frontiers in Blockchain

ORIGINAL RESEARCH
published: 21 September 2021
doi: 10.3389/fbloc.2021.725322

Analysis and Application of Verifiable Computation Techniques in Blockchain Systems for the Energy Sector

Andreas Zeiselmair [1,2], Bernd Steinkopf [1,3], Ulrich Gallersdörfer [3], Alexander Bogensperger [1,2] and Florian Matthes [3]*

[1]FfE, Munich, Germany, [2]TUM Graduate School, Technical University of Munich, Garching/Munich, Germany, [3]Department of Informatics, Software Engineering for Business Information Systems, Chair of Informatics 19, Technical University of Munich, Garching/Munich, Germany

OPEN ACCESS

Edited by:
Andreas Unterweger, University of Applied Sciences Salzburg, Austria

Reviewed by:
Günther Eibl, University of Applied Sciences Salzburg, Austria
Kaiwen Zhang, École de technologie supérieure (ÉTS), Canada
Daria Musikhina, University of Passau, Germany

***Correspondence:**
Andreas Zeiselmair
andreas.zeiselmair@tum.de

Specialty section:
This article was submitted to Blockchain for Good, a section of the journal Frontiers in Blockchain

Received: *15 June 2021*
Accepted: *06 September 2021*
Published: *21 September 2021*

Citation:
Zeiselmair A, Steinkopf B, Gallersdörfer U, Bogensperger A and Matthes F (2021) Analysis and Application of Verifiable Computation Techniques in Blockchain Systems for the Energy Sector. Front. Blockchain 4:725322. doi: 10.3389/fbloc.2021.725322

The energy system is becoming increasingly decentralized. This development requires integrating and coordinating a rising number of actors and small units in a complex system. Blockchain could provide a base infrastructure for new tools and platforms that address these tasks in various aspects—ranging from dispatch optimization or dynamic load adaption to (local) market mechanisms. Many of these applications are currently in development and subject to research projects. In decentralized energy markets especially, the optimized allocation of energy products demands complex computation. Combining these with distributed ledger technologies leads to bottlenecks and challenges regarding privacy requirements and performance due to limited storage and computational resources. Verifiable computation techniques promise a solution to these issues. This paper presents an overview of verifiable computation technologies, including trusted oracles, zkSNARKs, and multi-party computation. We further analyze their application in blockchain environments with a focus on energy-related applications. Applied to a distinct optimization problem of renewable energy certificates, we have evaluated these solution approaches and finally demonstrate an implementation of a Simplex-Optimization using zkSNARKs as a case study. We conclude with an assessment of the applicability of the described verifiable computation techniques and address limitations for large-scale deployment, followed by an outlook on current development trends.

Keywords: verifiable computation, blockchain, energy, peer-to-peer energy markets, zero knowledge proof (ZKP), trusted oracles, multi-party computation (MPC), ZkSNARKs

Abbreviations: DHT, Distributed Hash-Table; EEG, Renewable Energy Act (German: *Erneuerbare Energien Gesetz*); EPID, Enhanced Privacy Identifier; FHE, Fully Homomorphic Encryption; GO, Guarantees of Origin; GOR, Guarantees of Origin Register; LP, Linear Program; LFM, Local Flexibility Markets; MAC, Message Authentication Code; MPC, Multi-Party Computation; R1CS, Rank-1-Constraint-System; REC, Renewable Energy Certificates; sMPC, secure Multi-Party Computation; VC, Verifiable Computing; ZKP, Zero-Knowledge Proof; zkSNARK, Zero-Knowledge Succinct Non-Interactive Argument of Knowledge

1 INTRODUCTION AND MOTIVATION

The ongoing decentralization in the energy sector is driven by an increasing number of small-scale renewable energy plants. Focusing on the electricity sector, this trend of increasing granularity is reinforced by additional numbers of local energy consumers like heat pumps or electric vehicles. This is resulting in a growing field of additional market actors that need to be integrated and coordinated within the energy system. This means a significant change with regard to the historic development of the energy system. Starting with big power plants connected to the higher voltage levels of the electricity grid and its strictly defined top-down supply direction, it is necessary to change the system towards increased interconnectivity and optimized energy use under consideration of a variety of boundary conditions. New tools and platforms are already established and in further development that address these new tasks (Kloppenburg and Boekelo, 2018; Duch-Brown and Rossetti, 2020). Aggregators bundle the dispatch and load capacity of smaller energy units and optimize their usage depending on market signals. Grid operators need to forecast and dynamically adapt to the grid load depending on complex load flow calculations. The usage of flexibility is therefore a key factor to react to local load and generation peaks that need to be efficiently allocated through future local flexibility markets (LFM) (Villar et al., 2018; Bouloumpasis et al., 2019). These are only two examples of complex computational tools that are already established or in development. Recent advancements towards (close to) real-time peer-to-peer energy trading further elaborate these tendencies. Another relevant use case in this context is a detailed labeling and tracing of renewable energy certificates as a specific form of supply chain transparency. Since this forms a central research topic within the project InDEED,[1] the implementation example in **Section 4** is also based on this case study. As the fundamental enabler and main driver of these developments, increasing digitalization of the energy system provides improved data availability and direct asset control combined with increased computational power. On the other hand, there is also a trend of decentralization in the information technology sector. In order to gain data-sovereignty, resilience, and individual autonomy, distributed data networks come into consideration for the design of energy data management processes. Blockchain, as the most prominent representative of distributed ledger technologies (DLT), can provide a base infrastructure, enabling the transparent and tamper-resistant foundation for coordination schemes involving a large number of independent actors. By introducing smart contracts, blockchain technology has been elevated from being a mere digital currency to providing secure and trusted general-purpose computations which can be verified by anyone (Buterin, 2014). Smart contract platforms have found widespread adoption in many industry sectors, e.g., supply-chain management, finance, healthcare, and energy among others (Casino et al., 2019).

1.1 Need for Optimization in (Future) Energy Market Environments

As already mentioned, the increasing complexity and also the availability of sophisticated tools offers new opportunities for efficient energy management. The increasing number of new and small actors with access to energy markets opens the space for new market concepts in addition to traditional market places (Zeiselmair and Bogensperger, 2021). Additional knowledge and specifications also lead to the potential enrichment with quality features. Leading the energy market away from pure commodity trading towards the consideration of further specifications enables several business opportunities. Relatively new to energy markets, Roth (2015) observed and described these types of markets as "matching markets" in several business sectors.

Finally, this development also evolves the mainly used merit-order approach of energy allocation in most existing energy markets (i.e., energy spot or balancing markets Kirschen and Strbac (2005)) towards optimization-based market places. Even though alternative options exist, an optimization approach provides several advantages. Depending on the use case, alternatives range from heuristic algorithms (Heilmann et al., 2021) to iterative or multi-step negotiations (Morstyn et al., 2019; Sorin et al., 2019) within the market frameworks. Yet, these show significant drawbacks regarding the need to provide detailed (potentially sensitive) data (i.e., network topologies), an increase in computational effort, or additional communication and needed interaction between platform actors. Especially in short-term markets the latter aspects can be decisive to an efficient implementation.

Figure 1 gives a generic overview of functions and interactions in these market types, based on optimization as allocation logic.

There are already several examples of those optimization-based energy markets discussed and currently in development or pilot phase, with the following examples being the most prominent representatives:

- *P2P energy markets* represent an idealized form of energy trading. Prosumers—as a combination of consuming and producing market participants—can procure and divert locally generated energy (e.g., through photovoltaics). Further, they are able to potentially market their flexibility (Kubli et al., 2018). Instead of a central entity managing the energy distribution, all participants trade their energy directly. In the proposed setting, this is often enabled by secure smart-meters and a blockchain-based auction platform. With these tools, communities would become completely independent from commercial energy suppliers, as they could market and purchase their own energy locally (Long et al., 2018; Jogunola et al., 2020). Considered communities can be both virtual or organized in micro-grids.

[1]InDEED (www.ffe.de/indeed) is funded by the Federal Ministry for Economic Affairs and Energy (BMWi) (funding code 03E16026A).

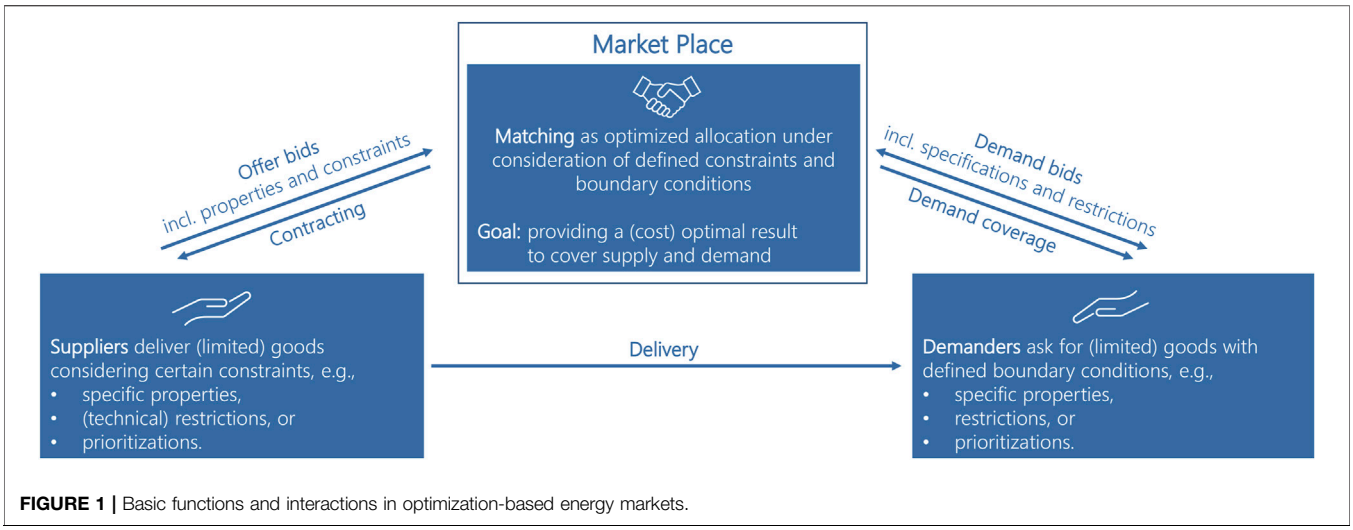

FIGURE 1 | Basic functions and interactions in optimization-based energy markets.

- *Local flexibility markets (LFM)* are recently discussed, new tools to access regional flexible power for grid congestion management. The network-supportive use of flexibility provided by decentralized energy units is allocated to relieve network congestions via a complex matching algorithm. This considers boundary conditions regarding specific effectivity (i.e., the specific impact of provisioned flexibility to a congested grid element) and technical constraints, e.g., time restrictions or call levels depending on type of flexible asset (Jin et al., 2020; Heilmann et al., 2021; Zeiselmair and Köppl, 2021).
- *Energy labeling*, the allocation of renewable energy certificates (REC) and the associated guarantees of origin (GO) represent an increasingly relevant subject in energy business models. Within Germany, the rising number of renewable energy plants reaching the end of their 20 years funding period according to the Renewable Energy Act (EEG) makes these plants available for issuing GOs and markets them as "regionally generated renewable energy." Efficiently allocating these energy volume and new business models under potential consideration of regional vicinity is currently subject to research (Bogensperger and Zeiselmair, 2020).

Many of these new market use cases are closely linked to the value proposition of distributed ledger technologies, such as blockchain. During several studies dealing with the identification of potential blockchain use cases, these were ranked as the most promising (Bogensperger et al., 2018; Hinterstocker et al., 2018; Andoni et al., 2019).

