification. However, transactions are verified by the other leader nodes too.

A. Adversary Model

We have used widely accepted Canetti and Krawczyk's (CK) adversary model [43] for the proposed scheme. The information is shared between the smart meter, grid operator, and miner node over a public channel. Some assumptions for the security and performance of the system are as follows.

1) The adversary Λ knows the communication channel is open and can compute the secret values if Λ has private credentials.
2) Consider an authorized miner node performing some malicious activities, e.g., trying to put a fake transaction in the block or a fake block. However, the consensus algorithm takes care of this.
3) A legitimate consumer can act as an adversary to make a DoS attack by sending multiple fake transactions. However, this act could move down the miner node but not the whole network, which makes the system fault-tolerant.
4) The adversary Λ can delay the response time for block confirmation to launch the double-spending and Liveness attack. Algorithm 3 protects the system from these attacks.

B. Type of Transactions

The energy is traded between the nodes via a smart grid energy distributor in the LEM. Three types of transactions are generated for trading energy, which are described below.

1) *PcoinCreate:* Pcoins are created by prosumers based on the generated electricity. The excess electricity is given to the power grid, which generates the equivalent coins. With respect to this coin generation, the transaction is generated, which is called the PcoinCreate transaction.
2) *PcoinTransfer:* The consumer who wants to get electricity from the smart grid first needs to buy some Pcoins. This happens via the auction process. After the Bid is allocated and the pricing of Pcoins is calculated between the prosumer and consumer, the Pcoins and money are transferred between them. This type of transaction is called "PcoinTransfer."
3) *PcoinConsume:* The consumer, or prosumer, can get electricity from the smart grid in exchange for Pcoins, and this transaction is called a "PcoinConsume" transaction. The consumed Pcoins will never be used again to get electricity.

C. Auction Mechanism

Below is the process of a modified double auction mechanism where both consumers and prosumers can bid to buy Pcoins and ask to sell Pcoins. Further, we are going to use the Vickrey auction or second-price auction to allow for truthful action. Whenever any prosumer wants to participate in the trading process, they must first register at the smart grid (certificate authority) for generating key pairs. After getting the key pair, the public key acts as an identity for the node. The auction is done so that prosumers can sell their Pcoins and consumers can buy those Pcoins. There are three important steps in the complete auction process.

- *Bid*: It is the first step in the auction process. It has a dedicated time phase during which prosumer and consumer share their bid with the smart grid. After the completion of this phase, the smart grid broadcasts all the collected bids to the peers in the network.
- *Allocate*: After receiving the bid details from the smart grid, all the nodes in the blockchain calculate the bid allocation and pricing scheme and share the final output

with the peers. Further, all the nodes verify the schemes and reach on consensus using the voting mechanism.

- *Energy Exchange*: Once the above steps are completed, the consumers can get their energy in exchange for Pcoins from the smart grid by invoking the PcoinConsume transaction.

Below is the detailed process of the auction mechanism.

1) Bidding Process: The auction mechanism is initiated by the bidding process. This bidding process is executed in a dedicated time phase or a slot. During this phase, all the consumers and prosumers who want to take part in the auction will share their bids with the smart grid. The consumer also shares its demand for the Pcoins that it requires, and the prosumers will share the number of Pcoins that it wants to sell. Once this phase of the bidding process is over, the Smart grid will then broadcasts the collected bids, consumer demand, and prosumer supply details with the peers in the network. These details are then used by the peers for the next steps.

2) Sorting Mechanism: In our case, the sorting mechanism is done off-chain, all the nodes conduct the auction mechanism, and the final allocation and pricing scheme is broadcasted to other nodes, unlike the central authority or auctioneer performing the Auction mechanism as per in [8], [9], [10]. The allocation and pricing scheme is chosen, which has been taken by 50 percent of the nodes in the network. Further, [10] tries to sort the buyers (Consumers) in descending order on the basis of their bid price. But on the consumer end, there is a possibility that a single buyer bids the highest for a large amount of Pcoins, so to avoid that, we created a sorting factor for the consumer as follows

$$sf(C_i) = \frac{bidC_i}{\sqrt{d_i}} \tag{12}$$

From the above equation, it can be seen that the sorting factor doesn't only depend upon the bid price but is inversely proportional to the square root of the Pcoins demanded. This keeps a single buyer from buying all the Pcoins by paying a high price. After this, we sort this Sorting Factor of consumers in descending order. In the case of the prosumers, we are trying to sort them in ascending order of their bid prices.

3) Bid Allocation Mechanism: We have implemented an optimal approach to allocating the consumer's bid to the prosumers. To avoid starvation, we are creating a threshold value on the number of Pcoins a prosumer could sell to a particular consumer. If consumer demand is not satisfied by the first prosumer, his demand will be satisfied by the next prosumer in the sorted order, and so on. The first prosumer, after satisfying one consumer's demands, will satisfy the next consumer in reverse sorted order of the sorting factor. Below is the formulation of the threshold value.

$$Th_s = \frac{\sum_{j=1}^{N} s_j}{M \cdot N} \tag{13}$$

In the Algorithm 1, we are considering that the consumers and prosumers are already sorted as per the sorting mechanism discussed above, and C_1 and C_M are the consumers having the highest and lowest sorting factors, respectively, and also P_1 and P_N are the prosumers having the lowest and highest bid price, respectively. The algorithm for bid allocation works as follows:

1) After Prosumers and Consumers are sorted as per the sorting mechanism, a threshold is calculated using (13).
2) Mapping is done between the consumers and the prosumers. The allocation matrix stores the number of Pcoins that needs to be transferred from j^{th} prosumer to the i^{th} consumer.
3) For loops at lines 2 and 3 are executed for every consumer and prosumer, respectively.
4) Below steps are allowed only if the bid amount of consumers is greater than the bid amount of prosumers.
5) If the d_i of the consumer is satisfied, it goes for the i+1 consumer.
6) If the j^{th} prosumer has no more Pcoins, then go for the j+1 prosumer.
7) If the d_i is pending and greater than or equal to the threshold value and also if the j^{th} prosumer has greater than or equal to the th_s value of Pcoin then Increment the allocation matrix of i^{th} consumer and j^{th} prosumer by threshold and decrement the s_j and d_i by threshold.
8) If the d_i is pending and less than the threshold value and also if the j^{th} prosumer has greater than or equal to the requested value of Pcoin by i^{th} consumer, then Increment the allocation matrix of i^{th} consumer and j^{th} prosumer by d_i and decrement the s_j by d_i and make d_i as 0.

