Security and Trust in Blockchains: Architecture, Key Technologies, and Open Issues

Peiyun Zhang, *Senior Member, IEEE*, and MengChu Zhou, *Fellow, IEEE*

***Abstract*— As a new promising distributed technology, blockchains have been widely applied since its inception. Its decentralization feature reduces the reliance on the trusted authorities and third parties. It can well solve the problem of data being tampered and increase data sharing. However, a blockchain system faces various security and trust issues, such as attacks against consensus mechanisms and propagation processes, which may make it store malicious information or delay data propagation. The work discusses the basic architecture of blockchains as well as its potential security and trust issues at data, network, consensus, smart contract, and application layers. Then, the related literature work is analyzed in terms of the issues at these layers. Some open issues are presented and discussed.**

***Index Terms*— Accounting, blockchains, consensus, data storage, propagation, security and trust, smart contract, verification.**

I. Nomenclature

BFT	Byzantine fault tolerance.
DAG	Directed acyclic graph.
DBFT	Delegated BFT.
D-H	Diffie–Hellman.
DoS	Denial-of-Service.
DPoS	Delegated Proof of Stake.
IoT	Internet of Things.
MitM	Man-in-the-Middle.
MPT	Merkle PatriciaTree.
N@S	Nothing at Stake.
PBFT	Practical BFT.
PoD	Proof of Delivery.
PoET	Proof of Elapsed Time.
PoL	Proof of Luck.
PoS	Proof of Stake.
PoT	Proof of Trust.
PoW	Proof of Work.
RSA	Ron–Shamir–Adleman.
SPV	Simplified payment verification.
SVM	Support vector machine.
TEEs	Trusted execution environments.
ZKP	Zero-knowledge proof.

Manuscript received January 7, 2020; revised April 16, 2020; accepted April 21, 2020. Date of publication May 19, 2020; date of current version June 10, 2020. This work was supported by the National Natural Science Foundation of China under Grant 61872006 and Grant 61472005. *(Corresponding author: MengChu Zhou.)*

Peiyun Zhang is with the School of Computer and Information, Anhui Normal University, Wuhu 241002, China (e-mail: zpyanu@ahnu.edu.cn).

MengChu Zhou is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: zhou@njit.edu).

Digital Object Identifier 10.1109/TCSS.2020.2990103

II. Introduction

A. Blockchains

A CENTRALIZED trusted party is often unavailable for data sharing due to tampered data and single point failures under external attacks [1]. Blockchain technology rightly emerges for decentralized data sharing with tamper-resistance across a (large) peer-to-peer network [2] . A typical blockchain system has multiple nodes without fully trusting each other. Some nodes show Byzantine behaviors, but the majority is trustworthy. The nodes record transactions and, at the same time, maintain shared global states. A blockchain is often considered as a distributed ledger. Each peer node may maintain a copy of the ledger with blocks.

Blockchains can safely and permanently store all kinds of certificates, licenses, registration forms, and records issued by government agencies. It can be easy to prove the authenticity and existence of certain data at any time. Blockchains can reduce the cost caused by middlemen and intermediaries and build trust.

B. Research Motivation

As a cross-domain technology, blockchain technology has attracted wide attention from academia to business circles since 2014. It has been applied to many fields besides encrypted currency, such as the IoT, medical treatment, finance, security, and logistics.

The International Organization for Standardization has also recently launched a technical committee on blockchain and distributed ledger technologies. A number of new groups have been formed to commit to blockchain technology, such as Blockchain Community Group. There are some typical blockchain systems, e.g., Ethereum, IBM HyperLedger Fabric, Factom, and Ripple. At present, Ethereum blockchain systems have a friendly interface for smart contracts. Based on contracts, a variety of distributed applications (DApp) can be built, which greatly reduces the threshold to develop blockchain applications. The existing Ethereum smart contract technology provides a good software technology foundation for the development of blockchain systems. Some blockchain technology companies, e.g., MetaX, have committed to the development and adoption of open platforms for the digital advertising industry. It allows a digital advertising supply chain to coordinate in a scalable, trustworthy, and secure way.

However, various attacks may cause security and trust issues in blockchains. There may be privacy leakage because the

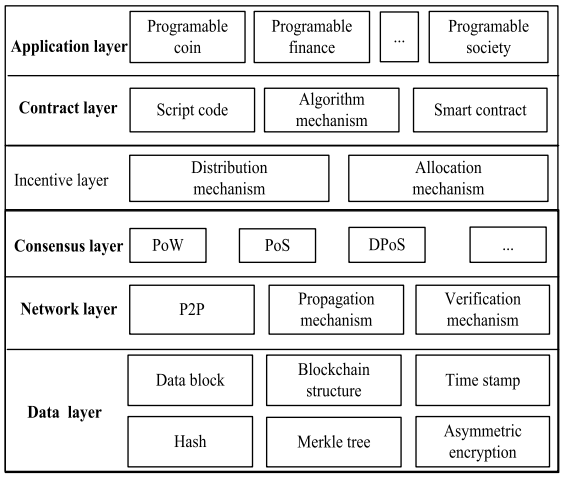

Fig. 1. Basic architecture of blockchains.

information is available to all participants [2] and a possibility of 51% attack in a public blockchain [3]; 51% attack means that the nodes can successfully tamper blockchain data by controlling 51% or more of the computation power of the whole network. Attacks against Bitcoin and Ethereum are surveyed in [4]. There are selfish mine attack, verifier's dilemma, and blockchain forks [3]. The Sybil attack is regarded as the most challenging attack in permission-less blockchain architectures, where an attacker disguises as multiple participants to gain unfair profit [5]. Malicious nodes may implement collusion attacks on blockchain systems [6]. In addition, blockchain systems are currently very fragile due to many implementation bugs in smart contracts. Bugs tend to lower the security and trust of blockchain systems. They may allow attackers to create "modified transactions" and then send currency directly to the attackers' wallet. Some security threats and attacks to blockchains are studied in [7]. Aitzhan and Svetinovic [8] identified 23 attacks and give detailed descriptions of them. Eight representative blockchain systems are researched in terms of their bug characteristics [9]. Many security and trust issues need to be captured and analyzed, which we focus on in this article.

The rest of this article is organized as follows. Section III proposes a basic architecture of blockchains, as well as its related security and trust issues. Section IV summarizes the related work dealing with these issues. Section V gives the related open issues and future work. Finally, in Section VI, some conclusions are drawn.

III. Security and Trust Issues in Blockchain Architecture

A basic architecture of blockchains generally consists of six layers [10], [11]: data, network, consensus, incentive, smart contract, and application layers, as shown in Fig. 1. This work focuses on security and trust issues at the five layers: data, network, consensus, smart contract, and application layers. Different from other surveys, this article presents research problems and their existing solutions in accordance with these five layers.

A. Research Problems

1) Data Layer: A data layer contains the data structure of blockchains. A blockchain has a special data structure for maintaining states and transaction history. Each block includes a hash that binds itself to the previous one. The data layer may face some attacks, such as tamper-proof data, preimage attack, second-preimage attack, and DoS, and suffer from low privacy of data storage.

2) Network Layer: A network layer contains the propagation mechanism of transactions/blocks. Nodes at this layer complete the following service functions [12].

a) *Bulletin Board Services:* Collect broadcast information, verify the legality of information, and release it to the blockchain network.
b) *Relay Services:* Propagate transactions and distribute them among nodes.

To achieve the abovementioned functions, some key capabilities for security are given in blockchain networks [13]. They include privacy protection, identity management, information security, trustworthiness, resistance to attacks, use of advanced cryptography, and decentralized access control.

The network layer may have problems of network delay, propagation and software errors, data missing, DoS, privacy leakage, and hackers' intrusion. It is due to: a) the network environments of blockchain systems are usually complex and b) there may exist malicious nodes or collusion attacks in blockchain networks.

3) Consensus Layer: The role of the consensus layer is to make all nodes agree on the blockchain content in a blockchain system. If a block is appended to a blockchain, the other nodes also append the same block to their copies of the blockchain when needed [14]. Hence, participant nodes in a blockchain system have the same confidence that their ledgers are both consistent and accurate with the same consensus [15].

A consensus for blockchain systems implies that all the honest nodes with high trust values agree on one value/ transaction. The value/transaction is generated by an honest/trusted node [16]. However, if there are malicious nodes or collusion attacks at the consensus layer, they may cause many risks. For example, the consensus mechanisms of PoS and PoW generally suffer from a "N@S" problem [17] and 51% attack [4], respectively. "N@S" means that block generators have nothing to lose by voting for multiple blockchain histories, which can prevent consensus from being achieved [17]. In addition, malicious nodes may verify false information and store it into blocks.

