Safe and Private Forward-Trading Platform for Transactive Microgrids

SCOTT EISELE, Vanderbilt University
TAHA EGHTESAD, University of Houston
KEEGAN CAMPANELLI, Vanderbilt University
PRAKHAR AGRAWAL, Vanderbilt University
ARON LASZKA, University of Houston
ABHISHEK DUBEY, Vanderbilt University

Power grids are evolving at an unprecedented pace due to the rapid growth of distributed energy resources (DER) in communities. These resources are very different from traditional power sources as they are located closer to loads and thus can significantly reduce transmission losses and carbon emissions. However, their intermittent and variable nature often results in spikes in the overall demand on distribution system operators (DSO). To manage these challenges, there has been a surge of interest in building decentralized control schemes, where a pool of DERs combined with energy storage devices can exchange energy locally to smooth fluctuations in net demand. Building a decentralized market for transactive microgrids is challenging because even though a decentralized system provides resilience, it also must satisfy requirements like privacy, efficiency, safety, and security, which are often in conflict with each other. As such, existing implementations of decentralized markets often focus on resilience and safety but compromise on privacy. In this paper, we describe our platform, called TRANSAX, which enables participants to trade in an energy futures market, which improves efficiency by finding feasible matches for energy trades, enabling DSOs to plan their energy needs better. TRANSAX provides privacy to participants by anonymizing their trading activity using a distributed mixing service, while also enforcing constraints that limit trading activity based on safety requirements, such as keeping planned energy flow below line capacity. We show that TRANSAX can satisfy the seemingly conflicting requirements of efficiency, safety, and privacy. We also provide an analysis of how much trading efficiency is lost. Trading efficiency is improved through the problem formulation which accounts for temporal flexibility, and system efficiency is improved using a hybrid-solver architecture. Finally, we describe a testbed to run experiments and demonstrate its performance using simulation results.

CCS Concepts: • **Computer systems organization** → *Embedded and cyber-physical systems*; • **Security and privacy**;

Additional Key Words and Phrases: smart grid, transactive energy, distributed ledger, privacy, decentralized application, cyber-physical system, smart contract, blockchain

ACM Reference Format:
Scott Eisele, Taha Eghtesad, Keegan Campanelli, Prakhar Agrawal, Aron Laszka, and Abhishek Dubey. 2020. Safe and Private Forward-Trading Platform for Transactive Microgrids. *ACM Transactions on Cyber-Physical Systems*, in press, 28 pages. https://doi.org/10.1145/3403711

Authors' addresses: Scott Eisele, scott.r.eisele@vanderbilt.edu, Vanderbilt University, Nashville, TN; Taha Eghtesad, teghtesad@uh.edu, University of Houston, Houston, TX; Keegan Campanelli, keegan.m.campanelli@vanderbilt.edu, Vanderbilt University; Prakhar Agrawal, Vanderbilt University; Aron Laszka, alaszka@uh.edu, University of Houston; Abhishek Dubey, Vanderbilt University, abhishek.dubey@vanderbilt.edu.

XXXX-XXXX/2020/1-ART1 $15.00
https://doi.org/10.1145/3403711

1 INTRODUCTION

The traditional setup of the power grid is rapidly changing. Solar panel capacity is estimated to grow from 4% in 2015 to 29% in 2040 [53], and with the decreasing costs of battery technology, it is becoming increasingly feasible to support almost 99% of total load with renewable sources, by balancing out the intermittence with batteries [39]. These changes are also leading to the development of a decentralized vision for the future of power-grid operations, in which local peer-to-peer energy trading within microgrids can be used to both reduce the load on distribution system operators (DSO) and help them plan better, leading to the development of transactive energy systems (TES) [13, 35, 45, 52]. A transactive energy system is a set of market-based constructs for dynamically balancing the demand and supply across the electrical infrastructure [45]. In this approach, customers connected by transmission and distributions lines can participate in an open market, trading and exchanging energy locally. Customers participating in these markets are known as *prosumers*. There are typically three phases in the operation of this market: posting offers to buy or sell energy, matching selling offers with buying offers, and synchronized energy transfer to and from the grid.

In theory, these interactions could happen in a centralized manner by communicating all offers to a centralized market, which would match the offers and broadcast the trades back to the individual prosumers. However, in a centralized solution [33], the market presents a single point of failure. Building a decentralized market for transactive microgrids is challenging because the system must satisfy several requirements, which are often in conflict with each other.

- First, the market must be *efficient*, *i.e.*, the system should maximize the utilization of local supply—in meeting local demand—by matching the prosumers in the microgrid, taking advantage of their temporal flexibility in production and consumption. This requirement is crucial since effective trading is the purpose of the system; all other requirements are supporting this one.
- Second, the market must be *safe*, *i.e.*, the system must reject trades that would endanger the stability or physical safety of the microgrid[1].
- Third, the market must be *privacy-preserving*, *i.e.*, the system should conceal information that could be used to infer the prosumers' energy usage patterns. The amount of energy produced, consumed, bought, or sold by any prosumer should be anonymous to other prosumers and limited to the monthly bill for the DSO. This is necessary because such information could be exploited, *e.g.*, to determine when a resident is at home.
- Fourth, the market must be *secure*, *i.e.*, the system must ensure authenticity, data integrity, and auditability for offers, trades, and bills.
- Finally, the market must be *resilient*, *i.e.*, it must retain availability even if some nodes or entities (*e.g.*, DSO) are unavailable.

Addressing the requirements of privacy, safety, and efficiency simultaneously in a decentralized system is essential because removing any of these requirements significantly simplifies the problem. For example, if we do not consider safety, then privacy is easy since all offers can be made anonymously. If we do not consider privacy, then safety is easy since the safety constraints are associated with the offers and can be checked. If we do not consider efficiency, then we can simply say no trades are allowed, preserving privacy and safety. If we allow a centralized market, then the centralized market can keep the offers confidential and can check the safety constraints. However, such a system has a single point of failure. Assurance that the system is secure and resilient are

[1] Note that this is orthogonal to the *physical enforcement of safety* which is provided by overcurrent protection units that limit the total current flowing through the microgrid.

crucial in practice, *e.g.*, because communities are facing increasing cyber threats as well as natural disasters that disrupt infrastructure.

The research community is increasingly advocating the use of distributed ledgers in the energy sector [1]. This is primarily because a distributed ledger can provide an immutable, complete, and fully auditable record of all transactions that have occurred within a system. However, there are still research gaps. For example, Andoni et al. [1] surveyed 140 applications of distributed ledgers and found that 33% were focused on decentralized energy trading, with privacy preservation being one of the key challenges that has not been addressed. They also highlighted balancing supply to demand (stability) as another critical issue. The work presented in this paper addresses the problem of privacy while ensuring that safety constraints can be enforced for trades.

Contributions: In this paper, we introduce TRANSAX[2], a blockchain-based decentralized transactive energy system that provides privacy while preserving safety without using a centralized market. The underlying communication substrate is implemented by using a distributed middleware, called Resilient Information Architecture Platform for Smart Grid (RIAPS) [18, 29, 60], which provides resilience against failure of individual services. The specific contributions of this paper are as follows.

(1) To enable safe and anonymous trading, we introduce production and consumption assets that allow us to dissociate safety from privacy (Section 4.3). To provide privacy, we integrate a decentralized mixing protocol [57] that enables prosumers to anonymize their production and consumption assets within their groups, and hence trade anonymously. This also enables privacy-preserving billing such that no information is disclosed to the DSO other than the billed amount.
(2) We show in Section 6.2 that prosumers can be anonymous within groups (Section 4.2) while preserving system safety; however, there is potential for some loss of trading efficiency. We provide analysis of much efficiency is lost and under what conditions.
(3) To improve trading efficiency, we provide prosumers with the ability to specify production and consumption offers with temporal flexibility (Section 4.4). We solve the trading problem as a linear program, maximizing the energy traded over a long time horizon[3], and introduce a hybrid architecture for solving this linear program.
(4) We show in Section 6.1.2 that the hybrid architecture can combine the resilience of distributed ledgers with the computational efficiency of conventional compute platforms when solving the energy allocation problem. This hybrid architecture ensures the integrity of data and computational results— if the majority of the ledger nodes are secure—while allowing the complex computation to be performed by a set of redundant and efficient solvers.
(5) In Section 7, we provide a testbed and experimental analysis of our proof-of-concept implementation of TRANSAX. We show that our approach is feasible for private blockchains in the grid-connected microgrid setting.

Outline: We introduce the background concepts in Section 2. TRANSAX components are discussed in Section 3. We present the energy trading approach, including extensions to support safety and privacy in Section 4. Then, we describe the protocol that implements the energy trading approach in Section 5. We analyze how we meet key requirements and discuss the tradeoff between privacy and efficiency in Section 6. We present an integrated testbed using GridLAB-D [9] and numerical results in Section 7. Finally, we present related research in Section 8 followed by conclusions in Section 9.

[2] This paper is a significant extension of our previously published conference papers [19, 36, 37]. [3] In Section 6.1.3, we show that temporal flexibility can improve trading efficiency.

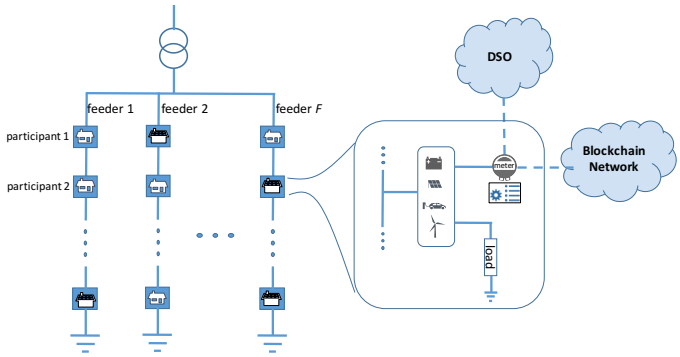

Fig. 1. Illustration of a multi-feeder system. Each feeder is protected by an overcurrent relay at the junction of the common bus. The inset figure shows that a node in the network has different kinds of loads, some of which can be scheduled, making it possible for a consumer to bid in advance for those loads. The smart meter ensures proper billing per node. The blockchain network is an immutable record of all transactions and is used for scheduling energy transfers between homes in the microgrid and between homes and the DSO.

2 BACKGROUND

To explain the concepts of TRANSAX, we first need to provide an overview of basic concepts and assumptions and how we use them in TRANSAX.

Assumption on the Microgrid Architecture. We consider a microgrid with a set of feeders. A feeder has a fixed set of nodes, each representing a residential load or a combination of load and distributed energy resources, such as rooftop solar and batteries, as shown in Fig. 1. Each node is associated with a participant in the local peer-to-peer energy trading market, and each participant is independent and has control over its energy utilization. The participants may be able to predict their future production and consumption based on historical data and anticipated utilization.

Resilient Information Architecture for Smart Grid. RIAPS is an *open application platform* for smart grids that distributes intelligence and control capability to local endpoints to reduce total network traffic, avoid latency, and decrease dependency on multiple devices and communication interfaces, thereby enhancing reliability. RIAPS also provides platform services to power-system applications running on remote nodes [63], including: (1) resource-management framework to control use of computational resources, (2) fault-management framework to detect and mitigate faults in all layers of the system, (3) security framework to protect confidentiality, integrity, and availability of a system under cyber-attacks, (4) fault-tolerant time-synchronization service, (5) discovery framework to establish the network of interacting actors for an application, and (6) deployment and management framework for administration of the distributed applications from a control room. In TRANSAX, RIAPS is used as the base middleware and application-management substrate, allowing all actors to communicate and the protocol—discussed later in this paper—to be implemented. As a note, actor interactions are supplemented with interactions with the distributed ledger and the smart contract. We will discuss these interactions in detail in the protocol section. For more details on RIAPS, we refer the interested reader to [20].

Distributed Ledgers. Distributed ledgers refer to distributed databases where nodes in a network simultaneously reconcile their copies of the data and sequence of actions through Byzantine

consensus to achieve a shared truth, so that data in the shared ledger can be verified and is tamper-aware. Data is added to the ledger via transactions and all transactions submitted to the ledger are associated with an account, which is typically not anonymous. The immutability of actions is crucial for providing safety and security, *i.e.*, after a transaction has been recorded, it cannot be modified or removed from the ledger. The ledger is distributed to enhance fault tolerance. Since a distributed ledger is maintained by multiple nodes, nodes must reach a consensus on which transactions are valid and stored on the ledger. This consensus must be reached both quickly and reliably, even in the presence of erroneous or malicious (*e.g.*, compromised) ledger nodes. We make no assumptions about the particulars of the consensus algorithm. In practice, a distributed ledger can be implemented using, *e.g.*, *blockchains* with proof-of-stake consensus or a practical Byzantine fault tolerance algorithm [8]. In TRANSAX, we use Ethereum [15] as the ledger.