Algorithm 1 Bid Allocation Mechanism

Require: M Consumers, N Prosumers,
d_i is the number of Pcoin i^{th} consumer(C_i) requires,
s_j is the number of Pcoin j^{th} prosumer(P_j) wants to sell
Ensure: *AM*
{The allocation matrix (AM) is the output of Algorithm}

1: $Th_s = \frac{\sum_{j=1}^{N} s_j}{M \cdot N}$
2: **for** $i \leftarrow 1$ to M **do**
3: **for** $j \leftarrow 1$ to N **do**
4: **if** $bidC_i >= bidP_j$ **then**
5: **if** $d_i == 0$ **then**
6: break;
7: **else if** $s_j == 0$ **then**
8: continue;
9: **else if** $d_i >= th_s$ **and** $s_j >= th_s$ **then**
10: $AM_{ij} += th_s$;
11: $d_i -= th_s$; $s_j -= th_s$;
12: **else if** $d_i < th_s$ **and** $s_j >= d_i$ **then**
13: $AM_{ij} += d_i$;
14: $s_j -= d_i$; $d_i = 0$;
15: **else**
16: $AM_{ij} += s_j$;
17: $d_i -= s_j$; $s_j = 0$;
18: **end if**
19: **end if**
20: **end for**
21: **end for**

Algorithm 2 Pricing Mechanism

Require: M Consumers, N Prosumers, AM,
$bidC_i$ is the bid price of One Pcoin of i^{th} consumer(C_i),
$bidP_j$ is the ask price of One Pcoin of j^{th} prosumer(P_j)
Ensure: PM
{The pay matrix(PM) is the output of the algorithm.}
1: **for** $i \leftarrow 1$ to M **do**
2: **for** $j \leftarrow 1$ to N **do**
3: **if** $i == M$ **then**
4: $PM_{ij} = AM_{ij} \cdot (bidP_j + bidC_i)/2$;
5: **else**
6: $PM_{ij} = AM_{ij} \cdot (bidP_j + bidC_{i+1})/2$;
7: **end if**
8: **end for**
9: **end for**

9) If the j^{th} prosumer has less Pcoins than the d_i or th_s then Increment the allocation matrix of i^{th} consumer and j^{th} prosumer by s_j and decrement the d_i by the s_j and make s_j as 0.

4) Pricing Mechanism: After the bid allocation is done, the next step is to create the pricing mechanism. We make use of the mean of consumer and prosumer bid prices and the Vickrey Auction to calculate the pricing mechanism. A Vickrey or second-price auction prevents the consumer from overpaying and also improves the competition, and optimizes future bids.

Suppose the i^{th} consumer receives x Pcoin from the j^{th} prosumer. Below is the amount the i^{th} consumer needs to pay to the j^{th} consumer.

$$pay_{ij} = x \cdot \left(\frac{bidP_j + bidC_{i+1}}{2} \right) \tag{14}$$

For the last consumer, the bid price paid will be the amount bided by them only. As per the Algorithm 2, a pay matrix (PM) is created, which contains the payment mapping of i^{th} consumer to the j^{th} prosumer. The pay matrix is updated with the help of (14) except for the last consumer, which pays the price bided by it. From the algorithm, it can be seen that the total payment received by the j^{th} prosumer is equal to the sum of the j^{th} column in the pay matrix. Also, the total payment given by the i^{th} consumer is equal to the sum of the i^{th} row in the pay matrix.

D. Temporary Leader Selection From Auction

Miner nodes are temporary leaders chosen from among all prosumers. One of these miner nodes is chosen as the super leader, who creates the block, and the other mining node is chosen as the verifier. The winners of the auction system are used to determine the temporary leaders. It is based on the original double auction method's concept. We are not using the sorting factor or threshold described above in this example. The goal of selecting the auction scheme leaders is to involve more prosumers and consumers in trade. The procedure for selecting a temporary leader through an auction is described below.

1) Sort the Consumers in decreasing order of their bid: $bidC_1 \geq bidC_2 \geq \cdots \geq bidC_m$.
2) Sort the Prosumers in increasing order of their bid: $bidP_1 \leq bidP_2 \leq \cdots \leq bidP_n$.
3) Let k be the largest "breakeven index" index such that $bidC_k \geq bidP_k$.
4) These first k prosumers are the winners of the double auction, and the group of these prosumers will act as our temporary leader for that time phase.

E. Super Leader Selection and Consensus Process

The "super leader" is the leader of all the temporary leaders. We have employed the proof of stake (PoS) consensus algorithm for block creation among the temporary leaders. The temporary leaders can increase their chances of being selected as super leaders by putting in a large amount of Pcoins. The temporary leaders put in a certain amount of Pcoins to indicate their interest in participating in the consensus process. Along with the amount of Pcoins, the concept of randomization is used to select the super leader. The probability of selection is influenced by the product of these two factors, which are the maximum Pcoins and the randomization. This means that temporary leaders with a large amount of Pcoins have a high chance of being selected as the super leader. By giving temporary leaders with large Pcoins a higher chance of being selected as super leaders, ETradeChain aims to encourage the nodes (prosumers) to generate the maximum possible green energy and avoid fossil fuel-based energy.

