IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 3, 2020, accepted September 8, 2020, date of publication September 15, 2020,
date of current version September 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3024254

When Blockchain Meets SGX: An Overview, Challenges, and Open Issues

ZIJIAN BAO[1], QINGHAO WANG[1,2], WENBO SHI[2], LEI WANG[3], HONG LEI[1], AND BANGDAO CHEN[1]

[1]Oxford-Hainan Blockchain Research Institute, Chengmai 571924, China
[2]Department of Computer Science and Engineering, Northeastern University, Shenyang 110001, China
[3]Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

Corresponding author: Hong Lei (leihong@oxhainan.org)

This work was supported in part by the Oxford-Hainan Blockchain Research Institute, in part by the National Natural Science Foundation of China under Grant 61472074 and Grant U1708262, in part by the Fundamental Research Funds for the Central Universities under Grant N172304023, and in part by the National Key Research and Development Program of China under Grant 2018YFB0803400 and Grant 2019YFB2101601.

ABSTRACT As a decentralized, public, and digital ledger technology in Peer-to-Peer network, blockchain has received much attention from various fields, including finance, healthcare, supply chain, etc. However, some challenges (e.g., scalability, privacy, and security issues) severely affects the wide adoption of blockchain technology. Recently, Intel software guard extensions (SGX), as new trusted computing technologies, have provided a new solution to the above challenges in the blockchain area. Although many studies have focused on using SGX technology to enhance their schemes in the blockchain areas, no comprehensive survey has systematically analyzed and delineated these studies. This article is the first to systematically discuss the application status of SGX in the blockchain area. In this article, we study the scheme designs, advantages, and disadvantages of the existing works using a six-layer hierarchical structure of the blockchain. We also summarize the functions of SGX and formally analyze the advantages and disadvantages of SGX. Finally, we review the remaining challenges and present a list of possible directions for future research.

INDEX TERMS Blockchain, Intel SGX, consensus algorithm, privacy protection, smart contract.

I. INTRODUCTION

Blockchain, as the underlying technology of digital cryptocurrency, is a decentralized, public, and digital ledger that stores all committed transactions in a chain of blocks. In contrast to a centralized digital ledger, blockchain uses consensus algorithms to synchronize the distributed data replicated across multiple users. Blockchain was first introduced with Bitcoin [1] to solve the double-spending problem without the need for a trusted authority or central server, which revolutionized the field of digital currencies. Subsequently, it was used in the other projects (e.g., Ethereum [2], a global, open-source platform for decentralized applications to write and execute arbitrary, programmable transaction logic in the form of smart contracts). With the widespread attention of society, blockchain shows enormous commercial value, and the market intelligence firm Tractica forecasts that increasing applications involving enterprise blockchain will drive the global market size to $20.3 billion by 2025 [3].

Although blockchain technology has great prospects in commercial applications, it is suffering from many technical challenges. As is well known, poor scalability is still the bottleneck for blockchain. In Bitcoin, the throughput is restricted to a rate of 7 transactions per second because the block size is limited to 1 MB and a block is created approximately every 10 minutes. However, larger blocks require a larger storage space and are associated with slower propagation in the network, which are challenging tradeoffs between block size and security. Moreover, the major consensus algorithms used in the contemporary blockchain systems encounter some serious problems. For example, proof-of-work (PoW) wastes considerable computing resources in solving the hard cryptographic puzzle. Although some existing works try to reduce the waste of the resources [4], [5], the result is not ideal

The associate editor coordinating the review of this manuscript and approving it for publication was Vivek Kumar Sehgal.

(e.g., esoteric resources, low recycling rates). Proof of stake (PoS) [20] is heralded as the most promising mechanism to replace PoW, but it still faces several vulnerabilities, e.g., **nothing at stake attack**, **long-range attack** [54]. Furthermore, because all transactions and balances of each public key (i.e., the account of a user) are visible to everyone, the link between the address and the identity can be obtained by analysis [6]–[8]. Therefore, determining how to solve these issues is an essential step in the development of blockchain technology.

Furthermore, Intel software guard extensions (SGX) [9], as trusted hardware technologies developed by Intel Corporation, have been integrated into Intel's commodity CPUs and provide the possibility of large-scale usage. SGX offers a secure container by leveraging trusted hardware. Remote users can upload the code and data into the secure container, and the process can be proven reliable. The secure container protects the confidentiality and integrity of data while the computation is being performed on it. The code and data loaded in the secure container cannot be tampered with by the outside world. At the same time, the built-in services (e.g., trusted random number, trusted time, and trusted monotonic counter) also provide powerful assurance for designing protocols [10]. Benefiting from the above characteristics, SGX technology can provide powerful support to solve the dilemma in the blockchain. Thus, utilizing a powerful tool such as SGX in the blockchain area has become a new research direction.

Although the potential vision is an integration of blockchain and SGX schemes, a myriad of research challenges remain to be addressed. On the one hand, due to the limitations of design, SGX has certain deficiencies. Current SGX technology only provides a finite-sized secure container (a maximum of 128 M). Although SGX allows system software to oversubscribe the container by securely evicting and loading the memory block, it will cause extra overhead [11]. Moreover, researchers have identified that Intel SGX has critical side-channel security flaws [12]–[14] that can capture confidential data from the secure container. The SGX-based schemes may inherit those deficiencies, which requires researchers to design effective solutions. On the other hand, applying SGX technology to the blockchain area may generate new challenges. For example, the communication of the SGX platform is always controlled by an untrustworthy party, which requires researchers to design the proper protocol to avoid vulnerabilities [69]. Besides, it is worth studying how to divide the trusted code loaded in the secure container properly. Thus, due to the deficiencies mentioned above, building an efficient and strong scheme in blockchain areas remains an open challenge.

The main contributions of this article are as follows:

- We systematically summarize the functions of SGX, including *secure container, intra-attestation, remote attestation, data sealing*, and *trusted functions*. In addition, we formally analyze the advantages and disadvantages of SGX.
- We introduce a six-layer hierarchical structure of the blockchain. Based on the blockchain layers, we then classify the cutting-edge studies on the application of SGX in the blockchain area and study the scheme designs, advantages, and disadvantages of these works.
- We summarize all of the SGX-based works in the different blockchain layers based on the functions and disadvantages of SGX, and research the current remaining challenges and future directions for the application of SGX in the blockchain area.
- To the best of our knowledge, this is the first survey related to the application of SGX in the blockchain area. This article can inspire subsequent researchers to understand the research actuality and future trends of the application of SGX in the blockchain area.

The rest of the paper is structured as follows. Section II presents an overview of blockchain layers. Section III introduces the Intel SGX technologies. Section IV discusses the application of SGX in the data layer of blockchain. Section V discusses the use of SGX in the consensus layer. In Section VI, we explore the application of SGX in the contract layer. Section VII presents topics in the application layer. Then, we summarize and describe prospects for research of SGX in the blockchain in Section VIII. Finally, we provide a conclusion in Section IX.

II. BACKGROUND ON BLOCKCHAIN

Blockchain is a decentralized digital ledger that is connected and secured using cryptography algorithms [15]. The blockchain nodes controlled by different users around the world form a vast peer-to-peer (P2P) network to maintain the blockchain system. In the blockchain system, users can record and review data, but they cannot modify or remove any of the previous data. The consensus algorithm maintains the consistency of data. To clearly illustrate the application of SGX in the blockchain area, we divide the blockchain system into six layers based on the previous works [16], [17], as shown in Figure 1, including the network, data, consensus, incentive, contract, and application layers.

The network layer. Most blockchain networks are P2P networks, where the roles of nodes in the network are logically entirely equal. Nodes broadcast or forward the proposed block on the P2P network in the consensus phase. Currently, we have seen the application of SGX for network communications, especially the SGX-based anonymity network [18], [19]. However, we have not seen any related attempts involving SGX for network optimization in the blockchain network, and we would like to point out that it is still a viable direction.

The data layer. The data (e.g., the transactions, the nonce relating to the mining competition, the timestamp, the Merkle root resulting from the hash tree of all transactions, and some other metadata) in the blockchain system are often encapsulated in blocks and stored in nodes. The latter block contains the hash value of the previous block, which resembles a chain structure. In the data layer, some blockchain improvement

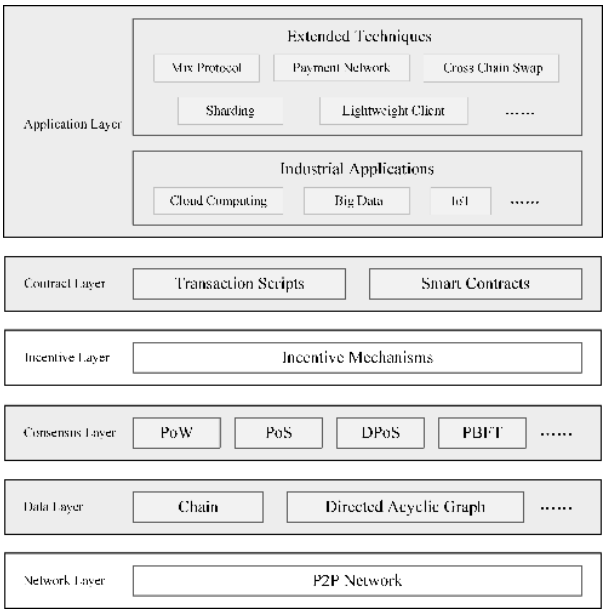

FIGURE 1. The architecture of the blockchain layer (the gray part of the figure contains the layers that we focus on in our analysis).

efforts produce the new data topologies by changing the organization, distribution, and status of the blocks, e.g., sharding [29]. We will discuss the related scheme based on SGX in Section IV.

The consensus layer. The consensus algorithm is the basis for quickly reaching a consensus on the validity of block data among highly dispersed nodes. Currently, there are several major consensus algorithms, including PoW [1], PoS [20], practical Byzantine fault tolerance (PBFT) [21], delegated proof-of-stake (DPoS) [22], etc. The Bitcoin system uses PoW, which means that the mining nodes perform a difficult task until one node first becomes the winner. PoS is an energy-saving alternative to PoW. PoS requires people to prove ownership of the amount of money without finding a nonce in an infinite space. For instance, Ethereum is planning to move from Ethash (a kind of PoW) to Casper (a kind of PoS). The Hyperledger Fabric[1] [16] uses PBFT (in version 0.6), which is a replication algorithm created to endure Byzantine faults. In PBFT, a primary is selected to determine a new block in each round. The entire process includes three phases: pre-prepared, prepared, and commit. In all phases, the node enters the next phase only after receiving 2/3 of the votes from all the nodes. More consensus algorithms are not described in detail here; we recommend references [23], [24]. However, most of the consensus algorithms have various limitations, e.g., resource waste and inefficiency (PoW), security issues (PoS), large-scale nodes are not supported (PBFT), and a centralized trend (DPoS). We will discuss the SGX-based schemes for the consensus layer in Section V.

