Joan Ferré Queralt

Hierarchical Smart Contracts to Prevent User Profiling in Decentralized Energy Distribution Systems

Final Master's Project

Directed by:

Dr. Jordi Castellà Roca and Dr. Alexandre Viejo

Master's Degree in Computer Security Engineering and Artificial Intelligence

UNIVERSITAT ROVIRA i VIRGILI

Escola Tècnica Superior d'Enginyeria

Tarragona, 2024

Contents

Abstract

This work presents an innovative energy trading system that leverages hierarchical smart contracts to enhance privacy protection for users in decentralized energy distribution networks. The proposed system utilizes blockchain technology to ensure the security, transparency, and immutability of transactions, thereby addressing key vulnerabilities present in traditional energy grid systems. Smart contracts within the system automate the execution of energy trades and enforce accurate verification of each transaction, eliminating the need for intermediaries and reducing the risk of fraud or errors. Furthermore, the hierarchical structure of these contracts is designed to obscure user identities and energy consumption patterns, thereby preventing unauthorized profiling or data breaches. In addition to these privacy-preserving features, the system incorporates mechanisms to detect and penalize dishonest behavior among users, maintaining the overall integrity and fairness of the energy market. The solution also emphasizes scalability and adaptability, making it suitable for implementation in various energy markets and capable of accommodating the evolving demands of modern energy systems. By combining robust privacy safeguards with advanced technological frameworks, this system offers a secure, efficient, and equitable approach to decentralized energy trading.

***Keywords*—** Smart grid, Smart contracts, Security, Privacy

1 Introduction

The advent of smart grid technology represents a transformation in the way electricity is generated, distributed, and consumed [14]. Unlike traditional electricity grids, which are characterized by a one-way flow of electricity from centralized power plants to consumers, smart grids leverage advanced information and communication technologies to enable a more interactive, efficient, and resilient energy system. This paradigm shift is driven by the increasing integration of renewable energy sources, the growing demand for more reliable electricity supply, and the necessity to improve energy efficiency and sustainability [9].

The transition towards a smart grid infrastructure is seen as a necessary evolution to accommodate the growing complexities of modern electricity demands. As urbanization intensifies and energy consumption continues to rise, traditional grid systems face significant challenges in meeting these demands while maintaining environmental sustainability. Smart grids, with their capacity for real-time data analysis and adaptive energy management, provide a viable solution to these challenges. By optimizing the balance between energy supply and demand, smart grids can reduce the reliance on fossil fuels, thus decreasing the carbon footprint of electricity generation. Additionally, the integration of renewable energy sources into the grid ensures a more diversified and sustainable energy portfolio, which is critical in the global effort to combat climate change.

At the core of the smart grid concept is the integration of distributed energy resources (DERs), such as solar panels, wind turbines, and energy storage systems [26]. These resources are often located close to the point of consumption, enabling prosumers—entities that both produce and consume electricity—to actively participate in the energy market [36]. The bidirectional flow of electricity and information in smart grids facilitates real-time monitoring, dynamic pricing, and load balancing, thereby enhancing the grid's ability to respond to fluctuations in supply and demand [30]. This interactivity and adaptability are essential for accommodating the variable nature of renewable energy sources and for optimizing energy usage patterns, ultimately leading to a more sustainable and resilient energy infrastructure.

The use of advanced metering infrastructure (AMI) in smart grids allows for detailed monitoring of energy consumption and production. Smart meters, which are key components of AMI, collect granular data on energy usage and transmit it to utilities and other entities in the grid [7]. This detailed data collection enables utilities to perform more accurate demand forecasting, optimize grid operations, and implement demand response programs that incentivize consumers to adjust their energy usage during peak periods. However, one of the fundamental challenges in smart grid systems is maintaining the privacy of prosumers during energy transactions. As more personal and usage data are generated and shared across the network, there is an increasing risk of privacy breaches. Prosumers' energy consumption patterns can reveal sensitive information about their daily routines, occupancy, and lifestyle. Ensuring that this data is protected from unauthorized access and analysis is critical to gaining user trust and fostering wider adoption of smart grid technologies [32].

Furthermore, the development of smart grid technologies is accompanied by a growing need for robust cybersecurity measures. As smart grids become more interconnected and reliant on digital technologies, they become increasingly vulnerable to cyber-attacks that could disrupt the stability and reliability of the electricity supply. Addressing these cybersecurity challenges requires a comprehensive approach that includes the implementation of secure communication protocols, the use of encryption technologies, and the development of advanced threat detection and response systems. Ensuring the cybersecurity of smart grids is not only vital for protecting critical infrastructure but also for maintaining public trust in these advanced energy systems.

In this work, we propose a novel energy trading system designed to enhance the privacy and security of prosumers within a smart grid. Our system leverages blockchain technology and smart contracts to facilitate secure and efficient energy transactions. We address key privacy concerns by implementing mechanisms that protect prosumers' identities and consumption data from unauthorized access and analysis.

1.1 Related work

Smart grid systems have been extensively studied in recent years, with various implementations focusing on different aspects such as dynamic pricing, electricity load balancing, microgrids (smaller groups of prosumers inside a smart grid), and trading between prosumers. In this work we focus on the privacy of prosumers during the electricity trading.

The concept of a smart grid was first introduced in a seminal report by the U.S. Department of Energy in 2007 [5]. This report outlined the vision for a modernized grid that could integrate various distributed energy resources, improve reliability and efficiency, and enable two-way communication between utilities and consumers. Later work by Reddy et al. [40] delved deeper into communication standards for smart grids, focusing on enabling real-time data exchange and control functionalities within the traditional centralized grid architecture. However, as smart grids evolve and incorporate more distributed elements, security concerns regarding centralized control become increasingly prominent. Distributed systems, with their inherent redundancy and tamper-proof nature, offer potential solutions to these security challenges, as discussed in [34] and listed in Table 1.

Objective	**Meaning**
Confidentiality	Users should be anonymous
Integrity	Records shouldn't be modified
Authentication	Records should be signed
Authorization	Only authorized users should be able to post records
Transparency	All records should be visible
Availability	System should be working at all times
Immutability	Records should be non-modifiable

Table 1: Distributed objectives

While some existing proposals, such as those presented in [46, 48], utilize distributed systems for facilitating peer-to-peer energy transactions, they often rely on a central entity to match consumers and producers. This introduces a single point of failure and replicates some of the security concerns associated with centralized solutions.

A significant step towards a decentralized electricity trading system was presented in [33]. The authors proposed an energy currency based on blockchain technology, using the work of Satoshi Nakamoto in [35]. This approach paves the way for seamless integration of blockchain and smart contracts within the smart grid, enabling secure and transparent peer-to-peer energy trading without the need for a central authority. Since this initial proposal, numerous studies have explored various blockchain architectures for smart grid implementation, as evidenced by the work presented in [1, 2, 20, 31, 33, 45]. These studies showcase the versatility and potential of blockchain technology in revolutionizing the energy sector.

However, the aforementioned blockchain solutions employ a Proof-of-Work (PoW) consensus mechanism. While PoW offers strong security guarantees, it is known for its high energy consumption due to the substantial computing power required to validate transactions [6]. This high energy footprint contradicts the principles of a sustainable smart grid and renders PoW unsuitable for large-scale adoption in this domain. Therefore, exploring alternative consensus mechanisms that are more energy-efficient while maintaining system integrity and security should be a priority.

There are other works which don't use PoW and instead use other consensus mechanisms, such as those presented in [18, 41], which also explore the integration of blockchain technology into energy systems without relying on PoW. However, these proposals often lack a strong focus on other aspects such as User anonymity, which refers to concealing the identity of participants in energy transactions, and user privacy, which encompasses the protection of sensitive information such as energy consumption patterns. These are critical aspects for ensuring user trust and adoption within a smart grid trading system. The lack of adequate privacy measures can lead to security vulnerabilities, as sensitive user data could be susceptible to unauthorized access or profiling, as highlighted in [49].

By analyzing user's transactions, even if they are anonymous, an attacker can learn patterns from all the available information and infer patterns that can identify certain users. Works such as [25] use anonymous transactions to provide privacy for users but don't conceal the electricity consumed or produced, which can lead to profiling attacks.

Permissioned blockchains, as implemented in [13, 12], offer a partial solution of the privacy requirement by restricting network participation to authorized entities. This approach can mitigate the risk of external attackers identifying users through transaction analysis. However, it does not address the concern of internal users potentially deducing user identities from within the permissioned network.

In response to that, other proposals such as [21], [17], [42] and [43] have emerged recently. In this way,

the solutions presented in [21], [17] and [43] take into account the anonymity of the users, however they do not verify whether the production or consumption claimed by the users is truly the one being specified. As a result, a dishonest user could potentially submit a fake energy production record, which can mislead the system and other users, leading to undesirable outcomes.

On the other hand, [42] uses a reputation system to punish those users who misbehave. However, in this scheme no one possesses the relationship between the smart grid accounts and real identity of the users and, hence, it is not capable of effectively applying coercing measures to the misbehaving user.

The latter problem also occurs in works such as [22, 27], where authors use anonymity to protect microgrid transactions and authorities are unable to identify a user in case of misbehavior.

Other works such as [28, 19] use aggregation methods with homomorphic encryption in blockchain. However, the exact electricity usage of users is lost in the ledger, hindering traceability and transparency.

1.2 Contribution and organization

This work proposes a new energy trading system based on a smart grid that empowers users to both consume and produce electricity using distributed energy resources. The proposed system is designed to address the limitations and vulnerabilities of traditional energy grids by leveraging the decentralized nature of blockchain technology.

One of the most significant improvements of this system over existing literature is its approach to user privacy. In conventional smart grid systems, detailed energy consumption data is often collected and analyzed by central entities, raising concerns about potential misuse or unauthorized access to sensitive information. Our system mitigates these risks by implementing privacy-preserving mechanisms that allow users to participate in energy trading without revealing their identities or consumption patterns. This is achieved through the use of hierarchical smart contracts, which ensure that while transactions are recorded and validated, the specific details that could lead to user profiling are obscured.

Furthermore, the system addresses the issue of accuracy in record-keeping. By utilizing smart contracts, the proposed solution guarantees that all energy transactions are automatically recorded in a secure and tamper-proof manner. These smart contracts are designed to handle complex logic, ensuring that every transaction is executed precisely as intended, without the possibility of manual errors or fraudulent alterations. This level of accuracy is critical in scenarios where the precise calculation of energy usage and debt settlement is necessary for maintaining the integrity of the system.

Additionally, the system incorporates mechanisms for identifying and penalizing misbehaving users.

In decentralized systems, the risk of users attempting to game the system—such as by falsely reporting energy production or consumption—can undermine the overall trust and efficiency of the network. Our approach includes a combination of automated monitoring through smart contracts and intervention by trusted entities, who have the authority to verify transactions and enforce penalties where necessary. This ensures that while user privacy is protected, the system remains secure and fair for all participants.

The rest of the paper is organized as follows. Section 2 provides a detailed background of the key technologies employed in this work, including blockchain, smart contracts, and distributed energy resources. Section 3 presents the system's architecture, outlining the various components and their interactions, along with an overview of the fundamental concepts and definitions necessary to understand the proposed scheme. In Section 4, we delve into the specific details of the system components and the step-by-step process involved in the energy trading system, from initialization to debt settlement. Section 5 conducts a thorough analysis of the security and privacy threats, evaluating how the proposed solution mitigates these risks and ensures the protection of user data. In Sections 7, 8 and 9, we assess the performance of the system through both distributed implementation and simulation studies, providing insights into its scalability, efficiency, and practical feasibility. Finally, Section 10 summarizes the findings, discussing the implications of the proposed work and suggesting potential avenues for future research and development.

2 Background

In this section, we introduce the technologies employed in our work, providing a foundational understanding of the key systems that have significantly influenced our research.

2.1 Smart Grid

A smart grid represents an advanced version of the traditional energy grid, characterized by the two-way flow of information and electricity. Unlike conventional grids, where electricity flows unidirectionally from producers to consumers, smart grids enable electricity to flow between consumers, who can also act as producers. These users are referred to as *prosumers*.

The concept of the smart grid (SG) originated with the idea of advanced metering infrastructure (AMI), aimed at enhancing demand-side management and energy efficiency. SGs also offer robust protection against malicious sabotage and natural disasters, thereby ensuring grid reliability and self-healing capabilities [39].

According to the National Institute of Standards and Technology [16] and further detailed in [8], the benefits of smart grids include:

- Improved power reliability and quality;
- Optimized facility utilization, reducing the need for backup (peak load) power plants;
- Enhanced capacity and efficiency of existing electric power networks;
- Increased resilience to disruptions;
- Predictive maintenance and self-healing responses to system disturbances;
- Expanded deployment of renewable energy sources;
- Accommodation of distributed power sources;
- Automated maintenance and operation;
- Reduced greenhouse gas emissions through the integration of electric vehicles and new power sources;
- Lower oil consumption by reducing inefficient peak usage generation;
- Enhanced grid security;
- Facilitation of the transition to plug-in electric vehicles and new energy storage options;
- Increased consumer choice;
- Enabling new products, services, and markets.

The potential benefits of smart grids address many of the current issues within the energy system. Recent years have seen numerous proposals for smart grid architectures aimed at realizing these benefits.

2.2 Blockchain

Blockchain technology emerged as a revolutionary innovation, providing decentralized, secure, and transparent solutions across various industries. Its foundations lie in cryptography and computer science, with the first practical implementation being Bitcoin, introduced in 2008 by the pseudonymous Satoshi Nakamoto [35].

A blockchain is a distributed, decentralized ledger that records transactions securely, tamper-resistantly, and verifiably. The architecture of a blockchain consists of several key components [29, 44]:

- **Blocks**: Each block contains a list of transactions, a timestamp, a reference to the previous block (parent block), and a unique identifier known as the block hash. The block hash, generated using cryptographic algorithms, ensures the integrity of the block's contents.

- **Chain**: Blocks are linked in a linear, chronological order, forming a chain. This structure ensures that altering a block's data would require modifying all subsequent blocks, which is computationally impractical, thus providing security against tampering.
- **Consensus Mechanism**: This set of rules and protocols governs how new blocks are added to the blockchain. Common mechanisms include Proof of Work (PoW) [35], Proof of Stake (PoS) [23], and Delegated Proof of Stake (DPoS) [24].
- **Nodes**: A blockchain network comprises nodes, which are computers that store a copy of the blockchain and participate in the consensus process. This distributed nature ensures no single point of failure, enhancing security and reliability.

Blockchain employs various security features to ensure system robustness [50]:

- **Cryptographic Hash Functions**: These functions generate unique block hashes and transaction identifiers, providing data integrity and security. Commonly used hash functions include SHA-256 (Bitcoin) and Ethash (Ethereum) [47].
- **Digital Signatures**: Transactions are secured using digital signatures generated with asymmetric cryptography, allowing users to sign and verify transactions with their private and public keys, ensuring data authenticity and non-repudiation.
- **Immutability**: The blockchain's structure ensures that once data is recorded, it is nearly impossible to alter, achieved through cryptographic hash functions, the linear chain structure, and the consensus mechanism.
- **Decentralization**: The distributed nature of blockchain systems eliminates the need for central authorities, reducing the risk of censorship, fraud, and single points of failure.

2.3 Smart Contracts

Among the various applications of blockchain technology, smart contracts stand out as one of the most notable. Smart contracts are self-executing computer programs that automatically execute when predetermined conditions are met, eliminating intermediaries and reducing the risk of fraud or disputes.

Notable features of smart contracts include:

- **Automation and Self-execution**: Smart contracts are self-executing and require minimal human intervention. They are coded with predetermined conditions, and once these conditions are met, the contract automatically executes the terms outlined in the agreement.

