Feature: Blockchain and Cyberphysical Systems

Blockchains for Transactive Energy Systems: Opportunities, Challenges, and Approaches

Scott Eisele, Carlos Barreto, Abhishek Dubey, Xenofon Koutsoukos
Vanderbilt University

Taha Eghtesad, Aron Laszka
University of Houston

Anastasia Mavridou
KBR / NASA Ames Research Center

***Abstract*—The emergence of blockchains and smart contracts have renewed interest in electrical cyber-physical systems, especially in the area of transactive energy systems. However, despite recent advances, there remain significant challenges that impede the practical adoption of blockchains in transactive energy systems, which include implementing complex market mechanisms in smart contracts, ensuring safety of the power system, and protecting residential consumers' privacy. To address these challenges, we present TRANSAX, a blockchain-based transactive energy system that provides an efficient, safe, and privacy-preserving market built on smart contracts. Implementation and deployment of TRANSAX in a verifiably correct and efficient way is based on VeriSolid, a framework for the correct-by-construction development of smart contracts, and RIAPS, a middleware for resilient distributed power systems.**

Index Terms: Electronic commerce, Middleware/business logic, Industrial control.

TRANSACTIVE ENERGY SYSTEMS

In the last decade, there has been an emphasis on decentralizing the operations of electrical power grids [1] due to their vulnerability to natural disasters, such as Hurricane Maria, and cyber threats, such as the Ukraine power grid attack. In the absence of centralized control, the "prosumers" (customers with both electrical energy production and consumption capability) can collaborate to dynamically balance the demand and supply across their microgrid, improving system reliability. However, this requires a financial

 Published by the IEEE Computer Society

market at the distribution level, where participants can trade energy assets. It also requires control strategies to keep local energy sources stable due to the low system inertia compared to a conventional grid [2]. This is the main concept behind transactive energy systems (TES) [3].

Prosumers that change consumption (demand response) as part of market-based transactive control were demonstrated in the Olympic Peninsula Project [4] in 2006. Both local production and consumption in a limited "transactive" system were demonstrated by the LO3 project in Brooklyn [5]. There are ongoing studies, such as the work done by Wörner et al. [6] in a town in Switzerland.

However, large-scale deployments are still missing. The primary reason for this is the complexity of integration between financial markets, predictive algorithms, information platforms, and physical control. While the research community has made progress in managing the control of the system [7] and developing predictive algorithms [8], the integration with a decentralized information architecture and market remains a challenge due to problems of trust, correctness, and privacy.

Our research team—and several other teams as shown by a recent survey [9]—proposed addressing the challenges of trust in TES through the use of blockchains. The motivation behind this is in part due to the success of Bitcoin, a prototypical example application of blockchains. Bitcoin stores transactions in a public distributed ledger, which is called a blockchain because the records are stored in blocks that are cryptographically linked to previous blocks, forming a chain. Any entity can read the ledger; however, to append a new block to the ledger, the Bitcoin network uses a probabilistic consensus protocol based on *proof-of-work* (PoW). This consensus protocol solves both trust and fault-tolerance issues since the majority of participants will reach consensus on the ledger state. Further, it provides censorship-resistant, immutable, tamper-proof, and transparent transactions, thus enabling trusted transactions without a trusted third party. Enabling trusted transactions without a trusted third-party is a crucial factor for TES. Some blockchain implementations also enable participants to implement *smart contracts*—programs that are stored and executed by the blockchain network, benefiting from its trust properties.

While the idea of integrating blockchains into TES is conceptually appealing, there are a number of challenges that must be addressed before protocols and implementations can live up to their potential. The outline for this article is as follows: first, we describe several of the key challenges which prevent the widespread adoption of decentralized TES. Then, we present TRANSAX, our solution for implementing blockchain-based TES and show how it addresses these challenges.

Challenges for Blockchains in TES

The key challenges of using blockchains in transactive energy systems can be summarized as (a) code complexity and immutability; (b) privacy issues; (c) high computation costs, especially when trying to process complex market operations through smart contracts; (d) integration challenges due to a lack of suitable patterns to interact with physical devices and to ensure time synchronization; and (e) security concerns of blockchain-based systems. Table 1 summarizes these challenges and how we address them.

Code Complexity and Immutable Bugs

Coding errors frequently occur due to incorrect assumptions about the execution semantics of smart contracts [12]. For example, Luu et al. [13] analyzed 19,366 smart contracts and found that 8,833 contracts had one or more security issues. These errors can result in devastating security incidents, such as the "DAO attack," where $50 million in cryptocurrency was stolen, and the multi-signature Parity Wallet library hack, where $280 million in cryptocurrency was lost.