Mixing. The use of blockchains in building a transactive energy platform is appealing also because they elegantly integrate the ability to immutably record the ownership and transfer of assets, with essential distributed computing services such as Byzantine fault-tolerant consensus on the ledger state as well as event chronology. The ability to establish consensus on state and timing is important in the context of TES since these systems are envisioned to involve the participation of self-interested parties, interacting with one another via a distributed computing platform that executes transaction management. However, this also leads to the problem of privacy as the records in blockchain can be attributed to the prosumers. The earliest approach to solve the privacy problem in blockchains was mixing. The key concept in mixing is to hide the linkage between the inputs and outputs of a transaction by combining them with other transactions. In TRANSAX, we use CoinShuffle [57]. A simplified example of how this works is that each participating prosumer provides an anonymous output account and shuffles them with the others, so that only the owner knows who owns a specific account. Then, if each prosumer inputs the same amount, all the transaction must do is transfer that amount from each of the public accounts to each of the anonymous accounts. Since the anonymous accounts were shuffled, no anonymous account can be linked conclusively to its owner. Note that this does not hide the quantity of assets stored in each anonymous account.

3 TRANSAX COMPONENTS

In this section, we introduce the components of TRANSAX (see Fig. 2).

Distribution System Operator. We assume the existence of a distribution system operator (DSO) that participates in the market and may use the market to incentivize timed energy production within the microgrid to aid in grid stabilization and promotion of related ancillary services [14] through updates to the price policy. The participants settle trades in advance using automated matching which allows them to schedule their transfer of energy into the local distribution system. The DSO meets the participants' residual demand and supply, *i.e.*, consumption and production that they did not trade in advance due to estimation error or lack of trade partners[4].

DSO is also responsible for handling financial operations, such as sending monthly bills, and functional operations, such as registering new smart meters[5]. Registration means adding a new smart meter to the platform when it is installed for a prosumer. Thus, DSO provides *safety* and *security* by limiting access to the distributed ledger to prosumers that have registered.

[4] Note that this requires the presence of a secondary controller that balances voltage and frequency in the microgrid as described in our prior work in [16]. [5] In practice, these smart meters must be tamper-resistant to prevent prosumers from "stealing electricity" by tampering with their meters. After a smart meter has measured the net amount of energy consumed by the prosumer in some time interval, it can send this information to the DSO for billing purposes. This way, the DSO has no fine-grained information on the energy profile of the consumer, the DSO only knows the amount that needs to be paid for the energy consumed during that cycle.

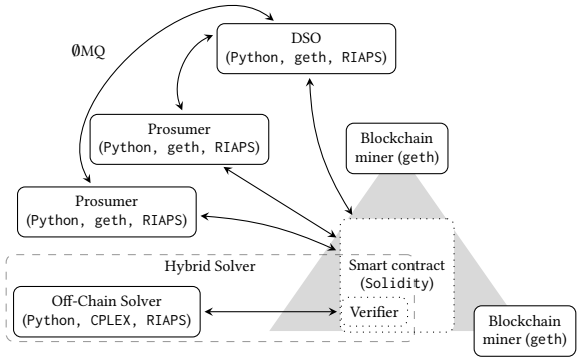

Fig. 2. Components of the energy trading system. In our reference implementation, we use Ethereum as the decentralized computation platform for smart contracts, and the other components interact with the blockchain network using the geth Ethereum client. The smart contract is implemented in Solidity, a high-level language for Ethereum, and it is executed by a private network of geth mining nodes. The off-chain solver uses CPLEX.

Producers and Consumers. The participants in the market are the producers and consumers of energy, collectively referred to as prosumers. The prosumers are built upon the RIAPS middleware which provides communication services via the ZeroMQ messaging library [6]. RIAPS additionally provides time synchronization between the prosumers allowing them synchronize production and consumption, ensuring that they remain balanced. The prosumers can construct offers and trade with other prosumers. The trades are submitted to a *Smart Contract* (described below) via a geth client. Geth [7] is the go implementation of the Ethereum protocol and is used to instantiate and interact with an Ethereum blockchain. The prosumers use a light client that interacts with the full clients that constitute the blockchain. Each prosumer has a smart meter that measures the prosumer's energy production and consumption. The smart meter aggregates this data and provides it to the DSO periodically, which is necessary element of privacy-preserving billing. Prosumers work together to achieve privacy via execution of mixing protocol.

Distributed Ledger. The blockchain in Fig. 2 provides the basis for the market functionality of TRANSAX. It provides immutable storage service for offers, solutions, safety constraints, and a notification log of events. Prosumers and solvers check for these events and perform actions based on them. Nodes in the network host full Ethereum clients to provide this substrate.

Hybrid Solver. The hybrid solver in Fig. 2 is an innovation of TRANSAX, which enables a distributed market architecture that has the auditability and resilience of blockchains through a smart contract, and can yet use high-performance computers to solve the computationally expensive optimization problem *off-blockchain*. The hybrid solver consists of smart contract elements as well as an off-chain solver.

- *Smart contract* — The smart contract in Fig. 2 provides the essential functionality of the market, such as financial transactions and enabling offers to be submitted and then matched into trades that satisfy the safety constraints of the system. The matching of the offers is a complex optimization problem, and since smart contracts are limited in the number of computations they can perform, we do not use the smart contract to actually match offers

[6] https://zeromq.org/ [7] https://geth.ethereum.org/downloads/

into trades. Rather, we only use it for validation of energy trading solutions provided by the *off-chain solver*.

- *Off-Chain Solver* — The off-chain solver in Fig. 2 consists of a set of solvers. Any participant of the system can act in the solver role since all offers posted on the blockchain are public. Prosumers are incentivized to act as solvers—especially if no dedicated solvers are available—since they can create trades that benefit them. Note that this is safe because the smart contract verifies each solution and accepts a new solution only if it is feasible and strictly better than the current solution. Each solver can use whatever strategy it chooses for solving, because the solutions will still be verified. In this work, we implement an efficient linear programming solver using CPLEX [28], which can be run off-blockchain, on any capable computer (or multiple computers for increased reliability). The solver is run periodically to find a solution to the energy trading problem based on the latest set of offers posted. Once a solution is found by the matching solver, it is submitted to the smart contract in a blockchain transaction, which is validated by the smart contract. Note that if new offers have been posted since the solver started working on its solution, the solution computed by the solver will still be considered valid by the smart contract because any solution that is valid for a set of offers is also valid for a superset of those offers. Since solvers may fail, the smart contract should accept solutions from multiple off-blockchain solvers to preserve the reliability provided by the blockchain. However, these solvers might provide different solutions. Thus, the smart contract must be able to choose from multiple solutions (some of which may come from compromised nodes).

4 ENERGY TRADING APPROACH

The distribution network infrastructure is a collection of feeders (Fig. 1). A feeder has a fixed set of nodes, each representing a prosumer, which is a combination of load and distributed energy resources, such as rooftop solar panels and batteries. We assume that the prosumers can estimate their future production and consumption based on historical data and anticipated utilization. The prosumers submit energy offers based on their estimates via automated agents that act on behalf of residents (*i.e.*, residents do not need to trade manually). The estimates do not need to be perfect because we assume the existence of a distribution system operator (DSO), which also participates in the market and can supply residual demand not met through the local market. The DSO may use the market to incentivize timed energy production within the microgrid to aid in grid stabilization and in the promotion of related ancillary services [14] through updates to the price policy. Since the trades record only the energy futures and do not control the actual exchange of energy, we include a smart meter at each prosumer to measure the prosumer's actual energy production and consumption. In practice, these smart meters must be tamper-resistant to prevent prosumers from "stealing electricity." After a smart meter has measured the net amount of energy consumed by the prosumer in some time interval, it can send the relevant information to the DSO for billing purposes to keeping the actual consumption private.

Our goal is to find an optimal match between energy production and consumption offers, which we refer as the *energy trading problem*. Each offer is associated with an identity that belongs to the prosumer that posted the offer. We refer to these identities as *accounts*, and prosumers may generate any number of them.

4.1 The Basic Problem Specification

Let $\mathcal{F}$ denote the set of feeders. On each feeder, there is a set of prosumers, who can make offers to buy and sell energy. We assume that time is divided into intervals of fixed length Δ, and we refer to the t-th interval simply as time interval t. For a list of symbols used in the paper, see Table 1.

Table 1. List of Symbols

Symbol	Description
	Microgrid
$\mathcal{F}, \mathcal{U}$	set of feeders and prosumers, resp.
C^e_f, C^i_f	maximum net (*external*) and total (*internal*) load *constraints*, resp., on feeder $f \in \mathcal{F}$
L^+_u, L^-_u	*production* (+) and *consumption* (-) *limits*, resp., of prosumer u
C^e_g, C^i_g	maximum net (*external*) and total (*internal*) load *constraints*, resp., on group $g \in \mathcal{G}$
EPA, ECA	asset granting permission to produce or consume, resp., a unit of energy
Δ	length of each time interval
T_{clear}	minimum number of time intervals between the finalization and notification of a trade
E^t_u	energy transferred by prosumer u in interval t
t_f	next interval to be finalized $t + 1 + T_{clear}$
	Offers
$\mathcal{S}_f, \mathcal{B}_f$	set of selling and buying offers, resp., from feeder $f \in \mathcal{F}$
$\mathcal{S}, \mathcal{B}$	set of all selling and buying offers, resp.
$\mathcal{S}^{(t)}, \mathcal{B}^{(t)}$	set of all selling and buying offers, resp., submitted by the end of time interval t
A_s, A_b	account that posted offers $s \in \mathcal{S}$ and $b \in \mathcal{B}$, resp.
E_s, E_b	amount of energy to be sold or bought, resp., by offers $s \in \mathcal{S}$ and $b \in \mathcal{B}$
I_s, I_b	time intervals in which energy could be provided or consumed by offers $s \in \mathcal{S}$ and $b \in \mathcal{B}$, resp.
R_s, R_b	reservation prices of offers $s \in \mathcal{S}$ and $b \in \mathcal{B}$, resp.
$\mathcal{M}(s), \mathcal{M}(b)$	set of offers that are matchable with offers $s \in \mathcal{S}$ and $b \in \mathcal{B}$, resp.
	Solution
$\varepsilon_{s,b,t}$	amount of energy that should be provided by s to b in interval t
$\pi_{s,b,t}$	unit price for the energy provided by s to b in interval t
$Feasible(\mathcal{S}, \mathcal{B})$	set of feasible solutions given sets of selling and buying offers $\mathcal{S}$ and $\mathcal{B}$
$\hat{\varepsilon}_{s,b,t}, \hat{\pi}_{s,b,t}$	finalized trade values
	Implementation Parameters
T_h	solve horizon; the number of intervals beyond the most recently finalized interval that are considered by the solver (offers beyond horizon $t_f + T_h$ are not considered by solver)
$\hat{\Delta}$	length of the time step used for simulating the real interval of length Δ

For feeder $f \in \mathcal{F}$, we let $\mathcal{S}_f$ and $\mathcal{B}_f$ denote the set of selling and buying offers posted by prosumers in feeder f, respectively.[8] A selling offer $s \in \mathcal{S}_f$ is a tuple (A_s, E_s, I_s, R_s), where A_s is the account that posted the offer, E_s is the amount of energy to be sold, I_s is the set of time intervals in which the energy could be provided, R_s is the reservation price, *i.e.*, lowest unit price for which the prosumer is willing to sell energy. Similarly, a buying offer $b \in \mathcal{B}_f$ is a tuple (A_b, E_b, I_b, R_b), where the values pertain to consuming/buying energy instead of producing/selling, and R_b is the highest price that the prosumer is willing to pay. For convenience, we also let $\mathcal{S}$ and $\mathcal{B}$ denote the set of all buying and selling offers (*i.e.*, we let $\mathcal{S} = \cup_{f \in \mathcal{F}} \mathcal{S}_f$ and $\mathcal{B} = \cup_{f \in \mathcal{F}} \mathcal{B}_f$).

We say that a pair of selling and buying offers $s \in \mathcal{S}$ and $b \in \mathcal{B}$ is *matchable* if

$$R_s \leq R_b \text{ and } I_s \cap I_b \neq \emptyset. \tag{1}$$

In other words, a pair of offers is matchable if there exists a price that both prosumers would accept and a time interval in which the seller and buyer could provide and consume energy. For a given selling offer $s \in \mathcal{S}$, we let the set of buying offers that are matchable with s be denoted by $\mathcal{M}(s)$. Similarly, we let the set of selling offers that are matchable with a buying offer b be denoted by $\mathcal{M}(b)$.