- **Trust and Security**: Built on blockchain technology, smart contracts operate in a decentralized and transparent environment, ensuring all parties involved have access to the same information, fostering trust, and reducing the likelihood of disputes.
- **Cost-efficiency**: By eliminating intermediaries and streamlining processes, smart contracts can significantly reduce transaction costs and administrative expenses.
- **Interoperability**: Smart contracts can interact with other blockchain-based systems and data sources, enabling seamless cross-platform integration and data exchange.
- **Customization**: They can be tailored to meet the specific requirements of various industries and applications, ensuring flexibility and adaptability.

2.4 IOTA

IOTA is a distributed ledger technology (DLT) specifically designed for the Internet of Things (IoT) and machine-to-machine (M2M) communication. It addresses the unique challenges of IoT systems by providing a secure, scalable, and feeless solution [38].

At the core of IOTA's infrastructure is a Directed Acyclic Graph (DAG) called the Tangle. The Tangle is a decentralized data structure that enables parallel transaction processing. Unlike traditional blockchains, where transactions are organized into blocks and processed sequentially, the Tangle allows multiple transactions to be confirmed simultaneously, enhancing scalability and enabling IOTA to handle numerous transactions even in environments with millions of interconnected IoT devices.

One of IOTA's key advantages is its feeless nature. Users validate two previous transactions before submitting their own, ensuring network security and integrity while eliminating transaction fees. This feeless structure makes IOTA particularly suitable for microtransactions and M2M communications prevalent in IoT ecosystems.

IOTA also facilitates real-time data sharing among devices and stakeholders within IoT systems. Leveraging IOTA's DLT, sensors, energy meters, and energy management systems can seamlessly exchange information, optimizing demand-response management, load balancing, and overall grid stability. This real-time data sharing capability is crucial for optimizing energy consumption, improving operational efficiency, and enabling intelligent decision-making in IoT environments.

Security is paramount in IoT systems, where data integrity and confidentiality are critical. IOTA's DLT ensures data integrity and security through cryptographic techniques and the Tangle's inherent structure. Each transaction is cryptographically linked to two previous transactions, making tampering computationally infeasible, thus providing robust security against malicious actors and ensuring system reliability.

Moreover, IOTA serves as a standardized platform for data exchange and communication among diverse stakeholders in IoT systems. It promotes interoperability between utilities, energy suppliers, and consumers, facilitating efficient coordination, energy trading, and seamless integration of different energy sources. This standardized approach fosters collaboration and innovation within the IoT ecosystem, leading to enhanced efficiency, reduced costs, and improved sustainability.

3 Architecture and system overview

In this section, we explore the architectural framework of our proposal, providing a detailed overview of the various actors and layers involved.

3.1 Basic definitions and concepts

In this section, we clarify key concepts and principles of our proposal. The accurate and consistent definition of these concepts is critical for understanding the architecture and functioning of the proposed energy trading system. We provide formal definitions and establish a standardized nomenclature for clarity, ensuring that the terms used throughout the paper have precise and unambiguous meanings.

Definition 1 (Electricity Transaction) *An electricity transaction $Tx(U_k, d, t)$ is defined as:*

$$Tx(U_k, d, t) = \langle U_k, \mathit{Date}, t, E_{d,t}^{U_k} \rangle$$

where U_k is a user identifier, Date is the transaction date, t is the time slot on that date, and $E_{d,t}^{U_k}$ is the quantity of electricity consumed or generated by user U_k.

In the context of this system, each electricity transaction represents a discrete interaction between a user and the grid, encapsulating the flow of energy either to or from the user's premises. These transactions form the fundamental units of data that are recorded on the blockchain, ensuring transparency and immutability. A day is divided into T time slots. The energy value $E_{d,t}^{U_k}$ is positive if the user is consuming electricity and negative if producing electricity. This distinction allows for the clear differentiation between consumption and production activities within the grid, which is crucial for accurate billing and energy balancing.

Definition 2 (Electricity Pattern) *A user's electricity pattern E^{U_k} is the daily average electricity consumption/production for each time slot t:*

$$E_t^{U_k} = \frac{1}{|D|} \sum_{d \in D} E_{d,t}^{U_k}$$

$$E^{U_k} = \{E_0^{U_k}, E_1^{U_k}, \ldots, E_T^{U_k}\}$$

where D is the number of days.

Furthermore, these electricity patterns allow users to be compared based on the similarity of their consumption profiles. Furthermore, these electricity patterns allow users to be compared based on the similarity of their consumption profiles. This comparison begins by defining a measure of similarity between two users.

Definition 3 (Distance) *The distance $D(a, b)$ between users U_a and U_b is defined as:*

$$D(U_a, U_b) = \sum_{t \in T} |E_t^{U_a} - E_t^{U_b}|$$

This metric quantifies their dissimilarity across all time slots.

The concept of distance is fundamental to the clustering of users into groups based on their electricity patterns. By quantifying the difference between two users' patterns, the system can group users with similar behaviors together.

Analyzing the consumption patterns of a large user group reveals distinct patterns based on user habits, preferences, and behaviors. This allows users to be grouped into categories called electricity profiles.

Definition 4 (Electricity Profile) *An electricity profile is defined as a set of users with similar consumption patterns. Let U be a set of users, each with an electricity pattern E^n:*

$$U = \{E^0, E^1, \ldots, E^{|U|}\}$$

The set U can be partitioned into $|\Gamma|$ profiles, where $|\Gamma| \leq |U|$.

Each profile γ is the mean electricity consumption/production μ_t^γ for each time slot among users in the group:

$$\mu_t^\gamma = \frac{1}{|\gamma|} \sum_{u \in \gamma} E_t^u$$

$$\gamma = \{\mu_0^\gamma, \mu_1^\gamma, \ldots, \mu_{|T|}^\gamma\}$$

Definition 5 (Objective Function) *To create $|\Gamma|$ profiles, we minimize the variance of electricity consumption within each time slot for all users in each profile:*

$$\Gamma(U) = \arg\min_{\Gamma} \sum_{\gamma \in \Gamma} \sum_{u \in \gamma} D(u, \gamma)^2$$

where Γ is the set of profiles, γ represents each profile, and u is a user in profile γ.

By optimizing this objective function, we assume that users with similar electricity consumption patterns can be grouped into coherent profiles. The minimization of variance within each profile ensures that the users in a given profile have highly similar electricity usage behaviors.

Definition 6 (Classification Function) *New or unknown users can be classified into existing profiles with the function:*

$$\gamma(U_k) = \arg\min_{\gamma_j} D(U_k, \gamma_j)$$

3.2 Privacy considerations

The electricity usage profile of a user contains valuable information that attackers can exploit to infer personal routines, occupancy patterns, and lifestyle characteristics. Analyzing consumption patterns can reveal, for example, when a user is typically home or away, potentially exposing vulnerabilities. These patterns, when aggregated over time, can also provide insights into more granular details, such as the types of appliances being used, the presence of electric vehicles, or even the likelihood of specific events occurring at a residence, such as vacations or prolonged absences. Variations in usage levels and patterns can indicate household occupancy, and unique consumption patterns can even identify specific types of premises, such as residential versus commercial properties. The risks associated with these inferences are not just hypothetical; they have real-world implications for the security and privacy of individuals.

Attackers can use this information for malicious purposes, compromising privacy and security. For instance, by knowing when a household is typically unoccupied, a potential burglar could plan a break-in. Similarly, businesses could exploit this data for targeted advertising, or worse, insurance companies might use it to adjust premiums based on inferred lifestyle risks. Therefore, it is crucial to ensure that such sensitive information is adequately protected in any system that records and processes electricity usage data. The challenge lies in balancing the need for detailed data to optimize grid performance and user experience with the imperative to protect individual privacy.

Given that information of Tx is public and it can lead to profile classifications, it is necessary to protect Tx information in order to prevent an attacker from correctly classifying a user into their electricity profile. The public nature of blockchain transactions, while providing transparency and security, also presents a significant privacy challenge. Since all transactions are visible on the blockchain, it becomes possible for an observer to analyze these transactions and potentially link them to specific users based on their unique electricity usage patterns. This possibility makes it imperative to develop a method that obfuscates the actual data being transmitted while maintaining the integrity and functionality of the transaction system.

Therefore, there's a need for a function F that protects Tx, giving as a result Tx'. This function F must be designed to ensure that even if an attacker gains access to the transaction data, they cannot accurately deduce the underlying electricity patterns of any individual user. The transformation applied by F should make it computationally infeasible to reverse-engineer the actual usage data or link it back to the original user with a high degree of certainty.

There are two primary categories of attackers seeking to extract this information:

- **External Attacker**: This attacker remains an external observer without participating in the system. Such an attacker could be a third party with access to the blockchain data, who tries to infer information about users based solely on publicly available transaction records. The external attacker does not have insider knowledge or special access, making their attacks reliant on statistical analysis and pattern recognition across the data set.
- **Internal Attacker**: This attacker actively engages with the system as another user. An internal attacker could be another prosumer within the smart grid who participates in energy trading. This attacker has more information than an external attacker, such as their own electricity usage data and possibly information about the group dynamics within a specific smart contract. This additional information allows the internal attacker to make more informed inferences about other users, potentially compromising their privacy.

In the context of privacy considerations, it is essential to account for both types of attackers. The system must be robust enough to prevent an external attacker from making accurate inferences based on publicly available data while also mitigating the risks posed by internal attackers who might exploit their position within the network.

3.2.1 Adversary Model

Let F be a function that takes an input Tx and outputs Tx', hiding the real electricity pattern of U_a. This function F serves as a privacy-preserving transformation that modifies the transaction data in a way that obscures the true electricity usage patterns from any observer, whether external or internal. The goal of F is to introduce uncertainty into the data, making it difficult for an adversary to accurately classify users or infer sensitive information based on the transformed data.

The adversary model can be described through the following steps:

1. The adversary A has oracle access to $\gamma(U_k)$. This means that the adversary can query a function γ that returns the profile classification for any user U_k based on their electricity usage pattern. This access allows the adversary to attempt to match transformed data Tx' with existing profiles.

2. The challenger has Tx and E^{U_a}, and computes $Tx' \leftarrow F(Tx)$. Tx' is given to A. The challenger represents the system, which applies the transformation F to the original transaction data Tx, resulting in Tx'. The adversary receives Tx' and attempts to analyze it.

3. Adversary A gets $E^{U'_a}$ from Tx'_i.

4. Adversary A continues to have oracle access to $\gamma(U_k)$ and outputs a profile $\gamma(E^{U'_a}) = \gamma'$. After obtaining $E^{U'_a}$, the adversary uses their access to the profile classification function γ to assign $E^{U'_a}$ to a specific profile γ'. This step represents the adversary's attempt to classify the user based on the reconstructed pattern.

5. The output of the experiment is defined to be 1 if $\gamma' = \gamma(E^{U_a})$. The experiment is considered successful (output is 1) if the adversary correctly matches the transformed data to the original profile of the user. The success rate of this experiment determines the effectiveness of the privacy-preserving function F.

For the system to be considered privacy-preserving in terms of the electricity usage of users, the following must hold:

$$Pr[\gamma(E^{U_a}) = \gamma' \mid E = E^{U'_a}] = Pr[\gamma]$$

This condition implies that the probability of the adversary correctly classifying the transformed electricity usage pattern $E^{U'_a}$ into its original profile $\gamma(E^{U_a})$ should be no better than random chance. In other words, the transformation F must be effective enough that even with access to the transformed data and classification function, the adversary cannot reliably infer the true profile of the user. If this condition holds, the system can be deemed to provide strong privacy guarantees, ensuring that users' electricity usage patterns remain protected against potential attackers.

3.3 Requirements

We now define a series of requirements that our system should meet. These requirements are divided into functional and security categories, each addressing critical aspects necessary for the effective and secure operation of the energy trading system.

3.3.1 Functional Requirements

The functional requirements ensure that the system operates as intended, facilitating efficient energy transactions while maintaining the necessary interactions between users and the grid. Each functional requirement is designed to address specific aspects of the energy trading process.

F1. Electricity Transaction Recording: The system must accurately record electricity transactions, including consumption and production data, within designated time slots. This requirement is fundamental to the operation of the system, as accurate transaction recording ensures that all energy exchanges are properly documented. This accuracy is crucial for the calculation of debts, credits, and overall energy balances within the system. Any discrepancy in recording could lead to disputes among users or between users and the grid, undermining trust in the system. The recorded transactions should be immutable and verifiable, ensuring transparency and accountability in the energy trading process.

F2. Smart Grid Coordination: The system should coordinate energy transactions among prosumers within localized groups. By facilitating direct energy exchanges between geographically proximate users, the system can reduce reliance on the main grid, lower transmission costs, and enhance overall grid efficiency. This coordination should be automated. The system should also handle scenarios where localized energy demand exceeds supply or vice versa, ensuring that energy flows are balanced.

F3. Main Grid Integration: It must integrate seamlessly with the main grid to facilitate energy exchange between localized prosumer groups and the bulk electricity distribution network. Integration with the main grid is crucial for ensuring the stability and reliability of the energy supply, especially during periods of high demand or low local production. The system should allow for bidirectional energy flows between prosumer groups and the main grid, enabling users to either draw energy from the grid when needed or contribute surplus energy back to the grid.

3.3.2 Security and Privacy Requirements

The security requirements are designed to protect the system from various threats, ensuring that user data and system operations remain secure. These requirements are essential for maintaining the integrity, confidentiality, and availability of the system, which are critical for user trust and regulatory compliance.

S1. Data Integrity: The system must ensure the integrity of electricity transaction records, preventing unauthorized modifications or tampering. Data integrity is paramount in a system where financial and operational decisions are based on recorded transactions. This immutability protects against fraud, errors, and external attacks that could compromise the accuracy of transaction records.

S2. User Authentication: It should authenticate users and devices participating in electricity transactions to prevent unauthorized access and ensure accountability. Strong user authentication mechanisms are necessary to ensure that only authorized individuals and devices can participate in the system. This prevents unauthorized access to sensitive data and functionalities, which could lead to data breaches, financial losses, or operational disruptions. The authentication process should be secure.

S3. Access Control: The system must enforce access control mechanisms to restrict access to sensitive data and functionalities based on user roles and permissions. Effective access control ensures that users can only access the information and perform the actions that are necessary for their role within the system. This minimizes the risk of accidental or intentional misuse of the system's capabilities. Access control policies should be clearly defined, regularly reviewed, and dynamically enforced to adapt to changing circumstances and emerging threats.

S4. Anonymity Preservation: It should preserve user anonymity within the system. Anonymity is a critical aspect of user privacy in the energy trading system. By ensuring that users' identities are not directly linked to their electricity transactions, the system can protect individuals from profiling, targeted attacks, and other privacy violations. This is particularly important in a decentralized system where transaction data is publicly accessible on the blockchain.

S6. Privacy Compliance: It should adhere to privacy regulations and standards, preventing adversaries from deducing individual consumption patterns from transaction records as in the adversary model. The system must implement measures that prevent unauthorized parties from inferring sensitive information about users, such as their daily routines or energy consumption patterns. This includes employing advanced cryptographic techniques, secure data storage, and rigorous access controls to ensure that user privacy is not compromised.

S7. Auditability: The system should support auditability features, allowing for the tracing and verification of transaction histories for accountability and regulatory compliance purposes. Auditability is crucial for maintaining transparency and trust in the system. It allows for the tracking of all transactions and system activities, providing a clear and verifiable record that can be reviewed in the event of a dispute or investigation. Audit trails should be secure, immutable, and accessible only to authorized personnel to prevent tampering or unauthorized access.

S8. Availability: Maintain high availability and ensure uninterrupted system function. The energy trading system must be reliable and always available to meet the needs of its users. High availability is critical for ensuring that users can access the system when they need to, especially during peak usage times or in the event of an emergency. The system should be designed to withstand failures, attacks, and other disruptions.

3.4 Architecture and Proposal Overview

In this section we introduce the basic architecture of our proposal as well as an introduction to the main steps of the system. The structure of this architecture is based on previous work done published [10].