1.2 Added Value Through Blockchain-Based Decentralization in an Increasingly Granular Energy System

Blockchain technology often plays an essential role in the discussion about a base framework for new energy platform solutions or decentralized business models. Due to its unique ability to transparently document the common state of information within a network, it can provide trust between non-trusting parties in a increasingly granular energy system. Indeed, the energy sector is already one of the most rapid adopters of blockchain technology (Wu and Tran, 2018). Its distributed nature, high security standards through avoidance of single-point-of-failures combined with shared responsibilities, and calls for more transparency could provide added value to the energy sector's demands compared to centralized system. Potential fields of energy blockchain applications are manifold and have already been discussed in various research projects (Chitchyan and Murkin, 2018; Strüker et al., 2018; Teufel et al., 2019). Hinterstocker et al. (2018) gives a basic overview of promising applications. These include the standardized settlement of electric vehicle charging, faster switching of electricity suppliers, labeling of green electricity, P2P trading (C2C and B2B) and proof of balancing power provision. Within the last years, first commercial use cases have also been implemented. Andoni et al. (2019) have given a comprehensive overview of the current state of blockchain use cases in the energy sector, breaking down individual projects into their components. The authors identified *P2P energy trading* as the primary driving force behind blockchain adoption with one-third of all projects utilizing it in some form. Other energy-specific use cases include *asset management* (11%), *metering & billing* (9%), *grid management* (8%), or *green certificate trading* (7%). Several works suggest the use of public ledgers as a control mechanism for the optimization algorithm so as to eliminate the need for a trusted third party (Munsing et al., 2017; Alskaif and Van Leeuwen, 2019). This approach appears especially promising in the context of P2P energy trading, where no central control entity exists but energy flow still needs to be optimized and reliably provided (Sousa et al., 2019).

Besides the already named potential advantages, the blockchain use is related to certain restrictions and drawbacks when dealing with complex computational operations, as described in the following section.

1.3 Challenges and Potential Solutions to Complex Computation in Energy Blockchain Architectures

Blockchain and smart contracts in particular are a relatively recent invention and as such are plagued by a host of different challenges that make their use in production systems less than ideal.

- Almost all public permissionless blockchains suffer from poor transaction throughput and do not scale with the amount of nodes in the network or the provided computational power (Bez et al., 2019). Compared to traditional systems like Visa processing thousands of transactions per second, Ethereum is only able to process about 15 transactions per second (Herrera-Joancomartí and Pérez-Solà, 2016).
- Even if only a small number of transactions can be facilitated at one point in time, the complexity for executing functions in smart contracts is severely limited as computation is expensive and limited in its execution time due to redundant execution. Storage and computational power is expensive in most networks.
- The transparency of blockchain networks often comes at the cost of missing privacy for transactions and data. To verify the correctness of transactions, anybody must be able to verify them.
- The energy consumption of public blockchains, especially Bitcoin and Ethereum, is very high (Stoll et al., 2019; Gallersdörfer et al., 2020). The PoW consensus mechanism in these networks incentivises miners to spend a large amount of money in participating and generating new blocks.

The energy sector's large scale and critical role in a society's infrastructure demands stringent requirements. Many use cases in the energy sector rely on high-performance calculations on sensitive data, and integrating them into a blockchain environment at this point in time would prove difficult due to the above-mentioned reasons. The blockchain community is aware of these shortcomings in current implementations and has been working on solutions to address them. We distinguish several scalability options related to the respective stack level. One of the most popular and transparent developments is Ethereum's discussion regarding "layer 1" or "layer 2" scalability options (Hafid et al., 2020). Because layer 1 refers to an improvement of the core protocol itself (see *ETH 2.0* in Ethereum), layer 2 options are developed "on top" of the base protocol. "Sharding" represents a currently discussed upgrade of the Ethereum protocol that intends to horizontally split the database and spread it across parallel chains, defined as "shards." Within each shard, state updates are propagated as usual, but communication between shards is limited to a simple synchronization mechanism. This way, shard data can be processed in parallel, significantly increasing the number of transactions per second. Recently, layer 2 options are gaining traction and numerous proposals and development projects emerge. Most of these protocols work with "off-chaining" functions, data, or computation. The validity is still provided by settling relevant proofs or references on the main blockchain in regular intervals or defined incidents. Popular approaches include zero-knowledge or optimistic rollups, state channels or micro-channels (e.g., *Plasma* (Poon and Buterin, 2017)).[2]

While these approaches focus on mere scalability in terms of transaction speed, the issue of providing complex computation in blockchain environments is only partly addressed by the proposed solutions. Also, considerably fewer solutions address the problem of privacy-preserving smart contracts. Classical techniques from cryptography, e.g., asymmetric encryption or hash- and reveal-schemes, can be used to secure private contract data, but their applications are limited and often require extensive key infrastructures already in place. Therefore, many critical industries opt instead to use private blockchains with restricted access to store their sensitive data (Kuo and Ohno-Machado, 2018). While this is a valid approach, it diminishes many of the initial advantages of the original blockchain idea. Open participation and distributed consensus are the very elements which give blockchains their strong security model.

Recently, a comprehensive approach has emerged, which tries to combine both performance and privacy by outsourcing sensitive data and extensive computational effort. The ideas of *verifiable computing (VC)* are relatively old but their particular application to blockchain environments has once again brought them to the attention of researchers worldwide. Instead of merely aggregating transactions, the entire smart contract is executed offline and its results are published. A verification algorithm subsequently ensures that the result was computed correctly. While not strictly defined that way, most VC schemes also keep the contract's data private during this process. Thus, verifiable computing seems like a suitable tool to bring the security of blockchain to the various sensitive and demanding uses cases of the energy sector by providing added value through increased transparency. As in the currently available literature, a compressed overview of VC techniques and—in specific—their application in energy blockchain architectures is still missing, our contribution provides an extensive literature review combined with an assessment in this field. Further, we introduce an implementation proposals as proof-of-concept. Within this paper, we address the following research questions:

- **RQ1** Which applicable verifiable computation techniques are available?
- **RQ2** How can these techniques be evaluated regarding key features and criteria?
- **RQ3** What added value can they provide in the application to optimization-based energy markets?
- **RQ4** How do they prove in practical implementation?

The paper is outlined as follows. In **Section 2**, we provide an overview on verifiable computation technologies, while in **Section 3** we evaluate the respective technologies. **Section 4**

[2]https://ethereum.org/en/developers/docs/layer-2-scaling/.

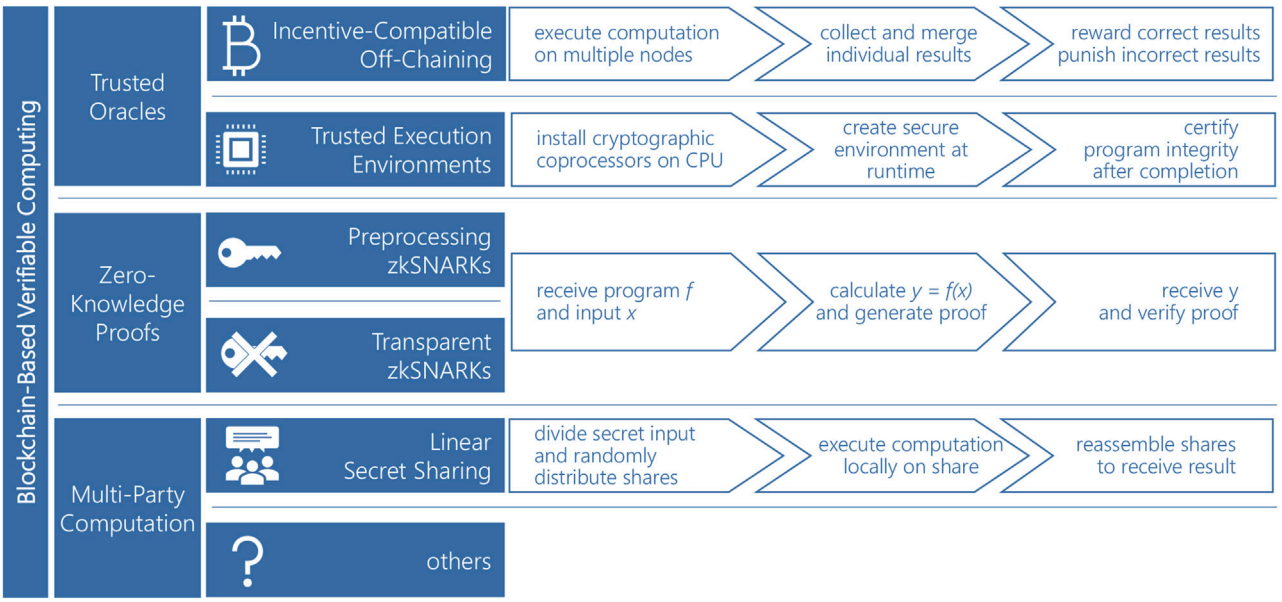

FIGURE 2 | Overview and function of presented verifiable computing techniques.

explains the application to optimization-based energy markets. We conclude the paper in **Section 5**.

2 OVERVIEW ON VERIFIABLE COMPUTATION TECHNOLOGIES AND THEIR APPLICATION IN BLOCKCHAIN ENVIRONMENTS

Verifiable Computation describes a method to outsource computations to external nodes with the ability to verify the correctness of the returned results. Within this paper, we further open the concept of VC to technologies that enable outsourcing expensive work from computationally weak nodes in a secure manner in general. Even though, by closer inspection not all aspects of verifiability can be provided by each portrayed technology, focus also lies on the applicability within the considered use cases. Further, our initial selection is based on the availability of sufficient (research) material and the relative maturity of these approaches. In the following, we focus on three relevant and emerging options of VC techniques, which are already in use or undergoing intensive further development (see **Figure 2**). Namely, these are "trusted oracles" as an already commercially applied option, "zkSNARKS"[3] as a promising and already applicable form of zero knowledge proofs, and "multi-party computation" as the most complex approach of this selection.

2.1 Trusted Oracles

One of the greatest limitations of smart contracts is their fully deterministic nature. When a contract's code is invoked, a majority of the blockchain's participants must agree on its result. This is a necessary requirement of the underlying consensus algorithm. If smart contracts were allowed to yield a randomized result for every participant, such a majority could not be guaranteed. Perhaps counterintuitively, this absence of non-determinism is not always ideal. A consequence of particular significance is the fact that smart contracts cannot access data external to the blockchain, e.g., public APIs.[4] Many potential blockchain use-cases are handicapped by this restriction, since they cannot depend on data from outside sources. Blockchain oracles (Al-Breiki et al., 2020) aim to mitigate this problem by on-chaining the data in a way that can be reconciled by the consensus algorithm.

All oracle implementations must contain at least one dedicated smart contract so that they can interact with the blockchain. This contract serves as a central hub where outside information is gathered and forwarded to other smart contracts. To inject data into the system, it must first be retrieved from the outside source and subsequently included in a transaction. The transaction is then sent to the oracle contract, which either stores it in a public field or passes it on to any interested parties. Packaging and transferring the data can either be done manually by a human operator or by an automated script. There are multiple implementation-specific ways to trigger an injection. Common ones include other smart contracts directly

[3]Zero-Knowledge Succinct Non-Interactive Argument of Knowledge.

[4]This is because the non-deterministic behavior of network calls cannot be uniformly processed by the consensus algorithm.

requesting the data, time-based injections, event-based injections, etc.

The simple oracle architecture we have provided above is already capable of successfully on-chaining outside data. Nonetheless, it suffers from one major flaw. A smart contract's main advantage over traditional programs is the guarantee that its code will be executed exactly as advertised. Users can rely on the blockchain's consensus algorithm to identify and remove any incorrect results. With outside data, however, the concept of a "correct result" is not clearly defined, since it appears as a black box to on-chain contracts. Hence, the consensus algorithm can only verify that an injection occurred and not the data itself. In the literature, this observation is referred to as the *Oracle Problem* (Egberts, 2019). The Oracle Problem makes naive implementations an easy target for manipulation attempts.

In practice, several approaches exist that endow oracles with a notion of trust akin to the one presented by blockchains. They range from simple redundancy checks to more sophisticated incentive schemes and even the use of advanced cryptographic hardware. Because trusted oracles can check the correctness of their data before it is sent to the blockchain, they make an excellent tool for verifiable computations (Heiss et al., 2019). Users can simply transmit their computation demand to an oracle service, where they are executed off-chain. After the computation has finished, the service provider on-chains the result via the oracle's contract. The trust mechanism employed by the oracle service lets users know that the result is genuine. As long they can accept the correctness of the mechanism, they can also accept the result. Within the domain of "trusted oracles," we distinguish two major approaches, "incentive-driven off-chain computation" and "software guard extensions."