A solution to the energy trading problem is a pair of vectors $(\boldsymbol{\varepsilon}, \boldsymbol{\pi})$, where $\varepsilon_{s,b,t}$ is a non-negative amount of energy that should be provided by offer $s \in \mathcal{S}$ and consumed by offer $b \in \mathcal{M}(s)$ in time

[8] To include the DSO in the formulation, we assign it to a "dummy" feeder.

interval $t \in I_s \cap I_b$ [9]; and $\pi_{s,b,t}$ is the unit price for the energy provided by offer $s \in \mathcal{S}$ to offer $b \in \mathcal{M}(s)$ in time interval $t \in I_s \cap I_b$.

A pair of vectors $(\boldsymbol{\varepsilon}, \boldsymbol{\pi})$ is a feasible solution to the energy trading problem if it satisfies the following two constraints. First, the amount of energy sold or bought from each offer is at most the amount of energy offered:

$$\forall s \in \mathcal{S} : \sum_{b \in \mathcal{M}(s)} \sum_{t \in I(s,b)} \varepsilon_{s,b,t} \leq E_s \quad \text{and} \quad \forall b \in \mathcal{B} : \sum_{s \in \mathcal{M}(b)} \sum_{t \in I(s,b)} \varepsilon_{s,b,t} \leq E_b \tag{2}$$

Second, the unit prices are between the reservation prices of the seller and buyer:

$$\forall s \in \mathcal{S}, b \in \mathcal{M}(s), t \in I(s,b) : R_s \leq \pi_{s,b,t} \leq R_b \tag{3}$$

The objective of the energy trading problem is to maximize the amount of energy traded. The rationale behind this objective is maximizing the load reduction on the bulk power grid. Formally, an optimal solution to the energy trading problem is

$$\max_{(\boldsymbol{\varepsilon},\boldsymbol{\pi}) \in \mathit{Feasible}(\mathcal{S},\mathcal{B})} \sum_{s \in \mathcal{S}} \sum_{b \in \mathcal{M}(s)} \sum_{t \in I(s,b)} \varepsilon_{s,b,t} \tag{4}$$

where $\mathit{Feasible}(\mathcal{S}, \mathcal{B})$ is the set of feasible solutions given selling and buying offers $\mathcal{S}$ and $\mathcal{B}$ (*i.e.*, set of solutions satisfying Equations (2) and (3) with $\mathcal{S}$ and $\mathcal{B}$).

The above formulation ensures feasibility, which takes the reservation prices into account. However, we do not address how to set the clearing prices in this paper. Clearing prices could be set using an existing approach, *e.g.*, double auction; however, this is part of our future work (Section 9).

4.2 Adding Safety Extensions to Problem Specification

To ensure the safety of the microgrid, we introduce additional constraints on the solution to the energy trading problem. Each prosumer u has independent production and consumption limits, which are denoted by L_u^+ and L_u^-, respectively. Further, each feeder $f \in \mathcal{F}$, has a transformer for incoming energy, which has a capacity rating. We let C_f^e denote the capacity of the transformer of feeder f. Similarly, the distribution lines and transformers within the feeder have capacity ratings as well. We let C_f^i denote the maximum amount of energy that is allowed to be consumed or produced within the feeder during an interval[10]. These constraints are physically enforced by the over-current relays of the circuit breakers and feeders.

Now we generalize and introduce the notion of groups. We note that groups can correspond to feeders and support the constraints that we introduced in the previous paragraphs. They allow us to support physical layouts other than strictly feeders, and it will be useful for privacy later. We define a *group* g to be a set of feeders (*i.e.*, $g \subseteq \mathcal{F}$). We let $\mathcal{G}$ be the set of all groups, and for each group $g \in \mathcal{G}$, we introduce group safety limits C_g^i and C_g^e, which are analogous to feeder limits. A solution is safe if it satisfies the following three constraints. First, the amount of energy transferred out of or into a prosumer is within the production and consumption limits in all time intervals:

$$\forall u \in \mathcal{U}, t : \sum_{s \in \mathcal{S}_u} \sum_{b \in \mathcal{B}} \varepsilon_{s,b,t} \leq L_u^+ \quad \text{and} \quad \forall u \in \mathcal{U}, t : \sum_{b \in \mathcal{B}_u} \sum_{s \in \mathcal{S}} \varepsilon_{s,b,t} \leq L_u^- \tag{5}$$

where $\mathcal{S}_u$ and $\mathcal{B}_u$ are the sets buying and selling offers posted by accounts owned by prosumer u.

[9] We require the both seller and buyer to produce a constant level of power during the time interval. This can be achieved by smart inverters. [10] In other words, limit C_f^e is imposed on the net production and net consumption of all prosumers in feeder f, while limit C_f^i is imposed on the total production and consumption of prosumers in feeder f.

Second, the amount of energy consumed and produced within each group is below the safety limit in all time intervals:

$$\forall g \in \mathcal{G}, t: \underbrace{\max\left\{\sum_{b\in\mathcal{B}_g}\sum_{s\in\mathcal{S}}\varepsilon_{s,b,t}, \sum_{s\in\mathcal{S}_g}\sum_{b\in\mathcal{B}}\varepsilon_{s,b,t}\right\}}_{\textit{max} \text{ of energy bought or sold } (X)} \leq C^i_g \tag{6}$$

This means that the sum of all the buying trades nor the sum of selling trades can exceed the safety limit.

Third, the amount of energy flowing into or out of each group is within the safety limit in all time intervals:

$$\forall g \in \mathcal{G}, t: -C^e_g \leq \underbrace{\left(\sum_{s\in\mathcal{S}_g}\sum_{b\in\mathcal{B}}\varepsilon_{s,b,t}\right) - \left(\sum_{b\in\mathcal{B}_g}\sum_{s\in\mathcal{S}}\varepsilon_{s,b,t}\right)}_{\textit{net} \text{ energy transfer } (N)} \leq C^e_g \tag{7}$$

Note that the maximum of bought or sold energy (X) in Eq. (6) is always greater than the net energy transferred (N) in Eq. (7), *i.e.*, $N < X$. This is important because it means that we need to consider only $C^e_g \leq C^i_g$. If we considered $C^i_g < C^e_g$, then $N < X \leq C^i_g < C^e_g$, which means that the internal limit will always trip ($X > C^i_g$) before the external limit, making the external limit irrelevant. This observation will be important in Section 6.2

4.3 Adding Privacy Extensions to Problem with Safety Specifications

To protect prosumers' privacy, we let them use anonymous accounts when posting offers. By generating new anonymous accounts, a prosumer can prevent others from linking the anonymous accounts to its actual identity, thereby keeping its trading activities private. However, anonymous accounts pose a threat to safety. Since the energy trading formalization with safety extension (see Equations (5) - (7)) discussed earlier requires the offers to be associated with the prosumer to enforce prosumer-level constraints and with the group from which they originated in order to be able to enforce group-level safety constraints. Without these associations, prosumers can generate any number of anonymous accounts. They can then post selling and buying offers for large amounts of energy without any intention of delivering and without facing any repercussions. A malicious or faulty prosumer could easily destabilize the grid with this form of reckless trading. Consequently, the amount of energy that may be traded by anonymous accounts belonging to a prosumer must be limited.

To enforce the prosumer-level constraints we introduce the concept of energy production and consumption assets, which allows us to disassociate the limiting of assets from the anonymity of offers. First, an *energy production asset* (EPA) is tuple $(E_{EPA}, I_{EPA}, G_{EPA})$, where

- E_{EPA} is the permission to sell a specific non-negative amount of energy to be produced,
- I_{EPA} is the set of intervals for which the asset is valid, and
- G_{EPA} is the group that the asset is associated with.

Second an *energy consumption asset* (ECA) represents a permission to buy a specific amount of energy and is defined by the same fields. For this asset, however, the fields define energy consumption instead of production. Each prosumer u is only permitted to withdraw assets up to the limits L^+_u and L^-_u into a non-anonymous account.

These assets can be moved to anonymous accounts in an untraceable way such as through an *anonymizing mixer*. The mixer ensures that accounts cannot be linked to the prosumer that owns

them. However, the anonymous accounts must retain their group association and the sum of the assets remains constant. Production assets are required to post a selling offer, and consumption assets are required to post a buying offer. For the offer to be valid, the account posting the offer must have assets that cover the amount and intervals offered. When a trade is finalized the assets are exchanged. We will provide more details on how they fit into the trading approach in Section 5.

To enforce group-level safety we only provide group-level anonymity, meaning that an offer can be traced back to its group of origin, but not to the individual prosumer within the group. When forming a group, the safety constraints need to be set appropriately. We will discuss how they should be set and the associated energy trading capacity costs in Section 6.2.

4.4 Introducing the Notion of Clearance Windows

In our basic problem formulation, we assumed that all buying and selling offers $\mathcal{B}$ and $\mathcal{S}$ are available at once, and we cleared the market in one take. In practice, however, the market conditions and the physical state of the DSO and prosumers may change over time, making it advantageous to submit new offers[11]. As new offers are posted we need to recompute the solution. While new offers can increase the amount of energy traded, the *trade values* $\varepsilon_{s,b,t}$ and $\pi_{s,b,t}$ need to be *finalized* at some point in time. At the very latest, values for interval t need to be finalized by the end of interval $t-1$; otherwise, participants would have no chance of actually delivering the trade.

Here, we extend the energy trading problem to accommodate a time-varying offer set (where offers can be unmatched, matched and pending, or matched and finalized), and a time constraint for finalizing trades. Our approach finalizes only trades that need to be finalized, which maximizes efficiency while providing safety. We assume that all trades for time interval t' (*i.e.*, all values $p_{s,b,t'}$ and $\pi_{s,b,t'}$) must be finalized and the trading prosumers must be notified by the end of time interval $t' - T_{clear} - 1$ (see Fig. 3), where T_{clear} is a positive integer constant that is set by the DSO. In other words, if the current interval is t, then all intervals up to $t + T_{clear}$ have already been finalized. Preventing "last-minute" changes can be crucial for safety and fairness since it allows both the DSO and the prosumers to prepare for delivering (or consuming) the right amount of energy. In practice, the value of T_{clear} must be chosen accounting for both physical constraints (*e.g.*, how long it takes to turn on a generator) and communication delay (*e.g.*, some participants might learn of a trade with delay due to network disruptions).

We let $\hat{\varepsilon}_{s,b,t}$ and $\hat{\pi}_{s,b,t}$ denote the finalized trade values, and we let $\mathcal{B}^{(t)}$ and $\mathcal{S}^{(t)}$ denote the set of buying and selling offers that participants have submitted by the end of time interval t. Then, the system takes the following steps at the end of each time interval t. First, find an optimal solution $(\boldsymbol{\varepsilon}^*, \boldsymbol{\pi}^*)$ to the extended energy trading problem:

$$\max_{(\boldsymbol{\varepsilon},\boldsymbol{\pi}) \in \mathit{Feasible}(\mathcal{S}^{(t)}, \mathcal{B}^{(t)})} \sum_{s \in \mathcal{S}^{(t)}} \sum_{b \in \mathcal{M}(s)} \sum_{\tau \in I(s,b)} \varepsilon_{s,b,\tau} \tag{8}$$

subject to

$$\forall\, \tau \le t_f : \varepsilon_{s,b,\tau} = \hat{\varepsilon}_{s,b,\tau} \tag{9}$$

$$\pi_{s,b,\tau} = \hat{\pi}_{s,b,\tau} \tag{10}$$

Second, finalize trade values for time interval t_f based on the optimal solution $(\boldsymbol{\varepsilon}^*, \boldsymbol{\pi}^*)$:

$$\hat{\varepsilon}_{s,b,t_f} := \varepsilon^*{}_{s,b,t_f} \tag{11}$$

$$\hat{\pi}_{s,b,t_f} := \pi^*_{s,b,t_f} \tag{12}$$

[11] Updating or cancelling offers could also be useful; however, we do not provide this functionality in the current version and leave it for future work.

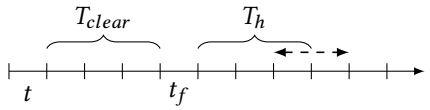

Fig. 3. Temporal parameters (t is current interval, t_f is the interval to be finalized).

By taking the above steps at the end of each time interval, trades are always cleared based on as much information as possible (*i.e.*, considering as many offers as possible)[12] without violating any safety or timing constraints. Note that here $\mathit{Feasible}(\mathcal{S}, \mathcal{B})$ now also includes the safety constraints (5), (6), and (7).