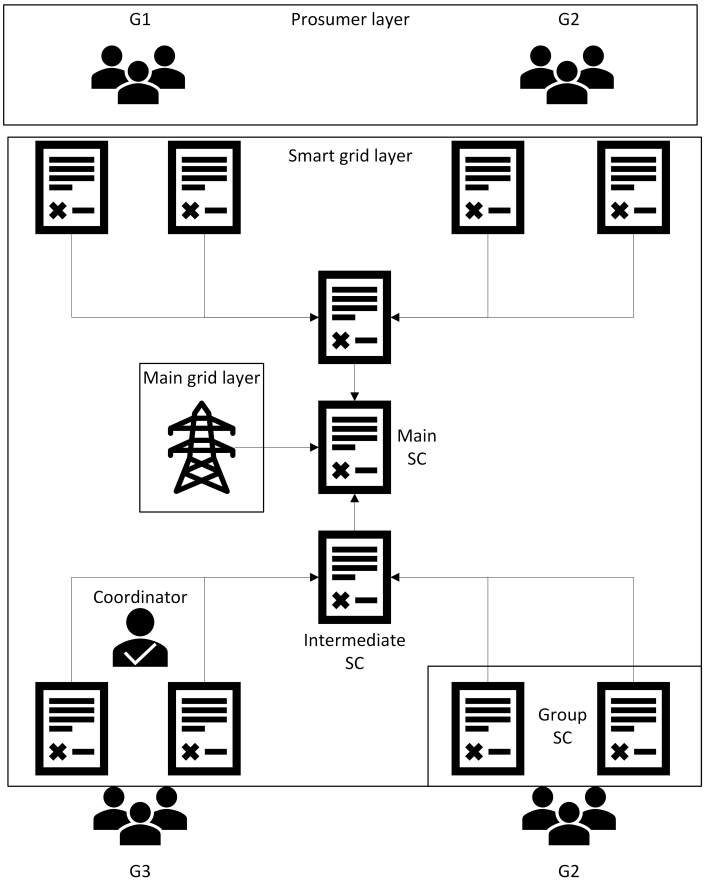

Figure 1: Architecture scheme

3.4.1 Architecture

The proposed architecture is depicted in Fig. 1. It consists of three main layers: *prosumers layer*, *smart grid layer*, and *main grid layer*. The following actors/components are involved:

- Smart meters (SM_i): devices who keep track of a prosumer's usage of energy (production and consumption).
- Prosumers: each system user that at a certain time can both produce and consume energy, depending on the balance between the production and consumption that prosumer can be: a *Consumer* (C_i) if she consumes more energy than produced; or a *Producer* (P_i) if she produces more energy than consumed. Each prosumer owns a smart meter and can have consumption, production facilities or both in their household.
- Main grid (MG): connects the main energy sources to the smart grid and is in charge of electricity distribution. Is the one in charge of energy consumption prediction and providing the extra electricity needed.
- Smart grid: keeps transactions between prosumers and between prosumers and the main grid. Also coordinates all payments.

- Coordinators (Co_i): responsible for verifying user's transactions, they own the smart meters used by the prosumers and, hence, they can verify that the prosumers are not lying about their consumption/production amounts. Also, if a prosumer fails to pay, the coordinator may take the necessary measures, although those are outside the scope of this proposal.
- Operators (O_i): are tasked with registering prosumers on the smart grid layer, connecting them, and giving them the authorization to send messages with the relevant trading information to their respective groups. They connect the prosumers to the smart grid physically (wiring) and digitally (giving them the necessary permissions).

The *prosumer layer* is composed of several groups of prosumers who are geographically close to each other. In these groups, they trade electricity among themselves to increase efficiency.

The *smart grid layer* is located between the prosumers layer and the main grid layer. This layer also has a coordinator for each group of prosumers, each coordinator verifies their group members transactions. This layer has the information of the energy usage of each group, which means that, in case a prosumer group needs more electricity or has a surplus, the smart grid layer communicates with the main grid level to buy or sell that electricity. In order to obtain that information, it records all transactions between prosumers in each group with the energy usage information.

Finally, the *main grid layer* is the system as we know of it today. It is composed of the main electrical companies with their bulk electricity generators and also control the distribution of energy.

3.4.2 Steps

The four main steps of the envisaged system are: *Initialization*, *Price calculation*, *Energy trading*, and *Debt liquidation*. Each of these steps plays a critical role in ensuring the smooth and efficient operation of the decentralized energy trading system. Together, they form a comprehensive process that manages the entire life-cycle of energy transactions, from initial user registration to the final settlement of debts.

In the *Initialization* step, prosumers (i.e., consumers and producers) should register with the operators and must be authorized to interact with the smart grid infrastructure. This step is foundational, as it establishes the identity and credentials of each participant in the system. During this phase, prosumers are required to provide necessary information, such as their identity, location, and energy production capabilities, which are verified by the system operators. These operators act as gatekeepers, ensuring that only legitimate users gain access to the smart grid. Once registered, each prosumer is provided with a smart meter that is integrated into the smart grid network. The successful completion of the initialization step ensures that the system has a clear and accurate understanding of all participants, laying the groundwork for subsequent steps.

The *Price calculation* step begins the day before energy trading, with a prediction of energy usage, a process already in use in most energy markets and whose development is beyond the scope of this work. During this step, the main grid performs a crucial function by estimating the overall energy demand for the upcoming trading day. This prediction is based on historical data, weather forecasts, and other relevant factors that might influence energy consumption. The grid then communicates with the main electricity producers to assess their production capacities and the prices at which they are willing to sell energy. Producers are ranked based on their pricing, with the highest price setting the ceiling for the trading price in each time slot. This method of price determination is similar to practices observed in various energy markets, such as those in the European Union [37]. The calculated prices are then distributed across different time slots throughout the day, reflecting the expected fluctuations in demand and supply. This step is critical for setting a transparent and fair market environment, where all participants can anticipate the cost of energy and make informed decisions about their trading activities.

The *Energy trading* step involves the direct interaction between prosumers and the smart grid layer, facilitated by the smart meters installed in each prosumer's home. This step is where the core functionality of the system comes into play. The smart meter continuously monitors the prosumer's electricity usage and production, recording this data in real time. These transactions are then securely transmitted to the smart grid layer via blockchain technology, ensuring that all data is tamper-proof and verifiable. The blockchain serves as a decentralized ledger that records every transaction, providing a transparent and immutable history of energy exchanges. The smart grid infrastructure uses this data to monitor the energy needs of each prosumer group. For example, if a group of prosumers generates more energy than they consume, the surplus can be sold back to the main grid or to other groups in need. Conversely, if a group requires more energy than it produces, it can purchase the necessary amount from the main grid. In this context, the main grid acts as an additional prosumer, participating in the energy trading process just like any other user. The coordinators, who own the smart meters, play a crucial role in this step. They act as trusted third parties (TTPs), verifying each transaction to ensure that the data being recorded is accurate and that no fraudulent activity occurs. This verification process is essential for maintaining the integrity and trustworthiness of the system.

Finally, in the *Debt liquidation* step, the smart grid aggregates all transactions made by each prosumer throughout the trading day and calculates the resulting debts or credits. For prosumers who consumed more energy than they produced, a debt is assigned, representing the amount of energy they need to pay for. Conversely, prosumers who generated excess energy and sold it to others will have a credit, representing the payment they are owed. The system then communicates these debts and credits to the respective prosumers. Once the debts are settled, the system automatically distributes the collected payments to the producers who provided the energy. This automated process ensures that there is no need for direct interaction between individual consumers and producers, reducing the complexity and potential for disputes. The final settlement of debts not only concludes the trading cycle but also reinforces the efficiency and fairness of the system, ensuring that all participants are compensated appropriately for their contributions.

This structured approach to managing energy transactions ensures that the system is both scalable and resilient, capable of handling the complexities of a decentralized energy market while maintaining high levels of security, transparency, and user satisfaction.

4 The proposed system in detail

This section first explains the blockchain-based smart contracts which are the core components of the proposed solution. After that, it details the four main steps: *Initialization*, *Price calculation*, *Energy trading*, and *Debt liquidation*.

4.1 Main smart contract

The main smart contract (SC_{MG}) is the first component to be deployed. This contract acts as an intermediary between each prosumer group (G) and the main grid.

Its initial task is to obtain and store the necessary information about each main grid producer (MG_P) so they can communicate their energy production capacity and price of production. To achieve this, each MG_P creates a blockchain account and registers it with the main SC. The SC then verifies the provided information and stores the address of each MG_P. Using the information provided by each MG_P and the knowledge possessed by the main grid (MG), the SC_{MG} can calculate the price of electricity at any given time of the day. In addition to that, the SC_{MG} needs the addresses of all the other SC so that they can communicate any missing or surplus energy in each group.

The MG_P keeps track of all transactions between each intermediate smart contract SC_I and the MG. The SC_I registers the willing to buy or the surplus of electricity to the SC_{MG}. This procedure is similar to the one followed between prosumers in each SC_{G_i} and between the connections between SC_I.

Once a SC_I needs energy from the MG or is willing to sell a surplus, the SC_{MG} registers the transaction with the quantity of energy and the price.

The SC_{MG} consists of a class which holds:

- *Date*: Date of the trading day of the SM_{MG}
- *Producer identities*: Public keys of MG_P which can interact with the SC_{MG} and sell electricity to the MG.
- *Coordinator identities*: Public keys of Co which can register SC into the SC_{MG}

- *Main Grid Authority*: Public key of the owner of the SC_{MG}.
- *Expected electricity consumption*: The predicted electricity consumption by the MG for the following day.
- *Production capacity*: List of the production capacity of the different MG_P and its price, divided by time slots.
- *Prices*: List of the price of electricity in each time slot.
- *Intermediate smart contracts*: Addresses of all the SC_I that can interact with the SC_{MG} and sell or buy energy from the MG.
- *Transactions*: List of all transactions made between the MG and the SC_I. Each transaction contains the amount sold/bought and its price.
- *Debts*: List of all debts from each SC_I and the MG.
- *Time slot*: Number of the current time slot. -1 before the trading and 24 when it's over.
- *Trading flag*: Boolean which is "True" when the trading in the MG_{SC} is available.
- *Producers flag*: Boolean which is "True" when MG_P are able to be registered into the MG_{SC} and make an offer.
- *Smart Contracts flag*: Boolean which is "True" when the SC are able to be registered into the MG_{SC}.

When a new SC_{MG} is deployed, the MG must provide the following arguments: i) *Date of the trading day*; ii) *Expected energy consumption*. By default, *Trading flag* is set to "False" and *Producers flag* is set to "True", *Time slot* is set to -1.

Once the SC_{MG} is deployed, the different actors can interact with it with the following methods:

- *newProducer*() method can only be called by the MG authority when the *Producers flag* is set to "True". It introduces a new MG_P which will be able to sell electricity in the trading day. The producer is added to *identities*.
- *newSC*() method can only be called by the MG authority when *Smart Contracts Flag* is set to "True". It registers all SC_I that will be able to communicate the need or surplus of electricity of each group. The new SC_I is added to *Group Smart Contracts*.
- *addOffer*() method which can be called by MG_P when *Producers flag* is set to "True". It communicates the available production and its price at a certain time slot. The offer is added to *Production capacity*.

- *beginAuction*() method called by the MG authority which, once all MG_P have registered and added their offers or the time limit has come, it determines the electricity price for each time slot. It sets *Producers Flag* to "False".
- *beginTrading*() method called by the MG when the trading day starts. It sets *Smart Contracts Flag* to "False" and *Trading Flag* to "True". It also communicates all the SC_I that the trading begins.
- *changeTime*() method called by MG, it increases *Time slot* by 1.
- *buyElectricity*() and *sellElectricity*() methods are called by the SC_I during the trading day, when *Trading flag* is set to "True". SC_I communicate the needed electricity or the surplus that has to be bought/sold to the MG. Both the amount and the price is stored in *Transactions*.
- *calculateDebts*() method called by the MG once the trading has ended. From all the registered transactions, it calculates the debt each SC_I and the MG has. Every debt is stored in *Debts*. It sets *Trading Flag* to "False". It also calls the method *calculateDebts*() from each SC.
- *payElectricity*() method called by each SC_I which has a debt with the MG. The SC_{MG} distributes the funds to the corresponding sellers (MG_P or other SC_I).
- *getDebt*() method is called by any actor and, if it exists, it returns the debt that she has to pay.

4.2 Intermediate smart contracts

After deploying the SC_{MG}, the next step is to deploy the SC_Is.

To ensure that every SC_I and SC can successfully deliver Tx to their respective SC_I, they must complete a registration process with the SC_I. The SC_I maintains a repository with all the requisite data for U and SC_I authentication. Additionally, the top SC_I must register with the SC_{MG} to acquire details regarding the current electricity pricing for each time slot. It is essential to note that the authorization records within each SC_I must be regularly updated. For instance, in cases where a new SC or SC_I joins the collective or departs, the record must undergo immediate revisions. This procedure guarantees that only sanctioned Tx are documented, thereby ensuring the precision of energy consumption and billing.

The SC_{MG} consists of a class which holds:

- *Date*: Date of the trading day of the SM_{MG}
- *Main Grid Authority*: Public key of the authority who can connect SC_I.
- *Prices*: List of the price of electricity in each time slot.
- *Parent SC*: parent node.

- *Children* SC: Addresses of all the SC_I or SC_G that can interact with the SC_I and sell or buy energy.
- *Transactions*: List of all transactions made between the children. Each transaction contains the amount sold/bought and its price.
- *Debts*: List of all debts from each child and the parent.
- *Time slot*: Number of the current time slot. -1 before the trading and 96 when it's over.
- *Trading flag*: Boolean which is "True" when the trading is available.
- *Smart Contracts flag*: Boolean which is "True" when the SC are able to be registered into the SC_I.

When a new SC_I is deployed, the MG must provide the following arguments: i) *Date of the trading day*; ii) *Parent* SC *address*. By default, *Trading flag* is set to "False" and *Time slot* is set to -1.

Once the SC_I is deployed, the different actors can interact with it with the following methods:

- *newSC*() method can only be called by the MG authority when *Smart Contracts Flag* is set to "True". It registers all SC_I or SC_G that will be able to communicate the need or surplus of electricity of each group. The new SC_I is added to *Children* SC.
- *beginTrading*() method called by the parent SC when the trading day starts. It sets *Smart Contracts Flag* to "False" and *Trading Flag* to "True". It also communicates all the children SC that the trading begins.
- *changeTime*() method called by parent SC, it increases *Time slot* by 1. It communicates the change to all the children SC. It also calculates the surplus energy or the needed energy that the children SC has had in the previous time slot and communicates it to the parent SC.
- *buyElectricity*() and *sellElectricity*() methods are called by the children SC during the trading day, when *Trading flag* is set to "True". Children SC communicate the needed electricity or the surplus that has to be bought/sold to the other SC. Both the amount and the price is stored in *Transactions*.
- *calculateDebts*() method called by the parent SC once the trading has ended. From all the registered transactions, it calculates the debt each child SC and the parent SC has. Every debt is stored in *Debts*. It sets *Trading Flag* to "False". It also calls the method *calculateDebts*() from each child SC.
- *payElectricity*() method called by each child SC which has a debt with the SC_I. The SC_I distributes the funds to the children and parent SC.
- *getDebt*() method is called by any child SC and, if it exists, it returns the debt that it has to pay.

4.3 Group smart contracts

After deploying the SC_I, the next step is to deploy the SC_Gs. Each prosumer group (G) has several SC_Gs that are responsible for tracking the energy usage and price for each user (U) within the G. Each group has its own SC_G in order to incentive first local electricity trading. Furthermore, each group has several SC_G in order to increase the privacy of the users by dividing their real electricity usage into different values.

To ensure that each U can send Tx to their respective SC_G, they must be registered in the SC_G. It possesses a table which contains the necessary information to authorize each user. Additionally, the SC_G must register with the SC_I in order to receive information on the price at which electricity is sold during each time slot. Moreover, each SC_G must possess the identity of the coordinator (Co) of the group. The coordinator is the only authority that can verify the Tx from each U.