Incentive-driven off-chain computation (IOC) (Eberhardt and Heiss, 2018; Heiss et al., 2019) is an approach to oracle-based off-chaining which has its origins in game theory. Instead of employing cryptographic techniques, an IOC system relies on reward and punishment rules to establish the correctness of computational results. In essence, the same computation is replicated over a number of participating oracle nodes. When all nodes have finished their work, they then vote on the correct outcome by submitting their result. The majority vote is eventually accepted as the sole solution. As their name implies, IOC systems employ an incentive scheme to enforce truthful behavior. Participants who publish the majority result receive rewards, which are usually monetary in nature. Minority voters either receive nothing or might even be punished for their failure to comply with the protocol. This incentive-driven voting game is based on many of the same principles found in the proof-of-work consensus algorithm used by early blockchain implementations (Nakamoto, 2008).

The security of such a system always rests on the assumption that there is an honest majority who will vote truthfully. To ensure the existence of an honest majority, the chosen reward scheme must be *incentive compatible*. This game theoretic term describes any system where participants must act in accordance with their true preference if they wish to maximize their profits. In an IOC setting most participating oracles have no personal preference for the outcome of a given computation, but they will instead answer truthfully, as this gives them the highest chance of being part of the majority. This is the case since each participant will naturally believe their result to be correct and they have no reason to assume that the other participants will answer incorrectly. As long as the network stays large enough, any party intentionally trying to manipulate the outcome of a computation will therefore be dwarfed by the votes of the honest players.

Software Guard Extensions (SGX) is a collection of cryptographic co-processors developed and introduced by Intel (Hoekstra et al., 2013; McKeen et al., 2013; Costan and Devadas, 2016). It provides a secure container where application code can run in isolation from the rest of its host system. To achieve this, SGX enforces code confidentiality and integrity at a hardware level during runtime. Not even privileged software, such as the OS or the BIOS, can access the program's memory contents. Application developers can use SGX to implement strong security mechanisms without needing access to advanced knowledge in cryptography. In the blockchain space, SGX is already a *de facto* standard and can be found in popular applications such as Hyperledger Sawtooth (Olson et al., 2018) and many trusted oracles.

2.2 zkSNARKs

Can we prove knowledge of a fact without revealing it? This is the fundamental problem which zero-knowledge proofs (ZKPs) seek to solve. Interestingly enough, the very concept of this technology dates all the way back to the 80s, where Goldwasser et al. (1989) gave the following definition:

"A zero-knowledge proof . . .

1. . . . must convince the verifier that the prover indeed knows the fact,
2. . . . may not be forged by a malicious prover without knowledge of the fact,
3. . . . may not allow the verifier to obtain knowledge of the fact."

Informally, these three properties have become known as *Completeness*, *Soundness*, and *Zero-Knowledge*, respectively.

zkSNARKs represent a special type of ZKPs, specifically designed as a cryptographic primitive for VC. It stands for **Z**ero-**K**nowledge **S**uccinct **N**on-interactive **AR**gument of **K**nowledge. This acronym expands on the initial definition of zero-knowledge proofs as: The proofs generated by a SNARK are short (succinct). Proof verification happens independently of the prover (non-interactive). An honest prover generating a valid proof will always be able to convince a verifier of its correctness (argument or soundness). Knowledge of a particular fact (witness) is included in the statement.

They combine state-of-the-art correctness arguments for mathematical functions with traditional zero-knowledge techniques by applying techniques used in existing non-interactive zero-knowledge proofs to hide the value of the prover's witness. Consequently, a verifier is only able to certify the inclusion of the witness in the requested computation, but remains incapable of its reconstruction. This grants users not only the ability to publicly verify arbitrary computations, and also lets

the provers hide any secret values that were used during said computation. Despite being a relatively young technology, great advancements in the realm of zkSNARKs have been made within the last decade. After a period of relative silence around the state of zero-knowledge proofs, their development has recently regained traction with the introduction of *zkSNARKS* (Gennaro et al., 2013). This new class of zero-knowledge proof supports additional functionality that is highly desirable for blockchain applications. Initially employed by Zcash (Hopwood et al., 2016) to hide transaction details for their cryptocurrency, zkSNARKs find increasingly widespread use in other blockchain projects. Arguably, their most promising application is the planned introduction of verifiable computing in Ethereum 2.0. As a foundation for the so-called "ZK-Rollups,"[5] the Ethereum developers hope to vastly increase their transaction throughput by placing expensive computations outside the chain. In the following, we take up this technique for our own verifiable computation approach (see **Section 4.3**).

2.3 Multi-Party Computation

In an ideal world, we would like our computations to be 1) correct, 2) private, and 3) include no single point of failure. *Multi-Party Computation (MPC)*[6] promises such an ideal world. The technology was first mentioned in the context of Yao's Millionaires' problem (Yao, 1982), where two millionaires envision a protocol to determine which is richer without divulging their real wealth. In a more general sense, MPC refers to a wide variety of different security protocols with the same core principle: The participants provide secret inputs, which are hidden using secure obfuscation schemes, and jointly execute operations on these values without revealing them. At the end of the protocol, its final output is made public. By looking at certain cryptographic artifacts created during runtime, the participants can later verify that the computation was indeed executed correctly. Throughout MPC's short history, most implementations have been realized using at least one of the following four concepts (Zhong et al., 2019):

- Garbled Circuits (Yao, 1982),
- Oblivious Transfer (Rabin, 1981),
- Linear Secret Sharing (Shamir, 1979),
- Fully Homomorphic Encryption (Gentry, 2009).

Of the listed technologies, linear secret sharing finds the most use in real-world applications (Cohen et al., 2013; Noyes, 2016; Bai et al., 2019; Sharma and Ng, 2020). Yao's original garbled circuits technique was only designed to serve two parties but it has since been extended to the *n*-party case. Nonetheless, garbled circuits exhibit poor performance at scale (Noyes, 2016), making them inferior to other MPC schemes. Oblivious transfer, on the other hand, is rarely used by itself and can instead be found in many existing protocols (Cohen et al., 2013; Bai et al., 2019; Sharma and Ng, 2020), as a way to secure communication channels. Lastly, fully homomorphic encryption (FHE) is an ongoing research effort that has gained much traction in recent years. In theory, FHE enables arbitrary computations on encrypted user data, preserving the applied operations when the data is finally decrypted. While this already sees some use within certain limits, current performance remains poor (Damgård et al., 2012; Zyskind et al., 2016).

Most real-world MPC systems rely on linear secret sharing as their main cryptographic protocol. The most prominent example of this is the *Enigma* platform (Zyskind et al., 2016; Zyskind, 2019). Enigma leverages the properties of Shamir's Secret Sharing (SSS) and SPDZ to build a privacy-preserving execution engine for smart contracts. SSS presents one of the earliest linear secret-sharing algorithms, which is still in use today (Shamir, 1979). The scheme, which was originally intended for secure storage of cryptographic keys, is also suitable for MPC because of its homomorphic properties. Specifically, the chosen representation of the secret-shared information exhibits an additive and multiplicative homomorphism (Zyskind et al., 2016). The sharing algorithm is founded on the principles of polynomial interpolation. SPDZ (Damgård et al., 2012) is a relatively recent multi-party protocol for secure general-purpose computations based on additive secret sharing. It supports arbitrary addition and multiplication operations. Further, SPDZ promises significant performance increases over earlier MPC implementations.

The protocol is capable of handling inputs from multiple joint parties, but always yields a single aggregated computation result. Authenticity of the inputs and any intermediate values obtained during execution is guaranteed by an embedded cryptographic MAC. In combination with a classical blockchain, this gives users the ability to have contracts with both private and public portions. The authors of Noyes (2016) introduced *Pandora* as an extension to Enigma. They adopted large portions of the computation engine, but made several contributions to the underlying network code, i.e., adding a "feed-forward execution loop for the pruning of (compute) nodes in individual securely private computational rounds." This further increases the performance of outsourced computations. Another system using linear secret sharing but with a different methodology is *Keep* (Luongo and Pon, 2017). It provides a distributed market for verifiable computations where buyers and sellers of computational power come together. Designed from the ground up as a black box service, Keep's main goal is ease of use for its clients. As such, clients may purchase execution time on worker nodes as abstract execution containers (the eponymous "keeps"). When a purchase occurs, a new keep is spawned for this specific instance of the computation. An underlying SPDZ protocol ensures that the computation is executed securely and privately. As for implementations of other MPC schemes, the authors of Pei et al. (2019) suggest an unnamed system based on oblivious transfer and homomorphic encryption, but their work is still academic and very high-level. A more practical alternative on the basis of oblivious transfer and garbled circuits was introduced in Benhamouda et al. (2019). By limiting their application to permissioned blockchains, the authors were able to circumvent performance and privacy challenges encountered by other MPC

[5]https://docs.ethhub.io/ethereum-roadmap/layer-2-scaling/zk-rollups/.

[6]Sometimes called secure Multi-Party Computation (sMPC).

TABLE 1 | Overview of the presented verifiable computation technologies, including pros and cons.

Trusted Oracles	Incentive-compatible off-chaining	+ Transparent aggregation
		+ Native execution performance
		– Slow aggregation
		– No cryptographic guarantees
		– Requires honest majority
	Trusted execution environments	+ Broad availability
		+ Near-native execution performance
		+ Encryption of private data
		– Vendor as trusted third party
		– No access to hardware (e.g., network)
		– Known side-channel attacks
Zero-knowledge proofs	Preprocessing zkSNARKs	+ Cryptographically secure
		+ Fully transparent
		+ Fast verification
		– Limited execution environment
		– Expensive proof generation
		– Requires trusted setup
	Transparent zkSNARKs	+ No trusted setup
		+ Same security as preprocessing variant
		+ Similar verification speeds
		– Immature technology
		– Proof generation even more expensive
		– Higher proof sizes
Multi-party computation	Linear secret sharing	+ Cryptographically secure
		+ Fully transparent
		+ Fully trustless
		– Immature technology
		– Limited execution environment
		– Poor execution performance

systems. Due to the better performance of permissioned environments, necessary data can be stored directly on-chain. Similarly, computations can be run as part of the built-in endorsement phase which replaces the consensus algorithm of permissionless networks. However, since we only consider verifiable computation techniques applicable to public blockchains, we will not pursue this approach further.

To conclude the specifications of the considered verifiable computation technologies presented before, **Table 1** gives an overview including respective advantages and disadvantages.

3 EVALUATION OF SOLUTION APPROACHES

Much literature exists that is specific to each of the three discussed technologies. This includes original specifications, improvement proposals, evaluations, and comparisons. In regard to analyzing blockchain-compatible verifiable computing on an inter-technological level, Eberhardt and Heiss (2018) were able to identify mostly the same categories of verifiable computing but discuss each approach at a much lower level of detail. As an article paper, their work naturally lacks any in-depth explanation of the different theoretical backgrounds and what real-world solutions are currently available. Additionally, the authors do not provide an implementation. Hence, we hope to contribute to their research by herein filling in some of the mentioned gaps. In order to provide a high-level comparison of the presented VC schemes, we developed an evaluation framework which considers the most important requirements for IT-infrastructures in energy applications. These include security, performance, and practicality. In total, there are nine criteria belonging to these three different categories. The evaluation criteria are not universal but were particularly chosen against the background of the introduced field of application within optimization-based energy markets. The evaluation of financial expenses related to the applied mechanisms was not considered as there are no relevant hardware expenses (with exception to SGX which is a standard component of newer Intel processors) and transaction fees are highly use-case specific. Since the fulfillment of many of these goals cannot be measured quantitatively, we employ an analytical approach based on a grading scale. The possible grades are *excellent (++)*, *good (+)*, *average (0)*, *fair (–)*, and *poor (––)*. A quantitative evaluation is only provided within the proof-of-concept implementation as performance metrics are quite use case specific depending on the applied VC technique. As focus lies on the application of the analyzed techniques within a specific field of energy use cases, a partly subjective evaluation is inevitable and therefore justified in our understanding. **Table 2** compares the verifiable computation schemes. The detailed explanation of the chosen criteria and evaluation is described in the following sections.