[1]Hyperledger Fabric is a platform for distributed ledger solutions underpinned by a modular architecture delivering high degrees of confidentiality, resiliency, flexibility, and scalability initiated by IBM.

The incentive layer. The incentive layer integrates economic incentives, which is significant for the decentralized blockchain system to work as a whole. For example, Bitcoin and Ethereum systems issued digital currency as the rewards for the nodes that add blocks to the blockchain. The incentive layer will not be covered further because of the design of the incentives parallels our study.

The contract layer. The contract layer provides the programmable characteristics for the blockchain. There are two representative technical implementations: the transaction scripts and smart contracts. The transaction scripts contained in the transaction provide simple programming features and do not support loop statements to prevent infinite loops from causing downtime. However, transaction script functions related to cryptography are relatively powerful, including hash functions, signature scripts, even multi-signature scripts [25]. Compared to the transaction scripts, a smart contract is a special protocol designed to provide, verify, and execute contracts with the Turing-complete feature that can not only build complex trading logic but also implement any complex application. However, transaction scripts and smart contracts are recorded on the blockchain and are publicly visible; thus, it is easy to divulge the information of users. Moreover, some applications of smart contracts (e.g., financial instruments) require data about the real-world state and events from trustworthy websites. As smart contracts lack network access, the data are usually relayed by an untrusted party. Thus, lacking a supplier of reliable data is also a critical obstacle to the evolution of smart contracts. We will discuss the SGX-based schemes for the contract layer in Section VI.

The application layer. We divide the application layer into two types: extended techniques and industrial applications. The extended techniques focus on the enhancements of performance and privacy in the blockchain, e.g., sidechain [26], the mix protocol [27], [28], and payment networks [30], [31]. The industrial applications of blockchain cover various fields, e.g., finance [32], [33], cloud storage [34]–[36], and the Internet of Things (IoT) [37]–[39]. We will discuss these studies in Section VII.

III. OVERVIEW AND ANALYSIS OF INTEL SGX

Intel software guard extensions (SGX) are the most popular trusted execution environment (TEE) products in commodity CPUs. The SGX-enabled platform owner can create a secure container, called the **enclave**, in the secure memory (maximum 128 M) provided by the SGX to protect the integrity and confidentiality of internal computation, although the privileged software (kernel, hypervisor, etc.) is malicious [40]–[42]. To create an enclave, the platform owner loads the code into the enclave and performs the initialization process by calling CPU instructions. If the enclave has been initialized, the internal code will not be tempered. Generally, developers will define some interfaces to call the codes in the enclave, and the interfaces are implemented by the CPU instructions.

SGX provides two kinds of attestations, named intra-attestation and remote attestation, to help an enclave to prove to other parties that the particular code is running securely in the SGX-enabled platform. Specifically, intra-attestation is used by the enclave to provide proof to another enclave on the same platform. Remote attestation is used by the enclave to provide proof to another platform.

SGX supports sealing the enclave data outside of secure memory. Concretely, SGX offers two sealing strategies: sealing to the enclave identity and sealing to the sealing identity. The former will seal the enclave data using a unique key; thus, the data cannot be decrypted by another enclave on the same platform. The latter will seal the data using a platform key that can be obtained by the enclave on the same platform, which enables data sharing between enclaves.

Note: The trusted execution environment (TEE) provides a secure container to prevent potentially malicious users from controlling or observing the data inside it. Some implementations of TEE are popular, e.g., ARM TrustZone [84] and Intel SGX [9]. TrustZone was motivated mainly by secure mobile needs. It is incorporated into the recent ARM processors to implement TEE. Different from TrustZone, Intel SGX is implemented on the commodity computer with Intel CPU, and it supports more complex operations such as multithreading. Thus, most of the TEE-based schemes that need secure nonmobile devices use SGX to implement the prototyping system.

A. THE FUNCTIONS OF SGX

In this section, we systematically summarize the functions of SGX, including *secure container, intra-attestation, remote attestation, data sealing*, and *trusted functions*.

- **Secure container (Enclave):** SGX provides the instruction for users to create an enclave, which can prevent the code and data in it from eavesdropping or tampering by the outside world.
- **Intra-attestation:** SGX provides the instruction that helps an enclave to attest to another enclave on the same platform directly. Specifically, the enclave will generate the certification called REPORT with the measurement (the hash and signature of the code and data generated by the hardware during loading application into the enclave), enclave deployment information, user custom data, etc. REPORT is signed by a special key, which can only be obtained by the enclave on the same platform. Then, the targeted enclave will receive the REPORT and acquire the special key to verify the REPORT.
- **Remote attestation:** SGX provides the instruction to generate the certification called QUOTE for an enclave to prove to the remote platform that the particular content is loaded in the enclave and the enclave is running in the SGX-enabled platform. Specifically, the enclave first performs intra-attestation with Quoting Enclave (QE, a special enclave provided by Intel). Then, QE converts REPORT into QUOTE by signing it with the enhanced privacy ID (EPID, a special key bound to the firmware version of the CPU and can only be obtained by QE) [43]. Finally, QUOTE is sent to the remote platform for verification. During attestation, the remote platform can establish a secure communication channel with the enclave by performing the key-exchange protocols in the user custom data [44]. The secure channel can be used to transfer the data into the enclave.
- **Data sealing:** SGX provides the instruction to seal an enclave data out of the secure memory. In particular, sealing the data using a unique key can help an enclave to reduce the consumption of secure memory, and sealing the data using a platform key permits an enclave to share data with other enclaves on the same platform.
- **Trusted functions:** Intel provides some trusted functions on Intel platforms (e.g., trusted random number, trusted time, and trusted monotonic counter) for SGX to support more complicated solutions [41], [83]. Specifically, SGX provides the **rdrand** instruction to generate the trusted random number protected by hardware, and it also provides trusted time and trusted monotonic counter through reaching out securely to the manageability engine on the Platform Control Hub (PCH).

B. ADVANTAGES AND DISADVANTAGES OF SGX

To better understand the application of SGX in the blockchain layers, we try to abstract SGX's advantages and disadvantages systematically. Note that the definitions apply to all the SGX-based schemes. The advantages of SGX are shown below.

- **Trusted execution:** As the enclave is isolated, the execution logic of the code loaded into the enclave cannot be tampered with by the external environment, which ensures the correct execution of its internal code.
- **Privacy protection:** The data in the enclave cannot be accessed or tampered by the outside world. Thus, the confidentiality of sensitive data generated during execution can be protected effectively. Moreover, the secure channel between the enclave and other parties guarantees the privacy of data delivered into the enclave.
- **Simplify the protocol process:** Due to the introduction of the SGX-based role as a trusted party, the SGX-based schemes can reduce the use of complex cryptographic tools and simplify the protocol process, thereby improving the efficiency.

Furthermore, the disadvantages of SGX are as follows.

- **Limited memory:** Currently, SGX only supports secure memory that is smaller than 128 MB. Although SGX offers instructions to allow system software to oversubscribe the secure memory by evicting and loading enclave pages securely, it requires additional operations to protect data privacy, which will result in significant overhead. Thus, the abuse of the enclave will incur the poor performance of the SGX-based schemes.
- **Availability failures:** The SGX-based platform is fully controlled by the platform owner. Thus, it is easy for the platform owner to terminate enclaves. Alternatively,

TABLE 1. The schemes based on SGX in the consensus layer.

Proposal	Based Consensus	Target Problem	SGX-based Role	Description
PoUW [46]	PoW	Resources waste on PoW	Miners	Transforming the computing of solving the cryptographic puzzle into useful work.
PoET [47]	PoW	Resources waste and inefficiency on PoW	Consensus nodes	Waiting the random time to represent the work of nodes required by PoW.
PoLK [48]	PoW	Resources waste and inefficiency on PoW, concurrent invocations issue of TEE	Consensus nodes	Using random number to confirm the leader and employing monotonic counter to prohibit concurrent invocations of the TEE.
Hybster [52]	PBFT	Inefficiency of PBFT	Administrator	Improving efficiency using parallelization techniques.
SPoS [55]	PoS	Security issues on PoS	Consensus nodes	Performing the consensus operations in an enclave to prevent against the attacks on PoS.

although the secure channel can encrypt the inputs of an enclave, the platform owner can intercept the partial inputs of the enclave, which may impact the security of schemes. For example, the platform owner can transfer the old inputs to the enclave for a favorable consequence, which is called the replay attack. Thus, an SGX-based scheme must tolerate such platform failures.

- **Side-channel attacks:** SGX is implemented on the Intel CPU architecture, and it shares some computing resources (e.g., page table, cache) with normal programs. Therefore, SGX suffers from side-channel attacks, in which the attacker observes the shared resources to obtain the control flow and the data access mode of the enclave program to infer the sensitive information in the enclave. Some side-channel attacks on SGX have exploded recently [19]. Although most of the side-channel attacks are relatively difficult to be exploited, sufficient profit may drive attackers to attack the SGX-based platforms.
- **Single-point attacks:** If an SGX-based solution relies on the credibility of a single SGX entirely, one compromised SGX will cause the solution to crash. Meanwhile, the SGX-based role in some schemes can maintain massive benefits, which stimulates the attacker to try to impose full control of the SGX, even exploiting the physical means. Therefore, a well-designed multiparty scheme should be able to tolerate the failures of one or more SGX machines.

IV. SGX IN THE DATA LAYER

The data layer includes different data structures encapsulated in blocks and stored in nodes. Some blockchain improvement efforts change the organization, distribution, and status of the blocks. In this section, we will discuss the related scheme based on SGX.

A. SHARDING

Sharding technology can be used to improve the scalability of the blockchain, which divides the blockchain nodes into smaller committees running consensus algorithms independently [29], [79]. However, those schemes rely on the unspent transaction output (UTXO) model and thus are only appropriate for the permissionless blockchain systems (e.g., Bitcoin).

Dang *et al.* [68] propose a sharded blockchain scheme (AHL) that extends sharding to permissioned blockchain systems and has enhanced performance. AHL implements the shard formation protocol to secure confirmation of the node-to-committee assignment. Specifically, the scheme uses the trusted random numbers generated by TEE (e.g., SGX) in each node to assign nodes to different committees, which ensures the activity and security of the protocol. Meanwhile, AHL introduces a reference committee as the coordinator of transaction processing, which is responsible for assigning transactions to different committees. In AHL, each committee runs a variant BFT algorithm that optimizes the overhead of communication by using TEE. However, the approach is still inefficient when the transactions require access to multiple committees in AHL.

V. SGX IN THE CONSENSUS LAYER

The consensus layer consists of consensus algorithms, which guarantees the reliability and consistency of the blockchain data. In this section, we discuss the solutions using SGX to solve the resource waste, inefficiency, and security issues in the existing schemes. We summarize these studies in Table 1.

A. RESOURCE WASTE ISSUES OF CONSENSUS ALGORITHM

In PoW [1], nodes concurrently solve cryptographic puzzles using vast computing power and compete for the longest chain. PoW spends massive computing resources to eliminate Sybil attacks [45], which can control the blockchain by spawning multiple synonyms. However, all the blocks will be discarded except for the last confirming blocks, which results in significant resource waste.