The Pcoins created in the last period of δT will be put at stake. The super leader gets remuneration for processing the transactions and creating the block. Algorithm 3 describes the PoS consensus for block creation and super leader selection. The algorithm for super leader selection and block creation works as follows:

1) All the nodes put their transactions in the transaction pool, which is accessible to all the other nodes. The transactions in the pool are verified by the temporary leaders.
2) The status of transactions in the pool is set to "pending" to indicate that they are waiting to be included in the next block creation. Once these transactions are included in the block, the status is updated to "processed."
3) All the temporary leaders put their recently generated Pcoins during δT time as a stake to participate in the super leader selection process. As per line 7 of the algorithm, the super leader is selected randomly out of all the temporary leaders; however, those who put in the maximum Pcoins will have the highest chance of being selected as the super leader. The selected super leader is called the validator.
4) We have given priority to nodes who put maximum Pcoins at stake because it will motivate the nodes (prosumers) to generate the maximum possible green energy and avoid fossil fuel-based energy.
5) The validator starts forging the block. First, it collects the verified transactions from the transaction pool, creates the block by following the intermediate processes

Algorithm 3 Super Leader Selection and Block Creation

Require: Pcoins {Pcoins are taken as stake for consensus}
Ensure: Leader Node, Block (BL)
{First, add all the transactions in the transaction pool and set their status as pending.}
while TRUE **do**
  TP.append ← new_transaction & TP.status ← pending
end while
{A loop to select the validator.}
while node!=validator **do**
  stake ← putStake(Pcoins)
  selectValidator(stake+randomization and node.status)
end while
if validator.isForging **then**
  stake.Locked(node.stake)
  while TP!=empty or no_of_transaction=516 **do**
    if TP.status == pending **then**
      Mrt ← createHashTree(TP.getAllTransactions)
      PBH ← H(PB) & BH = H(Mrt, T, PBH)
      BL ← createBlock(Mrt, T, PBH, BH)
      broadcast_block(BL)
      {return to the initial step again to create a new block.}
    else
      waits for new transactions.
    end if
  end while
end if
{Verifying the block created by the validator.}
while node.verifying **do**
  receive_block(BL) & verify_block(BL)
  if block == verified **then**
    stake.Unlocked(node.stake)
    node.status ← good
    reward ← transaction_fees
    TP.status ← processed
  else
    stake.deduct(node.stake) & node.status ← bad
  end if
end while

like Merkle roots, previous block hash, and block hash, and broadcasts the block in the network for verification.

6) After the verification, the newly created block is added to the existing blockchain, and the network is ready for the next round of blocks to be created, followed by the selection of the super leaders.
7) The verification is done by the other node in the network. If the other node finds any illegal activity by the leader, the first one to report it will receive the stake of the bad leader.

F. Remuneration Model

Block creation has been made more appealing to ETradeChain by compensating validators or super leaders for their efforts. The individual who contributes blocks to the chain receives a payment. Each block provided has a cost, which is referred to as transaction fees; hence, remuneration is decided by the blocks added by leaders. Therefore every transaction is subject to a transaction fee. If every block contains *N* transactions, then the general equation for total transaction fee (TTF) equals to

$$TTF = \left[\sum_{i=1}^{N} TF_i\right] \tag{15}$$

where *TF* is a transaction fee, and *N* is the number of transactions in some j^{th} block. For example, let there is one block created by the leader node, which contains 516 transactions, and the transaction fee is 0.1$ per transaction, then the *TTF* would be as $TTF = [\sum_{i=1}^{516}[0.1]]$, $TTF = 51.6\$$. This is how the remuneration of the miner node is done, and it gets benefits for the transaction processing and forging of the block.

G. Avoid Malicious Block Generation

In the previous section, we described the remuneration model to reward the leader node for creating the block and appending it to the blockchain. However, applying this approach increases the direct motivation of the leader node to bunch the verified transactions and put them into the block. However, it reduces the motivation of other verifier consensus nodes. So to solve this difficulty, ETradeChain employs a concept to encourage the verifier consensus nodes to honestly participate in the consensus process. Suppose any validator node has found that the newly created block is malicious or contains some malicious transactions (such as double-spent transactions). In that case, that node is awarded the total transaction fee *TTF* instead of the leader node, and the total Pcoins deposited by the leader node is also credited to that node. Additionally, the leader node is removed from the network as a punishment and marked as a malicious node. By applying this concept, all the miner nodes will actively participate in the consensus process, which will eliminate the possibility of various attacks.

V. Performance Analysis

The performance of ETradeChain is measured based on the five parameters: consensus delay, throughout, computational cost, energy consumption, storage, and communication cost. However, before the performance measurement, we have also detailed the implementation environment, i.e., our Testbed scenario.

A. Implementation Environment

We have developed a testbed that consists of 32 Raspberry Pi 4 model B (RP4) boards. Out of 32, 1 is represented as a smart grid (shown in the system model). 8 out of 32 RP4 boards act as prosumer nodes, and the rest of the RP4 boards act as consumers, where each prosumer is connected to 3 RP4 board consumers. The consumer RP4 board represents a device with the capability of a smart meter; the prosumer RP4 board acts as a smart meter and integrated unit. We chose the RP4 as the smart meter and integrated unit because it has

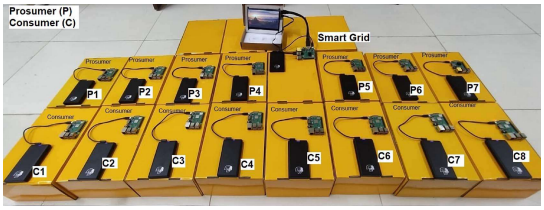


Fig. 2. Implementation environment for ETradeChain Testbed.

the capacity to process and store the security protocol. It has a micro-controller for data processing and a memory unit for data storage. The smart meter should also have these properties. The device also has the capability of communicating with other devices over a WiFi or Bluetooth wireless connection. We have used two types of RP4 boards with different configurations. The RP4 for consumers has 2 GB of memory and 4GB for prosumers. Except for the memory, both boards have the same software and hardware configurations.

The operating system installed in each RP4 is Raspberry Pi OS, also called Raspbian OS, a Debian-based operating system for RP4. Each RP4 is equipped with a 1.5 GHz Broadcom BCM2711 quad-core cortex-A72 processor. We have used the Java programming language to develop the programs for the testbed. Fig. 2 shows the implementation environment of ETradeChain. This figure shows seven prosumer RPs, eight consumer RPs, and one smart grid RP4 unit. We show only a few RP4s in our testbed image for clear visibility. Fig. 3 shows the internal working process of the proposed scheme in the RP4 module. Fig. 3(a) shows various features of the prosumer and consumer. Fig. 3(b) shows that the prosumer is generating the block. Fig. 3(c) shows the consensus process is taking place, i.e., the selection of a super leader.