4) Smart Contract Layer: A smart contract layer is mainly about script code and algorithms, which are encapsulated as smart contracts for a blockchain system. Smart contracts denote the business logic of blockchain applications. Malicious participants may intentionally introduce malicious code to smart contracts. Hence, if such contracts are not checked carefully, they may be untrustworthy and potentially even malicious [18].

5) Application Layer: An application layer is mainly about programmable applications for finance, healthcare, and society. At this layer, the privacy and confidentiality of data related

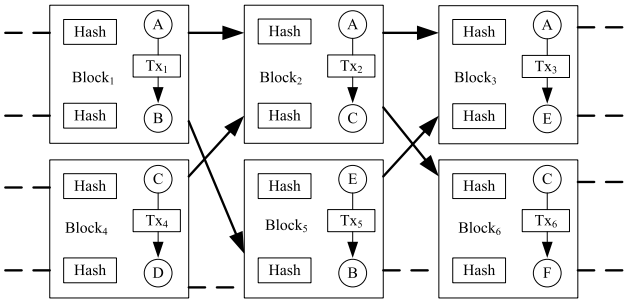

Fig. 2. Block structure of a TrustChain.

to business transactions should be guaranteed [19]. Privacy leakage may take place because every participant has the access to all the information in a public blockchain.

Researchers and practitioners need to solve these problems. Section IV generalizes the existing solutions.

IV. Methods for Enhancing Security and Trust at Blockchain Layers

This section focuses on the existing methods to address typical aforementioned security and trust issues about blockchains.

A. Security and Trust at the Data Layer

For a traditional blockchain, it has a chain-structure with blocks. Each block is connected to the previous one by cryptographic hash. Transaction data generally arranged in a Merkle tree and a timestamp are stored into each block. Its typical example is Bitcoin blockchain [20]. The following basic technologies are adopted to enhance trust and security at the data layer:

1) Asymmetric Encryption: One of the ways for decentralization is independent encryption. It helps prevent data and transactions from being tampered or forged. Asymmetric encryption is integrated into blockchains, including RSA, Elgamal, Rabin, D-H, and elliptic curve encryption algorithm [11]. They are mainly used for information encryption, digital signature, and login authentication.

2) Timestamp: Timestamp can be used to prove the existence of block data. It provides a basis for blockchain-based applications in time-sensitive fields, such as intellectual property registration. Besides the abovementioned basic technologies for blockchains, there are some other new studies to resist attacks to the data layer, such as modifying blockchain structures, resisting second-preimage attack, and adopting separated storage, as follows.

3) Modifying Blockchain Structure: A traditional blockchain structure can be modified to resist attacks, as follows.

a) TrustChain: To improve the ability of tampering-resistance, a Sybil-resistant scalable blockchain, called TrustChain [5], is designed to maintain participants' transactions. Each participant maintains a local blockchain for its own transactions, as shown in Fig. 2. Different from a traditional chain-structure, each block of a TrustChain has two incoming pointers and two outgoing pointers (some are clearly shown in arrow lines and others in dotted lines in instead in brief).

There are six blocks ($\text{Block}_1 \sim \text{Block}_6$) and six participants ($A \sim F$) in Fig. 2. $\text{Tx}_1 \sim \text{Tx}_6$ denote six transactions. When two parties (such as participants A and B in Block_1) transact, they both cryptographically sign transaction Tx_1. Transactions are chained together for blocks to point back to their previous ones. For example, Block_1 and Block_2 are chained together via the pointer at A. The block structure has the following benefits.

1) It can avoid tampering of blockchains because if one participant changes the blocks, the counterpart involved in the same transaction can know that.
2) It can resist double spending and replay because a fraud can be detected by the counterpart.
3) It can resist other forms of attacks, such as hiding blocks, refusal to sign, and whitewashing.

b) Directed acyclic graph: DAG-based distributed ledger technology shows the signs of being particularly good at overcoming the limitations of scalability inherent in chain-structured blockchains [21]. DAG as a graph is different from those with traditional chain structures. It can resist double-spending, penny-spending attack/transaction flood, Sybil attack, MitM attack, DoS attacks, and distribution of software patches. A DAG-based ledger is built from transactions connected through time by using trust-based algorithms. Its mechanisms can be used to monitor, detect, and defend against possible attacks [21].

c) Merkle tree and Merkle PatriciaTree: The Merkle tree is an important data structure for traditional blockchains, such as Bitcoin. It is used to quickly summarize and verify the existence and integrity of block data. The Merkle tree in a block has its limitation, i.e., light nodes cannot know the proof of state, such as smart contract state, balance, and account's existence. A light node only downloads and stores each block head (about 80 bytes) instead of each transaction and each block. Hence, MPT is proposed to decrease the operation time [22], [23].

An MPT has three Merkle trees in a blockchain instead of one [19], which are used to store transactions, receipts, and states. For example, the state Merkle tree is used to update balance and existence of old accounts, which may be inserted with a state after transactions are completed. The depth of the tree is limited, so as to decrease DoS attacks. Otherwise, attackers may create some transactions on purpose to increase the depth of the tree, which may make the updating process extremely difficult.

d) Resisting second-preimage attack: In terms of cryptography at data in blockchains, a preimage attack tries to find a message with a specific hash value. A second-preimage attack tries to find any second input with the same output as that of a specified input. Both hash-function and blockchain-transaction-record data format are possible under this attack, which may lead to privacy leakage of users' information.

e) Separated storage for different types of data to different blockchains: A two-blockchain mechanism is proposed to enhance privacy and security of data storage [24], [25].

Different data, such as accounts, smart contracts, and transaction data, have different privacy-sensitive characteristics. For example, information about accounts is much more sensitive than other data. However, they are now generally put into the same blockchain, which may easily result in the risks of privacy leakage of highly sensitive data. Hence, account data should be stored into an absolute blockchain [24]. In addition, consensus data are also sensitive and requires similar treatment. A shared main chain is used to store consensus data, and an individual chain is used for each node to record transaction data [25]. The mechanism may highly increase the cost due to cross-chain access.

B. Security and Trust at the Network Layer

Nodes at the network layer play a role in network routing, propagation of data, and discovery of new nodes. DoS is a very common attack at this layer, which may paralyze the network. The attackers may use a large amount of junk information to delay transactions, which makes accounting nodes unable to package other normal transactions into a block. In addition, the attack of a MitM may exist during the communications in a blockchain network [21]. There also exist tampering and overlay attacks [26]. Researchers propose the following methods to resist attacks at the network layer:

1) Improving propagation by clustering and classifying nodes at the network layer:

a) *Clustering Nodes:* Huang *et al.* [27] proposed a method to cluster behavior patterns with a univariate sequence for nodes at the network layer in a blockchain system. The method may be used to detect malicious nodes. However, it does not consider that each node may have a multivariate sequence.

b) *Classifying Nodes:* Pass and Shi [28] classified nodes into honesty and dishonesty ones. The former can send messages to all nodes. The latter may delay or rearrange the propagation of information. Both of them have no ability to modify the information. However, the latter may send malicious spam and attack other nodes. Hence, the latter should be prevented beforehand. In addition, the honest nodes should send messages to other honesty ones instead of the latter, so as to increase trust propagation.

2) Trust-based propagation: Trust can be used to reduce gray- and black-hole attacks, so as to improve data packet forwarding rate and reduce routing packet overheads. Node trust scores may be used for blockchain-based communications. The ranges of trust scores for different user types are given in blockchains [21]. Even if not enough honest nodes are selected to forward data, the less trusted nodes cannot undertake the task of propagation because these nodes may cause serious damage to the network [10]. A lightweight trust-enhanced routing protocol is proposed to select a feasible path without using untrustworthy nodes [29]. It can reduce the number of attacks and decrease the influence of distrusted nodes for information propagation. However, it cannot resist collusion attacks. Some prediction methods [30], [31], such as SVM, gray system theory, rough set theory, and the Markov chain, may be applied to trust-based propagation in a P2P network.

3) Trusted execution environments: TEEs can guarantee private key control not to be exposed to distrustful hardware and software. Lind *et al.* [32] designed TEEs to eliminate a large number of potential attacks and reduce transaction latency in the Bitcoin blockchain system. However, how to guarantee and improve the trust of an execution environment is a crucial problem in their method.