3.1 Security

Aspects related to information security have been grouped together in the *security* category. The concrete evaluation criteria are taken from (Cherdantseva and Hilton, 2013) and are mostly in line with common IT-security goals encountered in

TABLE 2 | Graded evaluation results of selected VC schemes regarding the following evaluation criteria: ***I**ntegrity*, ***T**ransparency*, ***C**onfidentiality*, ***P**rivacy*, ***T**ransaction **S**peed*, ***M**emory **C**onsumption*, ***M**aturity*, ***U**sability*, and ***E**xtensibility*.

	Security				Performance		Practicality		
	I	**T**	**C**	**P**	**TS**	**MC**	**M**	**U**	**E**
Trusted oracles (IOC)	–	++	––	0	0	++	++	++	++
Trusted oracles (SGX)	0	––	++	0	++	++	++	+	0
Zero-knowledge proof (Prepocessing)	+	++	0	0	0	+	0	+	–
Zero-knowledge proof (Transparent)	++	++	0	0	0	––	–	0	–
Multi-party computation (SPDZ)	++	++	++	+	––	+	–	–	–

other relevant literature on the topic. More specifically, we evaluate an application's ability to

- deny unauthorized modification *(**I**ntegrity)*,
- be monitored by an outside observer *(**T**ransparency)*,
- prevent leaking sensitive data *(**C**onfidentiality)*,
- protect the identities of its users *(**P**rivacy)*.

We choose to omit the other goals listed in Cherdantseva and Hilton (2013) because they are either 1) not deemed relevant to a blockchain environment, or 2) unaffected by the chosen VC scheme. Within (energy) market environments, these claims are of particular interest, as they intend to provide proper market operation through transparent allocation and non-discriminatory competition combined with keeping sensitive data secret.

3.1.1 Integrity

Generally, trusted oracles offer the least security guarantees among the provided solutions. Lo et al. (2020), for instance, analyzed the integrity of current oracle implementations with the help of "reliability scores" derived from fault tree diagrams. All examined oracles achieved scores between 0.99 and 0.93, with 1 being the theoretically achievable maximum of zero successful manipulations. Interestingly, the study found no difference between IOC and SGX oracles, despite the fact that SGX has been shown to be vulnerable to side-channel attacks (Weichbrodt et al., 2016; Wang et al., 2017; Cloosters et al., 2020). Especially timing-based attacks on SGX's cache infrastructure have seen some success in realistic scenarios. These attacks, however, are relatively novel, so widespread usage is not yet common.

Modern zkSNARKs, on the other hand, provide comparatively strong integrity guarantees. More specifically, the mathematical definition states that an adversary's chance of forging a proof shrinks superpolynomially with an increase in argument size. Unsurprisingly, this basic requirement is met by all implementations we examined, as they are all based on the same computational model. Preprocessing zkSNARKs, however, suffer from the fundamental flaw of requiring a trusted setup prior to generating a *common reference string (CRS)*, which ensures the protocol's correctness (Ben-Sasson et al., 2013). An attack targeting the setup directly (e.g., through social engineering) stands a much better chance of subverting the protocol. This problem is completely circumvented by transparent zkSNARKs, as they lack this setup. Transparent zkSNARKs thus remain secure, even in realistic scenarios, where one or more of the trusted setup nodes might be corrupted.

Lastly, SPDZ's message authentication code (MAC) scheme guarantees the integrity of outsourced computations with internal and external adversaries and even in a case where $n-1$ of the n participants act covertly. This is an exceptionally high level of security, since it basically allows anyone to make use of the protocol without having to trust the rest of the network, making SPDZ a completely trustless protocol.

3.1.2 Transparency

In terms of transparency, we find great differences between IOC and SGX. Because most IOC oracles fully exist as smart contracts on the blockchain, their execution paths are fully traceable by an outsider. This highly transparent design is a vital part of the original blockchain idea and leads to a strong notion of trust in the system. SGX, however, requires great trust in the manufacturer as a third party. While the employed cryptographic primitives are open source, they rely on several opaque services provided by a third party to function correctly, e.g., Intel's attestation API or Intel's enhanced privacy identifier (EPID).

zkSNARKs and SPDZ, on the other hand, have a strong background in academics and are designed to be used as cryptographic primitives. As such, all of their internal structure is publicly known and implementations feature an extensive technical documentation in their respective whitepapers. Additionally, most of them make their source code freely available online.

3.1.3 Confidentiality

Due to the public nature of IOC oracles, all of their internal state may be accessed freely by anyone. In fact, we did not encounter a pure IOC approach which allowed users to encrypt their private data prior to the computation. SGX circumvents this problem by assigning a permanent encryption key to each device. By encrypting sensitive data with these keys, users can keep their secrets private and still obtain valid computation results.

The original zkSNARK definition, on the other hand, has a strict confidentiality requirement, namely that no secret data is to be revealed at any point before, during, or after the computation. This condition is fulfilled by all examined implementations, albeit with a small caveat. Even though no outside observers may extract

private data from a proof, the prover itself still requires this data to execute the computation and generate the proof.

Similarly, SPDZ is based on many of the mathematical principles also found in zkSNARKs. As such, the protocol's encryption techniques also provide the same guarantees, meaning that private data is not leaked before, during, or after an execution. In contrast to zkSNARKs, however, no step of SPDZ requires the involvement of a trusted third party, since all data is secret-shared among its participants. This fact alone gives SPDZ the highest possible confidentiality score.

3.1.4 Privacy

All examined technologies are subject to the same privacy model as their respective blockchain. This includes privacy requirements regarding identification of users and potentially sensitive data. Within energy use cases, especially high-resolution consumption data could contain security-related information or trade and business secrets. Therefore, input parameters to the computation need to be kept secret and are only shared with the computational node. Regarding identities, this means that, even though these are hidden behind public identifiers, these identifiers remain unchanged for every transaction, presenting attackers with opportunities for correlation attacks. SGX in particular faces an additional challenge in keeping its unique hardware keys private through group signature schemes.

A similar shortcoming can be observed in zkSNARKs, which, because secret data cannot be re-extracted from a proof, must employ a special technique to show that it was indeed generated with the user-supplied data. Therefore, all participants initially publish a hash of their secret on the blockchain. The prover now recalculates these hashes as part of the proven computation and includes them in the output. Because forging the result of hash functions is computationally infeasible, this allows participants to verify the correctness of their inputs. However, since all of this communication is public, anyone can now also verify that a user took part in the protocol, giving rise to all kinds of new correlation attacks.

Unlike the other technologies, SPDZ offers a few advantages in terms of privacy. As it stands, current SPDZ protocols require the participation of all network nodes (Sharma and Ng, 2020). Thus, whereas the other verifiable computing techniques often have a clear initiator, SPDZ only runs once for all nodes. The secret-sharing algorithm makes it virtually impossible to distinguish between nodes who actually took part in the computation and the ones who did not. This mechanism effectively prevents correlation attacks based on pseudonymous encryption keys.

3.2 Performance

The *performance* category comprises criteria which directly impact the efficiency of an application. While performance is a fairly broad term, our evaluation framework is tailored to match relevant use cases in the energy sector. Even though current developments toward close-to-real-time energy markets exist, the introduced use case concepts still operate with a certain lead time. Therefore, a distinct quantitative evaluation of throughput metrics is not considered within this study. Thus, we consider only such goals with direct applicability to a blockchain environment. In our case, the selected relevant metrics are

- overall throughput on the blockchain *(**T**ransaction **S**peed)*,
- space constraints in the context of the underlying consensus protocol *(**M**emory **C**onsumption).*

3.2.1 Transaction Speed

IOC oracles are redundant by design. Individual results must be filtered and aggregated before a final answer can be obtained. This inherent communication delay leads to a performance bottleneck which cannot be remedied, e.g., with faster hardware. SGX oracles, on the other hand, require only a single node to execute the computation, without the need for redundant computations. For these reasons, IOC oracles generally exhibit much slower query times than their SGX counterparts (Lo et al., 2020).

Setty et al. (2013) includes a detailed comparison of recent zkSNARK implementations. The original version with a trusted setup (Gennaro et al., 2013) achieves the best performance with a constant verification time and quasilinear proof generation complexity. For transparent zkSNARKs, most schemes have either a linear or logarithmic verification time as well as proof generation complexities that are poly-logarithmic across all reviewed implementations.

Lastly, SPDZ exhibits dramatic performance differences between its two phases. The online computation phase is very efficient and runs in quasilinear time. In contrast, the preceding offline phase is highly complex and, depending on network specifics, can result in quadratic communication complexity. In a 2013 article (Damgård et al., 2013), the original authors introduced SPDZ2, which decreases the running time of both phases by a factor of two. Following this, further improvements to the offline phase were made in the *Overdrive* (Keller et al., 2018) and *TopGear* (Baum et al., 2020) papers by using lattice-based homomorphic encryption to speed up parameter generation.

3.2.2 Memory Consumption

Because of their design, neither oracle variant requires an increase in blockchain memory consumption. IOC does not introduce any new cryptographic primitives with additional space requirements. SGX allows encryption of transaction payloads, but this does not increase the message size and neither does key management, since it is fully handled off-chain.

The preprocessing zkSNARK from Gennaro et al. (2013) introduces a small memory overhead of 128 bytes for each on-chain proof, but this number seems negligible considering the block size limits of popular reference chains (1 MB for BitCoin, ~30 kb for Ethereum). Transparent zkSNARKs, on the other hand, need to embed additional information in their setup-less proofs, which in turn leads to massively inflated proof sizes. For most schemes, sizes of several hundred kilobytes are common, but some even reach upwards of tens of megabytes, which clearly makes them impractical in light of the mentioned block limits (Setty et al. 2013).

Publicly verifiable SPDZ requires several items to be published to the blockchain; most importantly, each participant's Pedersen commitment (Pedersen, 1992), as well as the final result and any intermediate result obtained during computation. Fortunately, all of these values are comparatively small (in the size of several bytes) and do not place excessive restrictions on the choice of an underlying blockchain implementation. Also, in the case of overly large inputs, Enigma provides a workaround by placing these off-chain in a distributed hash-table (DHT). When stored in this way, values can still be accessed during a computation, but their respective hash references have sizes which are negligible compared to the real data.

3.3 Practicality

Lastly, the category *practicality* groups various aspects related to the solution's implicit cost to scale. This includes the current state but also whether the solution is sufficiently future-proof to be considered for long-term projects. The concrete aspects that we look at in the following sections are

- whether the technology is currently in a practice-ready state *(**M**aturity)*,
- how accessible the system is for developers *(**U**sability)*,
- how easy it is to add new functionality *(**E**xtensibility)*.

As the energy sector is currently in fundamental change, there are relevant optimization potentials to be addressed through process digitalization. As the system evolves constantly, new functionalities and use cases need to be added dynamically. Nevertheless, establishing new processes also require long-term functionality and reliability as it is part of critical infrastructure.

3.3.1 Maturity

Since blockchain oracles are one of the oldest applications of the technology, most implementations have reached a high level of maturity. Several production-ready examples for IOC and SGX oracles exist, e.g., Chainlink, TownCrier, TrueBit, etc. All of these have been developed by an active community and have seen use in real-world applications, making them a safe choice for new projects.

zkSNARKs provide a middle ground between reliability and maturity. All examined implementations offer strong security and transparency, but the need for a trusted setup remains a concern, which the community must yet address. Conversely, performance remains lackluster but is steadily improving. The preprocessing variants especially find increasing use in real-world projects, such as the Zcash currency or Ethereum's ZK rollup protocol. The most complete zkSNARK library to date, libsnark, already supports all of the constructions outlined in Gennaro et al. (2013), Ben-sasson et al. (2014), and Groth (2016). Transparent zkSNARKs, on the other hand, are a much younger technology that is still in its initial stages of research, with many schemes lacking even a basic reference implementation (Bünz et al., 2020; Zhang et al., 2020). Additionally, their immense space requirements currently make them impractical for general use.