B. Consensus Delay and Throughput

When evaluating ETradeChain, we look at the number of transactions per second, its time to reach an agreement, and how fair it is when selecting new leaders. The assessment results demonstrate that, without compromising the security level, it is feasible to reduce consensus latency and increase throughput by altering the configuration of the miner nodes, i.e., two-level leader selection, first auction-based temporary leader selection, and a second super leader. Nearly every statistic across all of the tested characteristics shows that ETradeChain is superior to the competition. The State-of-the-art blockchain methods have a common weakness: fork creation and high consensus delay. ETradeChain addressed these weaknesses. To lower the propagation delay in the blockchain network, we fix the block size at less than or equal to 1 MB. Each block contains a maximum of 516 transactions, making the block size less than 1 MB. Each block contains three components: the block header, transactions, and transaction count. The size of the block header is 80 bytes, the size of the transaction count is 4 bytes, and the remaining space, i.e., 10,48,492 bytes, is occupied by the transactions. Each transaction has a maximum of 2000 bytes. Therefore, each block can contain a maximum of 516 transactions. Fig. 4a shows the consensus delay of ETradeChain compared with state-of-the-art schemes. The result shows that the consensus delay of ETradeChain is approximately 90% low as compared to other blockchain protocols.

TABLE II
EXECUTION TIME OF DIFFERENT CRYPTOGRAPHIC FUNCTIONS (IN MILLISECONDS)

Operation	Raspberry Pi 4GB	Raspberry Pi 2 GB
SHA256 Hash function: $T_{H()}$	0.005	0.045
AES Crypto-system 192-bit: T_{AES}	5.96	7.71
ECC Crypto-system: T_{ECC}	1.6	3.9
ECDSA signature: T_{sign}	2.94	6.13
ECDSA verification: T_{veri}	2.86	5.63
point multiplication: $T_{mul(.)}$	5.0	11.0
Modulo operation: $T_{mod(.)}$	2.0	5.0

Fig. 4b depicts ETradeChain's throughput. When compared to other blockchain systems, ETradeChain performed exceptionally well. Changing the time gap between leader nodes changed the throughput of the ETradeChain. As the number of forgers grows, the likelihood that various forgers will have an equal probability of generating a block increases. The ETradeChain is fairer than other platforms. Every miner node competes with the rest of the nodes in state-of-the-art schemes in order to win the competition for the block generation reward. Many nodes may create blocks concurrently as a result of this competition, causing significant congestion in the network and the formation of unnecessary forks. In ETradeChain, the architecture first chooses the forger as a temporary leader and then as a super leader. This reduces congestion and increases throughput.

C. Computational Cost

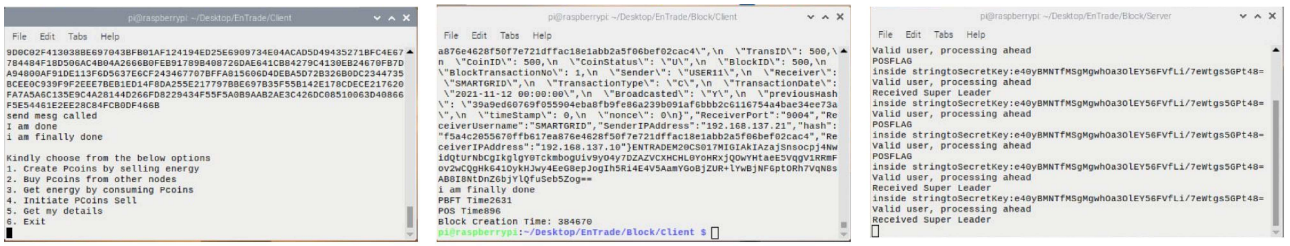
Computational time is the time taken by the Testbed for the execution. The measuring unit is milliseconds and is represented as T_{ct}. We illustrate the computational cost (also known as execution time) based on the operational costs for authentication, verification, communication, and block creation and verification. Table II shows different cryptographic functions and their computation time on both types of RP4 boards. (We have assumed that XOR operations take very negligible time compared to other operations). The time taken by all these cryptographic operations is based on the implementation environment, which we have defined in the previous section. Table III and Fig. 4c show the comparative computation cost of ETradeChain versus state-of-the-art schemes.

D. Energy Consumption

The energy consumption of a system is the amount of energy consumed by the system in order to carry out its operations of the system. During the execution process, the system requires a certain amount of energy in order to complete all of the phases of the scheme. The following is the formula for calculating energy consumption:

$$E_{rg} = T_{ct} * P_{CPU} \tag{16}$$

where P_{CPU} is the CPU power (5.6 W for RP4 4GB and 2.4 W for RP4 2 GB) and T_{ct} is the computational cost. The computation cost was already measured in the previous



(a) Prosumer Features (b) Prosumer Generating Block (c) Consensus algorithm processing

Fig. 3. ETradeChain working process on the RP4 devices. All of the above figures are the command line interface of RP4 doing all the required processing.

TABLE III
COMPARISON OF COMPUTATION COST OF ETRADECHAIN WITH STATE-OF-THE-ART SCHEMES

Scheme	Computation Cost Equation	Computation Cost on RP4 4GB (in Milliseconds)	Computation Cost on RP4 2GB (in Milliseconds)	Energy Consumption on RP4 4GB (in Joule)	Energy Consumption on RP4 2GB (in Joule)
[12]	$6T_{AES} + 2T_{sign} + 2T_{veri}$	47.36	69.78	265.22	167.47
[17]	$5T_{sign} + 5T_{veri}$	29	58.8	162.4	141.12
[18]	$6T_{H(.)} + 2T_{sign} + 2T_{veri}$	11.63	23.79	65.128	57.096
[19]	$11T_{H(.)} + 8T_{mul(.)}$	40.055	88.50	224.34	212.38
[29]	$5T_{H(.)} + 2T_{sign} + 2T_{veri}$	16.60	28.56	93.016	68.556
ETradeChain	$3T_{H(.)} + 1T_{sign} + 1T_{veri}$	5.815	11.895	32.54	28.54

TABLE IV
STORAGE FOR DIFFERENT CRYPTOGRAPHIC FUNCTIONS

Operation	Required storage
Secret key: S_k	40 Bytes
Hash function: $S_{H()}$	64 Bytes
Point multiplication: $S_{mul(.)}$	20 Bytes
ECC signature: S_{sign}	10 Bytes
ECC encryption/decryption: S_{ECC}	48 Bytes
AES encryption/decryption: S_{AES}	16 Bytes

section. Like computation cost, energy consumption is also measured on two platforms, i.e., RP4 with 4 GB and RP4 with 2 GB. Fig. 4d shows the comparison of ETradeChain with state-of-the-art schemes.