To solve this problem, Zhang *et al.* [46] propose the proof-of-useful-work (PoUW) scheme based on SGX, which is an improved PoW scheme. The core idea of PoUW is to replace the wasteful computation in PoW with the computation that is useful for an external goal, such as a user who needs to outsource computation. In PoUW, there are three roles, namely, agent, miner, and client. Figure 2 shows the interactions between the three roles to broadcast the new block to the blockchain network. In each round, miners gain the computing task from clients and load them into the enclave to execute. Once the task is done, the results will be returned to

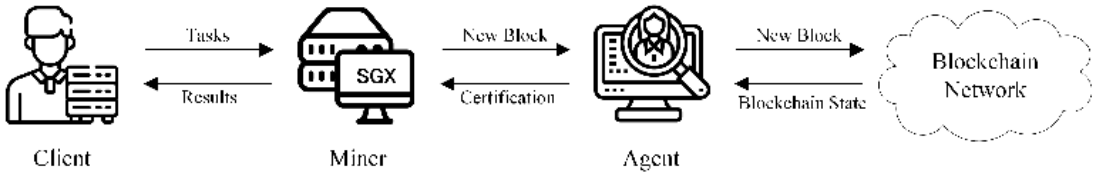

FIGURE 2. Overview of the PoUW.

the client. After each instruction of the useful computation, the enclave generates a random number and checks whether it is smaller than the target value (i.e., whether the miner wins the right of generating a new block). To improve efficiency, a check is performed at intervals, and the result is weighted by the total number of executed instructions. If a miner is determined to have created a new block, its enclave will generate a certification. Finally, the agent obtains the certification and the block from the miner and then broadcasts them to the blockchain network after checking. Benefiting from SGX, PoUW easily builds a trusted miner to perform the work required by clients, and remote attestation helps miners to prove the credibility of their platform to others. Obviously, the wasteful computation required by PoW is successfully replaced by useful work. However, if the attacker breaks the SGX-based node to change the execution result of the algorithm, the node will obtain the permanent right to generate blocks because of completely trusting the proof generated by enclave (i.e., the single-point attack). To relieve this problem, PoUW designs a statistical analysis scheme to test whether the mining time of the submitted blocks is different from the expected value. If so, the scheme will assert that the enclave of the miner has been compromised and the abnormal block will be rejected. Note that PoUW only makes wasted computation useful and still actually requires considerable computation. Furthermore, the users are unwilling to delegate some types of computing works (e.g., the big data or parallelized computations) to the miners, because the enclave memory is limited and PoUW requires that the useful work program must be single-threaded to ensure the security. Moreover, the users may need to make the arduous efforts to convert the useful program into an SGX-compliant version because the development languages supported by SGX are limited.

In the project of Sawtooth Lake,[2] Intel has proposed a new consensus algorithm, called the proof-of-elapsed-time (PoET) [47]. PoET reaches consensus depending on the wait time of nodes. In PoET, nodes need to generate a random number in each round, which represents the time that the node needs to wait next. The node that first completes the waiting process will obtain the right to create the new block and receive the reward. If the waiting process is completed, the enclave will return the waiting certificate broadcasted along with the block future. For security, PoET requires nodes to execute random number generation and the waiting process in the enclave provided by SGX. Moreover, PoET sets the timeout of the waiting certificate to mitigate the forks. Obviously, high computing resources are not consumed in PoET because nodes only perform the simple random number generation algorithm and the corresponding waiting process. Thus, PoET is not affected by the limited memory of SGX. However, PoET suffers from the single-point attack because it trusts the correctness of the algorithm executed in the single enclave. Furthermore, the only requirement to participate in PoET is that the device enables SGX. Thus, miners in PoET may gather a large number of cheap, outmoded SGX-enabled devices to promote the possibility of generating the new blocks, which is called the **stale chip problem**.

Based on the idea of PoET, Milutinovic *et al.* [48] propose the consensus algorithm called proof-of-luck (PoLK). Similar to PoET, the nodes of PoLK perform the random number algorithm in the TEE (e.g., SGX) in each round, and the node generating the smallest random number will win the right to create the new block. To reduce the network load, PoLK requires nodes to stop broadcasting their blocks when it receives the block with a smaller random number. However, a malicious participant may try to run multiple instances in parallel on the same platform to raise the probability of generating the smallest random number. To solve the problem, the enclave can create the maximum number of monotonic counters to prevent the participant from creating more enclave instances. Note that an SGX-enabled platform can only create the monotonic counters less than 256 at a time, and creating the monotonic counters exceeding 256 will return the error. Moreover, PoLK proposes to confirm a super-block that merges multiple normal blocks in each round to alleviate the dependence on the correctness of a single node, which prevents single-point attacks. However, PoLK also suffers from the **stale chip problem** because old devices can still be used for mining.

B. EFFICIENCY ISSUES OF CONSENSUS ALGORITHM

Pure Byzantine fault-tolerant (BFT) [49] has to employ $3f + 1$ service replicas to ensure that a service remains operational even if f of these replicas behave arbitrarily faulty. Some BFT protocols based on a hybrid fault model [50], [51] are able to tolerate arbitrary faults by employing a trusted subsystem, thereby reducing the costs significantly. However, those protocols of that class have to be performed largely sequentially and will not derive benefit from the parallel operations.

[2]Sawtooth Lake is a distributed ledger project with novel consensus and transaction handling mechanism, incubated under the Hyperledger project.

TABLE 2. The schemes based on SGX in the contract layer.

Proposal	Target Problem	SGX-based Role	Description
Hawk [56]	Privacy issue in the execution of smart contracts	Manager	Using cryptographic techniques and TEE to protect privacy in the execution of smart contracts.
ShadowEth [57]	Privacy issue of smart contracts in the existing public blockchain systems	Worker nodes	Executing smart contracts in the off-chain node based on TEE to protect privacy.
Ekiden [58]	Privacy and efficiency issues in the execution of smart contracts	Compute nodes	Executing smart contracts in the off-chain node based on TEE to protect privacy and minimizing the use of the blockchain to improve performance.
FastKitten [61]	Bitcoin does not support complex smart contracts	Operator	Implementing the TEE-based operator to execute off-chain smart contracts for Bitcoin.
TC [62]	Security and privacy issues of smart contracts' data source	Oracle	Building the secure connection between trustworthy websites and the enclave of the Oracle.

Thus, Behl *et al.* [52] propose Hybster, an improved BFT algorithm based on SGX, which enhances the performance by parallelizing the protocols. To prevent replay attacks where an attack saves the old state information to reset the subsystem, Hybster implements a trusted counter based on SGX for each node, which is called TrInX. The manager initializes all instances of TrInX before performing the protocol. Moreover, Hybster does not force all replications to perform consistency operations after a change operation. If the information is not essential to the security of the system, unsynchronized replications will be allowed to reduce the complexity of the algorithm.

C. SECURITY ISSUES OF CONSENSUS ALGORITHM

PoS is extremely energy efficient compared to PoW. In PoS, the probability of becoming the next block producer is determined by the proportion of assets owned by the node, and nodes do not need to solve the resource-intensive hard cryptographic puzzle. Thus, PoS does not waste a lot of computing resources. However, as the miner generating a block in PoS is simple, the optimal strategy for the miner is to mine on multiple forks to get their reward no matter which fork wins, which results in the **nothing at stake attack**. To implement the attack, an attacker may be able to send a transaction in exchange for another cryptocurrency, then start a fork of the blockchain from one block behind the transaction and send the money to themselves instead. In this way, the attacker's fork would win even with 1% of the total stake, because everyone else is mining on both. Moreover, as obtaining 51% of the stake in PoS is much easier than obtaining 51% of the computing power in PoW, PoS suffers from the **long-range attack** [54]. To implement the **long-range attack**, an attacker creates a fork from a confirmed block to alter the blockchain history. **Long-range attacks** require the attacker to control the majority of the stake in the blockchain. To achieve this condition more efficiently, the attacker may control the accounts that have enough stake at the past block height and create a fork from that height.

Li *et al.* [55] propose a secure PoS (SPoS) based on TEE (e.g., SGX) to relieve the abovementioned attacks in PoS. SPoS requires all nodes to support TEE. To join the blockchain network, a node needs to generate an account key pair in the enclave and configure the information of the node, which is easy to achieve by TEE. Considering the **nothing at stake attack**, SPoS uses the trusted monotonic counters to track the block height information and prevents a node that generates more than one block at the same block height. Meanwhile, the user's accounts are stored in TEE, and the confidentiality provided by TEE can prevent the attacker from obtaining others' accounts to obtain a sufficient stake, which alleviates the **long-range attack**. Thus, SPoS well resists the above attacks and guarantees the same fault tolerance as that of PoS. Note that SPoS is not affected by the **stale chip problem** because only the number and duration of assets that the node holds are related to the right of creating a block. Certainly, SPoS also inherits the problems of SGX (e.g., side-channel attacks).

VI. SGX IN THE CONTRACT LAYER

The contract layer includes various types of smart contract platforms. A smart contract provides powerful programming capabilities for the blockchain. However, it has privacy, efficiency, and security issues, which have been explained in Section II. This section discusses the SGX-based schemes to solve those issues in the contract layer. We compare several attributes of these studies in Table 2.

A. PRIVACY PROTECTION IN THE EXECUTION OF SMART CONTRACT

Some blockchain systems (e.g., Ethereum) implement the decentralized execution platform for smart contracts. However, the technology lacks transactional privacy. The entire sequence of actions is publicly visible since they are propagated across the network and recorded on the blockchain. Although users can create new pseudonymous public keys to increase their anonymity, the values of all transactions and balances for each public key are publicly visible. Some researchers [7], [8] demonstrated the de-anonymization attacks by analyzing the transaction graph structures of the blockchain.

Hawk [56] is a smart contract system using the zero-knowledge proof mechanism to protect privacy. The scheme implements a compiler to compile the smart contract to a Hawk program, which is a cryptographic protocol

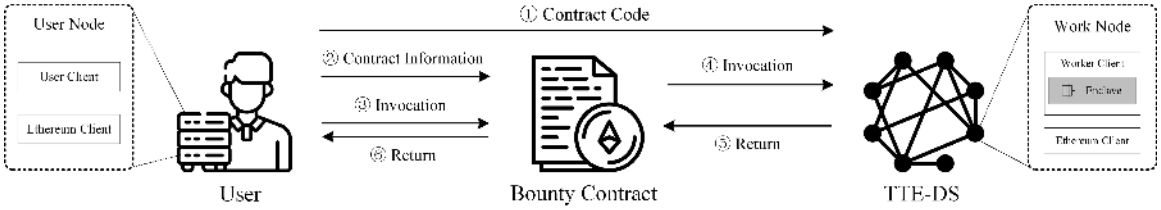

FIGURE 3. The architecture diagram of ShadowEth.

between the blockchain and the users. A Hawk program contains two parts, the private portion and the public portion. Hawk sets a manager to execute the private portion of the Hawk program and to facilitate the execution of the protocol. Due to the use of zero-knowledge proofs, the output of the private program will be hidden when submitted to the blockchain. However, the inputs of users are visible to the manager, which may reveal sensitive information. The authors recommend that the manager can be loaded in a TEE (e.g., SGX) to ensure that it does not disclose sensitive data from users. Obviously, the manager based on TEE can protect the confidentiality of data and the correctness of execution. However, side-channel attacks may break the confidentiality of the user's privacy in the manager. Moreover, the zero-knowledge proof results in high overhead, which limits the efficiency of Hawk.