4) Channel solution: Channel is a private communication subnet used to provide communications among multiple peers [33]. In a permitted blockchain network, channels can be used to restrict the propagation and distribution of confidential information exclusively to authorized nodes. Hence, channel technology can protect privacy for data and transactions at the network layer. Transactions and smart contracts may be visible only to a particular subset of participants belonging to the same channel [18]. Taking a payment service for example, it may be used to protect the privacy of a sender/recipient, values, paths, channel balance, and channel loads. The deadline of user transactions and the expiration time of the underlying channel need to be considered [1]. Specifically, P2P channels are built for each ongoing transaction. Hyperledger fabric reinforces confidentiality and privacy via its channel architecture [18]. However, how many channels should be created and what resources should be allocated are newly emerging concerns. In addition, the scalability and fault tolerant capability of a channel demand more studies.

5) Sharding solution: Sharding technology is an on-chain capacity expansion scheme for the network layer. Its core idea is "division and conquest." Nodes in the network are fairly and randomly divided into different shards [34], [35]. Each shard is processed in parallel to improve scale-out throughput. In a shard, each node only acquires a portion of transactions [25]. However, if a user is malicious, he/she may initialize another transaction to achieve double-spending attack. To prevent this attack, shards have to communicate with each other. In fact, this kind of communications may destroy the whole purpose of transaction sharding. In the case of state sharding, each shard keeps only a portion of a state during a network readjustment process. Problems may occur and lead to the failure of the entire system before the synchronization is completed [36]. In addition, this solution also suffers from the degraded capacity of fault tolerance. Some new technologies are needed to improve it.

C. Security and Trust at the Consensus Layer

According to consensus mechanisms whether trust is considered or not, main consensus mechanisms are classified into two types: no trust considered consensuses and trust considered ones, as listed in Table I.

More explanations about the consensus mechanisms in Table I can refer to [17]. This section summarizes trust and security issues of typical consensus mechanisms given as follows. First, we discuss the first type of consensus mechanisms.

TABLE I
MAIN CONSENSUS MECHANISMS AND THEIR TYPES

Consensus types	Consensus names
No trust considered consensuses	PoW [20][36], PoS[37], Delegated PoS (DPoS) [39], Byzantine Fault Tolerance (BFT) [2], Practical BFT (PBFT) [40], and Delegated BFT (DBFT) [3]
Trust considered consensuses	Ripple [41], Proof of Trust (PoT) [17][42], and Proof of Luck [43]

1) Proof of Work: Trust and security issues of PoW mainly include [4], [23] the following.

a) *Double Spending:* After a transaction is withdrawn, its issuer can reuse the coins for another transaction and the second transaction thus appears as "double spending."
b) *51% Attack:* Mine pools can cooperate to implement 51% attack to achieve double spending of bitcoins.
c) *Hijacked Attack:* Hackers use viruses to hijack other people's computers to dig a mine or overtake large "miners." Some other attacks include censorship resistance, resilience, to variations in mining power, key block, microblock forks, and wallet security [4], [44].

Eyal [45] and Eyal *et al.* [46] proposed an improved and extensible PoW protocol. An SPV is used to improve PoW by reducing its verification cost. However, because SPV does not have full records of all transactions, it causes the following new problems: 1) the vulnerability may be exploited by DoS and b) double-spending attacks may be against SPV nodes.

To defend against these attacks to SPV nodes, multiple SPV nodes need to be connected together. SPV nodes especially need to increase the probability of connecting to at least one reliable node. However, they may also be connected to spurious nodes or spurious networks [26]. Such random connections indicate that SPV nodes are also vulnerable to network partitions or Sybil attacks [47].

2) PoS and DPoS: PoW needs much computation time and energy, so it is impractical for many applications due to its low rate to create blocks and accept transactions. Therefore, researchers have developed alternative consensus mechanisms. PoS [37] is proposed based on stake as an alternative to solve the resource waste and security deficiencies of PoW. PoS can deal with the 51% attack to some extent. However, it is easily under the following risks: 1) the N@S attack when blocks are bifurcated and 2) lacking specialized accounting nodes. Proof of credit is a special PoS protocol with credits [38].

Not all nodes with PoS are likely to take part in an accounting process. Hence, DPoS [39] is proposed to improve it. DPoS participants vote for a certain number of delegated agent nodes, which are trusted to maintain security for a blockchain system. DPoS nodes can independently decide their trusted authorized agent nodes. The number of participant nodes for a verification process can be greatly reduced, so as to achieve rapid consensus verification. However, DPoS may be under the risk that delegated agents may be malicious. There are security challenges in the DPoS mechanism, such as stakeholder collusion voting. Both PoS and DPoS face security and trust issues.

3) BFT, PBFT, and DBFT: A permitted blockchain using BFT does not require costly mining [18]. For a given participant set, the PBFT algorithm [43] can reach consensus among less than one-third malicious participants, yet it does not scale well and need to maintain a list of participants or trust relationship among subgroups of participants [12], [14], [43]. Although BFT, PBFT, and DBFT can tolerate faults to some extent, they may also face inability services due to too many malicious nodes or collusion attacks [40].

Other consensus mechanisms without trust mechanisms include PoET, BFT-DPoS, Quorum Voting, Raft, Paxos, and PoD [48]. They face trust and security risks brought by malicious nodes. Next, we present the second type of consensus mechanisms.

1) Ripple: Ripple is one of the familiar trust consensus mechanisms, which requires multiple trusted verifiers to complete the accounting work together [41]. Although it takes into account trust values of nodes, it faces some problems as follows.

a) It needs more than 80% nodes to verify a transaction. Otherwise, the transaction has to keep waiting.
b) Communication cost may increase due to too many participation nodes. Hence, server load may be too heavy resulting in slow transaction processing rate and large cost.
c) It may have the risks of malicious or unexpected "consensus splitting" due to the fixed list of trusted nodes and lack of scalability.
d) It may be unavailable to maintain transactions or lead to bifurcation allowing attackers to execute invalid transactions.

2) Proof of Trust and Proof of Luck: In a PoT-based blockchain system, transaction verifiers are selected based on their trust values by secret sharing and Raft election. Peer nodes' trust is evaluated based on a trust graph managed by the blockchain itself [42]. To resist Sybil attacks, the work [43] uses TEEs to achieve PoL consensus and enforce correct processing of critical operations. It can restrain Sybil attacks from running under a single unit of hardware. They both face collusion attacks and trust and security issues in software [49].

D. Security and Trust at the Smart Contract Layer

Detected bugs of smart contracts include exception disorder, reentrancy, timestamp dependence, block number dependence, dangerous delegate call, and freezing Ether [50].

Smart contracts are easily attacked by hackers due to bugs and vulnerabilities [51]. A smart contract may be verified by single and M-of-N trusted verifiers [2]. However, the development of smart contracts lacks rigor and standardization. Program testing may be used to find out bugs' existence. However, it cannot indicate whether bugs exist or not. Vulnerabilities or bugs in programs of smart contracts may lead to catastrophic consequences given its related financial nature.

The formalized method can be used to improve smart contracts [52]. It may identify various mistakes and imprecisions

TABLE II
SOME TYPICAL LITERATURES ABOUT BLOCKCHAIN-BASED APPLICATIONS

Applications	Descriptions
Blockchain-based IoT	● Trust interoperability of IoT services [57] ● IoT-based E-commerce [58], traffic control [59], vehicular edge computing [60], and VANETs [19] ● Sensor networks and wireless sensor networks [61] ● An IoT-based smart manufacturing system [62]
Blockchain-based trading	● A localized P2P electricity trading system [10] ● Support secondary market trading [63] ● Secure P2P energy trading [64] ● Financial blacklist sharing [65]
Blockchain-based E-health	● E-Health systems [66][68] ● Medical data sharing [69]
Blockchain-based Cloud	● Secure encrypted signature data [70] ● Trust accountability [71]
Blockchain-based reputation and authentication	● A privacy-protection-based reputation system [72][73] ● Authentication [61][74]

in existing semantics. It may also prove whether the code mathematically meets the specification or not. Bigi *et al.* [53] validated the decentralized two-party smart contracts by formal methods and game theory. However, it does not consider how to realize the formalized verification of multiple-party smart contracts in blockchain systems.

E. Security and Trust at the Application Layer

The application layer is based on the aforementioned four layers. A new class of accountable blockchain-based application systems emerges as a publicly verifiable open ledger: Bitcoin, which allows users to transfer currency (bitcoins) safely without a centralized regulator [54]. However, more projects take blockchains to serve for other functions that require trusted computing and auditability [55].

1) Blockchain Applications: Jaoudel and Saadel [56] discussed 151 applications about blockchains, which are about: 1) IoT; 2) energy; 3) healthcare; 4) finance; 5) resource management; 6) government; 7) exchange; 8) transportation; 9) business process management; 10) rights management; 11) privacy; 12) supply chain; 13) smart cities; 14) insurance; 15) education; 16) data transfer; 17) social network; 18) fraud detection; 19) environment; 20) research; 21) decision making; 22) data accountability; and 23) access control. 78% of all retrieved published work has targeted at the first five applications. We list some typical blockchain-based applications in terms of security, privacy, trustworthiness, and authentication, as shown in Table II.