E. Storage Cost

Storage cost refers to the storage required for storing the values, such as the secret key, hash value, and other parameters. The measuring unit for storage cost is bytes and represented as S_{sc}. Table IV shows the storage requirement for various operations of the ETradeChain Testbed. In the ETradeChain, 1 hash $S_{H(.)}$, and one ECDSA signature S_{sign} are computed. Fig. 4e shows the comparative analysis of the storage of ETradeChain versus other schemes.

Table V shows the comparative storage cost of ETradeChain with state-of-the-art schemes. It is clear from the table that due to the very low number of lightweight cryptographic functions, the storage cost is very low in ETradeChain compared to the state-of-the-art schemes.

F. Communication Cost

Communication cost refers to the storage required for storing the communicating variables such as hash code, ciphertext, and other parameters. The measuring unit is a byte and is represented as C_{cc}. The operations and variables that have been used in communication between the smart meter and the miner node are the hash function and the ECDSA digital signature. Table V shows the comparative analysis of different mechanisms against ETradeChain. Fig. 4e shows the comparative analysis of the communication cost of ETradeChain versus other schemes.

VI. AUCTION AND SECURITY ANALYSIS

In this section, we have explained the auction property analysis and security analysis of ETradeChain. The security analysis of the ETradeChain justifies the prevention of double-spending attacks, the prevention of forking, fault tolerance, and non-repudiation. Below we have proven with the help of a lemma and proof that the system is protected against the attacks mentioned above. Table VI shows the security comparison of ETradeChain with the state-of-the-art approach where $\surd$ represents "Secure from Attack" and $\otimes$ illustrates "Not Secure from Attack".

The smart grid's self-defensive capability has been enhanced by many cyberattacks using a security framework based on blockchain, as described by Liang et al. [17]. The scheme used the consensus mechanism based on the PoW without any reward mechanism, i.e., mining is a competition among all miners, while there is no reward as an incentive for the miner who solves the puzzle problem first. The scheme seems to be secure from double-spending attacks. However, the scheme suffered from forking and non-repudiation attacks. Similarly, the scheme presented in [18] called Guardian used the PoW consensus algorithm. The scheme was also secured from double spending attacks but not from forking and non-repudiation attacks. The scheme is also not fault tolerant. The smart contract-based smart grid system was proposed in [19]. The scheme provides anonymous authentication; however, it is not secured from double-spending and fault tolerance. The issues of double-spending, forking, and non-repudiation were addressed in the scheme proposed in [29]. However,

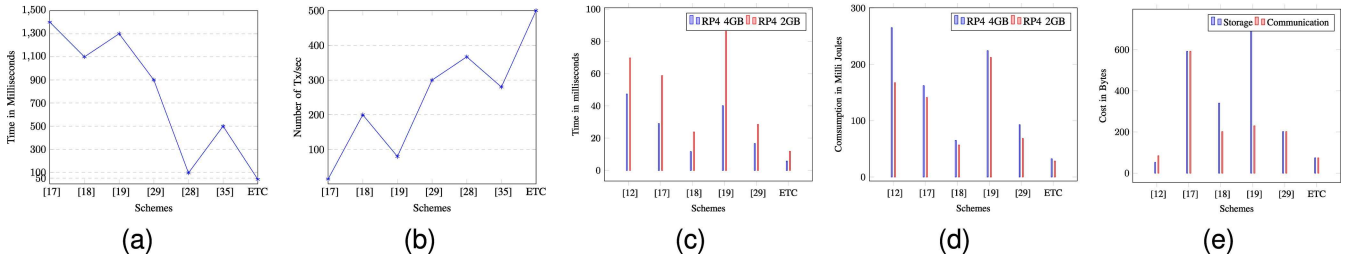


Fig. 4. (a) Consensus delay comparison of ETradeChain (ETC) with state-of-the-art schemes, (b) Throughput comparison of ETradeChain (ETC) with state-of-the-art schemes, (c) Computation cost of ETradeChain (ETC) vs state-of-the-art scheme, (d) Energy Consumption of ETradeChain (ETC) vs state-of-the-art scheme, (e) Storage and Communication cost of ETradeChain (ETC) vs state-of-the-art scheme.

TABLE V
COMPARISON OF STORAGE COST AND COMMUNICATION COST OF ETRADECHAIN WITH STATE-OF-THE-ART SCHEMES

Scheme	Storage Cost Equation	Storage Cost in MB	Communication Cost Equation	Communication Cost in MB
[12]	$2S_{AES} + 2S_{sign}$	52	$4S_{AES} + 2S_{sign}$	84
[17]	$8S_{H(.)} + 8S_{sign}$	592	$8S_{H(.)} + 8S_{sign}$	592
[18]	$5S_{H(.)} + 2S_{sign}$	340	$3S_{H(.)} + 1S_{sign}$	202
[19]	$10S_{H(.)} + 3S_{mul(.)}$	700	$ElementData = (230Bytes)$	230
[29]	$3S_{H(.)} + 1S_{sign}$	202	$3S_{H(.)} + 1S_{sign}$	202
ETradeChain	$1S_{H(.)} + 1S_{sign}$	74	$1S_{H(.)} + 1S_{sign}$	74

TABLE VI
SECURITY COMPARISONS OF ETRADECHAIN WITH STATE-OF-THE-ART SCHEMES

Attacks	[17]	[18]	[19]	[29]	[28]	[35]	*ETradeChain*
Double-Spending	√	√	⊗	√	⊗	⊗	√
Forking	⊗	⊗	√	√	⊗	√	√
Non-Repudiation	⊗	⊗	√	⊗	√	⊗	√

the scheme is not fault-tolerant for a large number of nodes (prosumers and consumers). The modified PBFT consensus algorithm was used on the scheme proposed in [28] and [35]. Both schemes are not secured from fault tolerance. The scheme [35] is also not secured from forking attacks. In contrast to the state-of-the-art scheme, the ETradeChain is secured from all four attacks.

Lemma 1: ETradeChain's auction is budget balanced.