Once the SC_Gs have been deployed and authorized, they can start collecting the Tx made by each U within the group. This data is important for ensuring fair and accurate billing according to the energy usage within the group.

SC_Gs can communicate with the SC_I. For example, if one group has a surplus of energy, they can communicate with the SC_I which can communicate with another group that needs more energy and sell their surplus.

Each SC_G consists of a class which holds:

- *Parent smart contract*: Address of the SC_I in which the SC_G is registered.
- *Date*: Date of the trading day of the SC_G.
- *Coordinator*: Public key of the SC_G's Co.
- *User Identities*: Public keys of all U of SC_G's G. Those users are able to sell/buy electricity.
- *Operator Identities*: Public key of operators who can register new U into the SC_G.
- *Prices*: List of the price of electricity in each time slot.
- *Transactions*: List of all transactions made between U. Each transaction contains the amount sold/bought and its price.
- *Debts*: List of all debts from each U.
- *Time slot*: Number of the current time slot. -1 before the trading and 96 when it's over.
- *Trading flag*: Boolean which is "True" when the trading in the SC_G is available.

When the SC_G is deployed, the Co must provide the following arguments: i) *Date of the trading day*; ii) *Parent smart contract address*. By default, *Trading flag* is set to "False", *Time slot* is set to -1.

Once the SC_G is deployed, the different actors can interact with it with the following methods:

- *newOperator*() method called by the Co which adds a new O public key which will be able to register U.
- *newUser*() method called by the O which adds a new U which will be able to sell/buy electricity.
- *beginTrading*() method called by the SC_I which sets the *Trading flag* to "True" and also communicates all prices.
- *buyElectricity*() and *sellElectricity*() methods are called by U during the trading day once every time slot, only when *Trading flag* is set to "True". U communicates the consumed or produced electricity to be bought/sold to others U. Both the amount and the price of the time slot is stored in *Transactions*.
- *verifyTransaction*() method is called by Co when, at the end of each time slot, the Co verifies each U transaction.
- *calculateDebts*() method is called by the SC_I once the trading day has ended. From all the registered transactions, it calculates the debt each U has.
- *payElectricity*() method called by each U which has a debt with the G. The SC distributes the funds to the corresponding sellers (SC_I or other U).
- *changeTime*() method called by Co, it increases *Time slot* by 1. It also calculates the surplus energy or the needed energy that the G has had in the previous time slot and communicates it to the SC_I.
- *getDebt*() method is called by any actor and, if it exists, it returns the debt that she has to pay.

4.4 Initialization

During the initialization stage some tasks must be performed by the Main Grid (MG), Operators (O), Coordinators (Co), and Users (U). These tasks for each entity are next detailed in Algorithm 1.

The user registration process is a crucial aspect of any smart grid system, as it ensures that only authorized individuals are able to access and perform transactions within the network. During the registration process, users are required to provide certain information such as their name, contact information, and other relevant details. This information is then stored by the coordinator, and it's linked to the user's blockchain address in a private storage.

Algorithm 1 Initialization Protocol

1: MG **Authority:**

1. Deploy SC_{MG} with corresponding arguments.
2. $SC_{MG}^{Address} \rightarrow Co, MG_P$.
3. Register MG_P to SC_{MG} using *newProducer()*.
4. Deploy each SC_I with corresponding arguments.
5. Communicate each child address to parent SC using *newSC*().

2: Co **for each group:**

1. Deploy SC_G with corresponding arguments.
2. Communicate SC_G address to SC_I using *newSC*().
3. Register O to SC_G using *newOperator()*.

3: O **for each user:**

1. Install sealed smart meter SM_k at U_k's domicile.
2. Connect SM_k to U_k generation and consumption's electricity line.
3. Connect SM_k to Co_k for validation.

4: U **for each user:**

1. U is given a blockchain account and private key.
2. Create n key pairs/addresses using Key Derivation Function.
3. Communicate public keys to O with identification.
4. O registers U into n of the SC_G using *newUser()* with group signature scheme.
5. O communicates U information to Co.
6. O sends U each SC address.

4.5 Price calculation

Algorithm 2 outlines the Price Calculation Protocol, a mechanism governing the dynamic energy marketplace. In this algorithm, energy MG_P interact with the central entity (SC_{MG}) through a bidding process. As the auction deadline approaches, SC_{MG} employs a systematic approach to select the most cost-effective energy producers (MG_P) for each time slot, ensuring a balance between expected consumption and production. This section provides a concise overview of the steps involved in determining optimal electricity prices within this competitive marketplace.

4.6 Energy trading

Presented as Algorithm 3 and Fig. 2, the Energy Trading Protocol governs the intricate dynamics of an energy trading system. Initiated by the MG authority, this protocol orchestrates the trading day by facilitating communication between various entities, including SC_{MG}, SC_I, SC_G, and U/SM. The

Algorithm 2 Price Calculation Protocol

1: **All available P:**
   1. Send offer to SC_{MG} using *addOffer*().
2: **When the deadline comes:**
   1. MG authority ends bidding time using *beginAuction*().
3: SC_{MG} **does the following:**
   1. **In each time slot:**
      (a) Sort offers from cheapest to most expensive.
   2. **Select the n cheapest MG_P in each time slot:**
      (a) Ensure expected electricity consumption meets production.
   3. **Determine the price in that time slot:**
      (a) Select the most expensive MG_P offer from the n selected.

protocol unfolds through steps that involve the initiation of trading, communication between U and SM, energy transaction handling, and verification processes.

Key aspects include the division of energy usage, the selection of SC_G entities, and transaction handling during each time slot. The verification process, led by Co, ensures the integrity of transactions, marking them as verified in SC_G. The protocol concludes with MG actions based on the difference between total electricity produced and consumed, potentially triggering energy transactions at the higher organizational level.

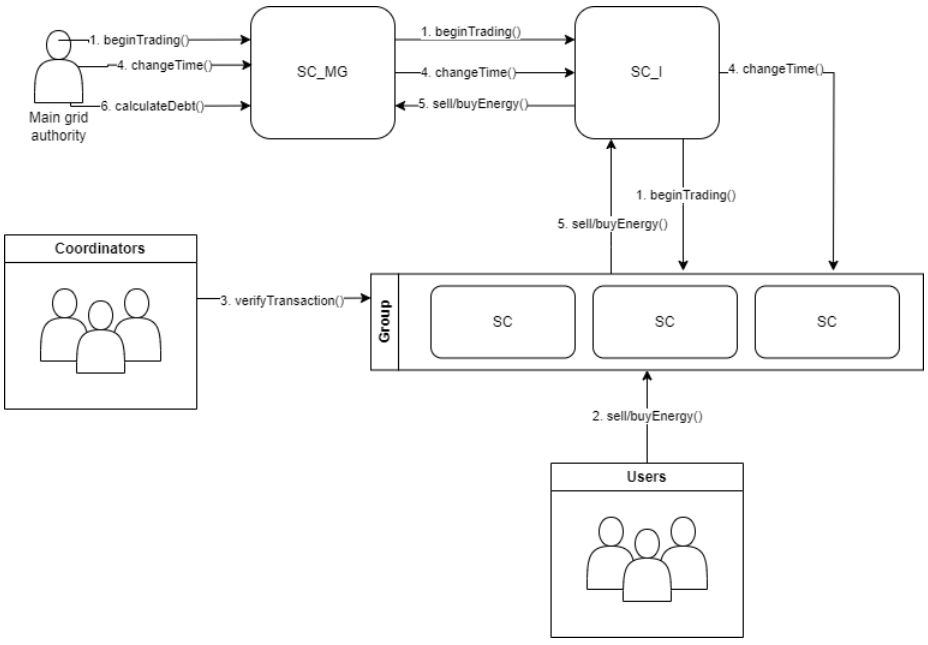

Figure 2: Energy trading

Algorithm 3 Energy Trading Protocol

1: MG **Authority:** Send *beginTrading*() to SC_{MG} when trading day starts
   1. SC_{MG}: Sends *beginTrading*() to all SC_I children.
   2. SC_I: Sends *beginTrading*() to the children recursively until SC_G.
2: U/SM **for each time slot:**
   1. Check electricity usage (positive/negative).
   2. Use *sellEnergy*() or *buyEnergy*() method.
   3. Randomly divide energy usage into n transactions.
   4. Send transactions to the n SC_G in which U is registered from k possessed by G.
   5. For each SC_G, add transaction to *Transactions* list with current price (unverified).
3: **Verification by** Co **each time slot:**
   1. Store energy usage information from each U of G.
   2. Send *verifyTransaction*() with transaction ID to SC_G.
   3. Mark transaction as verified in SC_G.
4: MG **Authority:** Send *changeTime*() to SC_{MG}
   1. SC_{MG}: Sends *changeTime*() to all SC_I children.
   2. SC_I: Sends *changeTime*() to the children recursively until SC_G.
5: SC **recursively sell/buy electricity:**
   1. SC_G calculates production/consumption difference and calls *sellEnergy*() or *buyEnergy*() in parent SC with total electricity produced or consumed.
   2. SC_I calculates production/consumption difference and calls *sellEnergy*() or *buyEnergy*() in parent SC with total electricity produced or consumed.
6: **End of Trading Day:** MG Authority calls *calculateDebt()* to end trading phase.

4.7 Debt liquidation

Algorithm 4 and Fig. 3 encapsulates the Debt Liquidation Protocol, a crucial mechanism within the energy trading ecosystem. The protocol commences with the MG Authority triggering the debt calculation process in SC_{MG}, initiating a series of actions across the hierarchical structure of the system.

The protocol unfolds with SC_{MG} orchestrating debt calculations for each child SC, followed by individual actions by each U entity. Debt payments and settlements are then executed in a cascading manner, involving interactions between U, SC_G, and SC_I. Notably, debt checks and subsequent payments ensure financial equilibrium within the system.

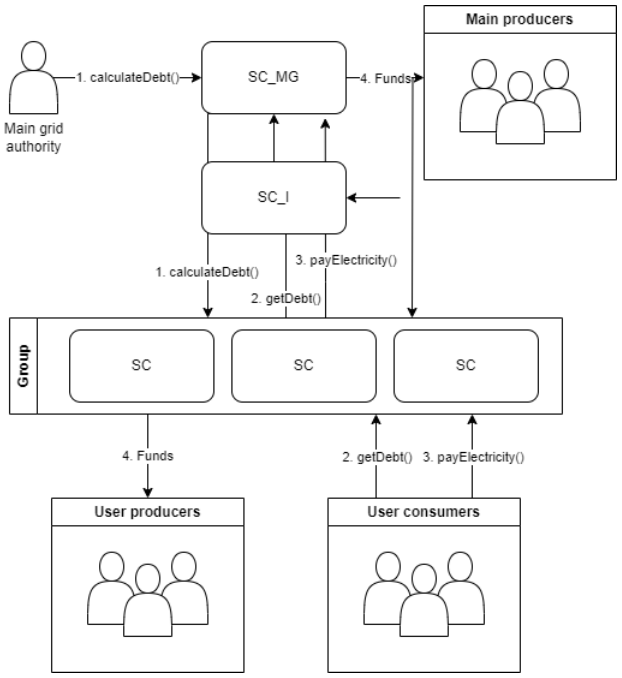

Figure 3: Debt liquidation

Algorithm 4 Debt Liquidation Protocol

1: MG **Authority:** Send *calculateDebt*() to SC_{MG} when trading day ends
   1. SC_{MG}: Sends *calculateDebt*() to all SC_I children.
   2. SC_I: Sends *calculateDebt*() to the children recursively until SC_G.
2: **For each** U**,** SC_G **and** SC_I**:**
   1. If consumed more than produced, call *getDebt()* from SC_G or parent SC.
3: **Each** U **and** SC **in debt**: call *payElectricity()* from SC_G or parent SC.
4: **Debt Payment:**
   1. **For each** SC **children and** U**:**
      (a) If they are producers, pay the corresponding funds.
   2. SC_{MG} pays every MG_P for energy produced, ending the third and final stage.

5 Security and Privacy study

In order to perform the privacy study of the proposed system, we first provide a comprehensive exploration of the concepts outlined in Section 3.1, fundamental definitions are established to create a precise nomenclature that underpins the architecture and functionality of our system. These definitions also detail the two types of attackers described in Section 3.2, which are critical to understanding the security and privacy measures implemented in our proposal. The clear articulation of these concepts is essential not only for the technical implementation but also for the formal analysis and evaluation of the system's performance in real-world scenarios.

We then evaluate the general requirements that ensure the overall privacy and security of the system. Following this, we conduct a detailed analysis of the two types of attackers that could potentially target the system, as described in Section 3.2. This comprehensive approach allows us to assess the robustness of the system's privacy-preserving features and identify any potential vulnerabilities.

5.1 Concepts based on our proposal

In order to provide a comprehensive exploration of the concepts outlined in Section 3.1, fundamental definitions are established to create a precise nomenclature that underpins the architecture and functionality of our system. These definitions also detail the two types of attackers described in Section 3.2, which are critical to understanding the security and privacy measures implemented in our proposal. The clear articulation of these concepts is essential not only for the technical implementation but also for the formal analysis and evaluation of the system's performance in real-world scenarios.

5.1.1 Definitions

As explained in Section 4, the system is divided into groups, which serve as the fundamental units of organization within the smart grid. Each group comprises multiple smart contracts that manage the interactions and transactions of the prosumers within that group.

Definition 7 (Group) *A group G_i is defined as the set of smart contracts σ_j^i forming that group:*

$$G_i = \{\sigma_0^i, \sigma_1^i, \ldots, \sigma_N^i\}$$

In our system, a group G_i represents a collection of smart contracts that govern the energy trading activities of a specific subset of prosumers. These groups are typically formed based on geographical proximity. By organizing prosumers into groups, the system can facilitate more efficient energy trading, reduce transmission losses, and enhance the overall resilience of the grid. The structure of each group is flexible, allowing for the dynamic addition or removal of smart contracts as prosumer participation changes over time.

Definition 8 (Smart Contract) *Each smart contract σ_s^i in G_i contains a subset of addresses $\{\alpha_{i,s}^{U_k}\}$, where U_k denotes the address owner, i is the smart contract group G_i, and s identifies the smart contract inside of the group G_i:*

$$\sigma_s^i = \{\alpha_{i,s}^{U_x}, \alpha_{i,s}^{U_y}, \ldots, \alpha_{i,s}^{U_z}\}$$

A smart contract σ_s^i is a self-executing digital contract that automates the negotiation, validation, and execution of energy transactions within a group G_i. Each smart contract contains a subset of user

addresses that represent the prosumers participating in that group. These addresses are pseudonyms that ensure the privacy of the users while enabling the contract to function correctly. The smart contract is responsible for managing the flow of energy between these addresses, recording transactions on the blockchain, and enforcing the terms of the agreement without the need for human intervention. This automation reduces the potential for errors, fraud, and disputes, and it ensures that all transactions are conducted fairly and transparently.

Definition 9 (User) *Each user U_k belongs to a certain group G_i and participates in a subset of the smart contracts of that group $\{\sigma_a^i, \sigma_b^i, \dots\}$. Therefore, users have a different address for each smart contract they participate in:*

$$A^{U_k} = \{\alpha_{i,a}^{U_k}, \alpha_{i,b}^{U_k}, \dots\}$$

In our system, each user U_k is a prosumer who both consumes and/or produces electricity. To maintain privacy and security, each user is assigned a unique address for every smart contract in which they participate. These addresses are pseudonymous, meaning that while they are linked to the same user, they do not reveal the user's actual identity. This structure allows users to participate in multiple smart contracts within a group without compromising their anonymity. The use of different addresses for each contract also helps to prevent profiling by making it more difficult for an attacker to link transactions across different contracts to the same user. This approach is a key component of our system's privacy-preserving design, ensuring that users can trade energy securely and confidently.