ShadowEth [57] establishes a platform to support the execution and storage of private smart contracts without breaking the integrity of existing public blockchains (e.g., Ethereum), which protects the contracts' confidentiality and security. In ShadowEth, the TEE-distributed storage platform (TEE-DS), comprised of the worker nodes based on TEE (e.g., SGX), is responsible for executing and storing the private contracts. The bounty contract is used to provide the workers with the target private contract and parameters as well as payment of remuneration. Figure 3 shows the workflow of ShadowEth. Specifically, to execute a private contract, the user node first puts the contract to TEE-DS using the user client and uploads the identification information of the contract to the bounty contract. Then, the user node sends the invocation request and relevant arguments to the bounty contract. A worker node will obtain the task from the bounty contract using the worker client and will load the contracts from TEE-DS into the enclave to execute. Finally, the worker node returns results to the bounty contract. For confidentiality, the arguments and returned value should be transferred by a secure channel between worker nodes and user nodes. However, side-channel attacks may reveal private information to malicious worker nodes, but it requires more demanding conditions. For example, in an auction scenario (an example program in ShadowEth), worker nodes can pre-execute the contract code to acquire the control flow of bids' comparison, which can infer the users' bids in an auction. Actually, it may require multiple attempts and need to steal the bids before the auction.

Cheng *et al.* [58] propose Ekiden to solve the privacy issues of smart contracts. As shown in Figure 4, Ekiden organizes all nodes into two groups: compute nodes and consensus nodes. The consensus nodes are responsible for maintaining the blockchain system and updating the state of the smart contract. The compute nodes, which are required to support TEE (e.g., SGX), are responsible for the execution of the smart contract in the enclave. Any node that supports TEE can join in the system as a compute node, which guarantees the scalability of the system. A quorum of compute nodes form a key management committee (KMC), which generates and manages keys for compute nodes by distributed protocols [59], [60]. KMC needs to generate a new pair of keys for each contract, and the user needs to encrypt the input using the corresponding public key before delivering it to a compute node. Similarly, the output will be transmitted through the secure channel between the compute node and the user, which ensures privacy. To tolerate failures of compute nodes, Ekiden only stores the status information of the contract on the blockchain and updates the information after each execution.

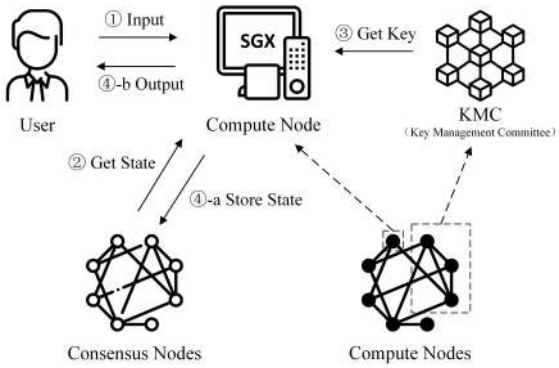

FIGURE 4. Overview of Ekiden architecture and workflow.

FastKitten [61] is a system based on TEE (e.g., SGX) to support executing complex smart contracts over the Bitcoin system. In FastKitten, an off-chain operator based on TEE (e.g., SGX) is used to execute smart contracts. To execute smart contracts, users first send the contract code and the information of participants to the operator to initialize the enclave. After the execution of the smart contract, the enclave will publish the final output transaction. To reduce the malicious behaviors, users must submit a deposit before executing

the contract, and then, the execution nodes must submit a deposit equal to the sum of deposits. If the protocol fails, the party that misbehaves will lose the deposits. To guard against the denial-of-service attacks caused by the controlled communication channel of TEE, FastKitten uses a time-out mechanism, which limits the maximum amount of execution steps that can be performed by the operator per one execution round. However, FastKitten only supports limited types of contracts and cannot prevent multiparty collusion from cheating regarding the deposits. Moreover, the privacy information in the smart contracts may be revealed by the side-channel attacks. Besides, FastKitten cannot tolerate enclave failures.

B. SECURITY AND PRIVACY IN THE DATA SOURCE OF SMART CONTRACT

The execution of smart contracts requires frequently obtaining real-world data. Currently, most of the data provided to smart contracts are obtained from trustworthy websites using smart contract data feeds (e.g., Oracles [2]), which are the smart contracts on the blockchain that serve data requests by other contracts. However, the data request for the smart contract is public, and the data provided by the data feeds are not trusted because of unreliable sources of information or man-in-the-middle attacks. Thus, determining how to provide reliable data for smart contracts and protect users' privacy regarding data feeds are challenging issues.

Zhang Fan *et al.* propose Town Crier (TC) [62], an SGX-based Oracle system, which builds a credible bridge between HTTPS-enabled websites and the Ethereum blockchain. In TC, there are three parts: **TC smart contract, Relay,** and **TC enclave,** as shown in Figure 5. The TC smart contract provides an interface for the other smart contracts to request data and acquires the corresponding result data. Relay provides communication services for the TC enclave because the enclave lacks direct network access. The TC enclave is responsible for analyzing the requests and returning the corresponding results. When a user requests data, the TC smart contract will be called. To obtain the user's request, the relay scrapes the blockchain to monitor the state of the TC smart contract. After obtaining the request from the TC smart contract, the relay will submit the request information to the TC enclave. Then, the TC enclave will contact the data source via HTTPS to obtain the requested data, and the relay is responsible for data passing. The data will be returned to

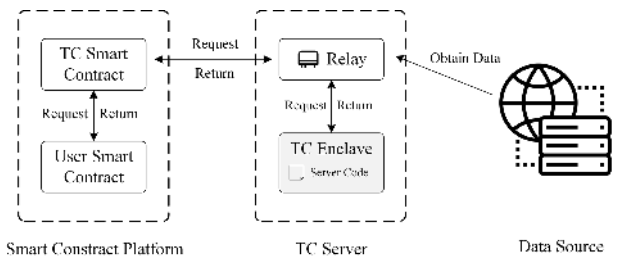

FIGURE 5. The architecture diagram of Town Crier.

TC smart contract in the form of a blockchain message, and the TC smart contract final return the data to the user's smart contract. TC provides the authenticated data to smart contracts without a trusted service operator. The data request can be delivered to the TC enclave through the secure communication channel to ensure the user's privacy. To protect against the compromise of a single TC server instance, the scheme proposes to build multiple TC server nodes. Users can request data from multiple TC servers to ensure the credibility of data. Meanwhile, to prevent the data providers from maliciously changing data, TC provides a possible solution that users can request the data from various data sources for fault tolerance. However, TC cannot guarantee the consistency of the data accessed by the different nodes. Note that TC has been a public service online [63].

VII. SGX IN THE APPLICATION LAYER

The application layer includes the extended techniques and the industrial applications of the blockchain. According to the extended techniques, we introduce the schemes based on SGX to enhance the performance and functions of the blockchain, e.g., payment network [64], cross-chain [65], lightweight client [66], and mix protocol [67]. For the industrial applications, we introduce the solutions combining SGX and the blockchain in other fields, e.g., cloud computing [69], big data [70], and IoT [71]. We summarize these schemes in Tables 3 and 4.

A. THE APPLICATION EXTENSIONS OF BLOCKCHAIN

1) PAYMENT CHANNEL NETWORK

The payment network [30], [31] is one of the solutions to solve the throughput problem in the blockchain. This network allows parties to build point-to-point payment channels to pay cryptocurrency off-chain, and only the settlement transaction needs to be synchronized to the blockchain. Thus, payment networks can operate with higher transaction throughput than the blockchain itself. However, existing schemes require parties to perceive and invalidate the aberrant settlement transaction that may be submitted by a malicious party in a limited period. Therefore, the network is vulnerable, as the malicious party can submit the settlement transaction that is of benefit to himself or herself and delay the other party's blockchain access to steal funds.

Lind *et al.* [64] propose Teechain, a secure payment network scheme based on TEE (e.g., SGX). In Teechain, all nodes, which are operated by different parties, construct a P2P network. Each node maintains a manager called the treasury in the enclave. It can be used to manage parties' funds and to execute the payment protocol. Before creating payment channels, a treasury generates a pair of key pairs as its cryptocurrency address, and the private key is stored in the enclave securely. The user then needs to send funds to the address in the form of a deposit. When a payment channel is built, the deposits will be updated according to the payment message. At any time, either party could close

TABLE 3. The application extensions based on SGX.

Proposal	Target Problem	SGX-based Role	Description
Teechain [64]	Existing payment channel schemes are insecure	Parties to the transaction	Building the secure payment channel between enclaves, which performs off-chain payment protocol to simplify payment.
Tesseract [65]	Existing cross-chain transaction schemes are insecure or inefficient	Exchange	Building the center exchange based on SGX that performs the sandbox trading in the enclave and synchronizes to the blockchain periodically.
BITE [66]	Privacy issue of lightweight client schemes	Full-nodes	Building the trusted full node based on SGX to verify transactions for lightweight clients.
Obscuro [67]	Malicious central mixer	Mixers	Performing mix in the central mixer based on TEE to protect privacy and security.

TABLE 4. The industrial applications of integrated blockchain and SGX.

Proposal	Target Problem	SGX-based Role	Description
Airtnt [69]	Trust issue in distributed cloud computing service	Provider nodes	Building the fair trade protocol between the user and server using smart contracts and TEE.
PrivacyGuard [70]	Data owner cannot ensure the authorized user to properly use the data	Off-chain contract execution node	Building the off-chain contract execution node based on TEE to securely execute access control contracts and recording the usage records of data in the blockchain.
TID [71]	Privacy issue of IoT data	Storage platform	Building the off-chain storage platform based on TEE to store IoT data and implementing the access control using smart contracts.

the payment channel by sending the settlement transaction to the blockchain. To tolerate nodes failures, the deposits are always built in the multi-signature transactions [25], which needs n signatures of m treasury addresses (those m treasuries will be called a committee, and m is greater than n). Thus, the settlement transaction does not depend on a single node. In a committee, treasuries jointly manage the deposit information using a replication protocol. To prevent against **roll back attacks**, Teechain employs hardware-based monotonic counters (e.g., provided by SGX) to guarantee state freshness. Moreover, the use of the side-channel-resistant cryptographic library alleviates side-channel attacks to some extent.