Blockchain-based platforms are designed to provide immutability, which prevents patching smart contracts or reverting malicious transactions. Developers can work around this by separating the code into distinct contracts, a "frontend" and a "backend," where the frontend references the backend library. Then, to change the functionality of the frontend, developers can simply change the reference to point to a new version of the backend. However, this can also erode trust since a contract may be changed and no longer satisfy its original terms. In more extreme cases, transactions can be reverted via a hard fork, but this requires the consensus of all the

Table 1: Summary of challenges integrating blockchain technologies with power systems and our relevant contributions.

Challenge	Description	Contributions
Immutable bugs	Blockchains' design guarantees immutability; however, this means bugs are also immutable	Build and verify smart contracts using VeriSolid [10]
Efficiency	Smart contracts require all verifier nodes to replicate the computations in a transaction	Limit the computations executed on the smart contract to checking correctness
Integration	Existing power grid equipment does not have the capabilities for managing a distributed set of blockchain nodes integrated with the power equipment	Use the middleware services (time synchronization, discovery) of RIAPS for integration [11]
Privacy	Transaction details can be open and attributable to prosumers	Energy assets, cryptographic mixing, and groups to provide k-anonymity to prosumers while ensuring feeder level safety
Cybersecurity	Although blockchains protect against some attacks, adversaries can compromise information before it is processed by the blockchain	Design policies to mitigate attacks (future work)

stakeholders and introduces security issues such as replay attacks.

To tackle these security risks and vulnerabilities in TRANSAX, we use formal methods developed by our team to generate code from the high-level, graphical, and FSM-based language to low-level smart contract code. Rooting the whole process in rigorous semantics allows the integration of formal analysis tools, which can be used to verify safety and security properties, thereby enabling the development of correct-by-design smart contracts.

Computational Efficiency

Smart contracts are not suitable for executing complex market mechanisms, because the majority of verifier nodes responsible for verifying the computation in a given transaction must perform the computation to ensure correct execution, making computations very costly. This is sometimes referred to as *on-chain* computation. To limit the potential for abuse of the network, Ethereum sets an upper bound on the amount of computation that may be performed in a single transaction.

To provide complex market functionality, the computation must be performed *off-chain* and only the results should be evaluated and verified by the smart contract on-chain. This is apparent in the implementation of transactive energy systems where the trades must be decided optimally based on a complex set of equations considering the feeder design and various power limits. Such complex computations are not possible to implement in smart contract languages like Solidity. Therefore, we have developed a novel hybrid solver pattern for TRANSAX where we integrate external solvers with smart contracts. This enables us to perform the computations off-chain and verify them on the blockchain.

Privacy Concerns

Although it is possible to make anonymous transactions with cryptocurrencies, energy trades may need information that reveals the traders' identities. For example, the trades must be associated with a specific feeder to ensure that the maximum power transferred through the feeder is less than the rated capacity. This poses a challenge for privacy, because a trader may need to reveal its location to permit constraint checks and validate trades.

If the information is available publicly, then the inference of energy usage patterns can be exploited, for example, to infer the presence or absence of a person in their home. Brenzikofer et al. [14] address privacy while incentivizing stability through dynamic grid tariffs. However, their safety checks are limited to total aggregated grid load rather than per feeder constraints, which are essential in a power network. In TRANSAX, we use the concept of tradeable and mixable energy assets in a transactive energy system to provide a level of anonymity to the users while ensuring that system calculations at the feeder level are still safe.

Integration Concerns

Integrating legacy infrastructure with blockchains is challenging since most existing smart meters lack the computational capabilities required to participate in a blockchain network [15]. An alternative to directly participating is for the

devices to send their data to nodes that are connected to the blockchain network. However, this requires configuring each device to connect to a suitable gateway and mechanisms to handle lost connections and gateway failures. Moreover, while the ledger provides consensus on when to produce or consume power, participants still need to time synchronize their energy transfer to avoid instabilities in the system.

TRANSAX solves the integration concerns using RIAPS (Resilient Information Architecture Platform for Smart Grids) [11], a platform for building distributed, fault-tolerant smart-grid applications. RIAPS provides key services, like time-synchronization and discovery. Discovery facilitates the integration of legacy hardware with blockchain applications by automating the network connections between them via RIAPS nodes, which have been developed to run on low-cost embedded devices. Each component in the TRANSAX is either a RIAPS node or interfaces with a RIAPS node.