4.5 Practical Considerations for Solving the Problem

To find the optimal solution efficiently, we frame the energy trading problem as a linear program. First, we create real-valued variables $\varepsilon_{s,b,t}$ and $\pi_{s,b,t}$ for each $s \in S, b \in \mathcal{M}(s), t \in I_s \cap I_b$. Then, the following reformulation of the matching problem is a linear program:

$$\max_{\boldsymbol{\varepsilon},\boldsymbol{\pi}} \sum_{s\in\mathcal{S}} \sum_{b\in\mathcal{M}(s)} \sum_{t\in I(s,b)} \varepsilon_{s,b,t} \tag{13}$$

subject to the constraint Equations, which can all be expressed as linear inequalities (2), (3), (5), (6), (7), and

$$\boldsymbol{\varepsilon} \geq \mathbf{0} \text{ and } \boldsymbol{\pi} \geq \mathbf{0}. \tag{14}$$

However, we must consider that even though Equation (4) can be formulated as a linear program and be solved efficiently (*i.e.*, in polynomial time), the number of variables $\{\varepsilon_{s,b,t}\}$ may grow prohibitively high as the number of offers and time intervals that they span increases. In practice, this may pose a significant challenge for solving the energy trading problem for larger transactive microgrids. A key observation that helps us tackle this challenge is that even though prosumers may post offers whose latest intervals are far in the future, the optimal solution for the finalized interval typically depends on only a few intervals ahead of the finalization deadline. Indeed, we have observed that considering intervals in the far future has little effect on the optimal solution for the interval that is to be finalized next (see Fig. 8).

Consequently, for practical solvers, we introduce a planning horizon T_h (see Fig. 3) that limits the intervals that need to be considered for a solution: for any $\hat{t} > t_f + T_h$, we set $\varepsilon_{s,b,\hat{t}} = 0$, where t_f is the earliest interval that has not been finalized. By "pruning" the set of free variables, we can significantly improve the performance of the solver with negligible effect on solution quality (see Fig. 8). This results in the following "pruned" objective function:

$$\max_{(\boldsymbol{\varepsilon},\boldsymbol{\pi}) \in \mathit{Feasible}(\mathcal{S},\mathcal{B})} \sum_{s\in\mathcal{S}} \sum_{b\in\mathcal{M}(s)} \sum_{\tau\in I_s\cap I_b\cap\{\tau;\tau\leq t_f+T_h\}} \varepsilon_{s,b,\tau} \tag{15}$$

Although solving linear programs is not computationally hard, it can be challenging with many variables and constraints in resource-constrained computing environments. Since computation is relatively expensive on blockchain-based distributed platforms[13], solving even the "pruned" energy trading problem from Equation (15) might be infeasible using a blockchain-based smart contract. Considering this, we choose to use our hybrid-solver approach since compared to finding optimal

[12] This includes offers for intervals beyond the finalization interval. Effectively, matches for an interval beyond finalization can be changed if a better solution is found; however, finalized matches are permanent and never changed. [13] Further, Solidity, the preferred high-level language for Ethereum, currently lacks built-in support for certain features that would facilitate the implementation of a linear programming solver, such as floating-point arithmetic [65].

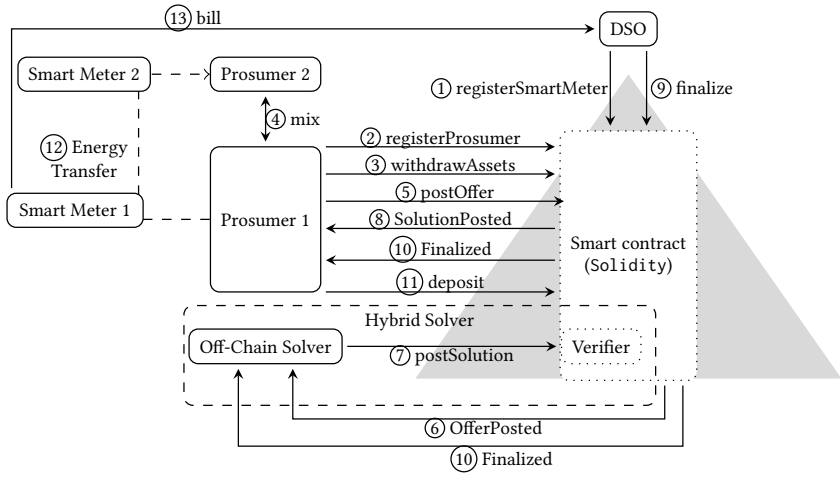

Fig. 4. Example workflow of TRANSAX. Nodes represent entities in the platform, and edges represent interactions, such as smart-contract function calls. In this example, prosumer 1 is selling energy to prosumer 2 and the dashed line represents the energy transfer.

trades, verifying the feasibility of a solution $(\boldsymbol{\varepsilon}, \boldsymbol{\pi})$ and computing the value of the objective function is computationally inexpensive and can easily be performed on a blockchain-based decentralized platform. Thus, the smart contract provides the following functionality:

- Solutions may be submitted to the smart contract at any time. The contract verifies the feasibility of each submitted solution, and if the solution is feasible, then the contract computes the value of the objective function. The contract always keeps track of the best feasible solution submitted so far, which we call the *candidate solution*.
- At the end of each time interval t, the contract finalizes the trade values for interval $t_f = t + T_{clear} + 1$ based on the candidate solution.[14]

5 TRANSAX PROTOCOL

We implement the practical solution approach described in the previous section as a protocol of interaction between the TRANSAX components (Section 3). The protocol is depicted in Fig. 4, and the activities are described below.

5.1 Registration

When a new customer is added to the grid, a smart meter is installed. The DSO registers the smart meter by calling ①[15] *registerSmartMeter* on the TRANSAX smart contract. This call sets the asset allocation limits for that customer and records which feeder it is located on in the grid. The customer then registers as a prosumer with TRANSAX by calling ② *registerProsumer*.

The registration information requires each prosumer to specify a smart meter, and to provide a DSO certified public address that corresponds to the specified smart meter for the DSO to use when allocating assets. Since the smart meter is associated with a specific feeder, the smart contract adds the prosumer to the group associated with that feeder. This is required to ensure that feeder-level safety constraints can be correctly applied. The registrations can happen asynchronously, allowing

[14] If no solution has been submitted to the contract so far, which might be the case right after the trading system has been launched, $\boldsymbol{\varepsilon} = \mathbf{0}$ may be used as a candidate solution. [15] The circled numbers correspond to the numbered edges in Fig. 4

new prosumers to join at any time, even long after trading has commenced. The registration process occurs only once for each smart meter and prosumer. Once registered, a prosumer may participate in the following trading protocol repeatedly.

5.2 Mixing

Once a prosumer has registered, it can withdraw assets into its public address (*i.e.*, the account registered at DSO) for future intervals by calling ③ *withdrawAssets*. After withdrawing assets, a prosumer could make offers using *postOffer*. However, if it made offers using its public account, then the trades could be traced back to the prosumer as all transactions in the distributed ledger are recorded publicly, thereby violating the privacy requirements. Instead, the prosumer creates an anonymous address, which is not registered with the DSO, and transfers the assets from its public address to the anonymous addresses via ④ mixing assets with other prosumers. Mixing can be done in assigned groups by executing a decentralized mixing protocol, such as CoinShuffle [57]. The goal of the mixing protocol is to transfer funds or assets from a set of accounts to a set of anonymous accounts without directly linking any of the accounts to each other. Due this mixing, even if an entity knows which prosumers participated in a mixing protocol (*i.e.*, based on their registered, public accounts) and what target anonymous accounts were used in the mixing, it cannot link any anonymous account to the prosumer who owns the account.

5.3 Trading

5.3.1 Posting Offers. Next, the prosumers can construct and post anonymous offers using their anonymous accounts by calling function ⑤ *postOffer*. The smart contract checks that the anonymous account used to post the offer has assets that cover the amount and intervals specified in the offer. If not, then the offer is rejected. If the offer is accepted, the smart contract emits event ⑥ *OfferPosted*, notifying the off-chain matching solvers.

5.3.2 Matching Offers. The matching solvers may wait for many prosumers to post many offers, but eventually, it pairs buying and selling offers and posts the solutions by calling function ⑦ *postSolution*. The smart contract checks the solution to make sure that it is feasible according to the feasibility requirements described in Section 4, including checking that the trades do not exceed the group capacity constraint. If the solution is valid, then smart contract saves it and emits event ⑧ *SolutionPosted*, notifying the prosumers of the current candidate solution. Additional solutions may be submitted by any solver, and if those solutions are valid and superior (*i.e.*, they trade more energy), then the smart contract will update the candidate solution. Offers can continue to be posted until the end of the trading interval when trades will be finalized.

5.4 Energy Transfer and Billing

As an interval comes to a close, the DSO calls[16] function ⑨ *finalize* which means that offers for interval t_f are no longer accepted and the smart contract transfers funds from the consuming offer's account to the producing offer's account. It also exchanges the *EPA* assets of the seller for the *ECA* assets of the buyer and vice versa for each of the matched offers. The call also emits the ⑩ *Finalized* event, notifying the solvers to update their solving interval, and the prosumers that the trades for interval t_f have been finalized. If a prosumer posts offers with many anonymous accounts, it will have to aggregate all the corresponding trades to determine how much energy it is expected to produce/consume during that interval when it arrives. Once the prosumers are notified

[16] Note that by default the DSO calls the *finalize* function to increment the current interval, but since this function is time guarded, any other entity can call it, which provides additional resilience.

of the trades, they call function (11) *deposit* to transfer all assets for the finalized interval from the prosumers anonymous accounts to an anonymous account owned by their smart meter.

The smart meter checks that the total amount of assets deposited matches the amount withdrawn for the finalized interval. This ensures that there are no trades that have not been accounted for. The smart meter also compares the total of all production assets that were deposited against the production originally withdrawn to compute the net energy sold ($\Delta EPA = EPA_{deposit} - EPA_{withdraw}$). When interval t arrives and the energy transfer begins (12), deviations from the allocated trades are covered by the DSO, including deviations due to prosumer failures. To provide billing information for the DSO, the smart meter must measure the deviations. To this end, it measures the net energy production E_u^t (negative values represent net consumption) of prosumer u at time interval t. The smart meter then computes the difference between the net energy sold and the net energy production to get the residual production (again, negative values are residual consumption). The residual production or consumption is multiplied by the selling or buying price of the DSO, respectively, to calculate what the prosumer owes the DSO for each interval. Every (13) billing cycle, the smart meter sums the cost of the residuals and sends that to the DSO for the monthly bill. The bill B_u^t of prosumer u for timeslot t, which will be paid by the prosumer to the DSO, is

$$B_u^t = \begin{cases} \left(E_u^t + \Delta EPA\right) \cdot \pi_t^S & \text{if } E_u^t + \Delta EPA < 0 \\ \left(E_u^t + \Delta EPA\right) \cdot \pi_t^B & \text{otherwise,} \end{cases} \tag{16}$$

where π_t^S is how much the DSO pays to purchase energy and π_t^B is how much the DSO charges for energy. The price schedule is set for each timeslot t by the DSO. The prices could be functions of $E_u^t + \Delta EPA$ to charge higher rates as the deviation from the traded amount increases. By designing the DSO prices to vary based on the deviation from the amount traded, we can provide strong incentives to prosumers to predict energy production and consumption accurately and to post conservative offers, so that the DSO and other prosumers can adjust their production or consumption preemptively, reducing the balancing that the DSO must provide due to unanticipated demand.

6 DISCUSSION AND ANALYSIS

In this section, we first describe how the TRANSAX design ensures the security, resilience and safety of the system. Then, we provide a discussion on the inherent trade-offs between efficiency, and privacy. Table 2 summarizes how each component in the architecture contributes to satisfying the system requirements.

6.1 Requirement Evaluation

6.1.1 Security and Safety. The underlying blockchain platform provides basic security features, so we are not concerned with the operations occurring on the blockchain. We are concerned with the secure and reliable operation of the solver. Similarly, the basic safety of the system is handled by the constraints described in Section 4.2. The safety constraints are applied correctly and reliably by the same contract. An adversary cannot force the contract to finalize trades based on an unsafe (*i.e.*, infeasible) solution since such a solution would be rejected. Similarly, an adversary cannot force the contract to choose an inferior solution instead of a superior one. In sum, the only action available to the adversary is proposing a superior feasible solution, which would actually improve energy trading in the microgrid.

6.1.2 Resilience. Now we show that our contract is reliable and can tolerate temporary disruptions in the DSO, solvers, or the communication network. First, since the *finalize* contract function is time guarded any entity can call it, and the system can progress without a DSO which is only required for registering new prosumers and their smart meters. Second, notice that any solution

Table 2. Summary of Component Functionality in TRANSAX

Requirement	Components	Approach
Security and Safety	Distribution System Operator and Smart Contract	DSO sets trading limits for prosumers and feeders, and the smart contract enforces them.
Resilience	Distributed Ledger and Hybrid Solver Architecture	Distributed ledger is resilient because of its distributed nature. Solvers are replicated to provide resilience.
Efficiency	Smart Contract and Solvers	Problem formulation allows temporal flexibility, smart contract enforces choosing best solution.
Privacy	Prosumers	Prosumers achieve privacy via a mixing protocol.