2) CROSS-CHAIN TRANSACTION EXCHANGE

Currently, many centralized exchanges help users to implement the swap between different cryptocurrencies, which are called cross-chain transactions. However, exchanges need to hold users' cryptocurrencies to complete the deal, which is insecure. The decentralized exchanges (e.g., EtherDelta [72]) settle transactions using smart contracts, which eliminates the risk in centralized exchanges. Nevertheless, they cannot support real-time cryptocurrency exchange.

Tesseract [65] is an exchange scheme that can implement secure and real-time cross-chain transactions. When making a deal, users first need to register an account and send funds as the balance into the Tesseract addresses that are generated and stored in the exchange's enclave. The enclave will create an address for each cryptocurrency. Then, users submit an order to the exchange's enclave via a secure channel. After receiving the message, the enclave issues the anonymous order to every user. When users choose the appropriate order to trade, the enclave will update the corresponding balance based on the transaction information, and it will periodically synchronize settlement transactions to the blockchain network. Tesseract builds a trusted exchange, which maintains the balance of users in the enclave to implement real-time cross-chain transactions. However, assuming that the enclave constructs two settlement transactions and synchronizes to the blockchain to achieve a cross-chain transaction, it will result in a unilateral payment if an attacker intercepts one of the settlement transactions. To solve this problem, Tesseract implements the fair settlement protocol to ensure the atomicity of two settlement transactions. To tolerant failure of the exchange node and avoid funds of users becoming stuck in contrast to the single exchange node, the authors build the multiple mutually Tesseract exchange nodes using the Paxos consensus protocol. Moreover, Tesseract also uses the side-channel-resistant cryptographic library to alleviate side-channel attacks.

3) LIGHTWEIGHT CLIENT OF BITCOIN

In the blockchain, nodes need to store complete data of the chain to perform transaction confirmation, which limits the use of the blockchain by some resource-constrained devices (e.g., mobile phones). Researchers [73], [74] propose lightweight clients that implement the simple payment verification (SPV). SPV requires clients to connect to a full node that stores the complete data of the chain. The full node will help clients to confirm the transaction. However, full nodes will acquire the user's privacy because they need to obtain transactions from clients to confirm them. Some schemes implement filters in the client [75], which provide a set of transactions with false positives that used to obscure the real transactions. Although these schemes protect the privacy of users, they require more communication overhead.

BITE [66] introduces the full node based on SGX to improve lightweight Bitcoin client privacy. In BITE, a client

will perform remote attestation and establish a secure channel to the enclave in a full node when it needs to verify a transaction. Then, the client sends the request (e.g., the transaction's verification information) to the enclave, and the enclave will return the information from the local store using a secure channel. The scheme proposes two solutions to obtain the required verification information: one solution is that the full node scans the whole blockchain and replies with the Merkle path of the target block; the client then verifies the correctness of the transaction through the Merkle path and the block header. The other solution is that the full node maintains a special version of the unspent transaction outputs (UTXO) database, and the enclave will access the database and directly return the corresponding result. To prevent against side-channel attacks, BITE uses the oblivious random access machine (ORAM) technology [76], a well-known technology that can hide access patterns, to access the local store. However, BITE may suffer from inefficiency because the cost of ORAM is linear with the size of the scanned data.

4) MIX PROTOCOL

The privacy protection scheme based on the mixer is one of the existing solutions to solve the privacy problem in Bitcoin. In these schemes, users send funds from the source Bitcoin addresses to a mixer's Bitcoin address, and the mixer then transfers the funds to the target Bitcoin addresses determined by users in a shuffled order. The miner disrupts the link between the source addresses and the target addresses, which protects privacy. However, in this case, it is difficult to constrain the malicious behavior of the servers. Moreover, some Bitcoin mixing protocols [77], [78] can complete the decentralized mix operation. However, they have to wait to find other mixing parties, which is inefficient.

Tran *et al.* [67] propose Obscuro, a centralized privacy protection scheme, to solve the problem above. The scheme relies on a centralized mixer based on TEE (e.g., SGX) to execute the mix operations in the enclave. Figure 6 shows the architecture and workflow of Obscuro. To mix a transaction, the user first broadcasts the transaction (including the target address information) to the Bitcoin network, which sends funds to the mixer's Bitcoin address. The mixer's address is generated by the enclave and stored in it securely; thus, the funds in the address are only used by the enclave. The mixer will synchronize the blockchain information to the enclave periodically, and the enclave will obtain the transactions related to the mix from the blockchain. The enclave then constructs the transactions that output funds from the mixer's address to the target addresses of users, and the mixer broadcasts the transactions to the Bitcoin network. In Obscuro, the mix information of users comes from the blockchain; thus, the mixer cannot intentionally prevent a part of the users from obtaining the service to reduce the anonymous set. Moreover, Obscuro does not need to introduce additional protocols (e.g., the deposit mechanism) to prevent the malicious behavior of the mixer because the enclave can ensure the trusted execution. However, Obscuro suffers from side-channel attacks because of completely trusting the enclave.

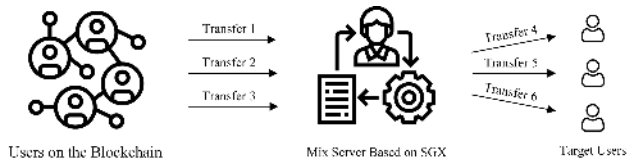

FIGURE 6. Overview of Obscuro's execution process.

B. THE APPLICATIONS OF BLOCKCHAIN IN OTHER FIELDS

1) DISTRIBUTED CLOUD COMPUTING SERVICE

Cloud computing is based on the centralized service model in that the cloud computing provider delivers computing services to users over the Internet. However, due to the rising computational requirements from the edge of the network (e.g., IoT devices), the centralized service model suffers from a tremendous demand for network communication. Moreover, cloud computing services (e.g., Amazon) act as large central points of failure and will be forced to interrupt many services if their data center breaks down. Decentralized computing infrastructures are an alternative to solve the above problem. However, it is challenging to build trust between users and computing nodes because any node can join the system.

Airtnt [69] is a distributed cloud computing solution that allows service providers to rent the calculations of TEE (e.g., SGX). The scheme designs a fair trade protocol using the remote attestation provided by TEE and smart contracts. Figure 7 illustrates the workflow between the three parties, service provider, service requester, and smart contract, respectively. Specifically, a requester first performs the remote attestation and constructs a secure communication channel with a provider's enclave. Then, the requester creates a payment channel between the requester's address and the provider's address, which is implemented by a smart contract. At this point, the requester can send the computing tasks to the provider, and the provider will load the code into the enclave to execute. The enclave returns the result encrypted by a

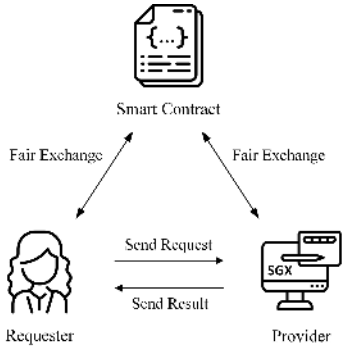

FIGURE 7. The architecture diagram of Airtnt.

temporary key, which is generated and stored in the enclave. After a check, the requester sends the payment information to the enclave, and the enclave will update the balance of the channel and return the temporary key to the requester. The requester can obtain the result using the key. Finally, the payment channel will be settled by the provider's enclave. Obviously, the fair trade protocol can guarantee the fairness of the rental. However, the payment channel requires the parties to maintain continuous communication during the rental process, which limits the availability of the scheme. Airtnt also suffers from side-channel attacks, which may reveal the user's privacy to the provider. Moreover, the single-point attack will permit the attacker to steal the funds without fulfilling any computing tasks because the provider's enclave can settle the payment channel.

2) PRIVACY PROTECTION AND CONTROLLED USE OF DATA

With the rise of big data, the ownership and privacy of data have received more attention from the public. For this purpose, researchers have proposed numerous schemes, e.g., the data access control schemes [80] are used to restrict the users of the data, and the data anonymization schemes [81] are used to protect the privacy of data. However, those schemes cannot guarantee that the authorized user uses the data properly (i.e., the authorized user can determine the use of the data freely).

To solve the problem above, Xiao Y *et al.* propose PrivacyGuard [70] to ensure that the data owners can control their data's access and use. In PrivacyGuard, the data owners can create a smart contract to set usage policies for their data (e.g., data consumers, data usage conditions, and the purpose of the data), which is always encrypted and stored in the cloud. When a data consumer uses the owner's data for an allowed purpose, the smart contract and the data will be loaded in an enclave provided by an off-chain contract execution party. After attestation, the owner will send the data key to the enclave using the secure communication channel, and the enclave will decrypt the data to execute the computing. At the moment, the usage records of data have been stored in the blockchain, which ensures the unchangeability and traceability. Finally, the result will be sent to the data consumer. In PrivacyGuard, the privacy of data is not revealed because the data are only decrypted in the enclave. Meanwhile, the smart contracts limit the abuse of the data, which guarantees that the data consumers will use the data properly. Moreover, the execution of smart contracts depends on the off-chain TEE, which improves efficiency. However, attackers can launch side-channel attacks to obtain private information in PrivacyGuard. Moreover, the off-chain contract execution party is trusted completely and consequently suffers from the single-point attack.

3) DATA SECURITY IN IoT

IoT devices have limited storage and computing capabilities, which usually leverage external services provided by the third party (e.g., the cloud). However, service providers often violate data privacy policies and exploit users' data for unauthorized purposes. To solve the problem, Ayoade G *et al.* [71] propose a decentralized IoT data management solution, TEE-based IoT database (TID). TID uses the hybrid storage architecture, which stores data digests on the blockchain and stores the original data on the TEE-enabled storage platform. The data will be encrypted by TEE to protect the off-chain data. In TID, the IoT devices first need to be registered in the blockchain through the IoT gateway, and the data's access control policy will be set in a smart contract. When a user needs to read the data, he or she first proves his or her authority to the blockchain. The smart contract will check if the user can access the data from the device using the device id and the address of the user. If the user passes the check, the smart contract will return the data digest, which is used to obtain the corresponding data from the TEE-enabled storage platform. Finally, the user will obtain the data. In general, the smart contract achieves the decentralized data access control for TID, and the TEE-enabled storage platform solves the privacy and security issues of storing on the third-party platform, which provides a new solution for the application of blockchain in IoT.

VIII. DISCUSSION AND FUTURE DIRECTIONS

In this section, we summarize the SGX-based schemes described above and discuss possible solutions to eliminate the disadvantages of SGX. Finally, we propose some future research directions that integrate blockchain and SGX.