Details about Table II are as follows.

a) Blockchain-based IoT applications: Blockchain with tempering-resistance provides assured authentication, nonrepudiation, and trust to IoT applications [57]. It is used to store encrypted or unencrypted IoT data, so as to protect device (such as vehicles) privacy and prevent devices from spreading forged messages. It can resist attacks, such as tampering, forging, tracking attacks, and DoS [19], as well as improve trust & authentication [61].

b) Blockchain-based trading applications: Consortium blockchains are proposed to achieve security and privacy preservation [64]. A blockchain assurance provider assesses the underlying cryptography, including private keys management and blockchain engine security [63].

c) Blockchain-based E-health and cloud applications: Blockchains can be used to protect privacy and resist tampering [66]–[69]. In addition, blockchains are used to store secure encrypted signature data against malicious service providers and malicious users in clouds [70], as well as provide trust accountability for cloud database systems [71].

d) Blockchain-based reputation and authentication: By using blockchain-based reputation and authentication, DoS, stale information attack, and double registration attack may be well defended [76]. It may resist bad-mouthing, ballot stuffing, Sybil attacks, and whitewashing [72]. It also may achieve anonymity preservation and improve the trustworthiness of messages relying on reputation [73].

2) Methods to Enhance Trust and Security for Applications:

a) Zero-knowledge proof: ZKP [74] ensures that a transaction is "valid" without revealing the actual purpose or other sensitive details of the transaction. Due to its potential of enhancing privacy and security for blockchain participants, ZKP arisen wide excitements in the blockchain community and provides a new way of using blockchains for financial circles recently.

b) Cryptographic schemes: An efficient homomorphic encryption and secure multiparty computation schemes may be used to realize multiparty ciphertext processing, which hide sensitive information, such as user transaction volume. A new encryption scheme for decentralized public key is proposed in [58]. Technically, the secure access of data may be controlled by providing dynamic keys to users based on blockchains [76]. The cryptography needs to be assessed, including managing private keys and maintaining blockchain security [63]. The flexibility of privacy protection and large-scale use need to be maximized [74].

c) Signature technologies: Multisignature technology in a blockchain can flexibly configure access rights to data [7]. Ring and group signature are used to protect the identity of signers. An attribute-based signature scheme with multiple authorities is proposed based on sharing the secret pseudo-random function seeds among authorities, so as to resist collusion attack [68]. A signcryption method is proposed to prevent adversaries from violating users' privacy, which has lower computing cost than that of signature and encryption sequentially [77].

d) Resisting Sybil attacks: To achieve the validity and integrity of transactions, Otte *et al.* [5] proposed a Sybil-resistant algorithm based on a blockchain to determine agent trustworthiness in an online community, so as to offer scalability, openness, and Sybil-resistance.

3) Access Control of a Management Platform to Enhance Trust and Security: A blockchain-based decentralized management platform [54] provides fine-grained access and privacy preservation for applications with two blockchains. Because the privacy of data transaction and access control transaction

are different, the platform deals with them differently and provides privacy preservation. In addition, data are encrypted to improve security and trust for applications. However, participants may need to retrieve data from multiblockchains. It may cause new problems about secure authentication and account access authorization due to cross-blockchain.

4) Evaluation Framework (Blockbench): Applications' performance needs to be tested. Blockbench [14] is an evaluation framework for analyzing private blockchains. It can be used to measure evaluation metrics, including consistency, availability, failure tolerance, scalability, latency, auditability, and DoS resistance [78]. Many studies focus on how to reveal and improve the trust and security of blockchains, but these issues are not fully solved. The current methods need more concrete evaluation on the effectiveness [3].

V. Open Research Issues and Future Work

A. Open Research Issues of Trust and Security

1) Resisting Collusion Attack for Consensus: Nowadays, one of the most popular consensus mechanisms is DPoS. Although delegated nodes are voted and selected by using DPoS, they may have unacceptable or malicious behaviors, such as downtime and delay. If collusion nodes are disguised as trustworthy ones, they may not be detected in time. It may cause malicious and false data to be stored into a blockchain. Some countermeasures need to be developed to detect such malicious attacks.

2) Improving the Security of Shard-Based Consensuses: Shards can provide parallel process for a blockchain. There are network sharding, transaction sharding, and state sharding. Via network sharding and transaction sharding, nodes in a blockchain system can be divided into different shards. Each shard can form an independent process and reach an agreement via its consensus mechanism for different transaction subsets. The public-randomness protocol may be used to resist bias and choose large representative shards. An efficient cross-shard submission protocol may be used to automatically handle cross-shards transactions [79]. However, sharding solutions suffer from the limited capacity of fault tolerance and degradation in decentralization [25]. They face Sybil attack by using PBFT consensus mechanism and double spending in shards [80]. Some novel technologies need to be developed to address the problems.

3) Improving Security and Trust of Blockchain-Based IoT: Despite its rapid development, IoT faces the problems of privacy and security. The security flaws of the IoT lie in the lack of a mutual trust mechanism among devices. All devices need check data in an IoT center. Once the database collapses, it may cause great damage to the whole IoT. The key benefit of using blockchain technology for IoT is to build trust between parties and devices, as well as reduce the risks of collusion attacks and tampering. The distributed network structure of a blockchain makes it possible to maintain consensus among IoT devices without verification in the center. Hence, even if one or more nodes are broken down, the data of the whole IoT system are still reliable and secure [30], [81].

A blockchain can provide a simple infrastructure for two devices to exchange messages, such as propagating currency or data directly via a secure and reliable time-stamped contract. Blockchain-based IoT devices may cooperate autonomously and smartly without any centralized authorization. There are several dominant topics about security and trust in blockchain-based IoT [56]: 1) enhancing security of interconnected devices; 2) maintaining anonymity; c) device management mechanisms and protocols; and d) network security.

TABLE III
Attacks to Blockchain-Based IoT

Attack names	The descriptions of attacks
DoS	Prevent connections between IoT devices and the blockchain network and being notified about new software updates.
Compromising software integrity	Impersonate a vendor to publish a malicious update software to the IoT devices.
Software downgrading attack	Cause an IoT device to downgrade its software to an outdated version including known bugs.

Attackers may read and send any transactions to a blockchain-based IoT. They may be passive to eavesdrop or active to inject, replay, or filter any messages involved. The following threats need to be considered in Table III [81]. DoS is a kind of attack mainly for a blockchain-based IoT network. The last two attacks are related to software vulnerability or bugs.

4) Meeting Users' Personalized Requirements With Blockchain Services: Users may have their personalized requirements that need different strategies to deal with. A blockchain system should take into account the characteristics, complexity, and economy of users' requirements, so as to offer high flexibility use.

5) Improving the Security and Trust for Apps: Some apps may be easily attacked by hackers due to their weak protection. It may result in users' information leakage and property loss. For example, there are some security problems about the encryption of digital wallet apps. Especially in the aspect of private key protection, there are serious shortcomings when implementing the desired security.

6) Improving Formalized Verification of Smart Contracts: Formal verification may be the most reliable way to implement the security and privacy of smart contracts. To strictly verify the required security of smart contracts, we need to formalize the semantics as well as the security properties of interest, especially at the level of the executed bytecode [52]. A number of security properties should be formally defined for smart contracts, including atomicity of invocation, independence, and integrity.

B. Proposed Future Work

As a distributed ledger, blockchains should have high trust and security, as well as meet users' personalized requirements. First, nodes with high trust values may be selected to deal with high priority requirements. Then, transactions are verified and then stored into the blocks by the trusted nodes. It can improve the credibility and efficiency of a blockchain system. A profile about a trusted-blockchain to meet trust and personalized requirements of users should be built.

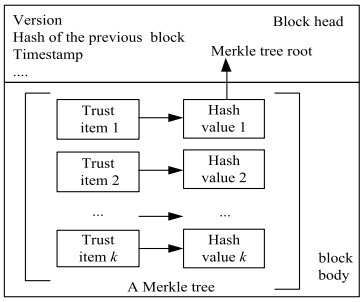

Fig. 3. One block containing trust items of nodes in a trusted-blockchain.

1) Constructing a Trusted-Blockchain: A trusted-blockchain is a blockchain that is different from other blockchains by storing important *trust items of nodes*, such as node data of their attributes and behaviors.

a) Node attributes include reputation, the number of transactions, the number of successful transactions, stable running time, waiting time, processing time, and trust values.
b) Node behaviors include those data from normal and malicious nodes, as well as data of collusion attacks (such as false verification, false accounting, and false propagation).