Security Threats

Research on power systems security has investigated cyber-attacks with different goals and strategies. Some attacks exploit the centralized nature of the system, for example, by compromising the utility's network to access control systems (such as in the attacks against Ukraine's power utilities). Other scenarios consider adversaries that target IoT or smart appliances to create disturbances in the system (e.g., turning all the A/Cs on at the same time).

The distributed nature of blockchain prevents some attacks that are feasible in centralized systems. For example, some *false data injection* attacks that modify utility's messages (e.g., price signals) may fail because the devices can verify such information with multiple sources (blockchain nodes). Hence, an adversary may have to compromise multiple blockchain nodes to deceive smart appliances. However, some attacks remain. Since prosumers must connect to the blockchain-based system through gateway nodes, an adversary can still attempt to "cut off" prosumers from the system by targeting these gateway nodes and making them unavailable. For example, an adversary can launch a (distributed) denial of service attack against a gateway node to prevent a set of bids from arriving at the market on time. Using this attack, the adversary, who may be affiliated with one of the market participants, can increase (or decrease) market prices by delaying a set of lower (or higher) price bids. We are still in the preliminary stages of developing active mitigation strategies in TRANSAX to prevent these attacks.

TRANSAX

TRANSAX is our solution for enabling transactive energy systems. Its architecture can be seen in Fig. 1, which describes all major components of the platform (middleware layers like RIAPS are not shown), including key smart contract functions and associated events. Each edge includes a circled number, *i.e.*, ⓗ, which indicates their sequence. The *Distribution System Operator* (DSO) regulates the microgrid and market. Prosumers are the participants who submit offers to produce or consume energy. Each prosumer has a smart meter, which is a secure device that measures the prosumer's energy flow and sends the monthly aggregate to the DSO for billing purposes. The smart meter also monitors the prosumer to detect any safety constraint violations. The smart contract provides the information system, enabling communications, and defines the offer format, as well as the rules for combining offers to form trades. The blockchain upon which the smart contract is deployed provides the storage for the smart contract data. The hybrid solver implements the market mechanism. We discuss these components and their interaction protocol below.

Smart Contract

The market is established via a *smart contract*, which enforces the system constraints and checks that trades do not violate them. It also defines the system's goal, represented as an optimization problem. The contract is deployed on a consortium blockchain. We use an Ethereum deployment with PoW consensus currently. However, this can be updated in future. To ensure the correctness of the smart contract, we use VeriSolid [10], an end-to-end, open-source framework for the *correct-by-design development and deployment of multiple interacting smart contracts* for blockchain-based CPS. VeriSolid helps

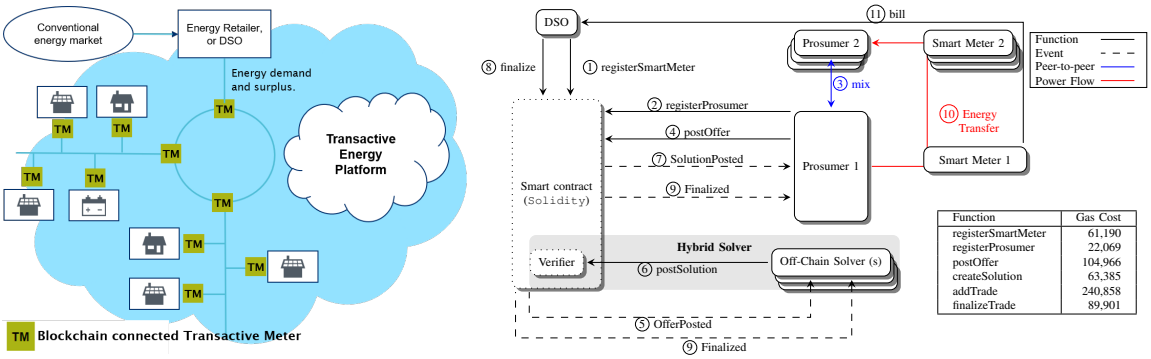

Figure 1: (Left) Physical microgrid topology in TRANSAX. Every node is managed by a smart meter, which has access to the blockchain and ensures proper billing per node. (Right) Information architecture of TRANSAX and the control flow of interaction between components. The gas costs for each function were estimated using the Remix editor and are shown in the inset table. The `postSolution` is a composite function that requires the solver to call `createSolution` followed by a number of `addTrade` invocations. Each `addTrade` specifies a seller, a consumer, the time interval, and the energy to be transferred. Function `finalize` is invoked a few intervals (can be configured) before the interval being finalized. The finalize call also requires the smart contract to check each trade that is part of that interval. The cost of single trade finalization is shown in the inset table. This cost is paid for each finalized trade.

developers to eliminate errors early at design time by raising the abstraction level and providing automated verification and code generation.