SPDZ, at least in theory, offers unparalleled security, i.e., integrity of computations, confidentiality of private data, and complete user privacy. Despite active development in recent years, however, SPDZ has received far less attention by the blockchain community than, e.g., zkSNARKs, resulting from poor performance and a low number of production-ready implementations. The technology is under constant development, but even newer variants (e.g., Fractal or SuperSonic) demonstrate less-than-optimal performance, rendering them unsuitable for labor-intensive tasks, such as scientific computing. Consequently, even the popular Enigma framework has switched its back end to zkSNARKs in a recent version (Eberhardt and Tai, 2018). Therefore, the current reference implementation for SPDZ and its variants is the SCALE-MAMBA framework,[7] which is, however, not blockchain-specific.

3.3.2 Usability

Trusted oracles offer the highest level of usability. Developer experience is straightforward, at least for IOC oracles, which operate off-chain, thus allowing any traditional software stack to be used. Developing for SGX is slightly more involved as it requires the use of Intel's APIs.[8]

All of the libraries and tool chains for zkSNARKs which we discussed previously are implemented on top of preprocessing zkSNARKs. This includes proof generators, verification systems, smart contracts, etc. No such universal tooling currently exists for any of the transparent alternatives. Fortunately, however, transparent zkSNARKs use the same rank-1-constraint-systems (R1CS)[9] format to represent their arithmetic circuits. Therefore, at least the existing circuit compilers can be reused without modification.

SPDZ also works on the same arithmetic circuit abstraction as zkSNARKs. This means that developers proficient in writing such circuits can expect a certain level of familiarity with SPDZ development. Enigma, for example, allows its private contracts to be written in high-level languages such as Rust which compiles to R1CS. Other implementations, however, do not support the use of intermediate formats. In this case, programs have to be compiled from a raw circuit description language, e.g., VHDL.[10]

3.3.3 Extensibility

Since IOC oracles run their computations in traditional software environments, their possible use cases are only limited by the restrictions of general software development. This gives oracle providers great freedom of choice in the services they might offer. While SGX also runs as part of a traditional application, the security features themselves are subject to certain constraints, such as limited clock access. This places some limitations on the possible number of use cases. Nonetheless, any pure computation task remains fully supported by the platform.

[7]https://github.com/KULeuven-COSIC/SCALE-MAMBA.

[8]https://software.intel.com/content/www/us/en/develop/topics/software-guard-extensions/sdk.html.

[9]R1CS are the *de facto* standard in this realm. They are a mathematical representation of a circuit's architecture, describing all of its properties, including input variables, output variables, and logic gates.

[10]Very High Speed Integrated Circuit Hardware Description Language.

All general-purpose zkSNARK schemes operate on the same computational model and are capable of running the same programs. This includes any algorithm with deterministic runtime which can be represented as an arithmetic circuit. As it stands, no constructions for complexity classes beyond this currently exist. This relegates zkSNARKs to purely mathematical tasks that do not make use of any hardware specific features, e.g., a node's network stack. While this somewhat limits their amount of possible use cases, they still remain useful for verifying many numeric problems, such as scientific calculations or payment schemes.

Due to their shared dependence on arithmetic circuits, the extensibility of SPDZ is analogous to its zkSNARK counterpart. As long as a program can be expressed as a circuit, it can be safely executed by the protocol. This is true for all algorithms with a fixed upper bound. Again, algorithms beyond this complexity class are not covered. Hence, only relatively simple and purely mathematical calculations may be evaluated and verified with SPDZ.

The preceding evaluation reveals the specific properties and the appropriate application scenarios of the presented VC techniques. Therefore, choosing the right technology is very use-case specific, i.e., regarding its requisitions to security, performance, and practicality (compare **Table 2**). In the following final section, we have applied our findings to an optimization-based market approach for REC that is currently discussed and analyzed in several (research) projects.

4 APPLICATION TO OPTIMIZATION-BASED ENERGY MARKETS

As already stated, optimization is a key tool of (future) energy market systems. In all described market platform approaches, we are faced by the central question of *market clearing*. Researchers have brought forth a number of numeric optimizations models that seek to provide market clearing mechanisms that are much more efficient than the traditional auction-based heuristics for several market frameworks (Byrne et al., 2017; Baroche et al., 2019; Bogensperger and Zeiselmair, 2020; Jin et al., 2020). By integrating additional boundary conditions and quality specifications, matching between offer and supply cannot only be represented by a pure cost function. In addition, the already mentioned requirements regarding privacy, due to the use of sensitive data, and scalability, in a system of several million market participants, need to be considered in a practical way. In the following section, we apply our findings of the previous general assessment to the selected application of REC allocation—a key use case in the research project InDEED.[11]

[11]www.ffe.de/indeed – Within the research project InDEED, the concept of a Blockchain-based distributed data platform is developed and scientifically evaluated. Platform use cases focus on energy "labeling" (i.e., transparent and manipulation-proof digital (temporal and spatial) mapping of feed-in, consumption, and storage) and "asset logging" (i.e., tamper-proof and time-discrete collection of operation, maintenance, and repair data of energy assets).

4.1 Case Study: Allocation of Renewable Energy Certificates

The current system of the German guarantees of origin register (GOR) and its European integration shows several issues regarding its transparent and efficient allocation of REC. The most relevant and often discussed drawbacks include the non-consideration of physical boundaries (e.g., import and export), lack of transparency, and therefore potential double spending. Relatively low certificate prices (on average 1–2 €/MWh GO) further reduce investment incentives for renewable energy systems (RES), especially in certificate importing countries. Stable trade relations between well-known actors and the lack of transparency, with regard to prices and pricing mechanisms, reduce price and allocation flexibility (Hauser et al., 2019). Finally, the ban on double marketing of EEG-subsidized RES excludes a vast share of available RES (Schweins, 2020). Last but not least, numerous manual interactions, combined with little automation lead to inefficiencies and therefore missing scalability (Bogensperger and Zeiselmair, 2020).

As an alternative platform solution, Bogensperger and Zeiselmair (2020) propose a modification and extension of today's trading system by a downstream linear optimization (see **Figure 3**). As a supplement to an ex ante process for trading and allocating estimated future GO certificates (1), the distribution of the acquired certificates within the acquiring party's customers (consumers) is performed by an additional optimization (2). This optimization is based on measured (almost) real time consumption and generation data. In a first step (1), utilities can acquire GOs from the producers directly or through the trading platform. In a second step (2), all measured data of generated and consumed energy is used to compute an optimized distribution. This allows high spatial and time resolution to provide regionalized and asset-specific certificates of origin without the need to completely overturn the existing process. So, additional use cases are feasible, including an optimized regional direct marketing or allocation according to customers' energy mix preferences and supply distance prioritization. Further, grid or regional (i.e., border interconnection) constraints can be considered.

The setup and affiliated implementation of the proposed architecture leads to a number of challenges. First, within peer-to-peer markets a large number of individual market actors need to be coordinated, including their specific demands and restrictions. Therefore, scalability is a must. Second, some kind of central entity needs to operate the market and consequently run the optimization to fairly allocate and orchestrate the supply and demand of GO. As the parties involved intend to act independently, they potentially show a certain distrust to a central and potentially monopolistic intermediary. Third, certain properties need to be fulfilled regarding data consistency, i.e., guarantees of origin and the corresponding electricity generated need to be documented inseparably and transparently. In addition, the data processed are potentially sensitive and may not be publicly available. A blockchain architecture could meet most of the required needs. Further, time sequences can be documented in high resolution and required processes can be automated using smart contracts.

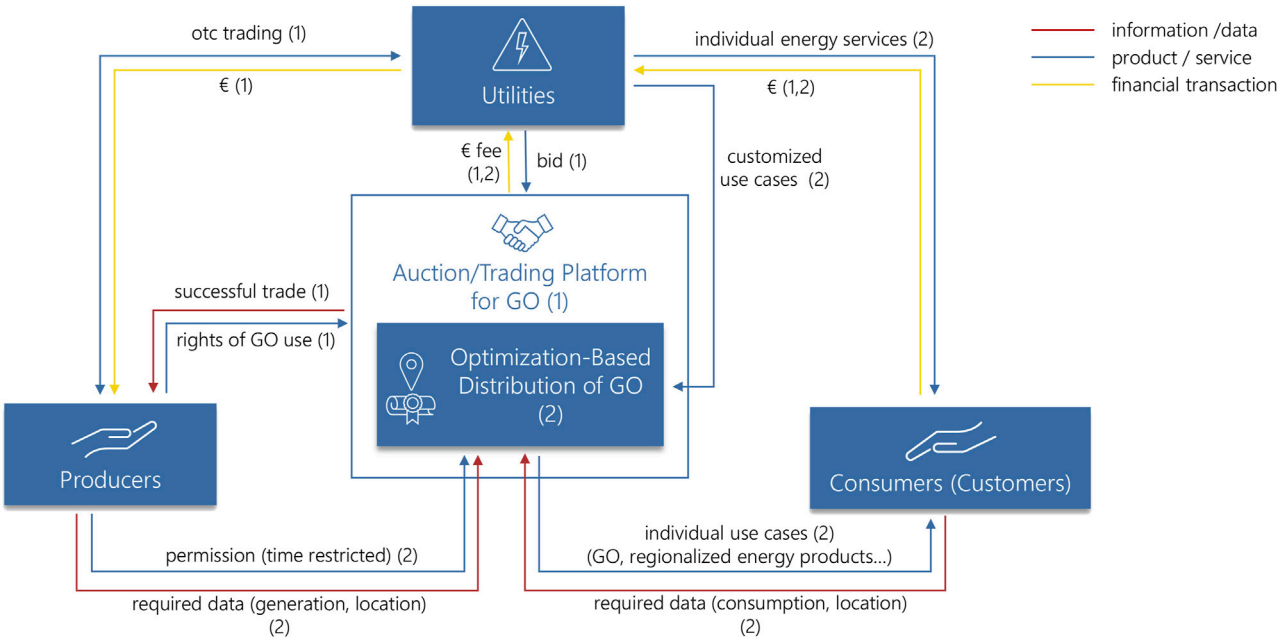

FIGURE 3 | e^3-value model describing the interactions of a potential auction platform for GO, divided into ex ante trading (1) and ex post optimized GO distribution (2), adapted according to Bogensperger and Zeiselmair (2020).

The challenge of managing complex optimization computations finally lead us to using VC techniques as analyzed in this paper. In order to select the most appropriate approach for the implementation, in the following we applied the evaluation results of **Section 3** to the selected case study of matching REC.

4.2 System Architecture Considerations

On the one hand, incentive-driven blockchain oracles are already available today and support the broadest spectrum of use cases due to their unconstrained execution model. On the other hand, distributed oracles largely lack support for secret user data. In systems that are intended for use by large swathes of the population but also (potentially critical) industry, preserving confidentiality is mission-critical. Transmitting information like energy consumption, location, or payments in the clear gives attackers an easy target or makes it possible to infer sensitive information. Correlating these data points could potentially allow them to stage large-scale attacks on the privacy of individual users.

Oracles based on the Intel SGX platform initially seem like a good fit for the energy industry. They provide decent security and an extensible execution model which can be molded to a multitude of use cases. Their biggest shortcoming, the reliance on the vendor as a trusted third party, is mitigated by the fact that most energy industrial systems are already operated by third parties that need to be trusted, e.g., governmental facilities. A tight cooperation between said government and the manufacturer can resolve some of these trust issues as the necessary trust is spread evenly among the participants. As a result, we can already see a few real-world examples for applications running on SGX (Araújo et al., 2018; Brenzikofer et al., 2019).

For the selected setting, however, we focus on technologies that require even less trust on the part of its users. This becomes important in situations where none of the system operators are trusted, like in P2P markets. In the case of one sole operator, they concentrate all the power and would have the ability to manipulate the market in their favor, even if the mentioned security measures are in place (Li et al., 2018). By conspiring with TEE vendors, for example, calculation results could be forged and prices artificially inflated.