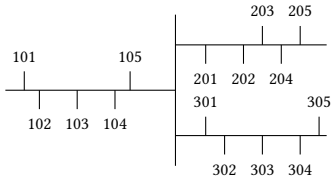

Fig. 5. Topology of a distribution network.

$(\boldsymbol{\varepsilon}, \boldsymbol{\pi})$ that is feasible for sets $\mathcal{S}$ and $\mathcal{B}$ is also feasible for supersets $\mathcal{S}' \supseteq \mathcal{S}$ and $\mathcal{B}' \supseteq \mathcal{B}$. As the sets of offers can only grow over time, the contract can use a candidate solution submitted during time interval t to finalize trades in any subsequent time interval $\tau > t$. In fact, without receiving new solutions, the difference between the amount of finalized trades and the optimum will increase only gradually: since the earlier candidate solution can specify trades for any future time interval, the difference is only due to the offers that have been posted since the solution was found and submitted. Thus, the system can continue making trades using older valid solutions

6.1.3 Trading Efficiency. The trading platform we have presented is able to support efficient trading through temporal flexibility. We show this through Example 1. As a reminder, this is due to prosumers being able to specify their production/consumption capacities and preferences (*i.e.*, reservation prices) via offers and the linear-program finding an optimal matching. In Section 7.3, we show using simulation that energy trading reduces the load on the power grid.

Example 1. Consider two prosumers (denoted by 102, 103) and one consumer (denoted by 101) from the community depicted in Fig. 5. We divide each day into 15-minute intervals. Let us assume that 102 has the ability to transfer 10 kWh into the feeder during interval 48, which translates to 12:00pm–12:15pm. Assume similarly that 103 can also provide 30 kWh to the feeder in interval 48, but it has battery storage. Since 103 has battery—unlike 102, who must either transfer the energy or waste it—103 can delay the transfer until a future interval, *e.g.*, interval 49. Now suppose that 101 needs to consume 30 kWh in interval 48 and 10 kWh in interval 49. A possible solution would be to provide all 30 kWh to 101 from 103 in interval 48. However, that will lead to the waste of energy provided by 102. Thus, a better solution will be to consume 10 kWh from 102 in interval 48 and 20 kWh from 103 in interval 48. Then, transfer 10 kWh from 103 in interval 49, which is more efficient than the first matching as it allows more energy (summed across the intervals) to be transferred. Thus, we see that permitting temporal flexibility can significantly increase trading volume, though it does increase the size of the optimization problem, increasing computational complexity.

6.1.4 Privacy. The platform provides pseudo-anonymity as the individual offers cannot be tied back to the prosumer who posted them since the offer is only affiliated with an anonymous address

and contains only the energy amount and reservation price. Additionally, the DSO does not know the total amount of energy utilized by the prosumers thanks to the anonymous billing via the smart meter. However, to preserve safety, some information about the prosumers needs to be public to allow checking of the offers to ensure that they are safe or limit the resources available to them.

In our design, we assume that the consumption (L^-) and production (L^+) limits of each prosumer are public information, as well as which feeder a prosumer is on. The group safety constraints C_g^i and C_g^e are also public. Recall that the smart contract ensures that no prosumer can withdraw more assets than the specified limits, and that any offer which violates the recorded safety constraints will be rejected. As a result, the only way to violate the safety requirements is if the asset limits or safety constraints are set incorrectly, which is not allowed by our design. However, as we will show below it is possible to improve privacy by choosing a conservative safety constraint for a group or a conservative limit on the maximum assets a prosumer can withdraw, which impacts the trading efficiency. Consider the following example for illustration.

Example 2. Consider the community depicted in Fig. 5. Let the prosumers denoted by 102 and 103 form a group g with an internal constraint of $C_g^i = 40$, where prosumers 102 and 103 have asset limits $L^+ = 10$ and $L^+ = 30$, respectively. Assume that the prosumers in this group have anonymized their assets. If the total assets traded by the group—which we denote T_f—is below 10, then there is no way to definitively say that either prosumer is trading. If the assets traded by f exceed 10, then we know that 103 is trading at least $T_f - 10$ since 102 can only produce 10. If $T_f > 30$, then we know that 102 is trading at least $T_f - 30$. If $T_f = 40$ or 0, then we know the full state of the feeder, either both prosumers are trading at their limit or not trading at all. To improve anonymity, the feeder as a whole should not trade more than 10. This however reduces trading efficiency considerably. Nonetheless, if both prosumers have $L^+ = 20$, then anonymity is improved until trading exceeds 20. Thus, it is important to select the constraints carefully. We discuss this in Section 6.2.

6.2 Tradeoff between Privacy and Efficiency

Note that the safety of the system is a strict requirement, which we cannot compromise. Thus, the only plausible tradeoff is between privacy and efficiency. This tradeoff can be achieved by creating groups, as we discussed in Section 4.3. However, groups and constraints must be created and set carefully to ensure that trading remains safe while also minimizing the loss in trading potential[17]. To better understand this problem, consider that when a group is mapped to a single physical feeder, the safety constraints are simply the feeder's constraints. However, in a group, we cannot tell which feeder the accounts belong to once the accounts are anonymous. Thus, to preserve safety, the constraints need to be adjusted. Therefore, the set of feeders are transformed into a group by treating all the prosumers in those feeders as if they were on a common feeder. Since the offers are anonymous at the group-level, the system can treat the group as a single feeder with two prosumers: one which posts production offers and one which post consumption offers (see Fig. 6).

To describe the methodology for selecting group constraints and the corresponding cost of privacy, we need to consider two cases.

6.2.1 Case 1 - There is a set of prosumers in the group that is capable of exceeding the safety constraint of the feeder they are on: Assume a microgrid with feeders $\mathcal{F}$ and groups $\mathcal{G}$, wherein L_u for each prosumer[18] and C_f for each feeder can have any value. Recall that we only need to consider $C_g^e \leq C_g^i$.

[17] The downside of grouping is that common feeder groups may result in lower energy trading limits due to modified aggregated constraints. We call this efficiency loss the cost of privacy. [18] Note that L_u^+ and L_u^- (Table 1) are the same type of constraint, representing production/outgoing or consumption/incoming limits, so we will use L_u to represent both in our analysis, but in each case the equations refer to both.

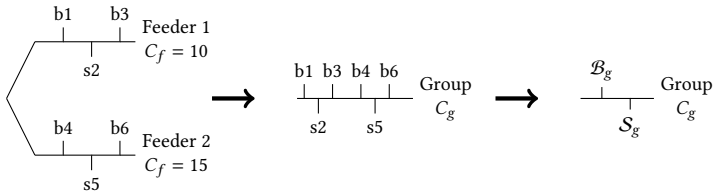

Fig. 6. Feeder conversion diagram.

For now, we consider when both constraints are violated simultaneously, setting $C^e_g = C^i_g$, and refer to it as the *feeder safety limit* C_g. For this system to be safe, the following condition on C_g must hold for every group g:

$$C_g \leq \min \left\{ C_f \middle| f \in g \text{ and } \sum_{u \in f} L_u \geq C_f \right\} \tag{17}$$

Proof. For the sake of contradiction, suppose that Equation (17) does not hold, but the system is safe. This means that $\exists C_f < \sum_{u \in f} L_u$ and $C_g > C_f$. Let $\sum_{u \in f} L_u = C_g$. Then, the prosumers in f can trade *EL* assets. However, this exceeds the feeder safety limit, so the system cannot be safe. Equation (17) must therefore be true. □

Thus, the best value for the group constraint is when Equation (17) is equality. This means that the group as a whole can at most produce the same amount as the single smallest of its internal feeders. The cost in this case is:

$$cost = \min \left\{ \sum_{\forall s \in \mathcal{S}_g} E_s, \sum_{\forall b \in \mathcal{B}_g} E_b \right\} - \min \left\{ \sum_{\forall s \in \mathcal{S}_g} E_s, \sum_{\forall b \in \mathcal{B}_g} E_b, C_g \right\}. \tag{18}$$

Thus, the cost is the amount by which the potential trades exceed the safety constraint.

6.2.2 Case 2 - No set of prosumers in any of the feeders in the group are capable of exceeding their feeders' safety constraint: Given a microgrid with feeders $\mathcal{F}$ and groups $\mathcal{G}$ where C_f can have any value and

$$\forall_g \forall_{f \in g} \sum_{u \in f} L_u \leq C_f, \tag{19}$$

group constraint should be set as $C_g = \sum_{f \in g} C_f$ to maximize trading, and trades can be done safely.

Proof. Assume a microgrid is not safe and Equation (19) is true. Then, $\exists f$ such that $\sum_{u \in f} L_u > C_f$. But, Equation (19) says this is not allowed. So, the system is safe. □

In this case, there is no cost to group privacy. Safety is ensured by the asset withdrawal limits rather than the group constraint. Note that Case 1 can be converted to Case 2 by reducing the prosumer asset limits so that the prosumers on a feeder cannot exceed their feeder's safety constraint[19]. To compute the cost of this conversion, instead of setting $C^i_f = C^e_f$ as we did in Case 1, we let $C^i_f > C^e_f$. This means that without privacy, the amount of energy that can safely be traded within the feeder is greater than the amount of energy that can be traded with other feeders. In this case, the maximum amount of energy that could potentially be traded is C^i_f. Even if the prosumers could exceed the internal constraint, those trades would not be permitted, so they are not a loss. Therefore,

[19] This can be enforced by the DSO during installation.

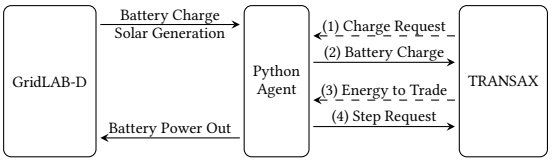

Fig. 7. Messages exchanged between simulator and TRANSAX.

we need to consider only the trades that could have been made but are no longer permitted, which is at most $C_f^i - C_f^e$.

6.2.3 Insights on Grouping. Based on the analysis of the effects of privacy on efficiency, the best strategy is to limit the trading assets of the prosumers such that they remain less than the feeder constraints. This means that all feeders can be safely grouped. The cost of grouping feeders is the loss of flexibility in trading due to the rigid asset limits. The cost will be at most the feeder limit minus the prosumer asset limit, if that prosumer has the capacity to reach the feeder limit, and if no other prosumers in its feeder are trading. This could be mitigated by an additional mixing and trading step within the feeder, but we have not examined this possibility in detail. There is a second criterion that may influence grouping decisions. There is information leakage, and at the extremes (max load, zero load) anonymity ceases to exist. We assume that generally this will not be the case, and the odds of that occurring diminish if there are many feeders in the group. Information leakage can be reduced by setting all the asset limits to the same value for all prosumers. The maximum system cost of this is the difference between the feeder limit and the sum of the prosumer limits. To reduce information leakage, groups should consist of feeders with similar limits.

7 EXPERIMENTAL EVALUATION

In this section, we present a simulation testbed[20] that we developed for evaluating TRANSAX, as well as our initial results illustrating the effectiveness of TRANSAX in reducing the load on the bulk power grid.

7.1 Testbed

The system to demonstrate the simulation platform has three major parts as shown in Fig. 7: the TRANSAX nodes (BeagleBone Blacks[21]) emulating the prosumers[22], the distribution system physics simulator (GridLAB-D [9], running on an x86 computer with a Core i7 processor and 24GB of RAM), and a Python agent to coordinate the hardware in the loop (emulated TRANSAX prosumers) integration with GridLAB-D. Messages and time steps between the Python agent and GridLAB-D are coordinated by the Framework for Network Co-Simulation [12].

The general message structure between GridLAB-D, the Python agent, and TRANSAX is shown in Fig. 7. While GridLAB-D is paused, TRANSAX agents request charge status for their batteries in the GridLAB-D simulation. They use this data, along with their predicted energy usage, to create a bid which is sent to TRANSAX. TRANSAX agents send the finalized trades back to the Python agent. The Python agent sets each simulated node's output for the next interval based on the finalized trades from TRANSAX by modifying GridLAB-D system parameters. In this demonstration, the Python agent meets the finalized trades only by modifying battery outputs. However, the Python agent has control over all the dynamically modifiable parameters in GridLAB-D. Consequently,

[20] The source code of the testbed is available at https://github.com/scope-lab-vu/transactive-blockchain [21] With limited computational capability and ARM architecture, these nodes are a good representation of embedded devices that we can expect to be used in real scenarios for managing energy trading within communities. [22] The control logic of prosumers is implemented in RIAPS.

future demonstrations could incorporate more control parameters, such as curtailments to solar output or curtailments to energy used by pure consumers.