Proof: From the Algorithm 2, we can see that the total payment received by the j^{th} prosumer is the sum of the j^{th} column in the pay matrix. Every column contains i rows. Therefore, the total payment received by j^{th} prosumer is $Pay_j = \sum_{i=1}^{M} PM_{ij}$. Similarly, total payment given by i^{th} consumer will be $Pay_i = \sum_{j=1}^{N} PM_{ij}$. The total payment received by all the Prosumers will be

$$TotalPay_j = \sum_{j=1}^{N}\sum_{i=1}^{M} PM_{ji} \tag{17}$$

Similarly, the total payment given by all the Consumers can be represented as,

$$TotalPay_i = \sum_{i=1}^{M}\sum_{j=1}^{N} PM_{ij} \tag{18}$$

From equation (17) and (18), we can see that the Total payment received by prosumers is equal to the total payment given by the consumers. Therefore, we proved that our system is budget balanced. ■

Lemma 2: ETradeChain's auction satisfies individual rationality.

Proof: Individual Rationality means no peer should lose by joining the auction. In our case, the utility of consumers will never be negative. As per the condition in Algorithm 1 line number 8, we are ensuring that in the allocation scheme, the bid price of the consumer should be greater than or equal to that of the prosumer. Also, considering the equation (14), we can say that the buyer will never pay more than the bid value, and the seller will never receive less than the asking value. Therefore, the utility of the buyer and seller will never be negative. It will be zero if they lose. Therefore, individual rationality is satisfied. ■

Lemma 3: The ETradeChain protocol prevents double-spending attacks.

Proof: Double-spending refers to the act of making two separate purchases with the same amount of money. Physical cash does not have this difficulty, but digital currency does. A transaction containing digital cash is broadcast to all nodes in the network for verification when using digital money. Before the transaction is confirmed, another node may duplicate it and rebroadcast it to the network. Due to the long confirmation time, the double-spending attack poses a serious threat to Bitcoin, Bitcoin-NG, and other blockchain protocols.

ETradeChain, on the other hand, consolidates all transactions into a single data block accessible only by the super leader node. No other node is permitted to construct a block while the leader node is generating it. As a result, in the ETradeChain, it is impossible to broadcast the same transaction multiple times. Additionally, ETradeChain also rewards honest miner nodes and punishes the malicious node, which encourages the honest node to find the malicious blocks of malicious transactions. Hence, the possibilities of double-spending are negligible. ■

Lemma 4: The ETradeChain protocol ensures no fork generation in the blockchain, or that there is no possibility of parallel block generation with the same set of transactions.

Proof: The fork generation can be easily understood. The minimum difference between the number of blocks generated by a malicious node and a fair node to confirm a transaction and lower the blockchain network's vulnerability probability than the threshold $\epsilon \in (0, 1]$ is defined as follows:

$$\alpha^*(\epsilon) = min\{\alpha : P(Q > \alpha) \leq \epsilon\} \tag{19}$$

$$\alpha^* = -\log(\epsilon)/\log(\lambda/\Lambda) \tag{20}$$

where P is the probability of a malicious node generating the block before the fair node. The high value of α^* has the efficiency to counterattack the intentional forks. To ensure blockchain network stability, fair node computing power must be increased. The computing power required to finish the forging in the allotted time frame is equal to the fair node's effective computational power. ETC's fair leader node has a lot more computing power than the malicious node in order to generate a block. As a result, an attacker is unable to fork the ETradeChain. ETradeChain also does not enable many nodes to forge the same block simultaneously, meaning that only the fair super leader node can forge the block at any one time. Thus, fork formation in the ETradeChain is less likely to occur.

However, ETradeChain has a time period in which the leader creates the blocks. This is in contrast to other blockchain protocols like Bitcoin-NG [44], and OmniLedger [45] that have fork generation problems. After the interval has passed, the next node in the network generates the block after selecting it as the super leader node. Due to this, ETradeChain does not have the problem of fork formation because two forger nodes cannot generate the block simultaneously. As a result, ETradeChain has no issues with fork creation. ■

Lemma 5: The ETradeChain provides non-repudiation of transaction transmission.

Proof: Every transaction and block is signed using the sender's private key, which is only known to the sender. Hence, after sending the transaction or block signed by a private key, the sender can not deny it. The ETradeChain supports non-repudiation of transactions and blocks, which can easily be justified using the proof of origin (PoO) and mandatory proof of receipt (PoR) [46]. In order to establish mutual non-repudiation, both the mandatory PoO and PoR must be included. Once messages are generated and received, the PoO must link their originator's identity with that information in order to not refuse it. The recipient's identification is likewise linked to the PoR. Neither must be able to be disproved or tampered with. The ECDSA-based digital signature has allowed us to achieve both of these goals. The proof can be described as follows:

$$PoO = Sign\big(SK_{pri}[\mathbb{O}_{M_{ID}}, M_{ID}, TL, Tr_i], T_1\big) \tag{21}$$

$$PoR = Sign\big(TL_{pri}[\mathbb{O}_{TL}, TL, M_{ID}, H(PoO)], T_2\big) \tag{22}$$

where, $\mathbb{O}_{M_{ID}}, \mathbb{O}_{TL}$ indicates that this structure is represented as PoO and PoR, respectively. M_{ID} and TL (temporary leader) are the originator and recipient (i.e., prosumer and consumer), respectively. SK_{pri} and TL_{pri} are the private key of the smart meter and miner node. T_1, T_2 are the timestamps, and Tr_i is the i^{th} the transaction. The prosumer and consumer identification is associated with the PoO and PoR. From the above two equations, it is obvious that ETradeChain supports non-repudiation. It is important to show it since it aids in identifying fraudulent nodes that attempt to broadcast false transactions. Hence, the ETradeChain withstands non-repudiation. ■

VII. Conclusion and Future Work

The ETradeChain we've introduced offers a secure and efficient energy trading system for the local energy market (LEM) within the smart grid. This LEM empowers renewable-energy-equipped residential units to benefit from energy trading amongst themselves. ETradeChain addresses various LEM security challenges, including transaction integrity, user authentication, double spending, forking, and non-repudiation. We've introduced a virtual coin, Pcoin, anchored to energy generation. These Pcoins, generated by prosumers, facilitate energy trading in the LEM. Our ETradeChain employs a modified double auction mechanism, using Pcoin as a stake to achieve consensus on energy transactions. By leveraging blockchain technology, ETradeChain showcases the feasibility of real-time P2P trading. Experimental results indicate that ETradeChain reduces consensus delay by up to 90%, achieves 60% higher throughput, and ensures 80% lower computational overhead, alongside 70-80% reductions in storage and communication overheads.