This can be prevented with verifiable computation techniques that only rely on cryptography for their security guarantees. As long as these guarantees hold, even powerful adversaries do not have the ability to tamper with the system. One such technique is MPC. By providing a completely transparent and trustless environment, the effects of power centralization are dissolved. User data is kept private with advanced secret-sharing techniques. All in all, MPC exhibits all the security properties which are desirable for applications in the energy industry. Unfortunately, its poor performance proves to be a real hindrance. Most energy-related applications deal with intricate large-scale systems containing numerous actors and variables. Thus, they require high performance and overall throughput to guarantee timely results (Wang et al., 2017). We believe that at this point, current MPC implementations do not satisfy this requirement and with the hesitant adoption of this technique, this is unlikely to change in the near future.

In contrast, we find that the balance between security and performance struck by zkSNARKs offers the most promising solution for our problem. Also based on the principles of cryptography, they exhibit many of the same security guarantees as MPC. Indeed, all zkSNARK constructions

operate transparently and mostly trustless. Unlike MPC, however, private user data has to be shared with the compute node. While this is undesirable for many confidential applications, it has limited impact on the practicality of energy-related uses cases. This is because relevant data is usually already available to the operators of the compute nodes. To reiterate on the peer-to-peer GO market example, users would have to give up their energy generation, consumption, and bids to calculate a market clearing price. Since the compute nodes are operated by regulated energy providers, they already know this information—no confidential data is leaked. Most importantly, however, zkSNARKs are the center of many ongoing research projects. This leads us to believe that the technology is likely future-proof. In the short time since their inception, many shortcomings have already been addressed. This includes the trusted setup requirement as well as their middling performance. Since development is backed by many significant players in the blockchain space, we expect this trend to continue. For these reasons, zkSNARKs are a good balance for the required features and thus, provide the preferred technological foundation for our prototype.

4.3 Implementation

In order to provide a proof-of-concept, we implemented a prototypical approach of a zkSNARK for an optimization algorithm. The implementation is not intended to be part of the evaluation but to provide insights into the current state of implementation frameworks. We have chosen the ZoKrates framework as introduced by Eberhardt and Tai (2018). ZoKrates provides us with an expressive language to formulate the optimization algorithm as well as the necessary tools to publish and verify the proofs using a blockchain platform. The program code is published along this paper as supplementary data.

In practice, the referred optimization problems can be formulated as *linear programs (LPs)*. Linear programming is a sub-category of mathematical optimization from operations research, where the problem's constraints are presented as equalities and inequalities of purely linear terms.

A generalized linear program is given by its canonical form

$$\text{maximize} \quad c^T x \tag{1}$$

$$\text{subject to} \quad Ax \leq b \tag{2}$$

$$\text{and} \quad x \geq 0, \tag{3}$$

where **x** is a vector containing the so-called *decision variables*, which represent the optimal input values after the program has been solved. Secondly, the vector **b** represents the total amount available of each required resource, constraining the possible values for each decision variable. The matrix A, on the other hand, contains each decision variable's coefficient in the different resource equations, where a higher coefficient means that the variable requires a larger share of the specific resource. Lastly, the vector **c** represents the decision variable's contribution to the targeted objective function, or the total revenue in the original setting. The solving of linear programs is an ever-evolving field with a substantial commercial interest and over the years many novel solution approaches have emerged. One of the earliest linear programming techniques was developed by George Dantzig in 1947 (Dantzig et al., 1955), dubbed the *simplex algorithm*. With its good performance and ease of use, it continues to form the basis of many commercial linear solvers, such as IBM ILOG CPLEX[12] or Gurobi.[13]

Like all other circuit compilers based on R1CS, ZoKrates is subject to the limitations of the format's computational model. Algorithms, which would normally be straightforward to implement on a von Neumann architecture, must be altered to fit this new model. The biggest challenge here is overcoming the bounded control flow restrictions of arithmetic circuits. In general, all arithmetic circuits have a fixed size and therefore finite runtime. Consequently, useful programming constructs like input-dependent loops and recursions are not allowed by ZoKrates (or any other R1CS compiler), since the compiler is unable to predetermine their exact runtime. Because such constructions could potentially repeat ad infinitum, this makes it impossible to unroll them into a fixed-size circuit layout. This restriction is often circumvented by disallowing recursion and giving a static upper bound to loops. ZoKrates follows the same approach by requiring that all loops run a fixed number of times. Additionally, all iterations must indeed be executed and cannot be skipped over with conditional statements, etc. This behavior is enforced in code by only allowing one type of looping construct: a modified version of the traditional *for-loop*. The index variable is always a single field element of the underlying zkSNARK construction and its range is defined by compile-time constants, which are also field elements. Knowing the maximum amount of iterations at compile-time ensures that the compiler can successfully unroll the loop. This peculiarity of arithmetic circuits has severe implications for our simplex algorithm. For example, a worst-case instance of the algorithm has to make one iteration for each decision variable in the tableau to reach an optimal solution. The famed efficiency of the algorithm only comes from the fact that typically this time can be cut short by skipping some of the variables. Therefore, in the average case, the simplex algorithm terminates in cubic time with respect to its input length. As ZoKrates requires our main execution loop to have a fixed duration, no shortcuts can be utilized to terminate sooner. This leads the algorithm to always have a worst-case performance. Even if an optimum is reached early, the remaining iterations of the loop must be taken. Accordingly, this is not a quality feature of our implementation but a property of this type of algorithm.

4.4 Results, Critical Review, and Outlook

Finally, we were able to proof the basic function of the implemented simplex algorithm as a zkSNARK in ZoKrates. The source code is provided open source as auxiliary data to this paper. We further conducted performance tests and sensitivity analyses (Steinkopf, 2020). Depending on the number of variables and the number of conditions in the

[12]https://www.ibm.com/products/ilog-cplex-optimization-studio.

[13]https://www.gurobi.com/.

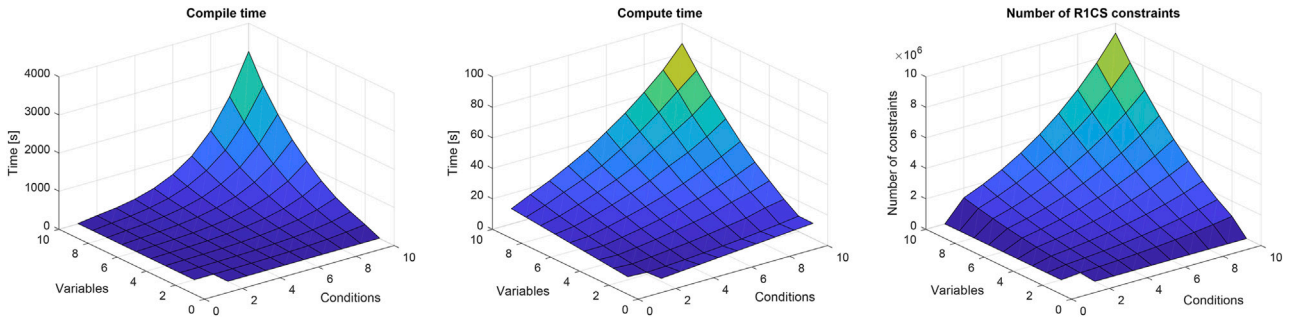

FIGURE 4 | Overview of performance metrics for different numbers of decision variables and conditions (Steinkopf, 2020).

optimization problem, three key evaluation metrics have been measured[14]: compile time for a single instance of our optimization circuit, compute time for a witness for the compiled circuit, and number of R1CS constraints (see **Figure 4**).

The resulting observations are mostly in line with the theoretical foundations of zkSNARKs, which state that circuit compilation should be much more expensive than witness derivation or even verification. Our results are not specific to certain optimizations, but hold true for all possible problem instances as our arithmetic circuit's runtime is fixed to that of a worst-case problem instance. The analysis revealed our zkSNARK to be viable for very small problem instances with parameter counts in the single digits. However, as problem size increases, all three examined circuit properties grow unfeasibly large, albeit at different rates. A first curve fitting approach—although to this quite limited number of experimental results—suggested an exponential dependency in the form of $f(x, y) = a \cdot exp(b \cdot x + c \cdot y)$.[15] Consequently, the rapid growth of complexity quickly leads to zkSNARKs that are not realistic in practical settings. Optimization problems in the real world often have hundreds or thousands of decision variables and conditions, which is clearly unsuitable for our construction. In a more practical settings, the obtained results do not support the idea of zkSNARKs as an optimization method with currently available frameworks. Nevertheless, the current developments in the field of ZKP are rapidly progressing and proposed alternative implementations promise to provide significant performance leaps. This allows us to look forward to further research, applying novel tools and frameworks (e.g., Circom[16]) that will also be considered in the context of the project InDEED.

5 CONCLUSION

The paper at hand provides an extensive overview of existing verifiable computing techniques with a focus on application in use cases related to the energy sector. Oriented to the initial research questions, we analyzed five different verifiable computation techniques in the fields of trusted oracles, zero-knowledge proofs, and multi-party computation regarding their respective advantages and disadvantages. Based on selected key features, we then evaluated the solution approaches regarding relevant criteria in security, performance, and practicality. We illustrated the added value in the field to optimization-based energy markets. By applying the evaluation results to decentralized optimization problems in the energy sector, we identified zkSNARKs as the most promising technology for the distinct case study of renewable energy certificate allocation. We implemented the simplex optimization algorithm as a zkSNARK to protect private user data while at the same time guaranteeing correctness of the optimization results. Even though current performance limitations are critical for large-scale deployment, recent research results show rapid progress in the development of more efficient algorithms.

To conclude: It will likely be feasible to run verifiable computing in a blockchain-based environment and profit from the respective properties of this technology in the future. Still, more research is required until this technology becomes practical. As an area with great potential, VC is receiving notable interest from the blockchain community, with many projects being actively developed. zkSNARKS are the field's current stars; nonetheless, new technologies are emerging constantly. With the recent advancements in machine learning, efficient methods for fully homomorphic encryption are receiving heightened interest and may eventually surpass competing approaches.

All in all, the introduced VC technologies yield great potential, especially in the energy sector. In order to ensure the security, reliability, and integrity of processes and data, many stakeholders are currently engaged in checking data provided by market participants. This includes the audit of processes in existing markets (e.g., REMIT) as well as the verification of provided services (e.g., frequency control) or the compliance with set boundary conditions (e.g., balance group management). With

[14]All tests were run on a worker node in an isolated environment. The node possesses an Intel Xeon Gold 6152 CPU with 22 Cores, 44 Threads and 30 MB of cache. Each core runs at a clock speed of 2.1 GHz. The used RAM is a 32 GB DDR4 Dual Rank RDIMM module with a clock rate of 2,666 MHz. Data is read from an SAS SSD with a transfer rate of 12 Gbps.

[15]with x = *number of conditions* and y = *number of variables* leaves a $R^2 = 0.9955$ for compile time ($a = 2.71$, $b = 0.4225$, $c = 0.2908$), $R^2 = 0.9861$ for compute time ($a = 1.49$, $b = 0.1961$, $c = 0.2175$), and $R^2 = 0.9888$ for the number of R1CS constraints ($a = 1.115e + 05$, $b = 0.2217$, $c = 0.2278$).

[16]https://github.com/iden3/circom.

increasing numbers of market participants due to decentralization, sector coupling, digitalization, etc., many of these processes might be revised and optimized in the future. VC is an integral part in order to deploy the blockchain-technology in these areas. zkSNARK especially show promising results for bringing trust in processes and ensure data minimization, privacy, and scalability. Overall, many areas can profit from the advances in verifiable computing technologies.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/**Supplementary Material**, further inquiries can be directed to the corresponding author.

AUTHOR CONTRIBUTIONS

AZ developed the concept and methodology with a focus on applications within the energy sector. BS elaborated the technical analysis of the verifiable computation techniques and developed the presented software. The writing of the manuscript was done by AZ and BS, supported by UG. UG and AB provided critical feedback, validation, and helped with the finalization. FM supervised the conducted research.