Definition 10 (Protected transaction) *The transaction Tx defined in Definition 1 in our proposal is divided into several protected transactions Tx', where instead of having U_k as the identifier of the user, the identifier is one of the addresses of the user $\alpha_{i,s}^{U_k} \in A^{U_k}$:*

$$Tx'(\alpha_{i,s}^{U_k}, d, t) = \langle\, \alpha_{i,s}^{U_k} \mid Date \mid t \mid E_{d,t}^{\alpha_{i,s}^{U_k}} \,\rangle$$

A protected transaction Tx' is a key mechanism in our system that enhances user privacy. Unlike a standard transaction, which might include identifiable information about the user, a protected transaction replaces the user identifier with one of the user's pseudonymous addresses. This ensures that the transaction is still valid and can be processed by the system, but without revealing the user's true identity. The protected transaction includes the time slot and date of the transaction, as well as part of the amount of energy consumed or produced. This design allows the system to maintain accurate records of energy transactions while preventing unauthorized parties from linking transactions to specific users. The use of protected transactions is crucial for safeguarding user privacy, particularly in a decentralized system where transaction data is stored on a public blockchain.

Definition 11 (Electricity Usage) *The electricity consumption $E_{d,t}^{U_k}$ for a user U_k during a given time slot t and date d is the sum of electricity consumption across all addresses $\alpha \in A^{U_k}$ held by the user*

U_k in her smart contracts:

$$E_{d,t}^{U_k} = \sum_{\alpha \in A^{U_k}} E_{d,t}^{\alpha}$$

Please, note that $E_{d,t}^{\alpha}$ is obtained from the set of protected transactions.

Electricity usage $E_{d,t}^{U_k}$ represents the total amount of electricity consumed or produced by a user U_k during a specific time slot and date. This value is calculated by summing the electricity usage recorded across all of the user's pseudonymous addresses involved in different smart contracts. By aggregating the energy data from multiple addresses, the system can accurately determine the user's overall energy consumption or production without compromising their privacy. This approach allows the system to perform necessary functions such as billing, energy balancing, and debt settlement while maintaining the anonymity of individual users. The ability to aggregate and protect electricity usage data is a cornerstone of our system's privacy-preserving capabilities, ensuring that users can engage in energy trading without fear of being profiled or tracked.

5.1.2 Adversary model

In our system, we recognize the importance of understanding and mitigating potential security threats posed by different types of attackers. Given the two types of attackers defined in Section 3.2, we consider those types in our system in the following manner. Each level of attack presents unique challenges and requires specific countermeasures to ensure the privacy and security of the users within the system.

- **External level**:

  At this level, the attacker only has access to the information stored in the blockchain. The blockchain consists of a series of smart contracts (SC), each containing multiple transactions from different users within each time slot. However, the attacker does not possess knowledge of the linkage between SC or the association between user addresses. The attacker's capabilities are limited to observing the public ledger, where they can analyze the frequency, timing, and value of transactions. Despite this access, the attacker's ability to extract meaningful information is constrained by the system's use of pseudonymous addresses and protected transactions. These privacy-preserving measures ensure that the attacker cannot easily correlate transactions to specific users or uncover their electricity usage patterns. Additionally, because the blockchain is decentralized and distributed, the external attacker lacks the computational power to break the cryptographic protections that secure the data, further limiting their potential impact.

  The primary risk at this level is that the attacker could attempt to perform statistical analysis or pattern recognition on the public transaction data to infer possible correlations between different addresses or to estimate group behaviors. For instance, by analyzing transaction volumes

or timing across different smart contracts, the attacker might try to identify which addresses are more active and hypothesize about the corresponding users' energy consumption habits. However, without direct knowledge of the users' identities or the structure of the smart contracts, these inferences are highly speculative and unlikely to yield accurate results.

- **Group level**:

  At the group level (internal), the attacker possesses the information available to the external attacker, as well as the ability to identify which smart contracts belong to the same group. Consequently, the attacker can determine that different addresses within these smart contracts correspond to the same group of users. This level of attack could occur if the attacker was a user within a specific group. By being a participant in the group, the attacker has enhanced access and insight into the group dynamics, which can be exploited to attempt to link transactions across different smart contracts more effectively.

  This internal position significantly increases the attacker's ability to perform linkage attacks, where they try to correlate pseudonymous addresses with specific users based on transaction history, behavior, or other metadata. The attacker could potentially exploit their insider knowledge, such as understanding the typical energy consumption profiles of group members or identifying the timing of specific transactions, to reduce the anonymity of the system.

  However, even at the group level, the system's design incorporates several mechanisms to mitigate these risks. For example, the use of multiple pseudonymous addresses for each user across different smart contracts makes it difficult for the attacker to definitively link all transactions to a single individual. Furthermore, the system's reliance on distributed consensus and cryptographic techniques ensures that even if the attacker gains knowledge of group-specific information, they cannot alter or manipulate the transaction records on the blockchain. This ensures the integrity of the data and prevents any unauthorized changes or tampering.

Overall, our system is designed to be resilient against both external and group-level attackers. By incorporating robust privacy-preserving measures and ensuring that critical information remains protected, we aim to maintain the security and anonymity of users, even in the face of sophisticated adversaries. The layered approach to security, with distinct defenses against different levels of attack, provides comprehensive protection and ensures that the system remains trustworthy and reliable.

5.2 Security requirements

The following claims outline the security requirements that our system meets. Each claim is supported by a proof that explains how the system fulfills the corresponding security objective.

Claim 1 *The system can detect unauthorized modifications or tampering of data. (S1)*

Proof 1 *The immutability of transactions within a blockchain ensures the integrity and authenticity of each transaction, eliminating the possibility of double-spending or unauthorized alterations. Once a transaction is recorded on the blockchain, it becomes a permanent part of the ledger, protected by cryptographic hashes that link each block to the one preceding it. This creates a chain of blocks where any attempt to modify a transaction would require altering every subsequent block, an infeasible task given the computational power required. Furthermore, the decentralized nature of the blockchain, where multiple nodes must reach consensus on the validity of each transaction, further enhances security by making it virtually impossible for a single entity to alter the ledger without detection. As a result, any unauthorized modifications or tampering with the data are effectively prevented, ensuring that all transactions remain accurate and trustworthy.*

Claim 2 *Modified messages from producers and the smart grid layer are prevented from participating in the system. (S1)*

Proof 2 *The system's security is underpinned by the use of blockchain, which ensures that only legitimate transactions are allowed to participate in the system. Each transaction is digitally signed by the sender using their private key, which provides a cryptographic proof of authenticity. If an unauthorized party attempts to modify a message sent by a producer or the smart grid layer, the digital signature would no longer match the modified content, rendering the signature invalid. The smart contracts automatically verify the validity of each signature before processing the transaction, ensuring that only genuine, unaltered messages are accepted. This mechanism not only prevents tampered messages from being processed but also protects the integrity of the entire system by ensuring that all actions are verifiable and traceable back to their origin.*

Claim 3 *The proposed system maintains high availability and is able to function without interruption. (S8)*

Proof 3 *The system achieves high availability by leveraging the decentralized nature of blockchain technology, which inherently supports redundancy and fault tolerance. In a decentralized blockchain, the ledger is replicated across multiple nodes, each maintaining a complete copy of all transactions. This ensures that even if some nodes fail or are taken offline, the system as a whole remains operational. Transactions can still be processed and verified by the remaining nodes, ensuring continuous access to the system's services. Additionally, the use of consensus algorithms such as Proof of Work (PoW) or Proof of Stake (PoS) ensures that the blockchain can continue to operate securely even in the face of partial failures or network partitions. By distributing the workload across a network of independent nodes, the system can provide reliable and uninterrupted service, making it resilient to both technical failures and malicious attacks.*

Claim 4 *The proposed system effectively restricts access to only authorized users, preventing unauthorized individuals from performing protected actions and accessing protected resources and functions. (S2-S3)*

Proof 4 *The system enforces strict access control mechanisms through the use of smart contracts that verify the identity of users participating in transactions. Each user must authenticate themselves by signing their transactions with their private key, which is unique to their digital identity within the system. The smart contract then verifies the signature before allowing the transaction to proceed, ensuring that only registered and authorized users can participate in energy trading. Additionally, the system uses role-based access control (RBAC) to limit access to sensitive data and functions based on the user's role within the network. For example, only coordinators may have the authority to audit transactions or identify misbehaving users, while regular prosumers are restricted to managing their own transactions. This layered approach to access control ensures that sensitive operations and data are protected from unauthorized access, thereby maintaining the security and integrity of the system.*

Claim 5 *Users cannot deny having performed a particular action or Tx. (S7)*

Proof 5 *The system provides non-repudiation by leveraging blockchain technology and smart contracts, which ensure that all transactions are traceable and tamper-proof. Every transaction is recorded on the blockchain with a unique identifier and timestamp, along with the digital signature of the user who initiated it. This immutable record serves as irrefutable proof that the transaction occurred, and that it was authorized by the user whose signature it bears. Because the blockchain is decentralized and transparent, any attempt to dispute or deny a transaction can be easily refuted by referencing the publicly accessible transaction history. Moreover, the use of cryptographic techniques ensures that the digital signatures are tied directly to the user's private key, which cannot be forged or repudiated. As a result, users are held accountable for all actions they perform within the system, providing a strong deterrent against fraudulent behavior.*

Claim 6 *The identity of a user and other personal information is not visible within the system. (S4)*

Proof 6 *The system is designed with privacy in mind, ensuring that users' identities and personal information are protected at all times. Each user in the system is represented by a pseudonymous address, which is used to interact with smart contracts and perform transactions. The link between a user's address and their true identity is known only by the trusted third-party coordinator (Co), who is responsible for user registration and identity verification. This separation ensures that even if an attacker gains access to the transaction data, they cannot easily determine the user's real identity. Furthermore, the system employs group signatures for operators, allowing them to sign transactions on behalf of the entire group without revealing the specific operator who performed the action. This adds*

an additional layer of anonymity, making it even more difficult for any external or internal attacker to trace transactions back to individual users. To further enhance privacy, users are encouraged to periodically update their addresses, thereby reducing the risk of long-term tracking or profiling by any potential adversary.

Claim 7 *The proposed system uses coordinators to verify the messages sent by prosumers and to identify misbehaving ones. (S7)*

Proof 7 *Coordinators play a crucial role in maintaining the integrity and security of the system. They are responsible for verifying each message sent by prosumers, ensuring that all transactions are legitimate and free from fraudulent activity, such as double spending or tampering. Coordinators have access to all transactions made by a particular user (U) and can analyze these transactions to detect any anomalies or suspicious behavior. If a coordinator identifies a pattern of fraudulent activity, they have the authority to trace the transactions back to the responsible user and take appropriate action, such as issuing penalties or excluding the user from the system. This monitoring capability is essential for maintaining trust within the network, as it ensures that all participants adhere to the rules and that any attempts to undermine the system are swiftly addressed. By providing a mechanism for the identification and punishment of misbehaving users, coordinators help to enforce the security and reliability of the entire energy trading ecosystem.*

5.3 Privacy Requirements

We now evaluate the privacy requirements, first analyzing the internal attacker followed by the external.

5.3.1 Internal attacker

An internal attacker is a user who is part of the system and has access to a subset of the information available within the smart contracts of their group. This attacker might attempt to link different elements within the system, such as smart contracts or user addresses, to de-anonymize users or gain unauthorized insights into their behavior. The following claims and proofs demonstrate how the system is designed to prevent such attacks.

Claim 8 *Smart contracts σ_s^i from the same group G_i cannot be linked together without the information of the Coordinators. (S6)*

Proof 8 *To understand why smart contracts within the same group cannot be linked by an internal attacker, let's first examine the information available within a single smart contract σ_s^i:*

- ***SC Address*** σ_s^i*: Each smart contract within the group has a unique address that is independent of any relationship with other smart contracts in the same group. This unique identifier does not reveal any direct linkage to other contracts.*
- ***Users' Addresses****: Each smart contract* σ_s^i *contains a set of user addresses* $\{\alpha_{i,s}^{U_k}\}$*, where each address is unique within that specific smart contract. These addresses do not have any inherent relationship or pattern that connects them to addresses in other smart contracts* σ_s^i *within the same group.*
- ***Users' Protected Transactions*** Tx'*: The electricity usage values* $E_{d,t}^{\alpha_{i,s}^{U_k}}$ *recorded in each protected transaction* Tx' *are designed to appear random, lacking any discernible patterns or correlations that could be used to infer relationships between users in different smart contracts or even within the same contract.*
- ***Number of Users****: The number of users within each smart contract can vary, and there is no fixed ratio or pattern that links the number of users across different contracts within the same group.*

Given this setup, the information contained within any two smart contracts σ_s^i *from the same group* G_i *does not provide any direct or indirect indicators that could be used to link them together. The unique addresses, lack of patterns in transactions, and varying numbers of users all contribute to making each smart contract an isolated entity from the perspective of an attacker.*

For an internal attacker attempting to link n *smart contracts within the same group, the number of possible combinations they would need to consider is given by the binomial coefficient:*

$$P_s = \binom{n}{|\sigma_s^i|}$$

Given a sufficiently large number of smart contracts n*, the number of possible combinations* P_s *becomes extremely large, making it computationally infeasible for an attacker to correctly link the contracts without the specific knowledge that only the Coordinators possess. This inherent complexity ensures that the system remains secure against internal attacks that attempt to link smart contracts within the same group.*

Claim 9 *Addresses* A^{U_k} *belonging to the same user* U_k *cannot be linked together without the information of the Coordinators, even when the attacker is part of the same group* G_a*. (S6)*

Proof 9 *To analyze this, let's consider the following context:*

The internal attacker is a member of the group G_a, which consists of n smart contracts (σ_s^a). The attacker has access to m of these contracts, where $m \leq n$. Each smart contract contains a number of user addresses, and each user U_k within the group is represented by a distinct address in each of the smart contracts they participate in.

The attacker, being part of G_a, attempts to link different addresses belonging to the same user by analyzing the smart contracts available to them. The attacker knows that each user may participate in multiple smart contracts, but the user's addresses across these contracts are pseudonymous and do not reveal any direct correlation.

To perform the attack, the attacker would have to systematically try all possible combinations of addresses from the m smart contracts to identify which addresses might belong to the same user U_k. The total number of combinations the attacker must consider is given by:

$$P_u = |\sigma|^{m-1}$$

Where $|\sigma|$ represents the average number of addresses (users) in each smart contract. The attacker must also consider the possibility that not every user is present in every smart contract, further increasing the complexity of the attack.

To potentially reduce the number of combinations, the attacker might employ several strategies:

1. *Time Slot Analysis: The attacker could analyze specific time slots (t) where energy usage patterns might provide clues. For example, during nighttime when solar energy generation is low, the attacker could attempt to exclude combinations that aggregate higher energy production, thereby narrowing down the possibilities.*
2. *Pattern Recognition: Users typically exhibit consistent energy consumption behaviors. By analyzing the consumption patterns across different time slots, the attacker could attempt to identify and exclude combinations that deviate from typical profiles, further reducing the number of plausible combinations.*

After applying these reduction strategies, the effective number of possibilities for user address linkage might be reduced to:

$$P'_u \leq P_u$$

Where P'_u is the reduced number of possibilities after applying the time slot analysis and pattern

recognition techniques.

The computational effort required for the attacker to test each possible combination is given by:

$$C = P'_u \cdot t \cdot O_c$$

Where t represents the number of time slots analyzed, and O_c represents the computational cost of testing a single combination.

Despite these strategies, the computation required to attempt all possible combinations remains significant, especially as the number of smart contracts n and the number of users in the group increase. To maximize privacy, the system should aim to increase both the number of smart contracts (σ) and the number of users. This increases the complexity and reduces the feasibility of the attack.

Therefore, even with the insider knowledge available to the attacker, linking addresses belonging to the same user across multiple smart contracts remains highly challenging without the additional information held by the Coordinators. This ensures that the system maintains strong privacy protections against internal attacks.

5.3.2 External attacker

In this section, we consider the threat posed by an external attacker—a party that has access to the public information available on the blockchain but lacks insider knowledge about the internal workings of the system, such as the specific mappings between users and their multiple pseudonymous addresses. The following claim and proof demonstrate how the system effectively protects users' privacy against such attackers.