Looking ahead, we aim to create a SmartGrid prototype to test ETradeChain's applicability in real-world settings. We also intend to incorporate modules for energy consumption prediction and adapt ETradeChain for various other domains.

References

[1] M. B. Mollah et al., "Blockchain for future smart grid: A comprehensive survey," *IEEE Internet Things J.*, vol. 8, no. 1, pp. 18–43, 2021.

[2] J. Bao, D. He, M. Luo, and K.-K. R. Choo, "A survey of blockchain applications in the energy sector," *IEEE Syst. J.*, vol. 15, no. 3, pp. 3370–3381, 2020.

[3] J. Ping, Z. Yan, and S. Chen, "A privacy-preserving blockchain-based method to Optimize energy trading," *IEEE Trans. Smart Grid*, vol. 14, no. 2, pp. 1148–1157, 2023.

[4] S. H. Alsamhi et al., "Blockchain-empowered security and energy efficiency of drone swarm consensus for environment exploration," *IEEE Trans. Green Commun. Netw.*, vol. 7, no. 1, pp. 328–338, 2023.

[5] X. Ma, D. Xu, and K. Wolter, "Blockchain-enabled feedback-based combinatorial double auction for cloud markets," *Future Gener. Comput. Syst.*, vol. 127, pp. 225–239, 2022.

[6] Z. Chen, W. Ding, Y. Xu, M. Tian, and H. Zhong, "Fair auctioning and trading framework for cloud virtual machines based on blockchain," *Comput. Commun.*, vol. 171, pp. 89–98, 2021.

[7] S. H. Alsamhi et al., "Drones edge intelligence over smart environments in B5G: Blockchain and federated learning synergy," *IEEE Trans. Green Commun. Netw.*, vol. 6, no. 1, pp. 295–312, 2022.

[8] T. Ha, D. Lee, C. Lee, and S. Cho, "VCG auction mechanism based on block chain in smart grid," in *2021 Int. Conf. Inf. Netw. (ICOIN)*. IEEE, 2021, pp. 465–468.

[9] S. Singla, A. Dua, N. Kumar, and S. Tanwar, "Blockchain-based efficient energy trading scheme for smart-grid systems," in *Proc. IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2020, pp. 1–6.

[10] H. T. Doan, J. Cho, and D. Kim, "Peer-to-peer energy trading in smart grid through blockchain: A double auction-based game theoretic approach," *Ieee Access*, vol. 9, pp. 49206–49218, 2021.

[11] D. Li, Q. Yang, W. Yu, D. An, Y. Zhang, and W. Zhao, "Towards differential privacy-based online double auction for smart grid," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 971–986, 2019.