FUNDING

Most of the research described within this paper was conducted as part of the project InDEED (www.ffe.de/indeed), funded by the Federal Ministry for Economic Affairs and Energy (BMWi) (funding code 03EI6026A).

ACKNOWLEDGMENTS

The authors would like to thank all project members, especially the following colleagues: Johannes Sedlmeir, Fabiane Völter, and Benjamin Schellinger of Fraunhofer Blockchain Lab and University of Bayreuth.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fbloc.2021.725322/full#supplementary-material

REFERENCES

Al-Breiki, H., Rehman, M. H. U., Salah, K., and Svetinovic, D. (2020). Trustworthy Blockchain Oracles: Review, Comparison, and Open Research Challenges. *IEEE Access.* 8, 85675–85685. doi:10.1109/ACCESS.2020.2992698

Alskaif, T., and Van Leeuwen, G. (2019). "Decentralized Optimal Power Flow in Distribution Networks Using Blockchain," in SEST 2019 - 2nd International Conference on Smart Energy Systems and Technologies, Porto, Portugal, 9-11 Sept. 2019, 1–6. doi:10.1109/SEST.2019.8849153

Andoni, M., Robu, V., Flynn, D., Abram, S., Geach, D., Jenkins, D., et al. (2019). Blockchain Technology in the Energy Sector: A Systematic Review of Challenges and Opportunities. *Renew. Sust. Energ. Rev.* 100, 143–174. doi:10.1016/j.rser.2018.10.014

Araújo, M. V. M., do Prado, C. B., Carmo, L. F. R. C., Ón, A. E. R. R., and Farias, C. M. (2018). Secure Cloud Processing for Smart Meters Using Intel SGX. *Anais do XVIII Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais* Porto Alegre, Brasil: SBC, 89–96.

Bai, L., Hu, M., Liu, M., and Wang, J. (2019). BPIIoT: A Light-Weighted Blockchain-Based Platform for Industrial IoT. *IEEE Access.* 7, 58381–58393. doi:10.1109/access.2019.2914223

Baroche, T., Moret, F., and Pinson, P. (2019). "Prosumer Markets: A Unified Formulation," in 2019 IEEE Milan PowerTech (IEEE), Milan, Italy, 23-27 June 2019, 1–6. doi:10.1109/ptc.2019.8810474

Baum, C., Cozzo, D., and Smart, N. P. (2020). "Using TopGear in Overdrive: A More Efficient ZKPoK for SPDZ," in International Conference on Selected Areas in Cryptography, Waterloo, ON, Canada, August 12–16, 2019, (Springer, Cham), 274–302. doi:10.1007/978-3-030-38471-5_12

Ben-Sasson, E., Chiesa, A., Genkin, D., Tromer, E., and Virza, M. (2013). "SNARKs for C: Verifying Program Executions Succinctly and in Zero Knowledge," in Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 8043 LNCS, San Francisco, CA, United States, August 20–22, 2014, 90–108. doi:10.1007/978-3-642-40084-1_6

Ben-Sasson, E., Chiesa, A., Tromer, E., and Virza, M. (2014). Succinct Non-Interactive Zero Knowledge for a von Neumann Architecture. *23rd USENIX Security Symposium USENIX Security 14* (San Diego, CA: USENIX Association), 781–796.

Benhamouda, F., Halevi, S., and Halevi, T. (2019). Supporting Private Data on Hyperledger Fabric with Secure Multiparty Computation. *IBM J. Res. Dev.* 63, 1–3. doi:10.1147/JRD.2019.2913621

Bez, M., Fornari, G., and Vardanega, T. (2019). "The Scalability Challenge of Ethereum: An Initial Quantitative Analysis," in Proceedings - 13th IEEE International Conference on Service-Oriented System Engineering, SOSE 2019, 10th International Workshop on Joint Cloud Computing, JCC 2019 and 2019 IEEE International Workshop on Cloud Computing in Robotic Systems, CCRS 2019, San Francisco, CA, United States, April 4–9, 2019 (Institute of Electrical and Electronics Engineers Inc), 167–176. doi:10.1109/SOSE.2019.00031

Bogensperger, A., Zeiselmair, A., Hinterstocker, M., and Dufter, C. (2018). Die Blockchain-Technologie – Chance zur Transformation der Energiewirtschaft? – Berichtsteil: Anwendungsfälle (Forschungsstelle für Energiewirtschaft e.V.). 94.

Bogensperger, A., and Zeiselmair, A. (2020). "Updating Renewable Energy Certificate Markets via Integration of Smart Meter Data, Improved Time Resolution and Spatial Optimization," in International Conference on the European Energy Market (EEM), Stockholm, Sweden, September 16–18, 2020. doi:10.1109/EEM49802.2020.9221947

Bouloumpasis, I., Steen, D., and Tuan, L. A. (2019). "Congestion Management Using Local Flexibility Markets: Recent Development and Challenges," in 2019 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe), Bucharest, Romania, September 29–October 2, 2019 (IEEE), 1–5. doi:10.1109/ISGTEurope.2019.8905489

Brenzikofer, A., Meeuw, A., Schopfer, S., Wörner, A., and Dürr, C. (2019). "Quartierstrom: A Decentralized Local P2P Energy Market Pilot on A Self-Governed Blockchain," in 25th International Conference on Electricity Distribution, Madrid, Spain, June 3–6, 2019, 3–6.

Bünz, B., Fisch, B., and Szepieniec, A. (2020). "Transparent Snarks from Dark Compilers," in *Advances in Cryptology – EUROCRYPT 2020*. Editors A. Canteaut and Y. Ishai (Cham: Springer International Publishing), 677–706. doi:10.1007/978-3-030-45721-1_24

Buterin, V. (2014). *A Next-Generation Smart Contract and Decentralized Application Platform*. Ethereum, 1–36.

Byrne, R. H., Nguyen, T. A., Copp, D. A., Chalamala, B. R., and Gyuk, I. (2018). Energy Management and Optimization Methods for Grid Energy Storage Systems. *IEEE Access.* 6, 13231–13260. doi:10.1109/ACCESS.2017.2741578

Casino, F., Dasaklis, T. K., and Patsakis, C. (2019). A Systematic Literature Review of Blockchain-Based Applications: Current Status, Classification and Open Issues. *Telematics Inform.* 36, 55–81. doi:10.1016/j.tele.2018.11.006

Cherdantseva, Y., and Hilton, J. (2013). "A Reference Model of Information Assurance & Security," in 2013 International Conference on Availability, Reliability and Security (IEEE), Regensburg, Germany, September 2–6, 2013, 546–555.

Chitchyan, R., and Murkin, J. (2018). Review of Blockchain Technology and its Expectations: Case of the Energy Sector. *arXiv* [Epub ahead of print].

Cloosters, T., Rodler, M., and Davi, L. (2020). "TEEREX: Discovery and Exploitation of Memory Corruption Vulnerabilities in SGX Enclaves," in Proceedings of the 29th USENIX Security Symposium, August 12–14, 2020, 841.

Cohen, G., Damgård, I. B., Ishai, Y., Kölker, J., Miltersen, P. B., Raz, R., et al. (2013). "Efficient Multiparty Protocols via Log-Depth Threshold Formulae," in Lecture Notes In Computer Science (Including Subseries Lecture Notes In Artificial Intelligence and Lecture Notes in Bioinformatics) 8043 LNCS, Santa Barbara, CA, United States, August 18–22, 2013, 185185–202202. doi:10.1007/978-3-642-40084-1_11

Costan, V., and Devadas, S. (2016). Intel SGX Explained. IACR Cryptology ePrint Archive: Report 2016/086, 1–118. doi:10.1159/000088809 Available at: https://eprint.iacr.org/2016/086.pdf (Accessed September 10, 2021)

Damgård, I., Keller, M., Larraia, E., Pastro, V., Scholl, P., Smart, N. P., et al. (2013). Practical Covertly Secure MPC for Dishonest Majority - or: Breaking the SPDZ Limits. *Esorics* 8134, 1–18. doi:10.1007/978-3-642-40203-6_1

Damgård, I., Pastro, V., Smart, N., and Zakarias, S. (2012). "Multiparty Computation from Somewhat Homomorphic Encryption," in Annual Cryptology Conference, Santa Barbara, CA, United States, August 19–23, 2012 (Springer), 643–662. doi:10.1007/978-3-642-32009-5_38

Dantzig, G., Orden, A., and Wolfe, P. (1955). The Generalized Simplex Method for Minimizing a Linear Form under Linear Inequality Restraints. *Pac. J. Math.* 5, 183–195. doi:10.2140/pjm.1955.5.183

Duch-Brown, N., and Rossetti, F. (2020). Digital Platforms across the European Regional Energy Markets. *Energy Policy* 144, 111612. doi:10.1016/j.enpol.2020.111612

Eberhardt, J., and Heiss, J. (2018). "Off-chaining Models and Approaches to Off-Chain Computations," in SERIAL 2018 - Proceedings of the 2018 Workshop on Scalable and Resilient Infrastructures for Distributed Ledgers, Rennes, France, December 10–14, 2018 (New York, New York, USA: ACM Press), 7–12. doi:10.1145/3284764.3284766

Eberhardt, J., and Tai, S. (2018). "ZoKrates-Scalable Privacy-Preserving Off-Chain Computations," in Proceedings - IEEE 2018 International Congress on Cybermatics: 2018 IEEE Conferences on Internet of Things, Green Computing and Communications, Cyber, Physical and Social Computing, Smart Data, Blockchain, Computer and Information Technology, Halifax, NS, Canada, July 30–August 3, 2018 (iThings/Gree (IEEE)), 1084–1091. doi:10.1109/cybermatics_2018.2018.00199

Egberts, A. (2019). The Oracle Problem - an Analysis of How Blockchain Oracles Undermine the Advantages of Decentralized Ledger Systems. *SSRN J.* doi:10.2139/ssrn.3382343

Gallersdörfer, U., Klaaßen, L., and Stoll, C. (2020). Energy Consumption of Cryptocurrencies beyond Bitcoin. *Joule* 4, 1843–1846. doi:10.1016/j.joule.2020.07.013

Gennaro, R., Gentry, C., Parno, B., and Raykova, M. (2013). "Quadratic Span Programs and Succinct Nizks without Pcps," in Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26–30, 2013 (Springer), 626–645. doi:10.1007/978-3-642-38348-9_37

Gentry, C. (2009). "Fully Homomorphic Encryption Using Ideal Lattices," in Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing, Bethesda, MD, United States, May 31–June 2, 2009, 169–178. doi:10.1145/1536414.1536440

Goldwasser, S., Micali, S., and Rackoff, C. (1989). The Knowledge Complexity of Interactive Proof Systems. *SIAM J. Comput.* 18, 186–208. doi:10.1137/021801210.1137/0218012

Groth, J. (2016). "On the Size of Pairing-Based Non-interactive Arguments," in Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 9666, 305–326. doi:10.1007/978-3-662-49896-5_11

Hafid, A., Hafid, A. S., and Samih, M. (2020). Scaling Blockchains: A Comprehensive Survey. *IEEE Access.* 8, 125244–125262. doi:10.1109/ACCESS.2020.3007251

Hauser, E., Heib, S., Hildebrand, J., Rau, I., Weber, A., Welling, J., et al. (2019). Climate Change 30/2019 - Marktanalyse Ökostrom II Marktanalyse Ökostrom und HKN, Weiterentwicklung des Herkunftsnachweissystems und der Stromkennzeichnung Abschlussbericht. Tech. rep., Umweltbundesamt. Available at: https://www.umweltbundesamt.de/publikationen/marktanalyse-oekostrom-ii (Accessed September 10, 2021).

Heilmann, E., Zeiselmair, A., and Estermann, T. (2021). "Matching Supply and Demand of Electricity Network-Supportive Flexibility: A Case Study with Three Comprehensible Matching Algorithms," in MAGKS Papers on Economics 202110 (Marburg, Germany: Philipps-Universität Marburg, School of Business and Economics).

