Computer Communications 136 (2019) 10–29

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Survey on blockchain for Internet of Things

Xu Wang [a,b,1], Xuan Zha [c,a,b,1,*], Wei Ni [d], Ren Ping Liu [b], Y. Jay Guo [b], Xinxin Niu [a,e], Kangfeng Zheng [a]

[a] School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing, China
[b] Global Big Data Technologies Centre, University of Technology Sydney, Australia
[c] China Academy of Information and Communications Technology (CAICT), Beijing, China
[d] Cyber-Physical System (CPS), Data 61, CSIRO, Sydney, NSW, Australia
[e] State Key Laboratory of Public Big Data, Guizhou, China

ARTICLE INFO

Keywords:
Blockchain
Internet of things (IoT)
Consensus protocol
Data structure

ABSTRACT

The Internet of Things (IoT) is poised to transform human life and unleash enormous economic benefit. However, inadequate data security and trust of current IoT are seriously limiting its adoption. Blockchain, a distributed and tamper-resistant ledger, maintains consistent records of data at different locations, and has the potential to address the data security concern in IoT networks. While providing data security to the IoT, Blockchain also encounters a number of critical challenges inherent in the IoT, such as a huge number of IoT devices, non-homogeneous network structure, limited computing power, low communication bandwidth, and error-prone radio links. This paper presents a comprehensive survey on existing Blockchain technologies with an emphasis on the IoT applications. The Blockchain technologies which can potentially address the critical challenges arising from the IoT and hence suit the IoT applications are identified with potential adaptations and enhancements elaborated on the Blockchain consensus protocols and data structures. Future research directions are collated for effective integration of Blockchain into the IoT networks.

Contents

* Correspondence to: Building 11, UTS, Ultimo, NSW, Australia.
E-mail address: xuan.zha@student.uts.edu.au (X. Zha).

[1] X. Wang and X. Zha contributed equally to this work.

https://doi.org/10.1016/j.comcom.2019.01.006
Received 7 August 2018; Received in revised form 6 January 2019; Accepted 11 January 2019
Available online 16 January 2019

1. Introduction

Traditional security mechanisms alone, such as cryptographic techniques [1], are not enough to preserve data integrity in this enormous scale, thus seriously restricting the adoption of IoT in the future. The Internet, on which IoT is based, is inherently insecure, where data security was an afterthought in the design, as can be evident from continual patches and manual handling [2]. Moreover, IoT has a substantially different architecture from the Internet, extending network connectivity and computing capability to objects with limited computing power, such as sensors and throw-away items, and allowing these devices to generate, exchange and consume data with minimal human interventions [3]. Simply extending computationally demanding and costly Internet security solutions to IoT is neither scalable nor practical [4]. Built on top of the Internet, cloud services have been widely adopted for the processing and storage of the large amount of IoT data [5]. In many cases, IoT data can be stored at different servers across a cloud, and processed and accessed in a distributed manner [6,7]. However, cloud services could inherent insecurity from the Internet, and are susceptible to cyber attacks such as SQL injection [8] and data tampering [9], and vulnerable to single-node failure [10]. In general, cloud services cannot ensure data integrity and availability [8], as expected for IoT applications.

Being a distributed, incorruptible and tamper-resistant ledger database, Blockchain has the potential to address the critical security issues of IoT, particularly on data integrity and reliability [11]. Blockchain allows software applications to send and record transactions/events in a trustworthy and distributed (peer-to-peer) manner. Blockchain is rapidly gaining popularity and used extensively for applications including smart contracts [12], distributed storage [13] and digital assets [14]. The potential applications of Blockchain in IoT include recording events (such as temperature, moisture or location changes) and creating tamper-resistant ledgers that are readable only to certain parties, e.g., specific participants in a supply chain.

With the Blockchain technologies, the security requirement of IoT can be fulfilled [10,15]. The following prominent features of Blockchain can contribute to the integrity of IoT applications and so enhance the IoT security:

- *Decentralization:* The peer-to-peer network setting of Blockchains is inherently suited for IoT networks which are typically distributed, for example, Blockchain in VANET [16,17]. Blockchains can record transactions between multiple parties without central coordination. This can provide flexible network configurations, and reduce the risks of single-point failures.
- *Integrity:* Blockchains are able to keep transactions permanently in a verifiable way. Specifically, the signatures of the senders in transactions can guarantee the integrity and non-repudiation of the transactions. The hash chain structure of Blockchains ensures that any recorded data cannot be updated, even partly. The consensus protocols of Blockchains can guarantee valid and consistent records. The protocols can also tolerate failures and attacks, e.g., attackers with less than $\frac{1}{2}$ hash power in Proof of Work (PoW), or less than $\frac{1}{3}$ of nodes in Practical Byzantine Fault Tolerance (PBFT) consensus protocol [18]. All these are critical to IoT applications, where IoT data can be generated and processed by heterogeneous devices or in heterogeneous network environments.
- *Anonymity:* Blockchains can use changeable public keys as users' identities to preserve anonymity and privacy [19]. This is attractive to many IoT applications and services, especially those which need to keep confidential identities and privacy [20].