The most important feature of this demonstration is its methodology for synchronizing time between GridLAB-D and TRANSAX, which is also responsible for time synchronization between GridLAB-D's variable-timestep solver and TRANSAX's matching solver. In the experiments described below, we use a solver time period of 15 minutes. Thus, the Python agent forces GridLAB-D's variable-timestep solver to pause at each logical 15-minute interval. Then, the prosumer nodes post offers for each 15-minute interval of logical time, and TRANSAX clears and finalizes trades. Next, the GridLAB-D simulation is advanced with actual energy transfer, allowing the impact to be measured. This process is repeated for the duration of the simulation's logical time. The time-synchronization strategy is scalable to any desired time period for the TRANSAX solver. The strategy also provides freedom to run experiments, such as assessing how the solver's time period affects the amount of energy traded, the stability of the finalized trades, or computational cost. Note that all physical nodes in the setup are time synchronized using the services provided by RIAPS [63] (Section 2).

7.2 Simulated Scenario

We run our simulations on the distribution topology described earlier in Fig. 5. It consists of substation feeding three main overhead lines that are connected to prosumers. The lines below the main lines represent prosumers with batteries and solar panels, which enables them to either consume or produce energy depending on the net output of the solar panels and batteries; and those above the main line represent prosumers with loads only (*i.e.*, they can never produce). For the demonstration, the simulation was built with 9 producer nodes and 6 consumer nodes.

The simulation was set up to run in logical time from 8AM to 8PM of the same day, for a total duration of 12 hours. During our experiments, we sped up the simulation by letting the real-time length of the time interval be $\hat{\Delta} < \Delta$ where $\hat{\Delta}$ is 2 minutes and Δ is 15 minutes. Note that $\hat{\Delta}$ is the amount of real time passed in the simulation before proceeding to the next interval; this allows us to speed up the experiments without compromising our results since running the system slower would be easier.

7.3 Results

We now discuss the results of three sets of experiments. The first experiment studies the impact of the solver horizon window T_h (Table 1). The second experiment demonstrates the benefit of the platform to the DSO. Finally, the third set of experiments studies how inaccurate predictions of energy consumption and productions affect the DSO.

7.3.1 Experiment 1 - Impact of T_h. It is expected that a longer time horizon will allow the TRANSAX solver to be more efficient and better match the producer and consumer offers. However, there is a tradeoff because a longer horizon also leads to higher computational cost. Thus, we varied the value of T_h and measured the memory usage, CPU usage, and amount of energy traded. In Fig. 8, we see that as the time horizon increases, so does the memory usage and energy traded until $T_h = 30$, at which point there is no additional gain to energy traded. The time horizon also impacts the CPU utilization of the solver (not shown). This demonstrates that we can select a finite time horizon and still obtain high-quality solutions.

7.3.2 Experiment 2 - Impact of TRANSAX on Load Serviced by DSO. To determine the impact of trading on the load supplied by the DSO, we ran several simulations. The simulation was first run without battery output and without any control by TRANSAX. This output was used to generate an energy profile for each prosumer for each interval. Then, the simulation was repeated

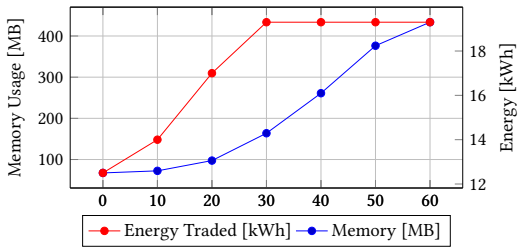

Fig. 8. Memory consumption and energy traded during a single interval of the simulation for various values of T_h using the CPLEX solver.

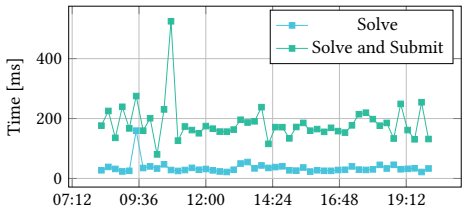

Fig. 9. *Solve* time is how long it took the solver to find a solution to the energy trading problem. *Solve and Submit* time is how long to took to find the solution and submit it to the smart contract.

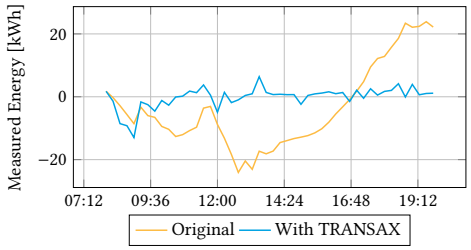

Fig. 10. DSO load with and without TRANSAX.

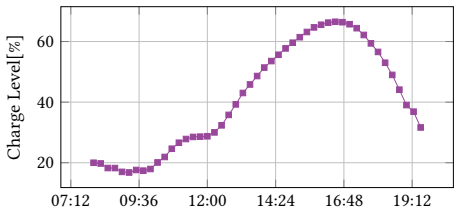

Fig. 11. Average battery charge level.

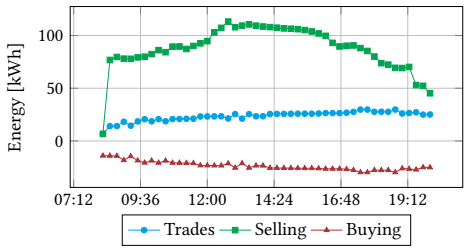

Fig. 12. Green: sum of all production offers for each interval. Red: negative sum of all consumption offers for each interval. Blue: sum of all energy traded in each interval, whose maximum value is the minimum of the production and consumption offers.

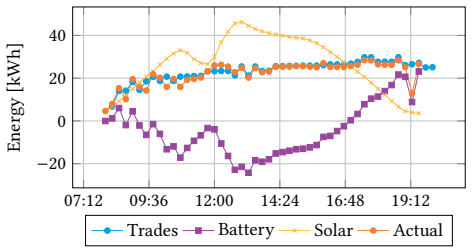

Fig. 13. Yellow: simulated solar profile. Purple: simulated battery charge level. Orange: simulated energy traded. Blue: total energy trades recorded in the market.

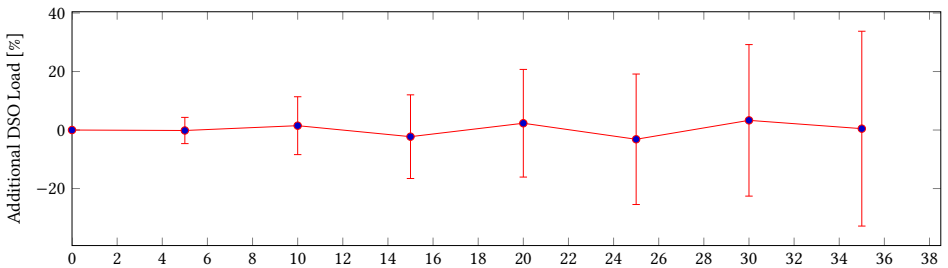

Fig. 14. Average additional load on DSO per day (calculated across 100 days) due to the difference between actual trades and the offers.

with the prosumers submitting offers matching their energy profile to the TRANSAX system, which represents an ideal scenario with accurate bids for each 15-minute interval. This simulation demonstrates how batteries can smoothen loads and utilize solar overproduction. Fig. 10 shows the comparison of DSO (utility substation) loads with and without TRANSAX. The horizontal axis is the simulated time since the start of the simulation. The vertical axis shows the load on the substation, negative values mean that the prosumers' generation exceeds their loads. Without TRANSAX, solar generation begins to outproduce the total load within the first interval. Solar production reaches its peak around 12:45PM. Finally, at 4:45PM, the load exceeds solar production, and the substation load becomes positive. The inclusion of TRANSAX dramatically reduces the need for the substation backup. From 8:00AM to 4:45PM, the overproduction of solar meant that the batteries were charging, which mitigated the negative load on the substation. After 4:45PM, the batteries discharged and mitigated the positive load on the system. Fig. 11 shows the average battery charge level across all 9 batteries. At the end of the simulation at 8PM, the average battery had only around 25% charge. This means that if the simulation were to go further into the night, there would not have been enough battery charge left to meet demand for the entirety of the night.

Fig. 12 shows the total amount of energy offered for each interval, as well as the total amount of energy recorded in trades. In Fig. 13, we see that the trades recorded (blue) are mostly consistent with the measured load (orange) on the system, with one notable exception at 2:15PM. The deviations occur because the prosumers currently assume that solar output remains constant over each interval, and this constant value is used when making offers. Fig. 9 shows the time required by the platform to find the optimal matching of a set of offers (green), as well as that time combined with the time required to submit that solution to the smart contract (blue). Most of the time spent is due to smart contract communications.

The results of the simulation with TRANSAX are promising. TRANSAX found energy trade solutions that resulted in an overall reduction of substation load. The distribution was however not completely independent of the substation feeder, and there is still a need for a connection to the larger distribution grid through a DSO.

7.3.3 Experiment 3 - Impact of Imprecision of Offer Prediction on DSO. Since prosumers must estimate their future production and consumption, we are interested in assessing the impact of estimation uncertainty on the stability of the system and on the load on the DSO. In an ideal case, prosumers will provide or consume the same amount of energy as they offered, and the DSO will know in advance how much energy it must provide to compensate for system stability. Any variations from the offered amount of energy result in uncertainty for the DSO. To study the effect of this uncertainty, we created scenarios where we added normally distributed error to energy produced or consumed by each prosumer (relative to the settled offers). The standard deviation of the error was scaled as a percentage of the prosumer's largest expected trade in a day, ranging from 0% to 35% of the energy traded. We chose this value because there are models for short-term cloud forecasting that have estimation errors of 20-30% for 15 to 45 minutes in the future [5]. We chose 35% as the upper bound. Fig. 14 shows the average daily difference between the energy that the DSO anticipated to provide from finalized trades and the energy it actually provided as a function of prosumer uncertainty. The averages were calculated over 100 simulated days. The uncertainty for the DSO increases with uncertainty in prosumer energy production and consumption. The standard deviation of the uncertainty for the DSO is 33% of the anticipated DSO load when prosumer trades are uncertain by a standard deviation of 35% of the offers; however, the average additional load remains near zero. The experiment demonstrates that while uncertainty in the offers will result errors in real-traded amount and eventually cause some uncertainty for the DSO, the net difference will remain small if the error is normally distributed.

8 RELATED WORK

Transactive strategies manage generators and loads based on market dynamics while ensuring system reliability. The earliest example of transactive control was demonstrated in the Olympic Peninsula Project [25]. An extension of these controls is seen in the management of building energy consumption [32]. Recently, with well-known grid failures, including the 2012 Sandy Storm and 2017 hurricane Maria in Puerto Rico there have been concerted efforts to build decentralized energy systems with transactive components [11, 46]. However, the existing platforms are not fully operational and, in most cases, cannot satisfy the three conflicting requirements of resilience, privacy and safety.

Existing energy trading markets, such as the European Energy Exchange [21] and project NOBEL in Spain, involve centralized database architectures which constitute single points of failure. The closest to a decentralized implementation that is required for resilience is Wörner et al. [66], who have developed an implementation of their peer-to-peer energy market and deployed it in a town in Switzerland. Their goal is to gather empirical evidence to answer the question of what the benefits of a blockchain system are in the electricity use case. However, the results have not yet been published. Similarly, the next phase of the LO3 project in Brooklyn [51] is working on extending the blockchain-based energy market. However, to the best of our knowledge, their focus has primarily been using blockchain as the resilient information store, and they are not using the decentralized architecture to implement a market. The blockchain there is simply a medium to store renewable energy attributes.

Since decentralized transactive energy system consists for various components including the markets, the controller and privacy mechanisms, we discuss them in detail below.

8.1 Markets

After prosumers presented their energy availabilities and demands in form of offers, these offers need to be matched. Researchers have proposed two approaches for this problem.

8.1.1 Stable Matching. Stable matching refers to matching of all possible buy and sell offers in a bipartite graph. Yucel *et al.* proposed a homomorphic encryption-based position hiding method [68] which protects users' privacy from adversary matchers. Nunna *et al.* [50] proposed the symmetrical allocation problem based on native auction algorithm to match buyers and sellers. PowerLedger [42] uses another mechanism to match offers. Offers are broken into equal portions and matched together e.g., when a new consumer arrives, it receives the equal allocation from the energy pool in the area.