A trusted-blockchain can be used to compute the following values: 1) computing trust values of nodes; 2) by using trust values, computing *collusion similarity* among collusion nodes; and 3) computing *collusion behavior degree* among collusion nodes.

Malicious nodes and collusion behavior among nodes may be detected based on the abovementioned computed values. A trusted-blockchain is needed to provide data for the abovementioned computation. A brief process of constructing a trusted-blockchain system is given as follows.

a) Designing a consensus with trust considered. It may be the combination of trust and DPoS/DBFT.
b) Generating blocks by using the consensus to a trusted-blockchain. Each block contains *trust items of nodes*, as shown in Fig. 3. The data of *trust items of nodes* is arranged as a Merkle tree in a block. When using a trusted-blockchain, another blockchain is needed to store transaction information of applications at the same time.

2) Automatically Collecting Data for a Trusted-Blockchain: A smart contract composed of automated script code may be deployed on an Ethereum blockchain system. After being triggered by blockchain transactions, the smart contract may automatically read and store data from or into a trusted-blockchain. Smart contracts are executed with no interference when the conditions are triggered. Thus, it may realize collecting *trust items of nodes* and then storing them into a trusted-blockchain automatically. The whole process is open and transparent, which makes the trust values more authoritative for a trusted-blockchain.

3) Detecting Malicious Nodes Based on a Trusted-Blockchain: Nodes with high trust values need to be selected for accounting, verification, and propagation. Therefore, it can improve efficiency and reduce the cost of a trusted-blockchain system. Hence, trusted nodes may be selected by detecting malicious ones. The following two methods are promising to detect malicious nodes.

a) *Detecting Malicious Outliers:* Malicious nodes may have some malicious behaviors, such as releasing false information or attacking their neighbor nodes. Hence, they may often be discarded by their neighbors. By analyzing the network structure of nodes in a blockchain system, malicious outliers may be obtained by a clustering algorithm [82].
b) *Detecting Malicious Nodes by Computing Trust Values:* Trust values of nodes can be calculated based on the key *trust items of nodes* from a trusted-blockchain by using some classification technologies, such as SVM or Bayes. Different from the reputation-based trust evaluation [83], trust values are computed by using behaviors of nodes. If the trust value of a node is very low, it may be a malicious one.

TABLE IV
COLLUSION TYPES

Types	Collusion names	Detailed descriptions
1	Collusion among validating nodes (verifiers)	False verification for the same transaction
2	Collusion among propagation nodes	The same false information propagation in the network or DoS attacks
3	Collusion between accounting nodes and validating nodes	False accounting, verification, and propagation

TABLE V
EXAMPLE OF USERS' PERSONALIZED REQUIREMENTS

Task types / Trust types	Urgent task	Non-urgent task
High trust values	The first priority	The third priority
Low trust values	The second priority	The fourth priority

4) Detecting Collusion Nodes: Due to the randomness, fuzziness, and unpredictability of a complex distributed blockchain environment, nodes may collude to obtain more profits. A collusion node is not easily detected based on its own trust value only. Hence, collusion behaviors of nodes need be detected. Collusion types based on node behaviors are given in Table IV.

In Table IV, the first two types of collusion attacks happen among nodes of the same roles, such as among verifiers or among propagation nodes. The third type of collusion comes from nodes of two different roles: accounting nodes and validating nodes. Based on Table V, the corresponding algorithms may be designed to detect collusion nodes, including calculating *collusion similarity* and *collusion behavior degree*, as shown in Fig. 4. The descriptions are given as follows.

a) Computing collusion similarity of nodes: Two malicious nodes may have the same or similar malicious behaviors for the same transactions, e.g., nonverification, false accounting, false verification, delayed accounting, delayed verifica-

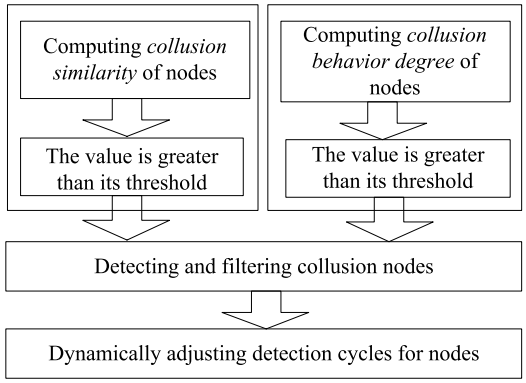

Fig. 4. Process of detecting collusion nodes.

tion, and delayed propagation. Based on a trusted-blockchain, characteristics of historical behaviors of malicious nodes can be obtained. They may be detected by collusion similarity, which may be calculated by fuzzy clustering methods. The method can detect type-1 and type-2 collusion in Table IV.

b) Computing collusion behavior degree of nodes: For malicious nodes, they may have the same patterns of malicious behaviors, such as false propagation -> false verification -> false accounting. The patterns of behaviors occur to the same transactions. By using behavior data in a trusted-blockchain, the *collusion behavior degree* can be computed based on the analysis of patterns of malicious behaviors. If the degree is larger than the given certain threshold, two nodes may be colluded if we know that one of them is surely malicious. It can detect the type-3 collusion from Table IV.

c) Detecting and filtering collusion nodes: According to the different trust requirements of users, collusion nodes may be detected and filtered out based on aforementioned two calculation results.

d) Dynamically adjusting detection cycles for nodes: Malicious nodes may be detected via a fixed or dynamic detection cycles. To find out malicious nodes more precisely at low cost, the cycle may be dynamically adjusted based on trust values, collusion similarity, and collusion behavior degree of nodes.

e) Selecting a set of nodes with high trust values for propagating transactions and blocks: Nodes with high trust values and higher propagation performance may be selected for propagating transactions and blocks. Then, a trusted optimization propagation network may be obtained quickly to reduce the propagation cost and improve security. It may also reduce the possibility of DoS, collusion, and bad mouthing attacks due to selected nodes with high trust values.

At the consensus and application layers, nodes with high trust values also need to be selected to come to trustworthy consensuses and applications. The abovementioned methods and steps may be coded into smart contracts and deployed on an Ethernet blockchain. By implementing these smart contracts, the detection of malicious and collusion nodes may be automatically implemented. Detection cycles may be adjusted adaptively and automatically as well.

f) Meeting user's personalized requirements: Blockchain-based applications need to meet users' personalized requirements [45], such as user trust and task urgency. Trust values of nodes may be classified as x types, and task urgency degree may be classified as y types. Then, the combination count of different trust values and urgency degree is xy. For example, trust values of users and task urgency degree may be classified into two types: high and low trust values and urgent and nonurgent tasks, respectively. Hence, there are four kinds of priorities for personalized requirements in Table V. In that, the first priority belongs to the urgent task from users with high trust values. The last priority belongs to the nonurgent task from users with normal trust values.

VI. Conclusion

This work analyzes a basic architecture of blockchains. The related work about security and trust issues is summarized. The open issues are discussed, and future work is proposed. Because blockchains are highly distributed, some security and trust mechanisms are greatly needed. Hence, the following needs arise.

New consensus mechanisms are needed. Some consensus mechanisms have already been designed, such as those in [42] and [43]. However, new mechanisms are needed to improve the automatic detection of threats to blockchains. It is important to design incentive-compatible consensus mechanisms so that self-interested nodes in a decentralized blockchain system can spontaneously implement block verification and accounting. It may eventually increase the cost of irrational behaviors in the system, so as to suppress security attacks and threats [8].

Industrial and government applications and cross-applications are needed to pay more attention based on the IoT using reputation and trust [84]–[86]. It is key to developing secure and encryption payment based on blockchains [87], [88]. In addition, new technologies combined with artificial intelligence and cloud computing [89]–[95] technologies are required to increase trust and security for blockchains when computing and storing a large amount of data.

References

[1] S. Wang *et al.*, "Blockchain-powered parallel healthcare systems based on the ACP approach," *IEEE Trans. Comput. Social Syst.*, vol. 5, no. 4, pp. 942–950, Dec. 2018.

[2] X. Xu *et al.*, "A taxonomy of blockchain-based systems for architecture design," in *Proc. IEEE Int. Conf. Softw. Archit. (ICSA)*, Gothenburg, Sweden, Apr. 2017, pp. 243–252.

[3] J. Yli-Huumo, D. Ko, S. Choi, S. Park, and K. Smolander, "Where is current research on blockchain technology?—A systematic review," *PLoS ONE*, vol. 11, no. 10, pp. 1–28, Oct. 2016, doi: 10.1371/journal.pone.0163477.