Claim 10 *The real electricity consumption value of user U_a can only be accurately calculated by summing the consumption across all their pseudonymous addresses. An external attacker, who lacks knowledge about the link between all of a user's addresses, will be unable to compute the true consumption value. Consequently, the attacker will only have access to seemingly random energy consumption values, with no meaningful relationship to the user's actual consumption profile. (S6)*

Proof 10 *To illustrate how the system protects user privacy from an external attacker, let's consider the adversary model outlined in Section 3.2. The attack scenario would proceed as follows:*

1. *The adversary A has access to $\gamma(U_k)$, which is the profile classification function. This function allows A to assign a user U_k to a specific electricity usage profile based on the observed data.*

However, A does not have access to the actual user identities or the mapping between users and their multiple pseudonymous addresses.

2. *The challenger (representing the system) possesses the true electricity consumption data E^{U_a} for user U_a. To protect U_a's privacy, the system applies a transformation function F that distributes E^{U_a} across multiple pseudonymous addresses $\alpha_{i,s}^{U_a}$, which are associated with different smart contracts σ_s^i. The transformation results in a set of seemingly random consumption values $E^{\alpha_{i,s}^{U_a}}$, each associated with a different address. The sum of these values across all addresses is equal to the true consumption value E^{U_a}, but individually, they appear random and unrelated.*

3. *The adversary A, still having access to the profile classification function $\gamma(U_k)$, attempts to classify the user based on the observed data. The adversary outputs a profile $\gamma' = \gamma(\alpha_{i,s}^{U_a})$, which is their best guess at the user's profile based on the transformed and distributed consumption values.*

4. *Because the values in $E^{\alpha_{i,s}^{U_a}}$ are random and do not directly correlate with U_a's true consumption pattern, the adversary is essentially making a random guess when attempting to classify the user. The probability that A correctly identifies the user's profile is given by:*

$$Pr[\gamma' = \gamma(U_a)] = 1/|\Gamma|$$

where $|\Gamma|$ is the total number of possible profiles. This probability is equivalent to the adversary choosing a profile at random, indicating that the attacker's ability to infer meaningful information is significantly diminished.

This approach ensures that individual electricity consumption data remains privacy-preserving. The key to this protection is the system's use of pseudonymous addresses and the random distribution of consumption values across multiple smart contracts. By doing so, the system effectively obscures the true consumption pattern of any individual user. Even though the attacker has access to all transactions on the blockchain, they are unable to link these transactions in a way that reveals the user's true electricity usage.

Moreover, this strategy prevents the external attacker from constructing a coherent or accurate profile of the user based on the observed data. The randomization and distribution of consumption values ensure that any analysis performed by the attacker will yield only noise, with no actionable insights into the user's habits or energy consumption patterns. This method is crucial for protecting users from being profiled or targeted based on their energy usage, thereby upholding the system's commitment to privacy and security.

6 Evaluation

In the following sections, we provide a comprehensive evaluation of our proposal to demonstrate its feasibility, scalability, and privacy-preserving capabilities within an IoT environment. The evaluation is divided into three key areas: experimental implementation, theoretical scalability analysis, and privacy assessment through simulation. Each area is designed to address specific aspects of the system's performance and security, ensuring that the proposal can be effectively deployed in real-world scenarios.

6.1 Distributed Implementation in an IoT Environment

To test the feasibility of implementing our proposal in an IoT environment, we will conduct a small-scale implementation using actual IoT devices. This setup will mimic a real-world environment, allowing us to evaluate the practical aspects of deploying our system. The primary goals of this experimental implementation are as follows:

- *Integration:* We will assess how seamlessly the proposed system can be integrated with existing IoT devices, such as smart meters.
- *Performance:* The performance of the system will be measured in terms of several key metrics:
  - **Gas Consumption:** In blockchain-based systems, "gas" refers to the computational resources required to execute transactions and smart contracts. We will measure the gas consumption associated with each operation in our system, particularly the execution of smart contracts and the storage of data on the blockchain. By analyzing these costs, we can determine the financial and computational feasibility of the system, especially when scaling to a larger number of users.
  - **Response Time:** We will evaluate the system's response time, which includes the time taken for a transaction to be processed, verified, and recorded on the blockchain. This metric is critical for assessing the system's usability in real-time energy trading scenarios, where delays in transaction processing could lead to inefficiencies or user dissatisfaction.

The outcomes of this experimental implementation will provide valuable insights into the practical challenges and benefits of deploying our system in real-world IoT environments, helping to refine the system's design and identify areas for optimization.

6.2 Theoretical Scalability Discussion

To complement the experimental evaluation, we will perform a theoretical study of the system's scalability. Scalability is a critical factor for the success of any large-scale deployment, particularly

in decentralized systems like ours, where the number of users and transactions can grow rapidly. Our analysis will focus on the following aspects:

We will analyze how the computational requirements of the system evolve as the number of users and transactions increases. This includes the processing time needed to execute smart contracts

The theoretical scalability analysis will provide a framework for understanding the system's limitations and guide the development of strategies to enhance its capacity to handle large-scale deployments.

6.3 Privacy Assessment through Simulation

Finally, to evaluate the privacy aspects of our system, we will simulate a large number of users in a controlled environment. This simulation allows us to assess the effectiveness of the system's privacy-preserving mechanisms without the need for a full-scale IoT deployment. The privacy assessment will focus on the following:

- *Data Anonymization:* We will test the effectiveness of the data anonymization techniques employed in our system, such as pseudonymous addresses and the distribution of user transactions across multiple smart contracts. The simulation will involve generating a variety of user behavior patterns and assessing how well the system can anonymize this data to prevent re-identification by an external or internal attacker.
- *Adversary Resistance:* The simulation will also assess the system's resilience against the adversary model defined in Section 3.2. We will simulate attacks by both external and internal attackers, attempting to link user addresses, infer electricity consumption patterns, and de-anonymize users. The effectiveness of these attacks will be measured by the success rate of the adversaries in compromising user privacy. By varying the attack strategies and the parameters of the system, we aim to identify any potential vulnerabilities and refine the system's defenses against privacy breaches.

The results of the privacy assessment will provide a rigorous evaluation of the system's ability to protect user privacy, ensuring that the proposed solution can withstand real-world adversarial conditions.

The combined insights from the experimental implementation, theoretical scalability analysis, and privacy assessment through simulation will offer a comprehensive understanding of our system's strengths and areas for improvement. This evaluation process is crucial for validating the feasibility, scalability, and privacy-preserving capabilities of our proposal, ensuring that it meets the requirements of a secure, efficient, and user-friendly decentralized energy trading system in an IoT environment.

7 Distributed implementation

In this section, we implement our proposal in a distributed environment to evaluate its feasibility and performance under real-world conditions. The implementation is designed to mimic a practical IoT-based smart grid system, allowing us to observe how the system behaves when deployed in a realistic setting with actual devices and network conditions.

7.1 Methodology

To conduct this evaluation, we opted for a small-scale IoT implementation. This approach allows us to gain an overall understanding of the system's functionality and cost behavior in a controlled yet realistic environment. Our primary objective is to assess the system's performance in a simulated real-world environment, focusing on key metrics such as integration with IoT devices, computational costs, and response times. This methodology also enables us to identify potential challenges and opportunities for optimization before considering larger-scale deployments.

7.2 Blockchain

In Section 1.1 of our work, we highlighted the importance of incorporating blockchain technology into our system. Blockchain provides a secure, transparent, and immutable framework for recording and validating transactions, making it an ideal technology for various applications, including decentralized energy trading in smart grids.

One example of a blockchain is the Bitcoin network, which has proven successful as a digital currency. However, Bitcoin's Proof of Work (PoW) consensus algorithm consumes a significant amount of computational power and energy, which can be a drawback for energy-efficient applications like the smart grid, particularly when IoT devices such as smart meters are involved.

Other blockchain platforms, such as Hyperledger Fabric [3], are designed for enterprise applications and offer greater privacy and control over data. However, these platforms typically require user identification and operate within a permissioned network, which does not align with the decentralized and open nature of our proposal.

Ethereum is another popular blockchain platform that offers programmable smart contracts and a more flexible framework [47]. Ethereum's recent transition to a Proof of Stake (PoS) consensus algorithm reduces the computational resources required, making it more suitable for IoT environments. Despite this improvement, Ethereum's gas fees and transaction throughput can still pose challenges in a high-frequency energy trading environment.

To address these concerns, we opted to use IOTA as our test environment blockchain. IOTA is a distributed ledger technology specifically designed for the Internet of Things (IoT). It utilizes a unique data structure called the Tangle, which promises low computational requirements, feeless transactions, and scalability, making it an excellent fit for smart grid systems. Comparative studies have shown that IOTA outperforms Ethereum in IoT scenarios in terms of performance, resource consumption, and scalability [4], further justifying our choice.

7.3 Dataset

Our dataset, sourced from [15], encompasses diverse user profiles spanning an entire year. It comprises records of electricity consumption and production in 15-minute intervals for 51 users, including a mix of households and a public building. Notably, some households are equipped with solar panels, contributing to the variability in consumption and production patterns.

We conducted simulation tests with different groups of users extracted from the dataset, gathering data such as gas consumption for each user's transactions and the electricity consumption of each smart meter. These simulations help us understand how the system behaves with real-world data and provide insights into its performance, scalability, and privacy-preserving capabilities.

7.4 Test environment

To explore the challenges inherent in smart grid implementations, we proposed a testing environment that simulates a distributed smart grid architecture. This environment enables us to develop and evaluate a demonstrator that meets the requirements of our proposal. The environment is based on a previous work done in [11].

The testing environment features a three-layer architecture comprising various actors and components: user interface nodes, distribution nodes, and a master node. These components are depicted in Fig. 4 and described as follows:

1. *User Interface Nodes*: This layer consists of prosumers and smart meters. Prosumers are the end users who generate and consume electricity, often from renewable sources like wind turbines and solar panels. They play a crucial role in the smart grid by balancing supply and demand. Smart meters are IoT devices that monitor and report electricity consumption and production in real-time, providing data for controlling and optimizing energy usage.
2. *Distribution Nodes*: The middle layer serves as an intermediary between the prosumers and the transmission layer of the grid, represented by the coordinators in our proposal. Distribution nodes manage the flow of data and energy within the smart grid, ensuring efficient communication and transaction processing between prosumers and the grid.

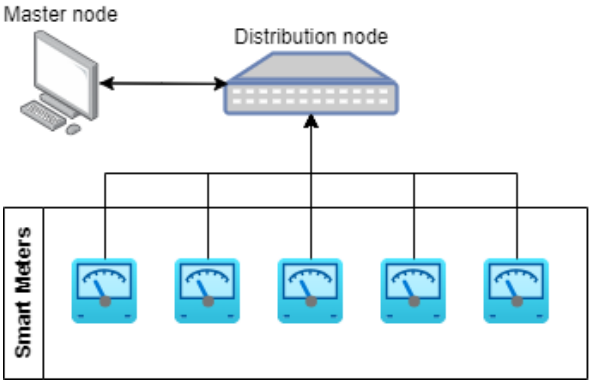

Figure 4: Distributed implementation architecture

3. *Master Node*: The master node is the central authority that oversees the entire operation of the smart grid. It regulates system parameters, adds or removes nodes, initiates or terminates processes, and performs comprehensive analysis based on data collected from distribution nodes and smart meters. The master node also deploys smart contracts on the IOTA test ledger, manages the simulation environment, and provides guidance to distribution and user interface nodes.

For the implementation, we used Raspberry Pi devices to simulate distribution nodes and user interface nodes. This choice was motivated by the need to closely replicate the computational capacity and energy consumption of typical IoT devices used in smart meters.

The master node was implemented on a personal computer to facilitate easy communication with all other nodes and to support data visualization and verification. The personal computer's higher computational power and enhanced connectivity options provided a stable platform for managing the simulation and analyzing the results.

To enable communication between all components, a Local Area Network (LAN) was established using WiFi. A predefined set of IP addresses and ports was assigned to each component to facilitate the exchange of messages.

7.4.1 Steps in the Demonstrator

Our demonstrator follows a series of steps to evaluate the proposed system: setup, test definition, electricity trading, and data gathering. Each step is explained in detail below.

7.4.2 Setup

In the setup phase, the necessary code and datasets are distributed to the respective nodes in the system. This phase ensures that all components are properly configured and ready for the simulation.

Each smart meter is equipped with the following:

- *Code for Data Collection*: Scripts for collecting electricity consumption and production data.
- *Anonymization Code*: Scripts for interacting with the IOTA distributed ledger and executing smart contracts.
- *Datasets*: Historical data for electricity consumption and production to simulate realistic scenarios.
- *Master Address*: The IP address of the master node to enable communication.

Each distribution node receives:

- *Management Code*: Scripts for managing the smart meters assigned to it.
- *Master Address*: The IP address of the master node.

The master node is provided with:

- *Initialization Code*: Scripts for initializing the simulation environment.
- *Registration Code*: Scripts for registering smart meters and distribution nodes.
- *Smart Contract Deployment Code*: Scripts for deploying smart contracts with specified parameters.
- *Management Code*: Scripts for managing the entire simulation process.

7.5 Test Definition

During the test definition phase, the master node initializes the simulation environment and registers the smart meters and distribution nodes. The detailed steps include:

1. *Initialization*: The master node starts the server and awaits connections from distribution nodes and smart meters.

2. *Registration*: Smart meters and distribution nodes register with the master node, providing their respective addresses and ports.

3. *Grouping*: The master node groups distribution nodes with smart meters based on predefined criteria. Each distribution node is assigned a set of smart meters.

7.5.1 Simulation Parameters

The master node configures the simulation parameters, which dictate the granularity and duration of data collection. These parameters include:

- *Timeslot Length*: The duration of each data collection period.
- *Time Interval*: The total duration of the simulation.
- *Number of Smart Contracts*: The number of smart contracts deployed per group.

7.6 Energy Trading

In the energy trading phase, smart meters begin transmitting their electricity consumption and production data to the assigned distribution nodes. The detailed process is as follows:

1. *Data Request*: Distribution nodes request data from their assigned smart meters, specifying the timeslot length and time interval.

2. *Data Transmission*: Smart meters respond by sending actual and anonymized data related to consumption and production, along with computation time and gas consumption. This data is transmitted at regular intervals based on the timeslot length.

7.7 Data Gathering

The final phase involves gathering and analyzing the collected data. The master node undertakes the following steps:

1. *Data Collection*: Distribution nodes forward the collected data to the master node.

2. *Data Processing*: The master node processes the received data, enabling it to perform detailed analysis and generate insights.

The data collected includes real and anonymized electricity consumption, gas consumption, response times, and other relevant metrics. This data is stored in JSON format on the master node for further analysis.

7.8 Parameters

To evaluate the performance and privacy of our proposal, we analyze how changes in certain system parameters impact the system's behavior. The key performance metrics considered include:

- *Gas Consumption*: We measure the gas consumption for each transaction and each user to evaluate the computational cost of transactions on the IOTA ledger.
- *Response Time*: We analyze the response time for each transaction to assess the feasibility and real-time performance of our system.

Given that the number of users is fixed at six, we will explore how varying the number of smart contracts per group affects these performance metrics. Increasing the number of smart contracts is expected to enhance the system's privacy, but we will also examine its impact on system performance.

The number of smart contracts per group in our evaluation ranges from two to 15. Simulations cover a full 24-hour period, with data collection occurring every 15 minutes. This setup provides a comprehensive view of how the system performs under different configurations and allows us to assess the trade-offs between privacy and performance in a real-world-like scenario.

7.9 Results

The results of our evaluation, as depicted in Fig. 5, reveal a clear linear relationship between the number of smart contracts deployed in the system and the corresponding increases in both gas consumption and time elapsed for transaction processing. This trend is expected given the structure of our system, where each user must interact with a growing number of smart contracts as the system scales.