Integrating Prosumers

The market is initialized, and constraints are established through the utility company (*i.e.*, DSO), which regulates who can participate in the market. Any new prosumer must perform the ① `registerSmartMeter` step, which specifies the asset limit for each prosumer based on the physical constraints of the prosumer and the supporting infrastructure. In addition, each prosumer must also register itself by calling ② `registerProsumer`, which specifies its feeder as well as the corresponding smart meter after which the prosumer can participate in all future trading intervals. The DSO is also responsible for making any changes to the systems' constraints (energy capacity of the feeders) stored in the blockchain and updating the smart contract if required.

Hybrid Solver Pattern

To achieve the system's goal, the market must solve an optimization problem. In the default implementation, we maximize the energy traded within the microgrid. This can be formulated as a Mixed Integer Linear Program (MILP). However, other optimization formulations are available [16]. Solving these optimizations is impractical with smart contracts. Thus, we use a hybrid-solver architecture where specialized *off-chain solver* nodes access the offers stored within the blockchain and find possible solutions to the market's optimization problem (we use IBM CPLEX to implement the solvers). The solvers submit the proposed solutions to the market by calling ⑥ `postSolution`.

The smart contract implements a trade verifier that computes whether a proposed trade is feasible. Using the system utility function defined in the smart contract, the proposal is then evaluated to determine its quality. Since there are many off-chain solvers, the verifier receives many solutions and keeps only the best one. Each off-chain solver is free to use any algorithm to pair offers, but they will be inclined to submit trades that the smart contract will select. Additionally, having many off-chain solvers means that reliability is preserved since the market continues to function as long as one submits a valid solution. Together, the solvers and smart contracts provide computation efficiency and ensure that system constraints are not violated.

Providing Privacy while Ensuring Safety

Both when an offer is made or a trade (specifying the net energy a prosumer has to produce or consume in a finalized interval) is computed and submitted by a solver, the smart contract verifies that no hardware constraints are violated. For example, each prosumer is limited in the amount of power that can be transferred through its line. This limitation is recorded through the smart contract when the DSO registers the smart meter for the prosumer. Similarly, each feeder has a protection relay that ensures the net load connected to that feeder remains below a certain limit. When a set of consumers connected to a feeder send their offers, the smart contract can check that the aggregated load imposed by those consumers on the feeder is below the safety limit. When multiple feeders are connected to each other in a radial pattern and the power is transferred from one feeder to the others (for a set of matched trades in an interval), we approximate the load flow using the superposition principle. That is, we aggregate the net load for each feeder line per power source and ensure that the total aggregates at each feeder is below the safety limit. The safety limit is calculated by accounting for any line drops that might occur. Note that the drops are negligible if the line distances are short as found in communities.

However, if the participants in the trade are anonymous for the sake of privacy, then the smart contract can no longer verify the system's constraints. In this scenario, a prosumer could behave maliciously and destabilize the grid without fear of repercussions due to the anonymity. To reconcile the dichotomy between ensuring grid stability and privacy, we implement *energy assets*, which represent permissions to buy or sell some amount of energy during a fixed time interval. During an interval, offers are made to exchange energy in future intervals, while energy is exchanged according to previous trades. To make offers for a given interval, a prosumer must have unused assets available for that interval.

To trade privately, the prosumers transfer their assets from their public accounts to anonymous ones using a mixing service [17], which collects all offers from within a feeder and mixes them. This ensures that anonymous accounts are not associated with a specific prosumer, but rather a specific feeder. Therefore, when trades are made using an anonymous account, feeder constraints can still be enforced by the smart contract, and prosumer constraints are enforced by the energy assets.

To increase privacy, we allow the feeders to form groups. Before submitting their offers, groups of prosumers can create anonymous addresses using a mixing protocol (see step ③). This protocol combines the credentials of several prosumers providing *k-anonymity*, *i.e.*, each address cannot be associated with a particular prosumer. The group then transfers assets from their public addresses to these anonymous addresses, which are used for making energy trading offers. Prosumers who participate in the mixing protocol must share their public blockchain address and a public key with the other prosumers.

Forming a group requires constructing a group constraint to ensure that trades within and across groups are safe. This approach sacrifices some trading efficiency to allow prosumers to have anonymity at the group level while still ensuring that trades are safe. The efficiency loss occurs when a trade that would otherwise be safe is rejected. This could occur if the limit for exchange *within* a feeder was greater than *across* the feeder, and two feeders have formed a group. Then, since the system cannot distinguish between trades within or across feeders, it must assume the lower limit. A system integrator can choose to create groups of one prosumer each, which will ensure that the system will work with highest efficiency possible – but without any privacy.