A. SUMMARY AND DISCUSSION

According to Section III, we summarize the SGX-based schemes mentioned in this article, see Table 5. The table shows that all schemes use the secure container and the attestation, which are the basic functions provided by TEE technology. The secure container ensures that the SGX-based party will follow the protocol definitions and pay honest users for their synchronization service without complex cryptographic protocol (i.e., SGX can be used to simplify the protocol process). For example, benefiting from the advantage, Obscuro [67] does not need to employ the punishment mechanism to prevent the mixer from breaking mix protocol and use the additional cryptographic schemes to guarantee the anonymity of users. The attestation can help the mistrust parties building trust with the low cost. For example, Airtnt [69] builds the trust between users and computing nodes, and the cost is simply to perform calculations in the enclave and complete the attestation process. Data sealing can help an enclave to swap data with another enclave or to ease memory pressure. Few schemes describe the function because it is generally relevant to the code implementation. Moreover, some schemes employ the trusted functions to improve security. For example, Teechain guarantees the state freshness using the trusted monotonic counters to prevent against **roll back attacks**. However, those schemes more or less suffer from the disadvantages of SGX, which indicates

TABLE 5. The functions and disadvantages of SGX-based schemes.

Proposal	Functions of SGX					Disadvantages of SGX			
	Secure container	Intra-attestation	Remote attestation	Data sealing	Trusted functions	Limited memery	Availability failures	Side-channel attacks	Single-point attacks
AHL [68]	●	●	●	○	●	○	○	○	○
PoUW [46]	●	●	●	○	○	◐	○	○	●
PoET [47]	●	●	●	○	●	○	○	○	●
PoLK [48]	●	●	●	○	●	○	○	○	●
Hybster [52]	●	●	●	○	●	○	○	○	●
SPoS [55]	●	●	●	○	●	○	○	○	●
Hawk [56]	●	●	●	○	○	○	●	◐	●
ShadowEth [57]	●	●	●	○	○	○	○	◐	●
Ekiden [58]	●	●	●	○	○	○	○	○	○
FastKitten [61]	●	●	●	○	○	●	○	◐	●
TC [62]	●	●	●	○	○	○	○	○	◐
Teechain [64]	●	●	●	○	●	○	○	○	○
Tesseract [65]	●	●	●	○	○	●	●	○	●
BITE [66]	●	●	●	●	○	○	○	○	○
Obscuro [67]	●	●	●	○	○	○	●	○	●
Airtnt [69]	●	●	●	○	○	○	○	○	●
PrivacyGuard [70]	●	●	●	○	○	○	●	●	●
TID [71]	●	●	●	●	○	●	○	○	●

● means that the scheme uses the function or suffers from the disadvantage. ○ means that the scheme does not use the function or suffer from the disadvantage. ◐ means that the scheme is influenced by the disadvantage, but not fatal.

that there still exist challenges to achieve a stable, robust, and practical usage of an integrated blockchain and SGX. We discuss the possible solutions to eliminate the disadvantages of SGX next.

The limitation of enclave memory is a natural barrier that affects the scalability of the solutions using SGX. In particular, it will result in a performance load that oversubscribes the secure memory. Therefore, researchers have to design reasonable ways to decrease the trusted code loaded in the enclave with security in mind. To this end, some studies can be used to improve the application's performance in the enclave. For example, sgx-perf [88] is an analysis tool to perform fined-grained profiling of performance-critical events in the enclave. This tool offers some recommendations on how to improve the application in the enclave. Moreover, multiple SGX nodes maintained by the consistent protocol can be employed to improve the solutions.

Availability failures are another factor that could sway SGX-based schemes. These failures can result in the SGX-base nodes becoming completely or partially disabled. To tolerate such failures, a solution is to keep a single node based on SGX stateless, which ensures that the nodes are interchangeable (i.e., an enclave instance can be replaced by another). To this end, the scheme may need to store the persistent state information in the trusted role, e.g., the blockchain or a committee composed of SGX-based nodes. The solution can also be used to alleviate single-point attacks.

Side-channel attacks can reveal the secret in an enclave, which may break the security and privacy of SGX-based schemes. Thus, the design of SGX-based schemes should consider side-channel attacks. Some studies have provided some solutions to defend against the side-channel attack in SGX, e.g., ORAM [85], addressing space layout randomization [86] and oblivious store operations [87]. Researchers can try to combine those solutions with their schemes to improve security and privacy.

B. FUTURE DIRECTIONS

1) MORE SCALABILITY ISSUES IN BLOCKCHAIN

Scalability is a critical barrier that limits the blockchain's scope of application. We analyze the scalability issue from the perspectives of the storage and the network.

Regarding storage, there are fewer SGX-based schemes to improve the storage capacity of the blockchain. We hold the opinion that there are broad prospects for exploiting SGX technology to solve the storage problem. For example, a simple idea for how to employ SGX to relieve the storage pressure of the blockchain is that the SGX-based roles (e.g., a centralized SGX-enabled server or a distributed platform consisting of SGX-enabled nodes) can be used to stores the original data off-chain (the data can be encrypted by the enclave and stored outside the enclave) and the data digests can be stored in the blockchain. The privacy of the data can be protected by SGX because the enclave can return the data secretly by the secure channel. In this way, the SGX looks just like ''an encryption machine''. Moreover, some smart contracts (e.g., the data access control) can be performed by the enclave off-chain, which can improve efficiency. Thus, the topic of using SGX to enhance the storage capacity of the blockchain is worth studying.

Regarding the network, as we mentioned in Section II, it is still a feasible direction to optimize the blockchain network

structure to speed up the spread of blocks and transactions by using SGX. For example, current blockchain systems predominantly use Gossip [91] as the block broadcast mechanism. Gossip uses the random network topology, which results in long broadcast time completing the message cover of all nodes in the network because the peer discovery process on the broadcast is random. The possible idea is that dividing the blockchain nodes into different groups based on geographic location. Each group has a leader node. The block will be broadcast in the initial group, and the leader node of the group will broadcast the block to the other leader nodes. Finally, the leader nodes broadcast the block in their groups. Adopting geographical proximity groups will relieve traffic congestion and reduce broadcast time. In this scenario, SGX can be used to protect the security of the leader nodes.

2) MORE SECURITY ISSUES IN CONSENSUS ALGORITHMS

The security of the blockchain is dependent upon the security of its underlying implementation in software and hardware as well as the protocols and messages required for it. The mechanism of consensus, while considered a way to ensure fairness and trust in an untrusted system, provides a target for would-be attackers. In Section V, we discuss the scheme using SGX to eliminate the attacks in existing consensus algorithms (e.g., SPoS). However, SGX has the potential to solve more attacks in the consensus layer. For example, the **selfish mining attack** is an attack strategy in PoW that permits the miner to obtain more rewards of creating blocks. To implement the **selfish mining attack**, a miner does not broadcast the generated block immediately and continues to generate the next block in a round until a new block is broadcasted. Obviously, the miner based on SGX can be forced to broadcast the generated block, which prevents the **selfish mining attack**. Thus, it is a potential direction that enables solving more attacks in the existing consensus algorithm by using SGX.

Moreover, the SGX-based consensus algorithms (e.g., PoET, PoLK) have the extra problem, as discussed in Section V. Due to the accessibility of SGX-support devices, those schemes have **the stale chip problem**, which will result in the centralization of rights. Thus, ensure the decentralized feature of the SGX-based consensus algorithms remains an open challenge.

3) TRUST ASSUMPTIONS IN CRYPTOGRAPHIC SCHEMES

Currently, some well-known cryptographic schemes are used to protect privacy in the blockchain. However, those schemes need a trusted party. For example, Zerocash [89] employs the zk-SNARKs protocol to protect privacy, which requires that a trusted party initializes the system. Hawk, mentioned in Section VI, uses the same zero-knowledge proofs protocol; thus, it needs a trusted manager to protect privacy. Moreover, Maxwell [90] implements the confidential transaction for cryptocurrency systems (e.g., Bitcoin) using the homomorphic encryption scheme, which requires a semi-honest party to calculate the ciphertext. In those scenarios, an SGX-based role is an excellent choice to serve as a trusted party. Thus, a potential direction is to exploit SGX to remove the trust assumptions of cryptographic schemes used in the blockchain.

4) SUPERVISION IN BLOCKCHAIN

Privacy protection schemes (e.g., mix protocol [77]) is the significant research direction of blockchain, which protects the privacy of users. However, those schemes may provide a safeguard for unlawful acts, e.g., scenarios where offenders can anonymously achieve money laundering by anonymous digital currency. Thus, a splendid blockchain system should supervise criminal acts while providing privacy protection. In this scenario, the SGX-based role can act as a qualified supervisor. For example, an SGX-based supervisor stores the secret that can track the real identity of users and holds the deposits of every user. When malicious behavior is discovered, the victim can provide relevant information to the SGX-based supervisor. The SGX-based supervisor then traces the real identity of the offender and performs the corresponding punishment. Thus, it is a potential direction that improving the supervision of blockchain using SGX.

IX. CONCLUSION

Intel SGX technology, featured with its security and confidentiality, have shown the great potential to solve the problems in the blockchain area. This article mainly investigated the application of SGX in the blockchain field. In this article, we first present the blockchain layers and summarize the functions, advantages, and disadvantages of SGX technology. Then, we analyze the problems and the SGX-based solutions in the data layer, the consensus layer, the contract layer, and the application layer. Finally, we summarize the SGX-based schemes and discuss the remaining challenges and future research directions on integrated blockchain and SGX schemes. We hope to have provided a survey that is able to elicit more discussion and inspiration regarding the convergence of the blockchain and SGX. As our future work, we will deploy the observations and analysis whose main focus is the effective works to solve the SGX's deficiencies inherited by the SGX-based works in the blockchain area.

ACKNOWLEDGMENT

The authors would like to thank the editors and the anonymous reviewers for their valuable comments. *(Zijian Bao and Qinghao Wang contributed equally to this work.)*

REFERENCES

[1] S. Nakamoto. (Jan. 2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: https://archive.is/rMBtV

[2] V. Buterin. (2014). *A Next-Generation Smart Contract and Decentralized Application Platform*. White Paper. [Online]. Available: https://www. weusecoins.com/assets/pdf/library/Ethereum_white_paper-a_next_ generation_smart_contract_and_decentralized_application_platformvitalik-buterin.pdf

[3] (2018). *Enterprise Blockchain Revenue Prediction*. [Online]. Available: https://www.businesswire.com/news/home/20180828005038/en/Enterprise-Blockchain-Revenue-Surpass-20-Billion-2025

[4] A. Miller, A. Juels, E. Shi, B. Parno, and J. Katz, "Permacoin: Repurposing bitcoin work for data preservation," in *Proc. IEEE Symp. Secur. Privacy*, May 2014, pp. 475–490.