[12] S. Li, K. Xue, D. S. Wei, H. Yue, N. Yu, and P. Hong, "SecGrid: A secure and efficient SGX-enabled smart grid system with rich functionalities," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1318–1330, 2020.
[13] K. Xue et al., "PPSO: A privacy-preserving service outsourcing scheme for real-time pricing demand response in smart grid," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2486–2496, 2018.
[14] Z. Guan et al., "APPA: An anonymous and privacy preserving data aggregation scheme for fog-enhanced IoT," *J. Netw. Comput. Appl.*, vol. 125, pp. 82–92, 2019.
[15] P. Gope and B. Sikdar, "Privacy-aware authenticated key agreement scheme for secure smart grid communication," *IEEE Trans. Smart Grid*, vol. 10, no. 4, pp. 3953–3962, 2018.
[16] S. Khan, R. Khan, and A. H. Al-Bayatti, "Secure communication architecture for dynamic energy management in smart grid," *IEEE Power Energy Technol. Syst. J.*, vol. 6, no. 1, pp. 47–58, 2019.
[17] G. Liang, S. R. Weller, F. Luo, J. Zhao, and Z. Y. Dong, "Distributed blockchain-based data protection framework for modern power systems against cyber attacks," *IEEE Trans. Smart Grid*, vol. 10, no. 3, pp. 3162–3173, 2018.
[18] A. Jindal, G. S. Aujla, N. Kumar, and M. Villari, "GUARDIAN: Blockchain-based secure demand response management in smart grid system," *IEEE Trans. Services Comput.*, vol. 13, no. 4, pp. 613–624, 2019.
[19] J. Wang, L. Wu, K.-K. R. Choo, and D. He, "Blockchain-based anonymous authentication with key management for smart grid edge computing infrastructure," *IEEE Trans. Ind. Informat.*, vol. 16, no. 3, pp. 1984–1992, 2019.
[20] Z. Ning, H. Chen, X. Wang, S. Wang, and L. Guo, "Blockchain-enabled electrical fault inspection and secure transmission in 5G smart grids," *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 1, pp. 82–96, 2022.
[21] S. Karumba, S. S. Kanhere, R. Jurdak, and S. Sethuvenkatraman, "HARB: A hypergraph-based adaptive consortium blockchain for Decentralized energy trading," *IEEE Internet Things J.*, vol. 9, no. 16, pp. 14216–14227, 2022.
[22] T. Wang, J. Guo, S. Ai, and J. Cao, "RBT: A distributed reputation system for blockchain-based peer-to-peer energy trading with fairness consideration," *Appl. Energy*, vol. 295, p. 117056, 2021.
[23] T. AlSkaif, J. L. Crespo-Vazquez, M. Sekuloski, G. van Leeuwen, and J. P. S. Catalão, "Blockchain-based fully peer-to-peer energy trading strategies for residential energy systems," *IEEE Trans. Ind. Informat.*, vol. 18, no. 1, pp. 231–241, 2022.
[24] J. Ping, Z. Yan, and S. Chen, "A privacy-preserving blockchain-based method to optimize energy trading," *IEEE Trans. Smart Grid*, vol. 14, no. 2, pp. 1148–1157, 2023.
[25] K. Christidis, D. Sikeridis, Y. Wang, and M. Devetsikiotis, "A framework for designing and evaluating realistic blockchain-based local energy markets," *Appl. Energy*, vol. 281, p. 115963, 2021.
[26] H. Wang, S. Ma, C. Guo, Y. Wu, H.-N. Dai, and D. Wu, "Blockchain-based power energy trading management," *ACM Trans. Internet Technol. (TOIT)*, vol. 21, no. 2, pp. 1–16, 2021.
[27] M. Foti, C. Mavromatis, and M. Vavalis, "Decentralized blockchain-based consensus for optimal power flow solutions," *Appl. Energy*, vol. 283, p. 116100, 2021.
[28] Q. Yang and H. Wang, "Privacy-preserving transactive energy management for IoT-aided smart homes via blockchain," *IEEE Internet Things J.*, vol. 8, no. 14, pp. 11463–11475, 2021.
[29] Z. Guan, X. Lu, W. Yang, L. Wu, N. Wang, and Z. Zhang, "Achieving efficient and privacy-preserving energy trading based on blockchain and ABE in smart grid," *J. Parallel Distrib. Comput.*, vol. 147, pp. 34–45, 2021.
[30] S. Chen et al., "A trusted energy trading framework by marrying blockchain and optimization," *Adv. Appl. Energy*, vol. 2, p. 100029, 2021.
[31] S. Chen, L. Zhang, Z. Yan, and Z. Shen, "A distributed and robust security-constrained economic dispatch algorithm based on blockchain," *IEEE Trans. Power Syst.*, vol. 37, no. 1, pp. 691–700, 2021.
[32] J. Yang, A. Paudel, and H. B. Gooi, "Compensation for power loss by a proof-of-stake consortium blockchain microgrid," *IEEE Trans. Ind. Informat.*, vol. 17, no. 5, pp. 3253–3262, 2020.
[33] Y. Cao, X. Ren, C. Qiu, X. Wang, H. Yao, and F. R. Yu, "A multi-agent reinforcement learning approach for blockchain-based electricity trading system," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2021, pp. 1–6.
[34] J. Yang, A. Paudel, J. Dai, and H. B. Gooi, "A mining-rewarding mechanism for peer-to-peer energy trading blockchain," in *Proc. IEEE PES Innovative Smart Grid Technol.-Asia (ISGT Asia)*. IEEE, 2021, pp. 1–5.
[35] J. Guo, X. Ding, and W. Wu, "An architecture for distributed energies trading in byzantine-based blockchains," *IEEE Trans. Green Commun. Netw.*, vol. 6, no. 2, pp. 1216–1230, 2022.
[36] Z. Shi, C. de Laat, P. Grosso, and Z. Zhao, "When blockchain meets auction models: A survey, some applications, and challenges," *arXiv preprint arXiv:2110.12534*, 2021.
[37] M. U. Hassan, M. H. Rehmani, and J. Chen, "Optimizing blockchain based smart grid auctions: A green revolution," *IEEE Trans. Green Commun. Netw.*, vol. 6, no. 1, pp. 462–471, 2021.
[38] C. Lin, D. He, X. Huang, and K.-K. R. Choo, "OBFP: Optimized blockchain-based fair payment for outsourcing computations in cloud computing," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 3241–3253, 2021.
[39] M. U. Hassan, M. H. Rehmani, and J. Chen, "DEAL: Differentially private auction for blockchain-based microgrids energy trading," *IEEE Trans. Services Comput.*, vol. 13, no. 2, pp. 263–275, 2019.
[40] H. Qiu and T. Li, "Auction method to prevent bid-rigging strategies in mobile blockchain edge computing resource allocation," *Future Gener. Comput. Syst.*, vol. 128, pp. 1–15, 2022.
[41] M. M. P. Chowdhury, C. Kiekintveld, S. Tran, and W. Yeoh, "Bidding in periodic double auctions using heuristics and dynamic monte carlo tree search." in *Int. Joint Conf. Artif. Intell. (IJCAI)*, 2018, pp. 166–172.
[42] T. D. Nguyen and M. T. Thai, "A blockchain-based iterative double auction protocol using multiparty state channels," *ACM Trans. Internet Technol. (TOIT)*, vol. 21, no. 2, pp. 1–22, 2021.
[43] A. P. Sarr, P. Elbaz-Vincent, and J.-C. Bajard, "A new security model for authenticated key agreement," in *Int. Conf. Security Cryptography Netw.*. Springer, 2010, pp. 219–234.
[44] I. Eyal, A. E. Gencer, E. G. Sirer, and R. Van Renesse, "{Bitcoin-NG}: A scalable blockchain protocol," in *13th USENIX Symp. networked Syst. design implementation (NSDI 16)*, 2016, pp. 45–59.
[45] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. Ford, "Omniledger: A secure, scale-out, decentralized ledger via sharding," in *Proc. IEEE Symp. Security Privacy (SP)*. IEEE, 2018, pp. 583–598.
[46] T. Coffey and P. Saidha, "Non-repudiation with mandatory proof of receipt," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 26, no. 1, pp. 6–17, 1996.

Umang Rajendra Barbhaya received the B.Tech. degree in computer engineering from the K. J. Somaiya College of Engineering, Mumbai, India, in 2018. He is currently pursuing the M.Tech. degree in computer science and engineering with the Indian Institute of Technology Jodhpur, Jodhpur, India. His research interests include blockchain, deep learning, cloud computing, and smart city applications.

Lokendra Vishwakarma (Graduate Student Member, IEEE) received the B.E. degree in information technology from Samrat Ashok Technological Institute, Vidisha, India, in 2009, and the M.Tech. degree in computer science and engineering from Rajiv Gandhi Prodyogiki Vishwavidyalaya, Bhopal, India, in 2014. He is a Ph.D. Research Scholar with the Department of Computer Science and Engineering, Indian Institute of Technology Jodhpur, Jodhpur, India. His research interests include cryptography, network security, blockchain, and smart city applications.

Debasis Das (Senior Member, IEEE) received the Ph.D. degree in computer science and engineering from the Indian Institute of Technology (IIT) Patna, India. He joined as an Assistant Professor with the Department of Computer Science and Engineering, IIT Jodhpur, Jodhpur, India, in 2019. He has published 100 research papers in top international journals and top conferences. His research interests include communication and networking, VANETs, machine learning, the Internet of Vehicles, blockchain, and network security.