Market Protocol

Fig. 1 describes the interaction sequence. The smart contract accepts offers for future trades during fixed time intervals (e.g., every 15 minutes). The prosumers submit offers using the available energy assets (withdrawn from the smart meter) by calling ④ `postOffer`, specifying the quantity and intervals during which the energy is available (e.g., prosumers with storage capability have more flexibility to execute the trades). Off-chain solvers monitor the blockchain data structure for ⑤ `OfferPosted` events and construct potential trades with the offers submitted. The solvers propose potential solutions by calling ⑥ `postSolution`, which include the number of

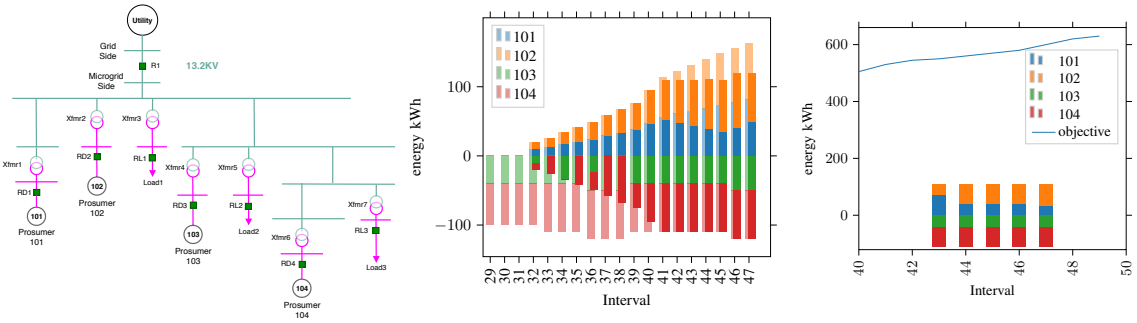

Figure 2: (Left) Microgrid used in the case study. Prosumers 103 and 104 are consumers. Prosumers 101 and 102 produce energy. Other loads are passive and are supplied by the utility. (Middle) Offers and subsequent trades made in the system. Lighter colors represent the offers that were made in an interval by a prosumer. Darker colors represent the actual trades that were matched and executed through TRANSAX in an interval. (Right) Objective and trades for an interval evolve over time before the interval is finalized.

trades, the total energy traded, the specific offers included, and the parties involved in the trades. The prosumers monitor the blockchain for ⑦ `SolutionPosted` events to determine whether their offers have been matched. Any unmatched assets are deposited back into the smart meter, enabling future offers.

The smart contract ranks the proposed solutions and accepts the best. The DSO calls ⑧ `finalize`, which closes the market, that is, instructs the smart contract to reject additional offers and solutions for the current interval. This function emits the ⑨ `Finalized` event to the blockchain data structure and also emits the final trades. Smart meters keep a balance of the future trades, and when the exchanging interval arrives, measure ⑩ actual *energy transfer* and check that it does not violate the safety constraints. The smart meter also computes the difference between the actual energy flow and the flow covered by trades to compute the prosumer's ⑪ *bill*, which it sends to the DSO on a monthly basis.

Multi-Interval Futures

If enabled, the platform allows the prosumers to specify start intervals and future end intervals for their offers. To understand the benefit of this, consider two producers $P1$ and $P2$ and a consumer C. Let us assume that during a particular interval (j) $P1$ can provide $10kWh$; while $P2$ can provide $30kWh$ and also has battery storage, which enables it to transfer the net energy across several future intervals. If C needs to consume $30kWh$ in interval j and $10kWh$ in interval $j+1$, then if we use a single interval market, $P2$ may be matched to provide the full amount; however, this means that the demand in interval $j+1$ will not be satisfied.

In a futures market, if the offer of $P2$ was valid for $j+1$, then the first trade for j will use only $20kWh$ from $P2$, leaving $10kWh$ for the next interval, maximizing the energy transferred. The challenge of a futures market though is the increased optimization complexity. TRANSAX is able to handle it because we separate the solver from the smart contract.

Security Concerns

Blockchain-based markets prevent some of the cyber threats as the distributed nature of the system prevents a single point of failure. Thus, an adversary would need more resources to spread false prices as shown for a non-blockchain system in [18]. Further, the authentication of prosumers prevents some false data injection attacks. Moreover, authentication and auditability create some accountability in the market; hence, prosumers may adopt better security practices.