[4] V. Gramoli, "From blockchain consensus back to Byzantine consensus," *Future Gener. Comput. Syst.*, vol. 107, pp. 760–769, Jun. 2020, doi: 10.1016/j.future.2017.09.023.

[5] P. Otte, M. de Vos, and J. Pouwelse, "TrustChain: A Sybil-resistant scalable blockchain," *Future Gener. Comput. Syst.*, vol. 107, pp. 770–780, Jun. 2020, doi: 10.1016/j.future.2017.08.048.

[6] M. Apostolaki, G. Marti, J. Muller, and L. Vanbever, "SABRE: Protecting bitcoin against routing attacks," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, Feb. 2019, pp. 24–27.

[7] R. Zhang, R. Xue, and L. Liu, "Security and privacy on blockchain," *ACM Comput. Surveys*, vol. 52, no. 3, Jul. 2019, Art. no. 51.

[8] N. Z. Aitzhan and D. Svetinovic, "Security and privacy in decentralized energy trading through multisignatures, blockchain and anonymous messaging streams," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 5, pp. 840–852, Sep. 2018.

[9] Z. Wan, D. Lo, X. Xia, and L. Cai, "Bug characteristics in blockchain systems: A large-scale empirical study," in *Proc. IEEE/ACM 14th Int. Conf. Mining Softw. Repositories (MSR)*, May 2017, pp. 413–424.

[10] J. Li, G. Liang, and T. Liu, "A novel multi-link integrated factor algorithm considering node trust degree for blockchain-based communication," *KSII Trans. Internet Inf. Syst.*, vol. 11, no. 8, pp. 3766–3788, 2017.

[11] Y. Yuan and F.-Y. Wang, "Blockchain: The state of the art and future trends," *Acta Autom. Sinica*, vol. 42, no. 4, pp. 481–494, Apr. 2016.

[12] Q. Dai, K. Xv, S. Guo, L. Dai, and Z. Zhou, "A private data protection scheme based on blockchain under pipeline model," in *Proc. 1st IEEE Int. Conf. Hot Inf.-Centric Netw. (HotICN)*, Aug. 2018, pp. 37–45.

[13] P. Fraga-Lamas and T. M. Fernandez-Carames, "A review on blockchain technologies for an advanced and cyber-resilient automotive industry," *IEEE Access*, vol. 7, pp. 17578–17598, 2019.

[14] T. Tuan Anh Dinh, J. Wang, G. Chen, R. Liu, B. C. Ooi, and K.-L. Tan, "BLOCKBENCH: A framework for analyzing private blockchains," Mar. 2017, *arXiv:1703.04057*. [Online]. Available: http://arxiv.org/abs/1703.04057

[15] K. Werbach, "Trust, but verify: Why the blockchain needs the law," *Berkeley Technol. Law J.*, vol. 284440, pp. 1–60, Nov. 2018. [Online]. Available: http://dx.doi.org/10.2139/ssrn

[16] D. Harz, "Trust and verifiable computation for smart contracts in permissionless blockchains," M.S. thesis, Roy. Inst. Technol., Stockholm, Sweden, 2017.

[17] J. Zou, B. Ye, L. Qu, Y. Wang, M. A. Orgun, and L. Li, "A proof-of-trust consensus protocol for enhancing accountability in crowdsourcing services," *IEEE Trans. Services Comput.*, vol. 12, no. 3, pp. 429–445, May 2019, doi: 10.1109/TSC.2018.2823705.

[18] S. Wu, Y. Chen, Q. Wang, M. Li, C. Wang, and X. Luo, "CReam: A smart contract enabled collusion-resistant e-auction," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 7, pp. 1687–1701, Jul. 2019.

[19] Z. Lu, W. Liu, Q. Wang, G. Qu, and Z. Liu, "A privacy-preserving trust model based on blockchain for VANETs," *IEEE Access*, vol. 6, pp. 45655–45664, 2018.

[20] R. Zhang and B. Preneel, "Lay down the common metrics: Evaluating proof-of-work consensus Protocols' security," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2019, pp. 1–19.

[21] "The trust chain consensus COTI: A decentralised and trust-based online payment system with arbitration service," White Paper, V. 3.0, Jul. 2018, pp. 1–58. [Online]. Available: https://coti.io/files/COTI-technical-whitepaper.pdf

[22] B. Qin, J. Huang, Q. Wang, X. Luo, B. Liang, and W. Shi, "Cecoin: A decentralized PKI mitigating MitM attacks," *Future Gener. Comput. Syst.*, vol. 107, pp. 805–815, Jun. 2020, doi: 10.1016/j.future.2017.08.025.

[23] S. Mathiyalahan, S. Manivannan, M. Nagasundaram, and R. Ezhilarasie, "Data integrity verification using MPT (Merkle patricia Tree) in cloud computing," *Int. J. Eng. Technol.*, vol. 7, no. 2.24, pp. 500–503, Apr. 2018.

[24] M. Wang and M. Duan, "The second-preimage attack to blockchain based on the structure of Merkle hash tree," *Netinfo Secur.*, vol. 1, pp. 38–44, Jan. 2018.

[25] W. Cai, L. Yu, R. Wang, N. Liu, and E. Deng, "Blockchain application development techniques," *J. Softw.*, vol. 28, no. 6, pp. 1474–1487, 2017.

[26] Z. Ren, K. Cong, T. Aerts, B. de Jonge, A. Morais, and Z. Erkin, "A scale-out blockchain for value transfer with spontaneous sharding," in *Proc. Crypto Valley Conf. Blockchain Technol. (CVCBT)*, Jun. 2018, pp. 1–10, doi: 10.1109/CVCBT.2018.00006.

[27] Q. Wang, B. Qin, J. Hu, and F. Xiao, "Preserving transaction privacy in bitcoin," *Future Gener. Comput. Syst.*, vol. 107, pp. 793–804, Jun. 2020, doi: 10.1016/j.future.2017.08.026.

[28] B. Huang, Z. Liu, J. Chen, A. Liu, Q. Liu, and Q. He, "Behavior pattern clustering in blockchain networks," *Multimedia Tools Appl.*, vol. 76, no. 19, pp. 20099–20110, Jan. 2017.

[29] R. Pass and E. Shi, "Hybrid consensus: Efficient consensus in the permissionless model," in *Proc. 31st Int. Symp. Distrib. Comput. (DISC)*, 2017, pp. 1–16, doi: 10.4230/LIPIcs.DISC.2017.39.

[30] H. Xia, J. Yu, C.-L. Tian, Z.-K. Pan, and E. Sha, "Lightweight trust-enhanced on-demand multi-path routing in mobile ad hoc networks," *J. Netw. Comput. Appl.*, vol. 62, pp. 112–127, Feb. 2016.

[31] M. Shen, X. Tang, L. Zhu, X. Du, and M. Guizani, "Privacy-preserving support vector machine training over blockchain-based encrypted IoT data in smart cities," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7702–7712, Oct. 2019.

[32] P. Zhang, S. Shu, and M. Zhou, "An online fault detection model and strategies based on SVM-grid in clouds," *IEEE/CAA J. Automatica Sinica*, vol. 5, no. 2, pp. 445–456, Mar. 2018.

[33] J. Lind, I. Eya, P. Pietzuch, and E. Sirer, "Teechan: Payment channels using trusted execution environments," in *Proc. 4th Workshop Bitcoin Block-Chain Res. (BITCOIN)*, 2017, pp. 1–14.

[34] F. Benhamouda, S. Halevi, and T. Halevi, "Supporting private data on hyperledger fabric with secure multiparty computation," *IBM J. Res. Develop.*, vol. 63, nos. 2–3, pp. 1–8, Mar. 2019.

[35] 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.*, New York, NY, USA, Oct. 2016, pp. 17–30.

[36] M. Al-Bassam, A. Sonnino, S. Bano, D. Hrycyszyn, and G. Danezis, "Chainspace: A sharded smart contracts platform," Aug. 2017, *arXiv:1708.03778*. [Online]. Available: http://arxiv.org/abs/1708.03778

[37] Y. Jia. (Mar. 2018). *Op Ed: The Many Faces of Sharding for Blockchain Scalability*. Bitcoin Magazine. [Online]. Available: https://www.nasdaq.com/article/op-ed-the-many-faces-of-sharding-for-blockchain-scalability-cm937322

[38] W. Li, S. Andreina, J. Bohli, and G. Karame, "Securing proof-of-stake blockchain protocols," in *Proc. Eur. Symp. Res. Comput. Secur. Int. Workshop Data Privacy Manage. Cryptocurrencies Blockchain Technol.*, 2017, pp. 297–315.