8.1.2 Auction. Another approach to match buyers' offers to sellers' is to use auctioning approaches. Majumder *et al.* [43] proposed a double auction mechanism before the era of blockchains where the controller doesn't need the users' preferences, but instead they use an incentive compatible auction mechanism to extract that information in the form of bids. In the era of blockchains, Kang *et al.* [31] and Guerrero [24] used double auctions to match parties and not goods in blockchains. To ensure integrity of results of matching, Wang [64] proposed a multi-signed digital certificate. Khorsani *et al.* [34] designed a greedy algorithm with the averaging auction mechanism to match buyers with higher price to sellers with lower prices. Zhao *et al.* [72] created a two-phase auctioning algorithm to find the optimal pricing for bids. Finally, Zhang *et al.* [70] developed a non-cooperative auctioning game and used it to find the optimal solution for the matching problem using the Nash equilibrium.

8.2 Grid Control and Stability

One integral part of smart grids are the microgrid controllers which ensure stability and resiliency of the microgrid. They enable transition of the microgrid from grid-connected to islanded [40, 61]

so that the failures in the grid do not cascade to other areas similar to the outage event back in 1999 in Sao Paulo, Brazil [71]. Currently, most of microgrid controllers are centralized [33] which are vulnerable to cyber-threats and privacy issues. A large spectrum of cyber-threats are applicable on centralized microgrid controllers with single-point-of-failure ranging from attackers eavesdropping on channels between the controllable resource and the centralized controller to steal critical information of the users or network infrastructure, performing DDOS attacks on the centralized controller, or manipulation of demand via IoT (MadIoT) attacks[26] to injecting malware into the market operation system and manipulate settings, such as DLMP limits or clearing time interval similar to the notable cyber-attack against Ukrainian power systems in December 2015 [38, 69].

Due to these drawbacks of centralized grid controls, industry is transforming from centralized to decentralized [58, 67]. The aim of TRANSAX is to create a decentralized transactive energy market which ensures privacy and security of users while maintaining stability and resiliency of the grid.

8.3 Security and Privacy

8.3.1 Communication Security. First step to preserve users' privacy and anonymity in a distributed system is to provide communication privacy. Without this, an adversary can discern who is making a function call or sending a message over the network based on the sender's MAC address, IP address, or route to destination. Existing protocols for low-latency communication anonymity include onion routing [54], the similar garlic routing [41], STAC [30], and the decentralized Matrix protocol [23]. However, Murdoch and Danezis [48] show that a low-cost traffic analysis is possible of the Tor-network, theoretically and experimentally. Communication security is an orthogonal research problem to TRANSAX.

8.3.2 Address Anonymity. Communication anonymity is necessary but not sufficient for anonymous trading, as the cryptographic objectives of authentication and legitimacy are not fulfilled. We suggest using cryptographic techniques from distributed ledgers, *blockchains*, and cryptocurrencies. The most adopted one, Bitcoin allows for very simple digital cash spending but has serious privacy and anonymity flaws [2, 4, 55]. Additionally, Biryukov and Pustogarov, 2015, show that using Bitcoin over the Tor network opens a new attack surface [6]. Solutions to the tracing and identification problems identified by these researchers have been proposed and implemented in alternative cryptocurrency protocols: mixing using ring signatures and zero-knowledge proofs [47, 62].

A proposed improvement to standard ring signatures is the CryptoNote protocol, which prevents tracing assets back to their original owners by mixing incoming transactions and outgoing transactions. This service hides the connections between the prosumers and the addresses. Mixing requires the possibility to create new wallets at will and the existence of enough participants in the network. Monero is an example of a cryptocurrency that provides built-in mixing services by implementing the CryptoNote protocol [49]. There are however alternative implementations of mixing protocols such as CoinShuffle [57] or Xim [7]. A variant of ring signatures, group signatures, were first introduced by Chaum and van Heyst, 1991, [10] and then built upon by Rivest *et al.*, 2001 [56]. The basis for anonymity in the CryptoNote protocol, however, is a slightly modified version of the *traceable ring signature* algorithm by Fukisaki and Suzuki, 2007 [23]. This allows a member of a group to send a transaction so that it is impossible for a receiver to know any more about the sender than that it came from a group member without the use of a central authority.

Some newer cryptocurrencies, such as Zerocoin [47], provide built-in mixing services, which are often based on cryptographic principles and proofs.

[23] https://matrix.org/docs/spec/

8.3.3 Smart Meters' Privacy. Most works discussing privacy look at it from the context of smart meters. McDaniel and McLaughlin discuss privacy concerns due to energy-usage profiling, which smart grids could potentially enable [44]. Efthymiou and Kalogridis describe a method for securely anonymizing frequent electrical metering data sent by a smart meter by using a third-party escrow mechanism [17]. Tan et al. study privacy in a smart metering system from an information theoretic perspective in the presence of energy harvesting and storage units [59]. They show that energy harvesting provides increased privacy by diversifying the energy source, while a storage device can be used to increase both energy efficiency and privacy. However, transaction data from energy trading may provide more fine-grained information than smart meter-based usage patterns [27].

9 CONCLUSIONS AND FUTURE WORK

In this paper, we described TRANSAX, a decentralized platform for implementing energy exchange mechanisms in a microgrid setting. Building on top of blockchains, we obtained decentralized trust and consensus capabilities, which prevent malicious actors from tampering with the shared system state. We found that satisfying the seemingly conflicting goals of safety and privacy can be reconciled using anonymity within a grid, though this may result in a loss of flexibility and trading volume if the prosumers within a feeder could exceed the feeder's limit. Using our hybrid-solver approach, which combines a smart-contract based validator with an open set of external solvers, we showed that we can clear offers securely, efficiently, and resiliently, submitting solutions to the contract within approximately 200ms. We also demonstrated using TRANSAX that private blockchain based transactive energy is feasible for communities on the scale of microgrids and smaller, though we have not determined the upper limit for scalability. We are able to ensure that trades are balanced, and that energy trading is able to reduce the load on the DSO.

In the current implementation we have not chosen a specific approach for *setting the clearing prices* for the prosumers' trades since the economics of setting the clearing prices is an orthogonal problem. Friedman and Rust [22] provide a survey of these mechanisms for governing trade, to which they refer as market institutions. One of the most commonly used mechanisms is the double auction. Note that we cannot apply the double auction directly because of the different time-interval attributes that the offers may specify. Prior work has extended the double auction to allow for multiple attributes; however, they typically (*e.g.*, [3]) require a function to combine the attributes into a single value, which is then used to order the offers. The difficulty of this approach is in identifying a meaningful function. A more straightforward approach is to perform the feasibility matching as we have presented, and then for each interval, use a double auction to set the clearing price for the matched offers. This approach provides a straightforward solution to the problem of setting clearing prices; however, it is not obvious whether it will preserve the properties that a simple double auction has, such as incentive compatibility. We leave the investigation of these mechanisms and how they are impacted by privacy to future work.

Further, we need to allow prosumers to *update or cancel offers*. The current formulation can support updating offers as long as the updates do not invalidate previous solutions; for example, a selling offer can increase the amount of energy to be sold or augment the set of intervals in which energy could be produced. To support restrictive changes or cancelling offers, we would need to introduce a deadline for when offers could no longer be updated or cancelled. Solvers could then wait for this deadline and start working only after the deadline. Lastly, in the current implementation, the DSO provides the missing energy when a prosumer fails; however, we may also consider the case when the DSO is not available. In this case, a potential solution is to maintain backup energy reserves to satisfy demand that was unmet due to a failure.

ACKNOWLEDGEMENTS

We thank the anonymous reviewers of our journal submission for their insightful comments and valuable suggestions. We especially thank Prof. Gabor Karsai from Vanderbilt University and Prof. Srdjan Lukic for their feedback and help with the RIAPS platform. This work was funded in part by a grant from Siemens, CT and in part by grants from NSF under award number CNS-1647015, CNS-1818901, and CNS-1840052. The views presented in this paper are those of the authors and do not reflect the opinion or endorsement of Siemens, CT and NSF.

REFERENCES

[1] Merlinda Andoni, Valentin Robu, David Flynn, Simone Abram, Dale Geach, David Jenkins, Peter McCallum, and Andrew Peacock. 2019. Blockchain technology in the energy sector: A systematic review of challenges and opportunities. *Renewable and Sustainable Energy Reviews* 100 (2019), 143–174.

[2] M Apostolaki, A Zohar, and L Vanbever. 2017. Hijacking Bitcoin: Routing Attacks on Cryptocurrencies. In *2017 IEEE Symposium on Security and Privacy (SP)*. 375–392. https://doi.org/10.1109/SP.2017.29

[3] Gaurav Baranwal and Deo Prakash Vidyarthi. [n.d.]. A fair multi-attribute combinatorial double auction model for resource allocation in cloud computing. 108 ([n. d.]), 60–76. https://doi.org/10.1016/j.jss.2015.06.025

[4] Simon Barber, Xavier Boyen, Elaine Shi, and Ersin Uzun. 2012. *Bitter to Better — How to Make Bitcoin a Better Currency*. Springer Berlin Heidelberg, Berlin, Heidelberg, 399–414. https://doi.org/10.1007/978-3-642-32946-3_29

[5] Florian Barbieri, Sumedha Rajakaruna, and Arindam Ghosh. 2017. Very short-term photovoltaic power forecasting with cloud modeling: A review. *Renewable and Sustainable Energy Reviews* 75 (2017), 242 – 263.

[6] A Biryukov and I Pustogarov. 2015. Bitcoin over Tor isn't a Good Idea. In *Proceedings of the 2015 IEEE Symposium on Security and Privacy*. 122–134. https://doi.org/10.1109/SP.2015.15

[7] George Bissias, A Pinar Ozisik, Brian N Levine, and Marc Liberatore. 2014. Sybil-Resistant Mixing for Bitcoin. In *Proceedings of the 13th Workshop on Privacy in the Electronic Society (WPES '14)*. ACM, New York, NY, USA, 149–158.

[8] Miguel Castro and Barbara Liskov. 1999. Practical Byzantine Fault Tolerance. In *Proceedings of the Third Symposium on Operating Systems Design and Implementation (OSDI '99)*. USENIX Association, Berkeley, CA, USA, 173–186.

[9] David P Chassin, K Schneider, and C Gerkensmeyer. 2008. GridLAB-D: An open-source power systems modeling and simulation environment. In *2008 IEEE/PES Transmission and Distribution Conference and Exposition*. IEEE, 1–5.

[10] David Chaum and Eugène van Heyst. 1991. Group Signatures. In *Proceedings of the 10th Annual International Conference on Theory and Application of Cryptographic Techniques (EUROCRYPT)*. 257–265.

[11] Sijie Chen and Chen-Ching Liu. 2017. From demand response to transactive energy: state of the art. *Journal of Modern Power Systems and Clean Energy* 5, 1 (2017), 10–19.

[12] Selim Ciraci, Jeff Daily, Jason Fuller, Andrew Fisher, Laurentiu Marinovici, and Khushbu Agarwal. 2014. FNCS: a framework for power system and communication networks co-simulation. In *Proceedings of the Symposium on Theory of Modeling & Simulation - DEVS Integrative*. 1–8.

[13] William Cox and Toby Considine. 2013. Structured energy: Microgrids and autonomous transactive operation. In *Proceedings of the 2013 IEEE PES Innovative Smart Grid Technologies (ISGT),*. 1–6.

[14] Oben Dag and Behrooz Mirafzal. 2016. On stability of islanded low-inertia microgrids. In *2016 Clemson University Power Systems Conference (PSC)*. IEEE, Clemson, SC, USA, 1–7.

[15] H Diedrich. 2016. *Ethereum: Blockchains, Digital Assets, Smart Contracts, Decentralised Autonomous Organisations*. CreateSpace Independent Publishing Platform. https://books.google.com/books?id=Y2YRvgAACAAJ

[16] Y. Du, H. Tu, S. Lukic, D. Lubkeman, A. Dubey, and G. Karsai. 2017. Implementation of a distributed microgrid controller on the Resilient Information Architecture Platform for Smart Systems (RIAPS). In *2017 North American Power Symposium (NAPS)*. 1–6. https://doi.org/10.1109/NAPS.2017.8107305

[17] Costas Efthymiou and Georgios Kalogridis. 2010. Smart grid privacy via anonymization of smart metering data. In *Proceedings of the 1st IEEE International Conference on Smart Grid Communications (SmartGridComm)*. 238–243.

[18] Scott Eisele, Abhishek Dubey, Gabor Karsai, and Srdjan Lukic. 2017. Transactive Energy Demo with {RIAPS} Platform. In *8th International Conference on Cyber Physical Systems (ICCPS)*. IEEE, Pittsburgh, PA, USA, 91.

[19] Scott Eisele, Aron Laszka, Anastasia Mavridou, and Abhishek Dubey. 2018. SolidWorx: A Resilient and Trustworthy Transactive Platform for Smart and Connected Communities. In *2018 IEEE International Conference on Blockchain (Blockchain-2018)*. IEEE, Halifax, NS, Canada, Canada, 1263–1272.