In practice, IoT devices lack resources that are required for participating in the computing-intensive consensus algorithms of many blockchains. Thus, prosumers have to connect to a blockchain-based system through *gateway* nodes, which creates a potential point of failure. For example, an adversary can launch

a *(distributed) denial of service* attack against a gateway node to prevent a set of bids from arriving at the market, changing the market's equilibria.

Delays in buyers' bids can also benefit the adversary because missing bids may lead to overestimation of the unresponsive loads. In other words, the DSO may assume that the prosumers who do not submit bids may accept any price. In such cases, the demand curve changes reflecting a higher willingness to pay for energy, which raises the prices.

To mitigate this, when a prosumer submits an offer, it can re-submit the bid to another gateway if it does not receive a confirmation within the expected time frame. The amount of time that a prosumer should wait to submit a bid depends on how frequently the blockchain blocks are generated. Another method to reduce the effectiveness of these attacks is to submit to gateways selected at random, so that the adversary has less control over which offers are dropped as shown in [19].

Correctness Concerns

We use VeriSolid [10] to develop the TRANSAX smart contract. VeriSolid is an end-to-end, open-source framework for the *correct-by-design development and deployment of multiple interacting smart contracts* for blockchain-based CPS. VeriSolid helps developers to eliminate errors early at design time by raising the abstraction level and providing automated verification and code generation

The VeriSolid verification approach can detect typical vulnerabilities, but it may also detect any violation of the required properties. In principle, a contract vulnerability is a programming error that enables an attacker to use a contract in a way that was not intended by the developer. To detect atypical vulnerabilities, developers must specify the intended behavior of a contract. VeriSolid enables developers to specify intended behavior in the form of safety and liveness properties, which capture important security concerns. Properties established at any step of the VeriSolid design flow are preserved in the resulting smart contracts, guaranteeing their correctness.

For example, in the TRANSAX smart contract, we checked that the `postSellingOffer` or `postBuyingOffer` cannot happen for an interval that has been finalized. We also checked that a new prosumer can only be registered if the TRANSAX is in setup mode and during this mode, all trading is halted.

Example

To illustrate effectiveness, we developed a closed-loop simulation (see Fig. 2) using OPAL-RT, a high-fidelity real-time power systems simulator. The case study has 10 feeder lines, passive loads, and four prosumers. Though not shown in the figure, the prosumer software and the TRANSAX software run separately on a cluster of BeagleBones and interact in real-time with the simulator.

Prosumers made offers, represented by the faded bars, for each interval. TRANSAX then found energy trade solutions for each interval, represented by the opaque bars, which resulted in overall mitigation of the load on the DSO (the remaining load is the gap between the offer and the actual trade). When matching offers to find trades, the solvers find solutions for many future intervals. This improves resilience to solver failure. Additionally, since the goal of the solver is to maximize the total energy traded, the solvers re-solve when new offers are posted.

Fig. 2 also shows how the trades evolved for interval 47 (chosen as an example). The magnitude did not change because no new offers were posed for interval 47 after the solver began matching offers (in interval 43) and because the posted offers were valid only for interval 47, eliminating the potential for shifting trades to a later interval. However, since new trades for other future intervals were added (not shown) the total energy traded continued to increase. This is why new solutions were accepted and the trade composition evolved, *i.e.*, the contribution of prosumer 101 decreased and replaced by prosumer 102. We also note that production exceeds consumption after interval 40. Since the consumption does not again exceed production in this example, the stored energy does not make a difference in improving trading efficiency in future intervals. However, readers can refer to [20] for an example of this.

Scalability

The scalability of TRANSAX is limited by the number of transactions that the distributed ledger supports, as well as the complexity of the multiple

solvers that are integrated into the market. The optimization complexity is determined by the number of feeders and the number of intervals that the platform looks into the future while matching trades. The largest system processed by TRANSAX is a 102-home community as described in [20]. The maximum time taken by solvers was less than 5 seconds to solve for the whole system during peak production. The increasing solver time is the result of increasing problem complexity, which is correlated with the number of variables and constraints in a problem, which in turn correlates with the number of selling offers.

Conclusion

Electricity markets based on blockchains inherit some desired properties, such as decentralization, robustness, and security (authentication, data integrity, and auditability). However, the characteristics of blockchains and the requirements of electricity markets also create significant challenges including privacy, computation efficiency, and integration concerns. Security and correctness concerns also exist.