[39] X. Han, Y. Yuan, and F.-Y. Wang, "A fair blockchain based on proof of credit," *IEEE Trans. Comput. Social Syst.*, vol. 6, no. 5, pp. 922–931, Oct. 2019.

[40] J. Kang, Z. Xiong, D. Niyato, D. Ye, D. I. Kim, and J. Zhao, "Toward secure blockchain-enabled Internet of Vehicles: Optimizing consensus management using reputation and contract theory," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2906–2920, Mar. 2019.

[41] I. Abraham, D. Malkhi, K. Nayak, L. Ren, and A. Spiegelman, "Solida: A blockchain protocol based on reconfigurable Byzantine consensus," Nov. 2016, *arXiv:1612.02916*. [Online]. Available: http://arxiv.org/abs/1612.02916

[42] D. Schwartz, N. Youngs, and A. Britto, "The Ripple protocol consensus algorithm," Ripple Labs, San Francisco, CA, USA, White Paper V. 1.0, 2014, pp. 1–8. [Online]. Available: http://www.naation.com/ripple-consensus-whitepaper.pdf

[43] L. Bahri and S. Girdzijauskas, "When trust saves energy: A reference framework for proof of trust (PoT) blockchains," in *Proc. Companion The Web Conf. Web Conf. (WWW)*, Lyon, France, Apr. 2018, pp. 1165–1169.

[44] 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)*, Dec. 2016, pp. 1–6.

[45] A. Gencer, "On scalability of blockchain technologies," Ph.D. dissertation, Cornell Univ., New York, NY, USA, 2017. [Online]. Available: https://doi.org/10.7298/X4SQ8XJ1

[46] I. Eyal, "Blockchain technology: Transforming libertarian cryptocurrency dreams to finance and banking realities," *Computer*, vol. 50, no. 9, pp. 38–49, 2017.

[47] I. Eyal, A. Gencer, E. Sirer, and R. Renesse, "Bitcoin-NG: A scalable blockchain protocol," in *Proc. Networked Syst. Design Implement. (NSDI)*, 2016, pp. 45–59.

[48] E. K. Kogias, P. Jovanovic, N. Gailly, I. Khoffi, L. Gasser, and B. Ford, "Enhancing bitcoin security and performance with strong consistency via collective signing," in *Proc. 25th USENIX Secur. Symp. (USENIX Secur.)*, Aug. 2016, pp. 279–296.

[49] H. R. Hasan and K. Salah, "Blockchain-based proof of delivery of physical assets with single and multiple transporters," *IEEE Access*, vol. 6, pp. 46781–46793, Aug. 2018.

[50] P. Zhang, Y. Kong, and M. Zhou, "A domain partition-based trust model for unreliable clouds," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 9, pp. 2167–2178, Sep. 2018.

[51] B. Jiang, Y. Liu, and W. K. Chan, "ContractFuzzer: Fuzzing smart contracts for vulnerability detection," in *Proc. 33rd ACM/IEEE Int. Conf. Automated Softw. Eng. (ASE)*, Paris, France, Sep. 2018, pp. 1–11.

[52] M. Rodler, W. Li, G. O. Karame, and L. Davi, "Sereum: Protecting existing smart contracts against re-entrancy attacks," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, Feb. 2019, pp. 1–15.

[53] I. Grishchenko, M. Maffei, and C. Schneidewind, "A semantic framework for the security analysis of Ethereum smart contracts," in *Proc. Int. Conf. Princ. Secur. Trust, Princ. Secur. Trust*, Apr. 2018, pp. 243–269.

[54] G. Bigi, A. Bracciali, G. Meacci, and E. Tuosto, "Validation of decentralised smart contracts through game theory and formal methods," in *Degano Festschrift* (Lecture Notes in Computer Science), vol. 9465. 2015, pp. 142–161, doi: 10.1007/978-3-319-25527-9.11.

[55] 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.

[56] M. Atzori, "Blockchain governance and the role of trust service providers: The TrustedChain network," Social Sci. Electron., Rochester, NY, USA, Tech. Rep., May 2017, pp. 1–23, doi: 10.2139/ssrn.2972837.

[57] J. Jaoude and R. Saade, "Business applications—Usage in different domains," *IEEE Access*, vol. 7, pp. 45360–45381, Mar. 2019, doi: 10.1109/Access.2019.2902501.

[58] W. Viriyasitavat, L. Da Xu, Z. Bi, and A. Sapsomboon, "New blockchain-based architecture for service interoperations in Internet of Things," *IEEE Trans. Comput. Social Syst.*, vol. 6, no. 4, pp. 739–748, Aug. 2019.

[59] C. Liu, Y. Xiao, V. Javangula, Q. Hu, S. Wang, and X. Cheng, "NormaChain: A blockchain-based normalized autonomous transaction settlement system for IoT-based e-commerce," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4680–4693, Jun. 2019, doi: 10.1109/JIOT.2018.2877634.

[60] L. Cheng *et al.*, "SCTSC: A semicentralized traffic signal control mode with attribute-based blockchain in IoVs," *IEEE Trans. Comput. Social Syst.*, vol. 6, no. 6, pp. 1373–1385, Dec. 2019.

[61] J. Kang *et al.*, "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4660–4670, Jun. 2019, doi: 10.1109/JIOT.2018.2875542.

[62] A. Moinet, B. Darties, and J.-L. Baril, "Blockchain based trust & authentication for decentralized sensor networks," Jun. 2017, *arXiv:1706.01730*. [Online]. Available: http://arxiv.org/abs/1706.01730

[63] Y. Zhang, X. Xu, A. Liu, Q. Lu, L. Xu, and F. Tao, "Blockchain-based trust mechanism for IoT-based smart manufacturing system," *IEEE Trans. Comput. Social Syst.*, vol. 6, no. 6, pp. 1386–1394, Dec. 2019.

[64] M. Li, D. Hu, C. Lal, M. Conti, and Z. Zhang, "Blockchain-enabled secure energy trading with verifiable fairness in industrial Internet of Things," *IEEE Trans. Ind. Informat.*, Feb. 2020, doi: 10.1109/TII.2020.2974537.

[65] Z. Li, J. Kang, R. Yu, D. Ye, Q. Deng, and Y. Zhang, "Consortium blockchain for secure energy trading in industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3690–3700, Aug. 2018.

[66] P. Treleaven, R. Gendal Brown, and D. Yang, "Blockchain technology in finance," *Computer*, vol. 50, no. 9, pp. 14–17, Sep. 2017.

[67] R. Guo, H. Shi, Q. Zhao, and D. Zheng, "Secure attribute-based signature scheme with multiple authorities for blockchain in electronic health records systems," *IEEE Access*, vol. 6, pp. 11676–11686, Feb. 2018, doi: 10.1109/ACCESS.2018.2801266.

[68] W. Liu, S. S. Zhu, T. Mundie, and U. Krieger, "Advanced block-chain architecture for e-health systems," in *Proc. IEEE 19th Int. Conf. e-Health Netw., Appl. Services (Healthcom)*, Oct. 2017, pp. 37–42.

[69] R. Guo, H. Shi, Q. Zhao, and D. Zheng, "Secure attribute-based signature scheme with multiple authorities for blockchain in electronic health records systems," *IEEE Access*, vol. 6, pp. 11676–11686, Feb. 2018, doi: 10.1109/ACCESS.2018.2801266.

[70] Q. Xia, E. B. Sifah, K. O. Asamoah, J. Gao, X. Du, and M. Guizani, "MeDShare: Trust-less medical data sharing among cloud service providers via blockchain," *IEEE Access*, vol. 5, pp. 14757–14767, Jul. 2017, doi: 10.1109/access.2017.2730843.

[71] Y. Zhang, R. H. Deng, J. Shu, K. Yang, and D. Zheng, "TKSE: Trustworthy keyword search over encrypted data with two-side verifiability via blockchain," *IEEE Access*, vol. 6, pp. 31077–31087, Jun. 2018.

[72] G.-H. Hwang and S.-K. Fu, "Proof of violation for trust and accountability of cloud database systems," in *Proc. 16th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput. (CCGrid)*, May 2016, pp. 425–433.

[73] A. Schaub, R. Bazin, O. Hasan, and L. Brunie, "A trustless privacy-preserving reputation system," in *Proc. Syst. Secur. Privacy Protection (SEC, ICT)*, 2016, pp. 398–411.

[74] Z. Lu, Q. Wang, G. Qu, and Z. Liu, "BARS: A blockchain-based anonymous reputation system for trust management in VANETs," in *Proc. 17th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun., 12th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, Aug. 2018, pp. 98–103.