[20] Scott Eisele, Istvan Madari, Abhishek Dubey, and Gabor Karsai. 2017. RIAPS: Resilient Information Architecture Platform for Decentralized Smart Systems. In *2017 IEEE 20th International Symposium on Real-Time Distributed Computing (ISORC)*. IEEE, IEEE, Toronto, ON, Canada, 125–132.

[21] European Power Exchange. 2017. EPEX SPOT Operational Rules. http://www.epexspot.com/en/extras/download-center/technical_documentation

[22] Daniel Friedman. [n.d.]. *The Double Auction Market Institutions, Theories, and Evidence: Proceedings of the Workshop on Double Auction Markets held June, 1991 in Santa Fe, New Mexico* (1 ed.). Routledge.

[23] Eiichiro Fujisaki and Koutarou Suzuki. 2007. *Traceable Ring Signature.* Springer, Berlin, Heidelberg, 181–200. https://doi.org/10.1007/978-3-540-71677-8_13

[24] J. Guerrero, A. C. Chapman, and G. Verbič. 2018. Decentralized P2P Energy Trading under Network Constraints in a Low-Voltage Network. *IEEE Transactions on Smart Grid* (2018), 1–1. https://doi.org/10.1109/TSG.2018.2878445

[25] Donald J Hammerstrom, Ron Ambrosio, Teresa A Carlon, John G DeSteese, Gale R Horst, Robert Kajfasz, Laura L Kiesling, Preston Michie, Robert G Pratt, Mark Yao, et al. 2008. *Pacific Northwest GridWise™ Testbed Demonstration Projects; Part I. Olympic Peninsula Project.* Technical Report. Pacific Northwest National Lab (PNNL).

[26] Bing Huang, Alvaro A. Cardenas, and Ross Baldick. 2019. Not Everything is Dark and Gloomy: Power Grid Protections Against IoT Demand Attacks. In *28th USENIX Security Symposium.* USENIX Association, 1115–1132.

[27] A Hussain, V H Bui, and H M Kim. 2017. A Resilient and Privacy-Preserving Energy Management Strategy for Networked Microgrids. *IEEE Transactions on Smart Grid* PP, 3 (2017), 2127–2139.

[28] IBM ILOG CPLEX. 2009. V12. 1: User's Manual for CPLEX. *International Business Machines Corporation* 46, 53 (2009).

[29] Institute for Software Integrated Systems. 2020. Resilient Information Architecture Platform for Smart Grid. https://riaps.isis.vanderbilt.edu.

[30] S Jebri, M Abid, and A Bouallegue. 2017. STAC-protocol: Secure and Trust Anonymous Communication protocol for IoT. In *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC).* 365–370.

[31] J. Kang, R. Yu, X. Huang, S. Maharjan, Y. Zhang, and E. Hossain. 2017. Enabling Localized Peer-to-Peer Electricity Trading Among Plug-in Hybrid Electric Vehicles Using Consortium Blockchains. *IEEE Transactions on Industrial Informatics* 13, 6 (Dec 2017), 3154–3164. https://doi.org/10.1109/TII.2017.2709784

[32] Srinivas Katipamula, David P Chassin, Darrel D Hatley, Robert G Pratt, and Donald J Hammerstrom. 2006. *Transactive controls: A market-based GridWiseTM controls for building systems.* Technical Report. Pacific Northwest National Lab, Richland, WA (United States).

[33] Amandeep Kaur, Jitender Kaushal, and Prasenjit Basak. 2016. A review on microgrid central controller. *Renewable and Sustainable Energy Reviews* 55 (2016), 338–345. https://doi.org/10.1016/j.rser.2015.10.141

[34] M. Khorasany, Y. Mishra, and G. Ledwich. 2017. Auction based energy trading in transactive energy market with active participation of prosumers and consumers. In *Proceedings of the 2017 Australasian Universities Power Engineering Conference (AUPEC).* 1–6.

[35] Koen Kok and Steve Widergren. 2016. A society of devices: Integrating intelligent distributed resources with transactive energy. *IEEE Power and Energy Magazine* 14, 3 (2016), 34–45.

[36] Aron Laszka, Abhishek Dubey, Michael Walker, and Doug Schmidt. 2017. Providing privacy, safety and security in IoT-based transactive energy systems using distributed ledgers. In *Proceedings of the 7th International Conference on the Internet of Things (IoT '17).* ACM, New York, NY, USA, 13:1—-13:8. https://doi.org/10.1145/3131542.3131562

[37] Aron Laszka, Scott Eisele, Abhishek Dubey, Gabot Karsai, and Karla Kvaternik. 2018. TRANSAX: A Blockchain-Based Decentralized Forward-Trading Energy Exchanged for Transactive Microgrids. In *2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS).* 918–927. https://doi.org/10.1109/PADSW.2018.8645001

[38] Robert M. Lee, Michael J. Assante, and Tim Conway. 2016. *Analysis of the cyber attack on the Ukrainian power grid: Defense use case.* Technical Report. Electricity Information Sharing and Analysis Center (E-ISAC).

[39] Timo Lehtola and Ahmad Zahedi. 2019. Solar energy and wind power supply supported by storage technology: A review. *Sustainable Energy Technologies and Assessments* 35 (2019), 25–31.

[40] N W A Lidula and A D Rajapakse. 2011. Microgrids research: A review of experimental microgrids and test systems. *Renewable and Sustainable Energy Reviews* 15, 1 (2011), 186–202. https://doi.org/10.1016/j.rser.2010.09.041

[41] Peipeng Liu, Lihong Wang, Qingfeng Tan, Quangang Li, Xuebin Wang, and Jinqiao Shi. 2014. Empirical Measurement and Analysis of I2P Routers. *Journal of Networks* 9 (2014), 2269–2278.

[42] Power Ledger Pty Ltd. 2019. Power Ledger Whitepaper. https://www.powerledger.io/wp-content/uploads/2019/05/power-ledger-whitepaper.pdf (Accessed on 28 August 2019).

[43] Bodhisattwa P Majumder, M Nazif Faqiry, Sanjoy Das, and Anil Pahwa. 2014. An efficient iterative double auction for energy trading in microgrids. In *2014 IEEE Symposium on Computational Intelligence Applications in Smart Grid (CIASG).* IEEE, Orlando, FL, USA, 1–7. https://doi.org/10.1109/CIASG.2014.7011556

[44] Patrick McDaniel and Stephen McLaughlin. 2009. Security and privacy challenges in the smart grid. *IEEE Security & Privacy* 7, 3 (2009), 75–77.

[45] Ronald B Melton. 2013. *Gridwise transactive energy framework.* Technical Report. Pacific Northwest National Laboratory, Richland, WA.

[46] Esther Mengelkamp, Johannes Gärttner, Kerstin Rock, Scott Kessler, Lawrence Orsini, and Christof Weinhardt. 2018. Designing microgrid energy markets: A case study: The Brooklyn Microgrid. *Applied Energy* 210 (2018), 870–880.
[47] I Miers, C Garman, M Green, and A D Rubin. 2013. Zerocoin: Anonymous Distributed E-Cash from Bitcoin. In *2013 IEEE Symposium on Security and Privacy*. IEEE, Berkeley, CA, USA, 397–411. https://doi.org/10.1109/SP.2013.34
[48] S J Murdoch and G Danezis. 2005. Low-cost traffic analysis of Tor. In *2005 IEEE Symposium on Security and Privacy*. 183–195. https://doi.org/10.1109/SP.2005.12
[49] Shen Noether. 2015. Ring Signature Confidential Transactions for Monero. Cryptology ePrint Archive, Report 2015/1098.
[50] HSVS Kumar Nunna and Suryanarayana Doolla. 2013. Multiagent-Based Distributed-Energy-Resource Management for Intelligent Microgrids. *IEEE Transactions on Industrial Electronics* 60, 4 (April 2013), 1678–1687.
[51] Lawrence Orsini, Scott Kessler, Julianna Wei, and Heather Field. [n.d.]. *How the Brooklyn Microgrid and TransActive Grid are paving the way to next-gen energy markets*.
[52] Farrokh A Rahimi and Ali Ipakchi. 2012. Transactive energy techniques: closing the gap between wholesale and retail markets. *The Electricity Journal* 25, 8 (2012), 29–35.
[53] Tom Randall. 2015. The way humans get electricity is about to change forever. Bloomberg.
[54] Michael G Reed, Paul F Syverson, and David M Goldschlag. 1998. Anonymous connections and onion routing. *IEEE Journal on Selected Areas in Communications* 16, 4 (1998), 482–494.
[55] Fergal Reid and Martin Harrigan. 2013. An Analysis of Anonymity in the Bitcoin System. In *Security and Privacy in Social Networks*. 197–223.
[56] Ronald L Rivest, Adi Shamir, and Yael Tauman. 2001. How to Leak a Secret. In *Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT)*. 552–565.
[57] Tim Ruffing, Pedro Moreno-Sanchez, and Aniket Kate. 2014. CoinShuffle: Practical Decentralized Coin Mixing for Bitcoin. In *Proceedings of the 19th European Symposium on Research in Computer Security (ESORICS)*. 345–364.
[58] J W Simpson-Porco, Q Shafiee, F Dörfler, J C Vasquez, J M Guerrero, and F Bullo. 2015. Secondary Frequency and Voltage Control of Islanded Microgrids via Distributed Averaging. *IEEE Transactions on Industrial Electronics* 62, 11 (November 2015), 7025–7038.
[59] Onur Tan, Deniz Gunduz, and H Vincent Poor. 2013. Increasing smart meter privacy through energy harvesting and storage devices. *IEEE Journal on Selected Areas in Communications* 31, 7 (2013), 1331–1341.
[60] H. Tu, Y. Du, H. Yu, A. Dubey, S. Lukic, and G. Karsai. 2019. Resilient Information Architecture Platform for the Smart Grid (RIAPS): A Novel Open-Source Platform for Microgrid Control. *IEEE Transactions on Industrial Electronics* (2019).
[61] Taha Selim Ustun, Cagil Ozansoy, and Aladin Zayegh. 2011. Recent developments in microgrids and example cases around the world – A review. *Renewable and Sustainable Energy Reviews* 15, 8 (2011), 4030–4041.
[62] Nicholas van Saberhagen. 2012. *CryptoNote v 2.0*. Technical Report. 3 pages. https://cryptonote.org/whitepaper_v2.pdf
[63] Peter Volgyesi, Abhishek Dubey, Timothy Krentz, Istvan Madari, Mary Metelko, and Gabor Karsai. 2017. Time Synchronization Services for Low-cost Fog Computing Applications. In *28th International Symposium on Rapid System Prototyping (RSP)*. IEEE, New York, NY, USA, 57–63.
[64] Jian Wang, Qianggang Wang, Niancheng Zhou, and Yuan Chi. 2017. A Novel Electricity Transaction Mode of Microgrids Based on Blockchain and Continuous Double Auction. *Energies* 10, 12 (2017). https://doi.org/10.3390/en10121971
[65] Gavin Wood. 2014. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper* 151 (2014).
[66] Anselma Wörner, Arne Meeuw, Liliane Ableitner, Felix Wortmann, Sandro Schopfer, and Verena Tiefenbeck. 2019. Trading solar energy within the neighborhood: field implementation of a blockchain-based electricity market. *Energy Informatics* 2 (2019). Issue S1. https://doi.org/10.1186/s42162-019-0092-0
[67] M Yazdanian and A Mehrizi-Sani. 2014. Distributed Control Techniques in Microgrids. *IEEE Transactions on Smart Grid* 5, 6 (November 2014), 2901–2909. https://doi.org/10.1109/TSG.2014.2337838
[68] F Yucel, E Bulut, and K Akkaya. 2018. Privacy Preserving Distributed Stable Matching of Electric Vehicles and Charge Suppliers. In *Proceedings of the 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*. 1–6.
[69] Kim Zetter. 2016. Inside the Cunning, Unprecedented Hack of Ukraine's Power Grid. WIRED, https://www.wired.com/2016/03/inside-cunning-unprecedented-hack-ukraines-power-grid/.
[70] Chenghua Zhang, Jianzhong Wu, Yue Zhou, Meng Cheng, and Chao Long. 2018. Peer-to-Peer energy trading in a Microgrid. *Applied Energy* 220 (2018), 1–12. https://doi.org/10.1016/j.apenergy.2018.03.010
[71] Liang Zhao, Kwangho Park, and Ying-Cheng Lai. 2004. Attack vulnerability of scale-free networks due to cascading breakdown. *Physical Review E* 70, 3 (sep 2004), 35101. https://doi.org/10.1103/PhysRevE.70.035101
[72] Shengnan Zhao, Beibei Wang, Yachao Li, and Yang Li. 2018. Integrated Energy Transaction Mechanisms Based on Blockchain Technology. *Energies* 11, 9 (2018). https://doi.org/10.3390/en11092412