As the number of smart contracts increases, each user must execute a transaction for each contract, leading to a proportional rise in the total number of transactions. Gas consumption, which represents the computational resources required to process these transactions on the IOTA ledger, thus scales linearly with the number of smart contracts.

The linear increase in gas consumption is a direct result of the system's architecture, where each smart contract requires a distinct transaction. Since gas fees are incurred with every transaction, adding more smart contracts directly translates to higher cumulative gas costs. While the predictable nature

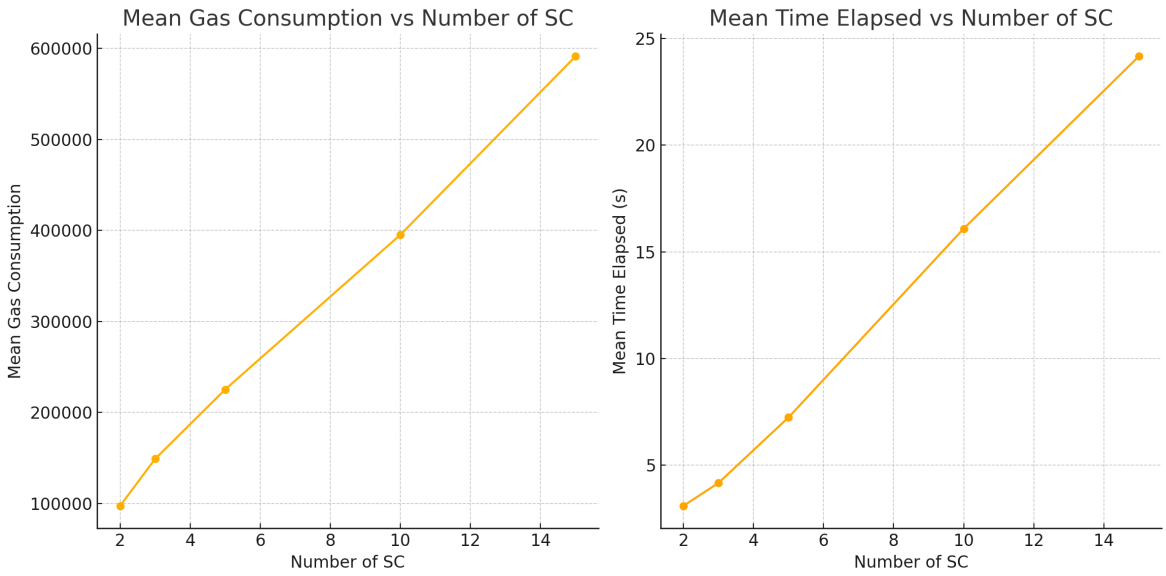

Figure 5: Gas consumption and time elapsed per time slot operation

of this increase simplifies forecasting, it also highlights the importance of optimizing smart contract interactions to minimize unnecessary gas consumption, especially in large-scale deployments where operational costs could become significant.

In addition to gas consumption, the time elapsed per transaction also shows a linear increase with the number of smart contracts. This is expected, as each additional smart contract introduces an incremental computational overhead. The more contracts a user needs to interact with, the longer it takes to process all required transactions. However, it is important to note that while the time per transaction increases, it remains within a manageable range, even at higher scales.

For example, when operating with 15 smart contracts, the mean transaction time recorded is 25 seconds. Given that each time slot in our simulation is 15 minutes, this transaction time represents a small fraction of the available time window. This indicates that, despite the linear increase in processing time, the system remains feasible for real-time operations in a smart grid environment. Users can continue to interact with the grid and execute transactions without experiencing significant delays, ensuring that the system can support timely energy trading and grid balancing activities.

In conclusion, the results from our evaluation demonstrate that while the system's gas consumption and processing time increase linearly with the number of smart contracts, the overall performance remains within acceptable limits for real-time operations.

These findings confirm the feasibility of our proposal for deployment in a real-world IoT-based smart grid environment, as well as verifying the requirements F1, F2 and F3 of our proposal.

8 Scalability discussion

To perform a theoretical scalability analysis based on the linear behavior of transactions in relation to the number of smart contracts and considering a hierarchical architecture with a tree structure, we formalize the system's scalability using a model that accounts for the tree-like organization of smart contracts. Below, we outline the key points of this analysis and provide a formalization of the system's scalability.

In a hierarchical smart contract architecture, the contracts are organized in a tree structure where:

- **Root Contract**: The top-level smart contract, which may manage high-level operations or oversee a group of subcontracts.
- **Intermediate Contracts**: These are the children of the root contract, responsible for managing sub-groups of other contracts.
- **Group Contracts**: The bottom-level smart contracts that directly interact with individual users or devices.

Each level of the tree introduces a layer of abstraction and management, allowing the system to scale by adding more branches (intermediate contracts) and leaves (group contracts).

8.1 Scalability Formalization

The scalability of the system can be analyzed by considering the following variables:

- N: The total number of users in the system.
- C: The total number of smart contracts.
- L: The number of levels in the hierarchical tree structure.
- B_i: The branching factor at level i, representing the number of child contracts each parent contract at level i has.
- T: The total number of transactions processed by the system.

Given the tree structure, the total number of smart contracts C can be expressed as:

$$C = \sum_{i=0}^{L} B_i$$

where $B_0 = 1$ (the root contract).

The number of transactions T scales linearly with the number of smart contracts, as each contract introduces a certain number of transactions. Therefore, for each user u, the number of transactions T can be formalized as:

$$T = N \times B_i^u$$

As the system scales, adding more contracts increases the total number of transactions linearly because each new contract requires additional transactions to manage its state, interact with other contracts, and process user requests.

8.2 Scalability Considerations

- **Branching Factor**: Increasing the branching factor B_i at any level i increases the number of smart contracts exponentially at lower levels of the tree. However, the linear relationship between the total number of contracts and the total number of transactions still holds due to the structure of the system.
- **Depth of the Tree**: Increasing the depth L of the tree adds more levels of contracts but does not necessarily increase the number of transactions per user. Instead, it redistributes the transactions across different levels, potentially reducing the load on any single contract.
- **Load Balancing**: Effective load balancing across the tree structure is essential to ensure that no single contract becomes a bottleneck. This can be achieved by optimizing the branching factors and the distribution of users among the leaf contracts.

8.3 Formalization of Scalable System

To ensure that the system scales effectively, we should aim for an optimal branching factor B at each level such that the load on individual contracts remains manageable while the overall system can accommodate a growing number of users.

We define the optimal branching factor B^* as the factor that maximizes the system's efficiency, balancing the trade-off between the depth of the tree and the number of contracts at each level:

$$B^* = \text{argmax}_B \left(\frac{\text{Total Transactions}}{\text{Total Contracts}} \right)$$

This optimization ensures that the system can handle an increasing number of users and contracts without degrading performance, maintaining a linear increase in transaction costs relative to the number of users.

8.4 Tree-Based Smart Contract System

The hierarchical architecture allows for a decentralized and scalable approach to managing smart contracts. By structuring the contracts in a tree format, we can distribute the load across multiple levels, avoiding bottlenecks and ensuring that the system can grow with the number of users.

The total computational overhead O of the system can be expressed as:

$$O = T \times C$$

Since T scales linearly with C, the system's computational complexity remains manageable as long as C grows in a controlled manner.

- **Linear Scalability**: The system maintains linear scalability by ensuring that each additional smart contract only introduces a proportional increase in the number of transactions.
- **Hierarchical Optimization**: The tree structure optimizes the distribution of transactions, reducing the load on any individual contract and allowing the system to scale horizontally by adding more branches or vertically by increasing the depth of the tree.
- **Performance Trade-Offs**: As the system grows, careful management of the tree structure, including branching factors and contract interactions, is required to maintain performance and scalability.

By leveraging a hierarchical tree structure for smart contracts, the proposed system can scale effectively while maintaining a linear relationship between the number of smart contracts and the total number of transactions. This architecture allows for efficient load distribution and enables the system to handle an increasing number of users and transactions without compromising performance. Formalizing the scalability of the system in this way provides a clear framework for understanding and optimizing its growth potential as more users are added to the network.

9 Simulation of the Profile Privacy Protection

In this section, we aim to empirically demonstrate the effectiveness of our system in preserving user privacy by concealing their electricity profiles from potential attackers. Through a series of simulations, we validate the theoretical conclusions drawn in Section 5.3.2, confirming the robustness of our privacy-preserving mechanisms against adversarial attempts to infer user profiles.

9.1 Objectives

The primary objective of this simulation is to empirically validate the theoretical probability:

$$Pr[\gamma' = \gamma(U_k)] = \frac{1}{|\Gamma|}$$

This probability represents the likelihood that an adversary can correctly infer the electricity profile of a user U_k, where $|\Gamma|$ is the total number of possible profiles. By simulating a scenario in which multiple users produce and consume electricity, we create an adversarial environment to assess how well our system protects user profiles from being accurately identified by an attacker.

9.2 Methodology

To simulate an adversarial environment, we analyzed the electricity profiles of several users and applied our proposed privacy-preserving protocol. The simulation involves both the generation of anonymized electricity consumption data and the simulation of an adversary attempting to de-anonymize this data. By comparing the adversary's success rate with the theoretical expectations, we can evaluate the effectiveness of our protocol in practice.

9.3 Dataset

For this experiment, we used the same dataset as described in Section 7.3. The dataset includes individual electricity consumption data for 51 users, recorded at 15-minute intervals throughout the day. This dataset provides a realistic basis for simulating electricity usage patterns and testing the privacy-preserving mechanisms of our system.

Figure 6 visualizes the daily electricity usage patterns of the users in the dataset. The y-axis represents the electricity consumption or production (E^{U_k}) of a user U_k, while the x-axis denotes the time slots (t) throughout the day. Distinct trends are evident, such as peak consumption periods during morning and evening hours and periods of electricity production during daylight hours for users with photovoltaic (PV) systems.

The visualization highlights the diversity of usage patterns across different users. Notably, some users, particularly those with PV systems, have periods of negative consumption (indicating electricity production), while others consistently consume electricity, resulting in profiles predominantly above the zero line. The dataset also includes profiles of a public building, which exhibit distinctive consumption patterns that could be easily recognizable without privacy-preserving techniques.

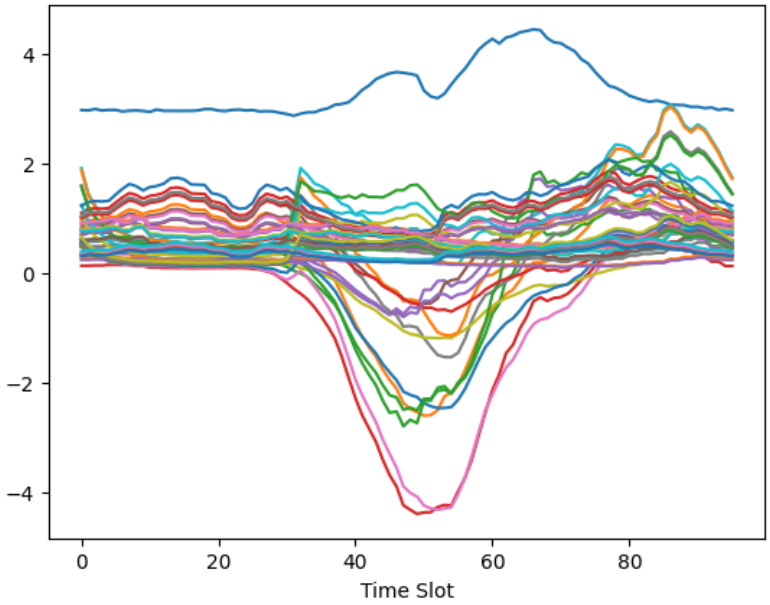

Figure 6: Visualization of Daily Electricity Usage Patterns

9.4 Profiling

To identify distinct electricity usage profiles within the dataset, we employed the k-means clustering algorithm. K-means is an iterative clustering algorithm that partitions the dataset into $|\Gamma|$ clusters, where each data point is assigned to the cluster with the nearest mean. This method allows us to recreate the profiling process described in Section 3.1.

By applying k-means to the electricity usage data, we divided the dataset into several profiles ($|\Gamma|$) based on the electricity patterns of the 51 users over an entire year. Each user was then assigned a profile γ corresponding to their typical electricity usage pattern.

9.5 Challenger Simulation

In the challenger simulation, we simulate the process of anonymizing user data by dividing each user's electricity usage pattern (E^{U_k}) into n random entries. These entries range from the minimum to the maximum values in the dataset, with their sum equaling the original E^{U_k}. This process generates $D \cdot n$ anonymized electricity profiles ($E^{U'_k}$), where D is the number of days in the dataset.

For each $E^{U'_k}$, we retain the information about the original user U_k from which it was derived. This setup allows us to track the effectiveness of the anonymization process and assess the adversary's ability to correctly identify the user's profile.

9.6 Adversary Simulation

The adversary simulation involves an attempt to de-anonymize the electricity usage data and correctly identify the user's profile. The adversary uses the k-means algorithm's predict function, denoted as $\gamma()$, to assign each anonymized electricity profile ($E^{U'_k}$) to a profile γ'.

The adversary's task is to compute γ' for each $E^{U'_k}$ and compare it to the original profile γ of the user U_k. The simulation outputs a value of 1 if the adversary correctly identifies the user's profile ($\gamma' = \gamma$) and 0 otherwise. By aggregating these results, we can measure the adversary's success rate and compare it to the theoretical expectation of $\frac{1}{|\Gamma|}$.

9.7 Parameters

In this subsection, we examine the parameters that vary in this simulation to evaluate the effectiveness of the adversary in determining the true profile of a user.

9.7.1 Number of Profiles

The number of electricity profiles ($|\Gamma|$) into which the dataset is divided is a critical parameter. Given the fixed number of users ($|U| = 51$), we studied how the number of profiles affects the adversary's success rate ($Pr[\gamma' = \gamma(U_k)]$) across a range of values ($|\Gamma| \in [2, 51]$). This analysis helps us understand the relationship between the granularity of profiling and the difficulty of de-anonymizing users.

9.7.2 Number of Smart Contracts

We also examined how the number of smart contracts per group ($|\sigma_s^i|$) affects the adversary's ability to correctly identify user profiles. We varied the number of smart contracts ($|\sigma_s^i|$) from 1 to 50 and measured the resulting success rate of the adversary. This analysis helps determine the minimum number of smart contracts required to effectively obscure user profiles and enhance privacy.

9.8 Results

Figure 7 [1] illustrates the accuracy of the adversary model across various configurations. The x-axis represents the number of clusters ($|\Gamma|$) selected for the k-means algorithm, while the y-axis indicates

[1]Note that the legend is incorrect, it should be 1, 2, 5, 15, 20, 50

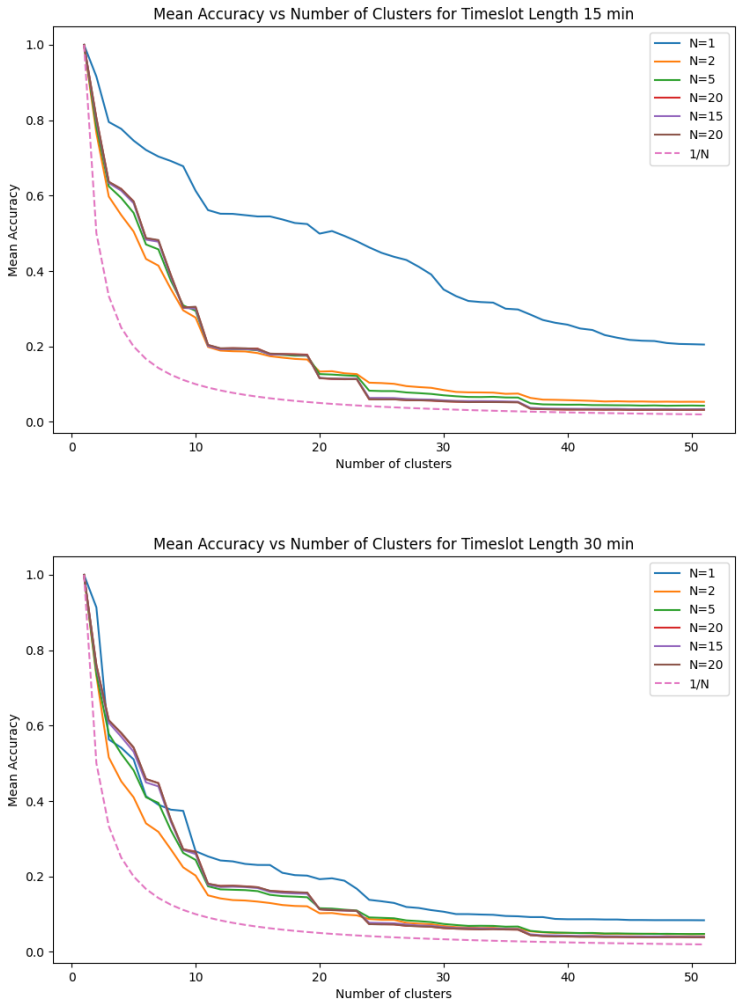

Figure 7: Adversary Success Rate Across Different Configurations

the proportion of instances where the adversary successfully identified the correct profile (γ'). Each line in the figure corresponds to simulations with different numbers of smart contracts per group ($|\sigma_s^i|$).