In this paper, we described our solution called TRANSAX for implementing TES. It integrates external solvers to reduce the computation load on smart contracts. The consensus algorithm is limited to the verification of trades calculated by external solvers, which means that prosumers can participate in the market with minor adjustments to their transactive technologies. This is important because most prosumer IoT devices or smart appliances have limited resources.

The ability to support multiple external solvers also improves the system reliability and enables the prosumers to post offers for a range of future intervals. This improves trading efficiency when compared to typical markets. We provide privacy by using the concept of tradeable and mixable energy assets. The integration and correctness concerns are handled by a middleware called RIAPS and formal design tool we have developed called VeriSolid.

In the future, we plan to continue the assessment of the scalability of this decentralized market and analyze potential vulnerabilities to cyber-attacks.

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 numbers CNS-1647015, CNS-1818901, and CNS-1840052 and the Advanced Research Projects Agency-Energy (ARPA-E), U.S. Department of Energy, under Award Number DE-AR0000666. The views presented in this paper are those of the authors and do not reflect the opinion or endorsement of ARPA-E, Siemens, CT and NSF.

REFERENCES

1. B. Römer, P. Reichhart, J. Kranz, and A. Picot, "The role of smart metering and decentralized electricity storage for smart grids: The importance of positive externalities," *Energy Policy*, vol. 50, pp. 486–495, 2012.
2. O. Dag and B. Mirafzal, "On stability of islanded low-inertia microgrids," in *2016 Clemson University Power Systems Conference (PSC)*. Clemson, SC, USA: IEEE, March 2016, pp. 1–7.
3. F. A. Rahimi and A. Ipakchi, "Transactive energy techniques: closing the gap between wholesale and retail markets," *The Electricity Journal*, vol. 25, no. 8, pp. 29–35, 2012.
4. D. J. Hammerstrom, R. Ambrosio, T. A. Carlon, J. G. DeSteese, G. R. Horst, R. Kajfasz, L. L. Kiesling, P. Michie, R. G. Pratt, M. Yao *et al.*, "Pacific Northwest GridWise™ testbed demonstration projects; Part I. Olympic Peninsula project," Pacific Northwest National Lab (PNNL), Tech. Rep., 2008.
5. L. Orsini, S. Kessler, J. Wei, and H. Field, *How the Brooklyn Microgrid and TransActive Grid are paving the way to next-gen energy markets*.
6. A. Wörner, A. Meeuw, L. Ableitner, F. Wortmann, S. Schopfer, and V. Tiefenbeck, "Trading solar energy within the neighborhood: field implementation of a blockchain-based electricity market," *Energy Informatics*, vol. 2, 2019.
7. Y. Du, H. Tu, S. Lukic, A. Dubey, and G. Karsai, "Distributed microgrid synchronization strategy using a novel information architecture platform," in *2018 IEEE Energy Conversion Congress and Exposition (ECCE)*. IEEE, 2018, pp. 2060–2066.