[75] C. H. Lee and K.-H. Kim, "Implementation of IoT system using block chain with authentication and data protection," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2018, pp. 936–940.

[76] N. Alexopoulos, J. Daubert, M. Muhlhauser, and S. M. Habib, "Beyond the hype: On using blockchains in trust management for authentication," in *Proc. IEEE Trustcom/BigDataSE/ICESS*, Aug. 2017, pp. 546–553.

[77] A. Lei, H. Cruickshank, Y. Cao, P. Asuquo, C. P. A. Ogah, and Z. Sun, "Blockchain-based dynamic key management for heterogeneous intelligent transportation systems," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1832–1843, Dec. 2017.

[78] J. Wang, M. Li, Y. He, H. Li, K. Xiao, and C. Wang, "A blockchain based privacy-preserving incentive mechanism in crowdsensing applications," *IEEE Access*, vol. 6, pp. 17545–17556, Mar. 2018, doi: 10.1109/ACCESS.2018.2805837.

[79] A. Anjum, M. Sporny, and A. Sill, "Blockchain standards for compliance and trust," *IEEE Cloud Comput.*, vol. 4, no. 4, pp. 84–90, Jul. 2017.

[80] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. Ford, "OmniLedger: A secure, scale-out, decentralized ledger via sharding," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2018, pp. 583–598.

[81] M. H. Manshaei, M. Jadliwala, A. Maiti, and M. Fooladgar, "A game-theoretic analysis of shard-based permissionless blockchains," *IEEE Access*, vol. 6, pp. 78100–78112, Dec. 2018, doi: 10.1109/ACCESS.2018.2884764.

[82] O. Leiba, Y. Yitzchak, R. Bitton, A. Nadler, and A. Shabtai, "Incentivized delivery network of IoT software updates based on trustless Proof-of-Distribution," in *Proc. IEEE Workshop Secur. Privacy Blockchain*, vol. 11, May 2018, pp. 29–39. [Online]. Available: http://arxiv.org/abs/1805.04282v1

[83] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, Feb. 2007.

[84] K. Chan, R. Tso, C. Chen, and M. Wu, "Reputation-based trust evaluation mechanism for decentralized environments and its applications based on smart contracts," in *Advances in Computer Science and Ubiquitous Computing* (Lecture Notes in Electrical Engineering 474). Singapore: Springer, Dec. 2017, pp. 310–314, doi: 10.1007/978-981-10-7605-3_51.

[85] G. Fortino, F. Messina, D. Rosaci, and G. M. L. Sarne, "Using blockchain in a reputation-based model for grouping agents in the Internet of Things," *IEEE Trans. Eng. Manag.*, early access, Jun. 2019, doi: 10.1109/TEM.2019.2918162.

[86] G. Fortino, F. Messina, D. Rosaci, and G. M. L. Sarne, "A reputation capital and blockchain-based model to support group formation processes in the Internet of Things," in *Proc. 6th Int. Conf. Control, Decis. Inf. Technol. (CoDIT)*, Paris, France, Apr. 2019, pp. 888–893.

[87] G. Fortino, F. Messina, D. Rosaci, and G. M. L. Sarné, "Using trust and local reputation for group formation in the cloud of things," *Future Gener. Comput. Syst.*, vol. 89, pp. 804–815, Dec. 2018.

[88] C. Lin, D. He, X. Huang, M. K. Khan, and K.-K.-R. Choo, "DCAP: A secure and efficient decentralized conditional anonymous payment system based on blockchain," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 2440–2452, Jan. 2020.

[89] H. Cui, Z. Wan, X. Wei, S. Nepal, and X. Yi, "Pay as you decrypt: Decryption outsourcing for functional encryption using blockchain," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3227–3238, Feb. 2020, doi: 10.1109/TIFS.2020.2973864.

[90] S. Gao, M. Zhou, Y. Wang, J. Cheng, H. Yachi, and J. Wang, "Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 2, pp. 601–614, Feb. 2019.

[91] P. Zhang, S. Shu, and M. Zhou, "Adaptive adjustment of dynamic detection cycle for fault detection in clouds," *IEEE Trans. Ind. Informat.*, to be published, doi: 10.1109/TII.2019.2922681.

[92] P. Zhang, Y. Kong, and M. Zhou, "A trust model based on double-blind anonymous evaluation in public clouds," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published, doi: 10.1109/TSMC.2019.2906310.

[93] P. Zhang and M. Zhou, "Dynamic cloud task scheduling based on a two-stage strategy," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 2, pp. 772–783, Apr. 2018.
[94] P. Zhang, M. Zhou, and G. Fortino, "The security and trust in fog computing environment," *Future Gener. Comput. Syst.*, vol. 88, no. 11, pp. 16–27, May 2018.
[95] P. Zhang, M. Zhou, and X. Wang, "An intelligent optimization method for optimal virtual machine allocation in cloud computing systems," *IEEE Trans. Autom. Sci. Eng.*, pp. 1–11, Mar. 2020, doi: 10.1109/TASE.2020.2975225.

Peiyun Zhang (Senior Member, IEEE) received the B.S. degree in applied electronics from Anhui Normal University, Wuhu, China, in 1998, the M.S. degree in computer science from Northwest University, Xi'an, China, in 2005, and the Ph.D. degree in computer science from the School of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing, China, in 2008.

She did post-doctoral research at the University of Science and Technology China, Hefei, China, from 2010 to 2013, and was a Visiting Scholar with the New Jersey Institute of Technology, Newark, NJ, USA, in 2016. She is a Professor with the School of Computer and Information, Anhui Normal University. Her research interests include blockchain, cloud computing, big data, trust computing, web service, and intelligent information processing. She has published over 50 articles in these areas.

MengChu Zhou (Fellow, IEEE) received the B.S. degree in control engineering from the Nanjing University of Science and Technology, Nanjing, China, in 1983, the M.S. degree in automatic control from the Beijing Institute of Technology, Beijing, China, in 1986, and the Ph.D. degree in computer and systems engineering from the Rensselaer Polytechnic Institute, Troy, NY, USA, in 1990.

He joined the New Jersey Institute of Technology (NJIT), Newark, NJ, USA, in 1990, where he is a Distinguished Professor of electrical and computer engineering. He has led or participated in over 50 research and education projects with total budget over U.S. $12 million, funded by the National Science Foundation, Department of Defense, NIST, New Jersey Science and Technology Commission, and Industry. He has over 800 publications, including 12 books, more than 500 journal articles (more than 400 in the IEEE TRANSACTIONS), 23 patents, and 29 book chapters. His research interests are in Petri nets, intelligent automation, the Internet of Things, big data, web services, and intelligent transportation.

Dr. Zhou is a fellow of the International Federation of Automatic Control (IFAC), the American Association for the Advancement of Science (AAAS), and the Chinese Association of Automation (CAA). He was a recipient of the Excellence in Research Prize and Medal from NJIT, the Humboldt Research Award for US Senior Scientists from Alexander von Humboldt Foundation, and the Franklin V. Taylor Memorial Award and the Norbert Wiener Award from the IEEE SMC Society. He has been among most highly cited scholars for years and ranked top one in the field of engineering worldwide in 2012 by Web of Science. He is also the Founding Editor of the *IEEE Press Book Series on Systems Science and Engineering* and the Editor-in-Chief of the IEEE/CAA JOURNAL OF AUTOMATICA SINICA. He served as an Associate Editor for the IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, and the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, and an Editor for the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING. He served as a Guest-Editor for many journals including the IEEE INTERNET OF THINGS JOURNAL, the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, and the IEEE TRANSACTIONS ON SEMICONDUCTOR MANUFACTURING. He is currently an Associate Editor of the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, the IEEE INTERNET OF THINGS JOURNAL, the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, and *Frontiers of Information Technology & Electronic Engineering*. He was the General Chair of the IEEE Conference on Automation Science and Engineering, Washington D.C., August 23–26, 2008, the General Co-Chair of the 2003 IEEE International Conference on System, Man and Cybernetics (SMC), Washington D.C., October 5–8, 2003, and the 2019 IEEE International Conference on SMC, Bari, Italy, October 6–9, 2019, the Founding General Co-Chair of the 2004 IEEE International Conference on Networking, Sensing and Control, Taipei, March 21–23, 2004, the General Chair of the 2006 IEEE International Conference on Networking, Sensing and Control, Ft. Lauderdale, Florida, USA, April 23–25, 2006, and the Program Chair of the 2010 IEEE International Conference on Mechatronics and Automation, August 4–7, 2010, Xi'an, China, the 1998 and 2001 IEEE International Conference on SMC, and the 1997 IEEE International Conference on Emerging Technologies and Factory Automation.