[5] S. King, "Primecoin: Cryptocurrency with prime number proof-of-work," Self-Published Papers. Accessed: Nov. 3, 2018. [Online]. Available: http://primecoin.io/bin/primecoin-paper.pdf
[6] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten, "SoK: Research perspectives and challenges for bitcoin and cryptocurrencies," in *Proc. IEEE Symp. Secur. Privacy*, May 2015, pp. 104–121.
[7] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, and S. Savage, "A fistful of bitcoins: Characterizing payments among men with no names," *Commun. ACM*, vol. 59, no. 4, pp. 86–93, Apr. 2016.
[8] D. Ron and A. Shamir, "Quantitative analysis of the full Bitcoin transaction graph," in *Proc. 17th Int. Conf. Financial Cryptogr. Data Secur.*, 2013, pp. 6–24.
[9] Intel, "Software guard extensions programming reference, revision 2," Intel, Santa Clara, CA, USA, Tech. Rep., 2014.
[10] V. Costan and S. Devadas, "Intel SGX explained," IACR Cryptol. ePrint Arch., Tech. Rep., 2016, vol. 2016, no. 86, pp. 1–118. [Online]. Available: https://eprint.iacr.org/2016/086/20170221:054353
[11] S. Arnautov, B. Trach, F. Gregor, T. Knauth, A. Martin, C. Priebe, J. Lind, D. Muthukumaran, D. O'keeffe, M. L. Stillwell, and D. Goltzsche, "SCONE: Secure Linux containers with Intel SGX," in *Proc. OSDI*, 2016, pp. 689–703.
[12] W. Wang, G. Chen, X. Pan, Y. Zhang, X. Wang, V. Bindschaedler, H. Tang, and C. A. Gunter, "Leaky cauldron on the dark land: Understanding memory side-channel hazards in SGX," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 2421–2434.
[13] S. Lee, M.-W. Shih, P. Gera, T. Kim, H. Kim, and M. Peinado, "Inferring fine-grained control flow inside SGX enclaves with branch shadowing," in *Proc. 26th USENIX Secur. Symp., USENIX Secur.*, 2017, pp. 557–574.
[14] F. Brasser, U. Müller, A. Dmitrienko, K. Kostiainen, S. Capkun, and A.-R. Sadeghi, "Software grand exposure: SGX cache attacks are practical," in *Proc. 11th USENIX Workshop Offensive Technol.*, Aug. 2017, pp. 1–10.
[15] G. Zyskind, O. Nathan, and A. '. Pentland, "Decentralizing privacy: Using blockchain to protect personal data," in *Proc. IEEE Secur. Privacy Workshops*, May 2015, pp. 180–184.
[16] E. Androulaki *et al.*, "Hyperledger fabric: A distributed operating system for permissioned blockchains," in *Proc. 13th EuroSys Conf.*, Apr. 2018, p. 30.
[17] R. Yang, F. R. Yu, P. Si, Z. Yang, and Y. Zhang, "Integrated blockchain and edge computing systems: A survey, some research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1508–1532, 2nd Quart., 2019.
[18] S. Kim, Y. Shin, J. Ha, T. Kim, and D. Han, "A first step towards leveraging commodity trusted execution environments for network applications," in *Proc. 14th ACM Workshop Hot Topics Netw. HotNets-XIV*, 2015, pp. 1–7.
[19] S. Kim, J. Han, J. Ha, T. Kim, and D. Han, "SGX-tor: A secure and practical tor anonymity network with SGX enclaves," *IEEE/ACM Trans. Netw.*, vol. 26, no. 5, pp. 2174–2187, Oct. 2018.
[20] P. L. Seijas, S. J. Thompson, and D. McAdams, "Scripting smart contracts for distributed ledger technology," IACR Cryptol. ePrint Arch., Tech. Rep. 2016 1156, 2016.
[21] C. Cachin, "Architecture of the hyperledger blockchain fabric," in *Proc. Workshop Distrib. Cryptocurrencies Consensus Ledgers (DCCL)*, Chicago, IL, USA, Jul. 2016, pp. 1–4.
[22] D. Larimer, "Delegated proof-of-stake (DPOS)," BitShare, White Paper, 2014. [Online]. Available: http://107.170.30.182/security/delegated-proofof-stake.php
[23] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim, "A survey on consensus mechanisms and mining strategy management in blockchain networks," *IEEE Access*, vol. 7, pp. 22328–22370, 2019.
[24] T. M. Fernández-Caramés and P. Fraga-Lamas, "A review on the use of blockchain for the Internet of Things," *IEEE Access*, vol. 6, pp. 32979–33001, 2018.
[25] K. Okupski, "Bitcoin developer reference," Dept. Comput. Sci., Technische Univ. Eindhoven, Eindhoven, The Netherlands, Tech. Rep., Jul. 2016. [Online]. Available: http://enetium.com/resources/Bitcoin.pdf
[26] A. Back, M. Corallo, L. Dashjr, M. Friedenbach, G. Maxwell, A. Miller, A. Poelstra, J. Timón, and P. Wuille, "Enabling blockchain innovations with pegged sidechains," Blockstream Inc., Victoria, BC, Canada, Tech. Rep. 5620e43, Oct. 2014. [Online]. Available: http://kevinriggen.com/files/sidechains.pdf
[27] G. Maxwell, "CoinJoin: Bitcoin privacy for the real world," in *Proc. Post Bitcoin Forum*, 2013. [Online]. Available: https://bitcointalk.org/index.php
[28] L. Valenta and B. Rowan, "Blindcoin: Blinded, accountable mixes for bitcoin," in *Financial Cryptography and Data Security. FC* (Lecture Notes in Computer Science), vol. 8976, M. Brenner, N. Christin, B. Johnson, and K. Rohloff, Eds. Berlin, Germany: Springer, 2015, doi: 10.1007/978-3-662-48051-9_9.
[29] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, "A secure sharding protocol for open blockchains," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2016, pp. 17–30.
[30] J. Poon and T. Dryja, "The bitcoin lightning network: Scalable off-chain instant payments," in *Proc. Lightning Labs*, San Francisco, CA, USA, vol. 9, Jan. 2016, p. 14.
[31] C. Decker and R. Wattenhofer, "A fast and scalable payment network with bitcoin duplex micropayment channels," in *Proc. Symp. Self-Stabilizing Syst.*, Cham, Switzerland: Springer, 2015, pp. 3–18.
[32] A. Tapscott and D. Tapscott, "How blockchain is changing finance," *Harvard Bus. Rev.*, vol. 1, no. 9, pp. 1–5, 2017.
[33] E. Hofmann, U. Strewe, and N. Bosia, *Supply Chain Finance and Blockchain Technology. The Case of reverse Securitisaton*. Berlin, Germany: Springer, 2018.
[34] B.-K. Zheng, L.-H. Zhu, M. Shen, F. Gao, C. Zhang, Y.-D. Li, and J. Yang, "Scalable and privacy-preserving data sharing based on blockchain," *J. Comput. Sci. Technol.*, vol. 33, no. 3, pp. 557–567, May 2018.
[35] E. Gaetani, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, and V. Sassone, "Blockchain-based database to ensure data integrity in cloud computing environments," in *Proc. 1st Italian Conf. Cybersecur. (ITASEC)*, 2017, pp. 2–11.
[36] X. Liang, S. Shetty, D. Tosh, C. Kamhoua, K. Kwiat, and L. Njilla, "ProvChain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability," in *Proc. 17th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput. (CCGRID)*, May 2017, pp. 468–477.
[37] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for IoT security and privacy: The case study of a smart home," in *Proc. IEEE Int. Conf. Pervas. Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2017, pp. 618–623.
[38] Y. Zhang and J. Wen, "The IoT electric business model: Using blockchain technology for the Internet of Things," *Peer Peer Netw. Appl.*, vol. 10, no. 4, pp. 983–994, Jul. 2017.
[39] J. Sun, J. Yan, and K. Z. K. Zhang, "Blockchain-based sharing services: What blockchain technology can contribute to smart cities," *Financial Innov.*, vol. 2, no. 1, p. 26, Dec. 2016, doi: 10.1186/s40854-016-0040-y.
[40] F. McKeen, I. Alexandrovich, A. Berenzon, C. V. Rozas, H. Shafi, V. Shanbhogue, and U. R. Savagaonkar, "Innovative instructions and software model for isolated execution," in *Proc. 2nd Int. Workshop Hardw. Architectural Support Secur. Privacy HASP*, 2013, p. 10.
[41] I. Anati, S. Gueron, S. P. Johnson, and V. R. Scarlata, "Innovative technology for CPU based attestation and sealing," in *Proc. ACM 2nd Int. Workshop Hardw. Archit. Support Secur. Privacy*, New York, NY, USA, vol. 13, 2013, pp. 1–7.
[42] M. Hoekstra, R. Lal, P. Pappachan, V. Phegade, and J. Del Cuvillo, "Using innovative instructions to create trustworthy software solutions," in *Proc. 2nd Int. Workshop Hardw. Architectural Support Secur. Privacy HASP*, 2013, Art. no. 2487726.
[43] E. Brickell and J. Li, "Enhanced privacy id: A direct anonymous attestation scheme with enhanced revocation capabilities," in *Proc. ACM Workshop Privacy Electron. Soc. WPES*, 2007, pp. 21–30.
[44] K. Krawiecka, A. Kurnikov, A. Paverd, M. Mannan, and N. Asokan, "SafeKeeper: Protecting Web passwords using trusted execution environments," in *Proc. World Wide Web Conf. World Wide Web WWW*, 2018, pp. 349–358.
[45] J. R. Douceur, "The sybil attack," in *Proc. Int. Workshop Peer Peer Syst. (IPTPS)*, 2002, pp. 251–260.
[46] F. Zhang, I. Eyal, R. Escriva, A. Juels, and R. van Renesse, "REM: Resource-efficient mining for blockchains," in *Proc. 26th USENIX Secur. Symp. (USENIX Secur.)*, Vancouver, BC, USA, Aug. 2017, pp. 1427–1444.
[47] G. Prisco, *Intel Develops Sawtooth Lake: Distributed Ledger Technology for the Hyperledger Project*. Nashville, TN, USA: Bitcoin Magazine, 2016.
[48] M. Milutinovic, W. He, H. Wu, and M. Kanwal, "Proof of luck: An efficient blockchain consensus protocol," in *Proc. 1st Workshop Syst. Softw. Trusted Execution SysTEX*, 2016, p. 2.
[49] M. Abd-El-Malek, G. R. Ganger, G. R. Goodson, M. K. Reiter, and J. J. Wylie, "Fault-scalable Byzantine fault-tolerant services," *SIGOPS Oper. Syst. Rev.*, vol. 39, no. 5, p. 59, Oct. 2005.