8. C. Yang, A. A. Thatte, and L. Xie, "Multitime-scale data-driven spatio-temporal forecast of photovoltaic generation," *IEEE Transactions on Sustainable Energy*, vol. 6, no. 1, pp. 104–112, 2014.
9. M. Andoni, V. Robu, D. Flynn, S. Abram, D. Geach, D. Jenkins, P. McCallum, and A. Peacock, "Blockchain technology in the energy sector: A systematic review of challenges and opportunities," *Renewable and Sustainable Energy Reviews*, vol. 100, pp. 143–174, 2019.
10. A. Mavridou, A. Laszka, E. Stachtiari, and A. Dubey, "VeriSolid: Correct-by-design smart contracts for Ethereum," in *23rd International Conference on Financial Cryptography and Data Security (FC19)*, February 2019.
11. H. Tu, Y. Du, H. Yu, A. Dubey, S. Lukic, and G. Karsai, "Resilient information architecture platform for the smart grid (RIAPS): A novel open-source platform for microgrid control," *IEEE Transactions on Industrial Electronics*, pp. 1–11, 2019.
12. H. Chen, M. Pendleton, L. Njilla, and S. Xu, "A survey on Ethereum systems security: Vulnerabilities, attacks and defenses," *arXiv preprint arXiv:1908.04507*, 2019.
13. L. Luu, D.-H. Chu, H. Olickel, P. Saxena, and A. Hobor, "Making smart contracts smarter," in *23rd ACM Conference on Computer and Communications Security (CCS)*. New York, NY, USA: Association for Computing Machinery, 2016, p. 254–269.
14. A. Brenzikofer, A. Meeuw, S. Schopfer, A. Wörner, and C. Dürr, "Quartierstrom: A decentralized local P2P energy market pilot on a self-governed blockchain," in *25th International Conference on Electricity Distribution*, June 2019.
15. N. U. Hassan, C. Yuen, and D. Niyato, "Blockchain technologies for smart energy systems: Fundamentals, challenges, and solutions," *IEEE Industrial Electronics Magazine*, vol. 13, no. 4, pp. 106–118, 2019.
16. S. Eisele, A. Laszka, A. Mavridou, and A. Dubey, "SolidWorx: A Resilient and Trustworthy Transactive Platform for Smart and Connected Communities," in *2018 IEEE International Conference on Blockchain (Blockchain 2018)*, Halifax, Canada, Jul. 2018.
17. T. Ruffing, P. Moreno-Sanchez, and A. Kate, "CoinShuffle: Practical decentralized coin mixing for Bitcoin," in *19th European Symposium on Research in Computer Security (ESORICS)*, M. Kutyłowski and J. Vaidya, Eds. Cham: Springer International Publishing, 2014, pp. 345–364.
18. C. Barreto and X. Koutsoukos, "Attacks on electricity markets," in *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Sep. 2019, pp. 705–711.
19. C. Barreto, T. Eghtesad, S. Eisele, A. Laszka, A. Dubey, and X. Koutsoukos, "Cyber-attacks and mitigation in blockchain based transactive energy systems," to appear in 3rd IEEE International Conference on Industrial Cyber-Physical Systems (ICPS 2020), 2020. [Online]. Available: http://aronlaszka.com/papers/barreto2020cyber.pdf
20. A. Laszka, S. Eisele, A. Dubey, G. Karsai, and K. Kvaternik, "TRANSAX: A blockchain-based decentralized forward-trading energy exchanged for transactive microgrids," in *2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)*, Dec 2018, pp. 918–927.

Scott Eisele is a graduate student in electrical engineering department at Vanderbilt University. His research interests are in cyber-physical systems, and distributed computing. He completed an undergraduate degree in Mechanical Engineering at Brigham Young University, Utah. He is member of the IEEE. Email: scott.r.eisele@vanderbilt.edu

Carlos Barreto is a postdoctoral scholar at Vanderbilt University. His research interests include security and resiliency of cyber-physical systems, risk analysis, and game theoretic analysis of security problems. He received the Ph.D. in computer science from the University of Texas at Dallas. He is member of the IEEE. Email: carlos.a.barreto@vanderbilt.edu

Abhishek Dubey is an Assistant Professor of Electrical Engineering and Computer Science at Vanderbilt University, Senior Research Scientist at the Institute for Software-Integrated Systems. His research is in the area of resilient cyber-physical systems with a focus on transportation and electrical networks. He received his PhD in electrical Engineering from Vanderbilt University. He is a senior member of IEEE. Email: abhishek.dubey@vanderbilt.edu

Xenofon Koutsoukos is a professor with the department of electrical engineering and computer science and a senior research scientist with the Institute for Software Integrated Systems, Vanderbilt University. His research work is in the area of cyber-physical systems with emphasis on security and resilience, control, diagnosis and fault tolerance, formal methods, and adaptive resource management. He received the Ph.D. degree in electrical engineering from the University of Notre Dame. He is a Fellow of the IEEE. Email: xenofon.koutsoukos@vanderbilt.edu

Taha Eghtesad is a graduate student in the Department of Computer Science at the University of Houston. His research interests are in cybersecurity and artificial intelligence. He has a B.Sc. in Computer Engineering from Shahid Beheshti University. Email: teghtesad@uh.edu

Aron Laszka is an Assistant Professor in the Department of Computer Science at the University of Houston. His research interests revolve around cyber-physical system, cybersecurity, and applications of artificial intelligence. Previously, he was a Research Assistant Professor at Vanderbilt University from 2016 to 2017, and a Postdoctoral Scholar at the University of California, Berkeley from 2015 to 2016. He graduated summa cum laude with a Ph.D. in Computer Science from the Budapest University of Technology and Economics in 2014. Email: alaszka@uh.edu

Anastasia Mavridou is a Computer Scientist in the Robust Software Engineering group at the NASA Ames Research Center, employed by KBR. Her research interests lie in the area of model-based design, system analysis and verification with a focus on correct-by-construction techniques. Previously, she was a Postdoctoral Scholar at Vanderbilt University. Anastasia received her Ph.D. in Computer Science from École Polytechnique Fédérale de Lausanne (EPFL), Switzerland. Email: anastasia.mavridou@nasa.gov