Heiss, J., Eberhardt, J., and Tai, S. (2019). "From Oracles to Trustworthy Data On-Chaining Systems," in Proceedings - 2019 2nd IEEE International Conference on Blockchain, Blockchain, Atlanta, GA, United States, July 14–17, 2019, 496–503. doi:10.1109/Blockchain.2019.00075

Herrera-Joancomartí, J., and Pérez-Solà, C. (2016). "Privacy in Bitcoin Transactions: New Challenges from Blockchain Scalability Solutions," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 9880 LNAI, Sant Julià de Lòria, Andorra, September 19–21, 2016, 26–44. doi:10.1007/978-3-319-45656-0_3

Hinterstocker, M., Dufter, C., Von Roon, S., Bogensperger, A., and Zeiselmair, A. (2018). "Potential Impact of Blockchain Solutions on Energy Markets," in International Conference on the European Energy Market, Lodz, Poland, June 27–29, 2018 (EEM), 1–5. doi:10.1109/eem.2018.8469988

Hoekstra, M., Lal, R., Pappachan, P., Phegade, V., and Del Cuvillo, J. (2013). "Using Innovative Instructions to Create Trustworthy Software Solutions," in HASP '13: Proceedings of the 2nd International Workshop on Hardware and Architectural Support for Security and Privacy, Tel-Aviv, Israel, June 23–24, 2013, 2487726–2488370. doi:10.1145/2487726.2488370

Hopwood, D., Bowe, S., Hornby, T., and Wilcox, N. (2016). *Zcash Protocol Specification*. San Francisco, CA: GitHub.

Jin, X., Wu, Q., and Jia, H. (2020). Local Flexibility Markets: Literature Review on Concepts, Models and Clearing Methods. *Appl. Energ.* 261, 114387. doi:10.1016/j.apenergy.2019.114387

Jogunola, O., Wang, W., and Adebisi, B. (2020). Prosumers Matching and Least-Cost Energy Path Optimisation for Peer-To-Peer Energy Trading. *IEEE Access* 8, 95266–95277. doi:10.1109/access.2020.2996309

Keller, M., Pastro, V., and Rotaru, D. (2018). "Overdrive: Making SPDZ Great Again," in Lecture Notes In Computer Science (Including Subseries Lecture Notes In Artificial Intelligence And Lecture Notes In Bioinformatics) 10822 LNCS, Tel-Aviv, Israel, April 29–May 3, 2018, 158–189. doi:10.1007/978-3-319-78372-7_6

Kirschen, D., and Strbac, G. (2005). *Fundamentals of Power System Economics*. John Wiley & Sons. doi:10.1002/0470020598

Kloppenburg, S., and Boekelo, M. (2019). Digital Platforms and the Future of Energy Provisioning: Promises and Perils for the Next Phase of the Energy Transition. *Energ. Res. Soc. Sci.* 49, 68–73. doi:10.1016/j.erss.2018.10.016

Kubli, M., Loock, M., and Wüstenhagen, R. (2018). The Flexible Prosumer: Measuring the Willingness to Co-create Distributed Flexibility. *Energy Policy* 114, 540–548. doi:10.1016/j.enpol.2017.12.044

Kuo, T.-T., and Ohno-Machado, L. (2018). Modelchain: Decentralized Privacy-Preserving Healthcare Predictive Modeling Framework on Private Blockchain Networks. *arXiv*.

Li, Z., Kang, J., Yu, R., Ye, D., Deng, Q., and Zhang, Y. (2017). Consortium Blockchain for Secure Energy Trading in Industrial Internet of Things. *IEEE Trans. Ind. Inf.* 14, 1. doi:10.1109/TII.2017.2786307

Lo, S. K., Xu, X., Staples, M., and Yao, L. (2020). Reliability Analysis for Blockchain Oracles. *Comput. Electr. Eng.* 83, 106582. doi:10.1016/j.compeleceng.2020.106582

Long, C., Wu, J., Zhang, C., Thomas, L., Cheng, M., and Jenkins, N. (2017). "Peer-to-peer Energy Trading in a Community Microgrid," in IEEE Power and Energy Society General Meeting, Chicago, IL, United States, July 16–20, 2017. doi:10.1109/PESGM.2017.8274546

Luongo, M., and Pon, C. (2017). The Keep Network: A Privacy Layer for Public Blockchains. Tech. rep., KEEP Network, Tech. Rep., 2018. Available at: https://coinpare.io/whitepaper/keep-network.pdf (Accessed September 10, 2021).

McKeen, F., Alexandrovich, I., Berenzon, A., Rozas, C. V., Shafi, H., Shanbhogue, V., et al. (2013). "Innovative Instructions and Software Model for Isolated Execution," in HASP'13: Proceedings of the 2nd International Workshop on Hardware and Architectural Support for Security and Privacy, Tel-Aviv, Israel, June 23–24, 2013. doi:10.1145/2487726.2488368

Morstyn, T., Teytelboym, A., and McCulloch, M. D. (2019). Bilateral Contract Networks for Peer-To-Peer Energy Trading. *IEEE Trans. Smart Grid.* 10, 2026–2035. doi:10.1109/TSG.2017.2786668

Munsing, E., Mather, J., and Moura, S. (2017). "Blockchains for Decentralized Optimization of Energy Resources in Microgrid Networks," in 1st Annual IEEE Conference on Control Technology and Applications, CCTA 2017, Maui, HI, United States, August 27–30, 2017, 2164–2171. doi:10.1109/CCTA.2017.8062773

Nakamoto, S. (2008). Bitcoin: A Peer-To-Peer Electronic Cash System. Available at: https://bitcoin.org/bitcoin.pdf (Accessed September 10, 2021).

Noyes, C.. (2016). Blockchain Multiparty Computation Markets at Scale.

Olson, K., Bowman, M., Mitchell, J., Amundson, S., Middleton, D., and Montgomery, C. (2018). *Sawtooth: An Introduction*. The Linux Foundation.

Pedersen, T. P. (1992). "Non-interactive and Information-Theoretic Secure Verifiable Secret Sharing," in Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 576, Santa Barbara, CA, United States, August 16–20, 1992, 129–140. doi:10.1007/3-540-46766-1_9

Pei, X., Sun, L., Li, X., Zheng, K., and Wu, X. (2018). "Smart Contract Based Multi-Party Computation with Privacy Preserving and Settlement Addressed," in Proceedings of the 2nd World Conference on Smart Trends in Systems, Security and Sustainability, WorldS4 2018, London, United Kingdom, October 30–31, 2018, 220–229. doi:10.1109/WorldS4.2018.8611588

Poon, J., and Buterin, V. (2017). *Plasma: Scalable Autonomous Smart Contracts*. Whitepaper, 1–47.

Rabin, M. O. (1981). *How to Exchange Secrets with Oblivious Transfer. Technical Report TR-81, Aiken Computation Lab*. Cambridge, MA, United States: Harvard University, 1–5.

Roth, A. E. (2015). *Who Gets What - and Why: The New Economics of Matchmaking and Market Design*. New York: Houghton Mifflin Harcourt Publishing Company.

Schweins, J. (2020). Rechtliche Rahmenbedingungen der Kennzeichnung von regionalen Ökostromprodukten - Legal Requirements for Labeling Regional green Power Products. (14), 47.

Setty, S., Braun, B., Vu, V., Blumberg, A., Parno, B., and Walfish, M. (2013). Resolving the Conflict between Generality and Plausibility in Verified Computation, in Proceedings of the 8th ACM European Conference on Computer Systems (EuroSys), 71–84. doi:10.1145/2465351.2465359

Shamir, A. (1979). How to Share a Secret. *Commun. ACM.* 22, 612–613. doi:10.1145/359168.359176

Sharma, S., and Ng, W. K. (2020). Scalable, On-Demand Secure Multiparty Computation for Privacy-Aware Blockchains. *Commun. Comp. Inf. Sci.* 1156, 196–211. doi:10.1007/978-981-15-2777-7_17

Sorin, E., Bobo, L., and Pinson, P. (2019). Consensus-Based Approach to Peer-to-Peer Electricity Markets with Product Differentiation. *IEEE Trans. Power Syst.* 34, 994–1004. doi:10.1109/TPWRS.2018.2872880

Sousa, T., Soares, T., Pinson, P., Moret, F., Baroche, T., and Sorin, E. (2019). Peer-to-peer and Community-Based Markets: A Comprehensive Review. *Renew. Sust. Energ. Rev.* 104, 367–378. doi:10.1016/j.rser.2019.01.036

Steinkopf, B. (2020). Analysis and Implementation of Verifiable Computation Techniques for Energy Blockchain Applications. Master's Thesis. Munich, Germany: Technical University of Munich.

Stoll, C., Klaaßen, L., and Gallersdörfer, U. (2019). The Carbon Footprint of Bitcoin. *Joule* 3, 1647–1661. doi:10.1016/j.joule.2019.05.012

Strüker, J., Albrecht, S., and Reichert, S. (2018). "Blockchain in the Energy Sector," in *Business Transformation through Blockchain: Volume II*. Cham, Germany: Springer International Publishing, 23–51. doi:10.1007/978-3-319-99058-3_2

Teufel, B., Sentic, A., and Barmet, M. (2019). Blockchain Energy: Blockchain in Future Energy Systems. *J. Electron. Sci. Tech.* 17, 100011. doi:10.1016/j.jnlest.2020.100011

Villar, J., Bessa, R., and Matos, M. (2018). Flexibility Products and Markets: Literature Review. *Electric Power Syst. Res.*, 154, 329–340. doi:10.1016/j.epsr.2017.09.005

Wang, Y., Wang, S., and Wu, L. (2017). Distributed Optimization Approaches for Emerging Power Systems Operation: A Review. *Electric Power Syst. Res.* 144, 127–135. doi:10.1016/j.epsr.2016.11.025

Weichbrodt, N., Kurmus, A., Pietzuch, P., and Kapitza, R. (2016). "AsyncShock: Exploiting Synchronisation Bugs in Intel SGX Enclaves" in Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes In Bioinformatics) 9878 LNCS, Heraklion, Greece, September 26–30, 2016, 440–457. doi:10.1007/978-3-319-45744-4_22

Wu, J., and Tran, N. K. (2018). Application of Blockchain Technology in Sustainable Energy Systems: An Overview. *Sustainability* 10, 1–22. doi:10.3390/su10093067

Yao, A. C. (1982). "Protocols for Secure Computations," in 23rd Annual Symposium on Foundations of Computer Science (sfcs 1982), Chicago, IL, United States, November 3–5, 1982, 160–164. doi:10.1109/sfcs.1982.38

Zeiselmair, A., and Bogensperger, A. (2021). "Development of a System Cartography and Evaluation Framework for Complex Energy Blockchain Architectures," in Internationaler ETG-Kongress 2021 (Wuppertal: VDE ETG), March 18–19, 2021.

Zeiselmair, A., and Köppl, S. (2021). Constrained Optimization as the Allocation Method in Local Flexibility Markets. *Energies* 14, 3932. doi:10.3390/en14133932

Zhang, J., Xie, T., Zhang, Y., and Song, D. (2020). "Transparent Polynomial Delegation and its Applications to Zero Knowledge Proof," in 2020 IEEE Symposium on Security and Privacy (SP) (IEEE), San Francisco, CA, United States, May 18–21, 2020, 859–876. doi:10.1109/SP40000.2020.00052

Zhong, H., Sang, Y., Zhang, Y., and Xi, Z. (2019). "Secure Multi-Party Computation on Blockchain: An Overview," in International Symposium on Parallel Architectures, Algorithms and Programming, Singapore, January 26, 2020 (Springer), 452–460.

Zyskind, G. (2018). "Enigma: Decentralized Computation Platform with Guaranteed Privacy," in *New Solutions for Cybersecurity*, 1–14. doi:10.7551/mitpress/11636.003.0018

Zyskind, G. (2016). Efficient Secure Computation Enabled by Blockchain Technology. PhD thesis. Cambridge, MA, United States: Massachusetts Institute of Technology.

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.