[50] G. S. Veronese, M. Correia, A. N. Bessani, L. C. Lung, and P. Verissimo, "Efficient byzantine fault-tolerance," *IEEE Trans. Comput.*, vol. 62, no. 1, pp. 16–30, Jan. 2013.
[51] T. Distler, C. Cachin, and R. Kapitza, "Resource-efficient byzantine fault tolerance," *IEEE Trans. Comput.*, vol. 65, no. 9, pp. 2807–2819, Sep. 2016.
[52] J. Behl, T. Distler, and R. Kapitza, "Hybrids on steroids: SGX-based high performance BFT," in *Proc. 12th Eur. Conf. Comput. Syst.*, Apr. 2017, pp. 222–237, doi: 10.1145/3064176.3064213.
[53] J. Martinez. (2018). *Understanding Proof of stake: The Nothing at Stake Theory*. [Online]. Available: https://medium.com/coinmonks/understanding-proof-of-stake-the-nothing-at-stake-theory-1f0d71bc027
[54] P. Gazi, A. Kiayias, and A. Russell, "Stake-bleeding attacks on Proof-of-Stake blockchains," in *Proc. Crypto Valley Conf. Blockchain Technol. (CVCBT)*, Jun. 2018, pp. 85–92.
[55] W. Li, S. Andreina, J.-M. Bohli, and G. Karame, "Securing proof-of-stake blockchain protocols," in *Proc. Int. Workshops Data Privacy Manage., Cryptocurrencies Blockchain Technol. (ESORICS)*, Oslo, Norway, Sep. 2017, pp. 297–315.
[56] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2016, pp. 839–858.
[57] R. Yuan, Y.-B. Xia, H.-B. Chen, B.-Y. Zang, and J. Xie, "ShadowEth: Private smart contract on public blockchain," *J. Comput. Sci. Technol.*, vol. 33, no. 3, pp. 542–556, May 2018.
[58] R. Cheng, F. Zhang, J. Kos, W. He, N. Hynes, N. Johnson, A. Juels, A. Miller, and D. Song, "Ekiden: A platform for confidentiality-preserving, trustworthy, and performant smart contracts," in *Proc. IEEE Eur. Symp. Secur. Privacy (EuroS&P)*, Jun. 2019, pp. 185–200.
[59] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung, "Proactive secret sharing OR: How to cope with perpetual leakage," in *Proc. CRYPTO*, Santa Barbara, CA, USA: Springer, 1995, pp. 339–352.
[60] D. Schultz, B. Liskov, and M. Liskov, "MPSS: Mobile proactive secret sharing," *ACM Trans. Inf. Syst. Secur. (TISSEC)*, vol. 13, no. 4, p. 34, 2010.
[61] P. Das, L. Eckey, T. Frassetto, D. Gens, K. Hostáková, P. Jauernig, S. Faust, and A.-R. Sadeghi, "Fastkitten: Practical smart contracts on bitcoin," in *Proc. 28th USENIX Secur. Symp. (USENIX Secur.)*, 2019, pp. 801–818.
[62] F. Zhang, E. Cecchetti, K. Croman, A. Juels, and E. Shi, "Town crier: An authenticated data feed for smart contracts," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2016, pp. 1–20.
[63] F. Zhang, E. Cecchetti, K. Croman, A. Juels, and E. Shi. (2017). *Town Crier*. [Online]. Available: http://www.town-crier.org/
[64] J. Lind, O. Naor, I. Eyal, F. Kelbert, E. G. Sirer, and P. Pietzuch, "Teechain: A secure payment network with asynchronous blockchain access," in *Proc. 27th ACM Symp. Operating Syst. Princ.*, Oct. 2019, pp. 63–79.
[65] I. Bentov, Y. Ji, F. Zhang, L. Breidenbach, P. Daian, and A. Juels, "Tesseract: Real-time cryptocurrency exchange using trusted hardware," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2019, pp. 1521–1538.
[66] S. Matetic, K. Wst, M. Schneider, K. Kostiainen, G. Karame, and S. Capkun, "BITE: Bitcoin lightweight client privacy using trusted execution," in *Proc. 28th USENIX Conf. Secur. Symp.*, Santa Clara, CA, USA, Aug. 2019, pp. 783–800.
[67] M. Tran, L. Luu, M. S. Kang, I. Bentov, and P. Saxena, "Obscuro: A bitcoin mixer using trusted execution environments," in *Proc. 34th Annu. Comput. Secur. Appl. Conf.*, Dec. 2018, pp. 692–701.
[68] H. Dang, T. T. A. Dinh, D. Loghin, E.-C. Chang, Q. Lin, and B. C. Ooi, "Towards scaling blockchain systems via sharding," in *Proc. Int. Conf. Manage. Data SIGMOD*, 2019, pp. 123–140.
[69] M. Al-Bassam, A. Sonnino, M. Król, and I. Psaras, "Airtnt: Fair exchange payment for outsourced secure enclave computations," 2018, *arXiv:1805.06411*. [Online]. Available: http://arxiv.org/abs/1805.06411
[70] Y. Xiao, N. Zhang, J. Li, W. Lou, and Y. T. Hou, "PrivacyGuard: Enforcing private data usage control with blockchain and attested off-chain contract execution," 2019, *arXiv:1904.07275*. [Online]. Available: http://arxiv.org/abs/1904.07275
[71] G. Ayoade, V. Karande, L. Khan, and K. Hamlen, "Decentralized IoT data management using BlockChain and trusted execution environment," in *Proc. IEEE Int. Conf. Inf. Reuse Integr. (IRI)*, Jul. 2018, pp. 15–22.
[72] Profitgenerator. (2017). *EtherDelta*. [Online]. Available: https://steemit.com/ethereum/@profitgenerator/etherdelta-decentralized-token-exchange
[73] B. Bünz, L. Kiffer, L. Luu, and M. Zamani, "FlyClient: Super-light clients for cryptocurrencies," in *Proc. IEEE Symp. Secur. Privacy (SP)*, San Francisco, CA, USA, 2020, pp. 928–946, doi: 10.1109/SP40000.2020.00049.
[74] (2018). *BitcoinJ*. [Online]. Available: https://bitcoinj.github.io/
[75] Ethereum. (May 2019). *Light Ethereum Subprotocol (LES)*. Accessed: Oct. 19, 2019. [Online]. Available: https://github.com/ethereum/devp2p/blob/master/caps/les.md
[76] E. Stefanov, M. van Dijk, E. Shi, C. Fletcher, L. Ren, X. Yu, and S. Devadas, "Path ORAM: An extremely simple oblivious RAM protocol," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. CCS*, 2013, pp. 299–310.
[77] G. Bissias, A. P. Ozisik, B. N. Levine, and M. Liberatore, "Sybil-resistant mixing for bitcoin," in *Proc. 13th Workshop Privacy Electron. Soc. WPES*, 2014, pp. 149–158.
[78] J. H. Ziegeldorf, F. Grossmann, M. Henze, N. Inden, and K. Wehrle, "CoinParty: Secure multi-party mixing of bitcoins," in *Proc. 5th ACM Conf. Data Appl. Secur. Privacy CODASPY*, 2015, pp. 75–86.
[79] M. Zamani, M. Movahedi, and M. Raykova, "RapidChain: Scaling blockchain via full sharding," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Jan. 2018, pp. 931–948.
[80] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving secure, scalable, and fine-grained data access control in cloud computing," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.
[81] N. Li, T. Li, and S. Venkatasubramanian, "T-closeness: Privacy beyond K-anonymity and L-diversity," in *Proc. IEEE 23rd Int. Conf. Data Eng.*, Istanbul, Turkey, 2007, pp. 106–115, doi: 10.1109/ICDE.2007.367856.
[82] T. Nolan. (2019). *Alt Chains and Atomic Transfers*. [Online]. Available: https://bitcointalk.org/index.php?topic=193281.msg2224949#msg2224949
[83] I. Intel. (2019). *Trusted Time and Monotonic Counters with Intel Software Guard Extensions Platform Services*. [Online]. Available: https://software.intel.com/content/www/us/en/develop/download/trusted-time-and-monotonic-counters-with-intel-software-guard-extensions-platform-services.html
[84] T. Alves and D. Felton, "Trustzone: Integrated hardware and software security," ARM, Cambridge, U.K., White Paper, 2004, vol. 3.
[85] A. Ahmad, B. Joe, Y. Xiao, Y. Zhang, I. Shin, and B. Lee, "OBFUSCURO: A commodity obfuscation engine on intel SGX," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2019, pp. 1–15.
[86] J. Seo, B. Lee, S. Kim, M.-W. Shih, I. Shin, D. Han, and T. Kim, "SGX-shield: Enabling address space layout randomization for SGX programs," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2017, pp. 1–15.
[87] A. Rane, C. Lin, and M. Tiwari, "Raccoon: Closing digital sidechannels through obfuscated execution," in *Proc. 24th USENIX Secur. Symp.*, Washington, DC, USA: USENIX Association, Aug. 2015, pp. 431–446.
[88] N. Weichbrodt, P.-L. Aublin, and R. Kapitza, "Sgx-perf: A performance analysis tool for intel SGX enclaves," in *Proc. 19th Int. Middleware Conf.*, Nov. 2018, pp. 201–213.
[89] E. B. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized anonymous payments from bitcoin," in *Proc. IEEE Symp. Secur. Privacy*, May 2014, pp. 459–474.
[90] G. Maxwell. (2016). *Confidential Transactions*. [Online]. Available: https://people.xiph.org/ greg/confidential_values.txt
[91] J. Leitão, J. Pereira, and L. Rodrigues, "Gossip-based broadcast," in *Handbook of Peer-to-Peer Networking*, X. Shen, H. Yu, J. Buford, and A. Akon, Eds. Boston, MA, USA: Springer, 2010, doi: 10.1007/978-0-387-09751-0_29.

ZIJIAN BAO received the bachelor's and master's degrees from Northeastern University, China. He is currently a Researcher with the Oxford-Hainan Blockchain Research Institute. His research interests include blockchain, cryptography, and trusted hardware.

QINGHAO WANG is currently pursuing the master's degree with Northeastern University, China. His current research interests include blockchain and Intel SGX technology.

WENBO SHI received the M.S. and Ph.D. degrees from Inha University, Incheon, South Korea, in 2007 and 2010, respectively. He is currently a Professor with Northeastern University, Qinhuangdao. His research interests include the cryptographic protocol, cloud computing security, artificial intelligence security, data security, privacy protection, and denial of service attack defense.

LEI WANG received the bachelor's degree from Shanghai Jiao Tong University and the Ph.D. degree from The University of Electro-Communications. He is currently a Research Professor with Shanghai Jiao Tong University. His research interests include cryptography and blockchain.

HONG LEI received the bachelor's degree in technology and apparatus for measurement and control and the master's degree in testing and measurement technology and instrument from Beihang University, Beijing, China, in 2006 and 2009, respectively, and the Ph.D. degree from the Department of Electrical and Computer Engineering, Michigan State University, in 2015. He continued as a Postdoctoral Researcher with the Smart Microsystems Laboratories, where he investigated the modeling and design of artificial lateral lines systems for underwater robots. He joined Schweitzer Engineering Laboratories, in March 2016, and then joined the Department of Electrical and Computer Engineering, Portland State University, as a Tenure-Track Assistant Professor, in July 2018. He is currently an Associate Dean of the Oxford-Hainan Blockchain Research Institute. His research interests include TEE and blockchain.

BANGDAO CHEN received the bachelor's degree from the Department of Computer Science, Shanghai Jiao Tong University, and the M.Sc. and D.Phil. degrees in computer science from the University of Oxford. His main research interests include decentralized identification and authentication, payment security, and communication security. His current research interests include cyber security, blockchain related technology research, and product development.