The figure includes results for timeslot lengths of 15 and 30 minutes. As the timeslot length increases, the adversary's accuracy approaches the theoretical benchmark of $\frac{1}{|\Gamma|}$, represented by the dashed light blue line. This trend suggests that larger timeslots provide better privacy by aggregating data, though at the cost of reduced granularity in the readings.

1. *Impact of Smart Contracts on Privacy*: The results show that when the number of smart contracts per group exceeds one, the adversary's success rate in correctly identifying user profiles decreases significantly. This confirms that using multiple smart contracts effectively enhances privacy by dispersing and obfuscating user data across several contracts.

2. *Timeslot Length and Privacy*: The simulations indicate that longer timeslots (e.g., 30 minutes) contribute to better privacy protection, as the adversary's success rate approaches the theoretical

minimum of $\frac{1}{|\Gamma|}$. This finding suggests that timeslot aggregation can be a useful technique for improving privacy, though it may also reduce the precision of energy consumption data.

3. *Profile Granularity*: As the number of clusters ($|\Gamma|$) increases, the adversary's accuracy naturally decreases. This is expected, as a larger number of profiles make it more challenging for the adversary to match anonymized data to the correct profile.

The results from our simulation confirm the effectiveness of the proposed system in preserving user privacy. By increasing the number of smart contracts per group and adjusting the timeslot length, the system can significantly reduce the accuracy of an adversarial attack, bringing it close to the theoretical expectation of random guessing. These findings align with our theoretical predictions and underscore the importance of these parameters in safeguarding user privacy in smart grid systems.

10 Conclusions and future work

In this work, we have explored the effectiveness of hierarchical smart contracts in preventing user profiling within decentralized energy distribution systems. The primary goal was to demonstrate that through the careful design and implementation of a hierarchical structure of smart contracts, it is possible to maintain user privacy while ensuring efficient energy trading within a smart grid environment. While extending the work done in [10] and using the testing environment in [11] published in conference papers.

We began by analyzing the current landscape of smart grids, emphasizing the critical importance of privacy in modern energy distribution networks. As smart grids increasingly integrate distributed energy resources and IoT devices, the exchange of data between various entities becomes essential for optimizing energy distribution and consumption. However, this increased data exchange also heightens the risk of user data being exposed and exploited by malicious entities, making privacy a paramount concern.

To address these challenges, we proposed a solution based on hierarchical smart contracts organized in a tree structure. This architecture enables the aggregation and anonymization of user data at various levels within the energy distribution hierarchy. By implementing this method, we successfully obscured individual user data, significantly reducing the ability of potential attackers to accurately profile users based on their energy consumption patterns.

The theoretical foundation of our approach was supported by a detailed scalability analysis, which formalized the relationship between the number of smart contracts, the depth of the hierarchical tree, and the system's overall performance. Our analysis demonstrated that by optimizing the branching factors and carefully managing the tree structure, the system could scale effectively, maintaining a linear relationship between the number of smart contracts and the total number of transactions. This scalability is crucial for deploying the system in real-world smart grid environments where the number of users and transactions can grow rapidly.

To validate our approach, we conducted a series of simulations designed to empirically test the system's privacy-preserving capabilities. The results of these simulations were encouraging, showing that as the number of smart contracts within the system increased, the level of privacy improved correspondingly. This improvement was evident from the decreased success rate of adversary models in correctly identifying user profiles. Specifically, the implementation of multiple smart contracts per group led to a significant reduction in the probability of successful internal attacks. The increased complexity and randomness introduced by the hierarchical structure effectively obscured the correlation between user behavior and the observed data, making it more challenging for adversaries to de-anonymize users.

Our findings have significant implications for the development of secure and privacy-preserving smart grids. As smart grid technology continues to evolve, the need for robust privacy measures will

become increasingly critical. Our approach provides a viable pathway for enhancing user privacy without compromising the functionality and efficiency of the grid. The hierarchical smart contract architecture not only protects user data but also ensures that the system can scale to accommodate a growing number of users and transactions.

Moreover, this study underscores the importance of continued research into the integration of blockchain technologies and smart contracts within IoT ecosystems. The success of our approach suggests that further exploration into hierarchical structures and distributed ledger technologies could yield even more advanced privacy solutions.

In conclusion, our work demonstrates the potential of hierarchical smart contracts as a powerful tool for enhancing privacy in decentralized energy distribution systems. The scalability, effectiveness, and flexibility of this approach make it a promising direction for future research and development in secure and privacy-preserving smart grid technologies.

10.1 Future Work

There are several avenues for future work that could build upon the findings of this research. First, the scalability of the proposed solution should be tested in larger, more complex smart grid environments. This will help to better understand the limitations and potential of hierarchical smart contracts in real-world applications.

Second, further investigation into the trade-offs between privacy and system performance is necessary. While our results show a clear benefit in terms of privacy, the impact on system latency and computational overhead needs to be thoroughly evaluated.

Finally, expanding the study to include different types of adversary models and attack scenarios would provide a more comprehensive assessment of the system's robustness. Understanding how the system behaves under various conditions will be essential for deploying this technology in diverse and potentially hostile environments.

Bibliography

[1] Anak Agung Gde Agung and Rini Handayani. "Blockchain for smart grid". In: *Journal of King Saud University - Computer and Information Sciences* 34.3 (2022), pp. 666–675. ISSN: 1319-1578.

[2] Nurzhan Zhumabekuly Aitzhan and Davor Svetinovic. "Security and Privacy in Decentralized Energy Trading Through Multi-Signatures, Blockchain and Anonymous Messaging Streams". In: *IEEE Transactions on Dependable and Secure Computing* 15.5 (2018), pp. 840–852.

[3] Christian Cachin et al. "Architecture of the hyperledger blockchain fabric". In: *Workshop on distributed cryptocurrencies and consensus ledgers*. Vol. 310. 4. Chicago, IL. 2016, pp. 1–4.

[4] Xuan Chen et al. "A comparison of distributed ledger technologies in iot: Iota versus ethereum". In: *2021 20th International Symposium on Communications and Information Technologies (ISCIT)*. IEEE. 2021, pp. 182–187.

[5] US Congress. "Energy independence and security act of 2007". In: *Public law* 2 (2007), pp. 110–140.

[6] Alex de Vries. "Bitcoin's energy consumption is underestimated: A market dynamics approach". In: *Energy Research & Social Science* 70 (2020), p. 101721. ISSN: 2214-6296.

[7] Soma Shekara Sreenadh Reddy Depuru, Lingfeng Wang, and Vijay Devabhaktuni. "Smart meters for power grid—Challenges, issues, advantages and status". In: *Renewable and Sustainable Energy Reviews* 15.6 (2011), pp. 2736–2742.

[8] Xi Fang et al. "Smart Grid — The New and Improved Power Grid: A Survey". In: *IEEE Communications Surveys & Tutorials* 14.4 (2012), pp. 944–980.

[9] Xin Fang et al. "Smart grid—The new and improved power grid: A survey". In: *IEEE communications surveys & tutorials* 14.4 (2012), pp. 944–980.

[10] Joan Ferré-Queralt, Jordi Castellà-Roca, and Alexandre Viejo. "Smart Contracts for a Secure and Privacy-Preserving Smart Grid". In: *International Conference on Risks and Security of Internet and Systems*. Springer. 2023, pp. 103–118.

[11] Joan Ferré-Queralt et al. "Leveraging a Smart Grid Framework for Interoperability with a Distributed Ecosystem". In: *XVIII Reunión Española sobre Criptología y Seguridad de la Información*. RECSI. 2024, TO APPEAR.

[12] Keke Gai et al. "Permissioned blockchain and edge computing empowered privacy-preserving smart grid networks". In: *IEEE Internet of Things Journal* 6.5 (2019), pp. 7992–8004.

[13] Feng Gao et al. “A blockchain-based privacy-preserving payment mechanism for vehicle-to-grid networks”. In: *IEEE network* 32.6 (2018), pp. 184–192.

[14] Clark W Gellings. *The smart grid: Enabling energy efficiency and demand response*. CRC press, 2009.

[15] Calvin Goncalves et al. “Dataset of an energy community’s generation and consumption with appliance allocation”. In: *Data in Brief* 45 (2022), p. 108590.

[16] Avi Gopstein et al. *NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 4.0*. Feb. 2021.

[17] Zhitao Guan et al. “Privacy-Preserving and Efficient Aggregation Based on Blockchain for Power Grid Communications in Smart Communities”. In: *IEEE Communications Magazine* 56.7 (2018), pp. 82–88.

[18] Jianchao Hou, Haicheng Wang, and Pingkuo Liu. “Applying the blockchain technology to promote the development of distributed photovoltaic in China”. In: *International Journal of Energy Research* 42.6 (2018), pp. 2050–2069.

[19] Chunqiang Hu et al. “Smart contract assisted privacy-preserving data aggregation and management scheme for smart grid”. In: *IEEE Transactions on Dependable and Secure Computing* (2023).

[20] Jiawen Kang et al. “Enabling Localized Peer-to-Peer Electricity Trading Among Plug-in Hybrid Electric Vehicles Using Consortium Blockchains”. In: *IEEE Transactions on Industrial Informatics* 13.6 (2017), pp. 3154–3164.

[21] Rabiya Khalid et al. “A Blockchain-Based Load Balancing in Decentralized Hybrid P2P Energy Trading Market in Smart Grid”. In: *IEEE Access* 8 (2020), pp. 47047–47062.

[22] Sarmadullah Khan and Rafiullah Khan. “Multiple authorities attribute-based verification mechanism for Blockchain mircogrid transactions”. In: *Energies* 11.5 (2018), p. 1154.

[23] Sunny King and Scott Nadal. “Ppcoin: Peer-to-peer crypto-currency with proof-of-stake”. In: *self-published paper, August* 19.1 (2012).

[24] Daniel Larimer. “Delegated proof-of-stake (dpos)”. In: *Bitshare whitepaper* 81 (2014), p. 85.

[25] Aron Laszka et al. “TRANSAX: A blockchain-based decentralized forward-trading energy exchanged for transactive microgrids”. In: *2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE. 2018, pp. 918–927.

[26] Chao Liu et al. “Cyber security and privacy issues in smart grids”. In: *IEEE Communications Surveys & Tutorials* 14.4 (2012), pp. 981–997.

[27] Xin Lu et al. “Blockchain-based distributed energy trading in energy Internet: An SDN approach”. In: *IEEE access* 7 (2019), pp. 173817–173826.

[28] Azhar Mahmood et al. “An efficient and privacy-preserving blockchain-based secure data aggregation in smart grids”. In: *Sustainable Energy Technologies and Assessments* 60 (2023), p. 103414.

[29] Sarah Manski. “Building the blockchain world: Technological commonwealth or just more of the same?” In: *Strategic Change* 26.5 (2017), pp. 511–522.

[30] Weiyi Meng, Yue Zheng, and Vincent WS Wong. “Smart grid neighborhood area networks: a survey”. In: *IEEE Network* 27.1 (2013), pp. 8–13.

[31] Esther Mengelkamp et al. “A blockchain-based smart grid: towards sustainable local energy markets”. In: *Computer Science-Research and Development* 33 (2018), pp. 207–214.

[32] Aaron R Metke and Randy L Ekl. “Security Technology for Smart Grid Networks”. In: *Proceedings of the IEEE*. Vol. 98. 1. IEEE. 2010, pp. 2–5.

[33] Mihail Mihaylov et al. “NRGcoin: Virtual currency for trading of renewable energy in smart grids”. In: *11th International Conference on the European Energy Market (EEM14)*. 2014, pp. 1–6.

[34] Muhammad Baqer Mollah et al. “Blockchain for Future Smart Grid: A Comprehensive Survey”. In: *IEEE Internet of Things Journal* 8.1 (2021), pp. 18–43.

[35] Satoshi Nakamoto. “Bitcoin whitepaper”. In: *URL: https://bitcoin. org/bitcoin. pdf-(: 17.07. 2019)* (2008).

[36] Yael Parag and Benjamin K Sovacool. “Electricity markets design for the prosumer era”. In: *Nature Energy* 1.4 (2016), p. 16032.

[37] Ioana PETCU. “How are EU electricity prices formed and why have they soared?” In: . (2022). URL: https://www.eurelectric.org/in-detail/electricity%5C_prices%5C_explained/ (visited on 09/30/2022).

[38] Serguei Popov. “The tangle”. In: *White paper* 1.3 (2018), p. 30.

[39] Farrokh Rahimi and Ali Ipakchi. “Demand Response as a Market Resource Under the Smart Grid Paradigm”. In: *IEEE Transactions on Smart Grid* 1.1 (2010), pp. 82–88.

[40] K.S. Reddy et al. “A review of Integration, Control, Communication and Metering (ICCM) of renewable energy based smart grid”. In: *Renewable and Sustainable Energy Reviews* 38 (2014), pp. 180–192. ISSN: 1364-0321.

[41] Omaji Samuel and Nadeem Javaid. “A secure blockchain-based demurrage mechanism for energy trading in smart communities”. In: *International Journal of Energy Research* 45.1 (2021), pp. 297–315.

[42] Omaji Samuel and Nadeem Javaid. “GarliChain: A privacy preserving system for smart grid consumers using blockchain”. In: *International Journal of Energy Research* 46.15 (2022), pp. 21643–21659.

[43] Ye-Byoul Son et al. “Privacy-preserving peer-to-peer energy trading in blockchain-enabled smart grids using functional encryption”. In: *Energies* 13.6 (2020), p. 1321.

[44] Melanie Swan. *Blockchain: Blueprint for a new economy*. " O’Reilly Media, Inc.", 2015.

[45] Blockchain-Based Microgrids Energy Trading. “DEAL: Differentially Private Auction for Blockchain-Based Microgrids Energy Trading”. In: ().

[46] Ifiok Anthony Umoren et al. “Blockchain-based energy trading in electric-vehicle-enabled microgrids”. In: *IEEE Consumer Electronics Magazine* 9.6 (2020), pp. 66–71.

[47] Gavin Wood et al. “Ethereum: A secure decentralised generalised transaction ledger”. In: *Ethereum project yellow paper* 151.2014 (2014), pp. 1–32.

[48] Wenti Yang et al. “Autonomous and privacy-preserving energy trading based on redactable blockchain in smart grid”. In: *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE. 2020, pp. 1–6.

[49] Peng Zhuang, Talha Zamir, and Hao Liang. “Blockchain for Cybersecurity in Smart Grid: A Comprehensive Survey”. In: *IEEE Transactions on Industrial Informatics* 17.1 (2021), pp. 3–19.

[50] Aviv Zohar. “Bitcoin: under the hood”. In: *Communications of the ACM* 58.9 (2015), pp. 104–113.