Interests in applying Blockchain to IoT networks have already emerged in academia and industry, with the goal of providing security [21–27]. In this sense, cloud can provide distributed storage for IoT applications, while Blockchain can secure the integrity of the storage and prevent data tampering. Blockchain and cloud can be integrated as Blockchain-based Distributed Cloud [28].

However, existing Blockchain technologies can be inefficient for IoT applications, due to the aforementioned massive deployment of IoT devices, non-homogeneous network structure with strong partitioning, and subsequently huge sensory data and demands for high capacity

in Blockchain (i.e., high transaction or block generation speed) [29]. Particularly, physical characteristics of IoT devices and networks, such as limited bandwidth and connectivity, non-trivial network topology, and unpredictable link delays, can cause discrepancy or inconsistency between the records maintained in a distributed fashion at different locations. In fact, the record generation speed needs to be restrained by the propagation speed of blocks, which are the data units of Blockchains. Existing Blockchain technologies, which nearly unexceptionally operate at the application layer and neglect these physical aspects of networks and devices, substantially reduce the block generation speed to be far slower than the propagation, thus resulting in inefficient uses of Blockchain.

In this survey, we investigate the key challenges and the benefits of Blockchain in IoT applications. The state-of-the-art Blockchain technologies in terms of consensus protocols and data structures are analyzed. The limitations of the current Blockchain technologies for IoT applications, as well as future potential research directions, are presented.

It is worth pointing out that there have been a number of recent surveys on Blockchain in general [30,11,31–33], and Blockchain for IoT applications [14,12]. Those surveys are heavily emphasized on design and application. In contrast, in this survey, we emphasize on theoretic background of Blockchain. We are particularly interested in identifying the limitations and gaps of existing theories and understanding their impacts on the scalable design of Blockchain for IoT applications.

2. Survey methodology

This survey summarizes recent research breakthroughs on Blockchain and Blockchain-based IoT applications. It is based on a total of 244 research references identified from Google Scholar, Web of Science, IEEE Xplore, Elsevier, as well as online resources, e.g., webpages and developer communities, to provide a timely review of Blockchain technologies. The references are categorized holistically in accordance with the following five important aspects of Blockchain IoT technologies.

Aiming at the IoT security issues, the survey first summarizes the characteristics of IoT and performs security analysis on IoT in Section 3. Special attention is given to the unique features of IoT networks and applications, such as mobility, low cost, high throughput requirement, a large number of devices, big IoT data, decentralized network architecture, and unstable connections. By reviewing the latest security researches on IoT, the paper identifies the security issues in IoT, such as attacks to end devices, attacks to communication channels, attacks to network protocols, attacks to sensory data, denial of service attack and software attacks.

By taking the first Blockchain application, i.e., Bitcoin, for an example, preliminaries on Blockchain are provided in Section 4, including the chained data structure, Byzantine Generals' Problem, and consensus protocols. By reviewing theoretical attack models and analyzing existing Blockchain attacks, this survey presents security analyzes on Blockchain. Typical attacks include double spending attack, consensus protocol attack, eclipse attack, and distributed denial-of-service attack. Blockchain also suffers from programming fraud, vulnerability of smart contract, and leakage of private key, as discussed in Section 4.

In Section 5, this survey reviews industrial Blockchain-based IoT applications and projects, with an emphasis on two typical structures for Blockchain based IoT applications, namely, IoT involved Blockchain and Blockchain as a service for IoT. Critical challenges of applying Blockchain in IoT are presented by studying Blockchain performance and IoT requirements. Potential Blockchain designs and technologies which can be applied in IoT applications are examined, followed by discussions on privacy, identity and access control.

The survey further elaborates on the suitability of the latest Blockchain technologies in IoT from three dominating categories of Blockchain, namely, public Blockchain, private Blockchain, and hybrid Blockchain. Popular block validation mechanisms, such as proof of work, proof of X and Byzantine fault tolerance, are explained in detail. In addition to the chain structure, other data structures which improve the Blockchain performance and benefit the application of Blockchain in IoT, including DAG, GHOST and others, are presented. Impactful projects and technologies are compared from the aspects of capacity, scale, and specific features. Details are provided in Section 6.

Last but not least, the survey points to a number of research directions and opportunities to bridge the current gap between the requirements of IoT applications and the limitations of the current Blockchain technologies. The potential research directions include side chain, IoT-specify consensus algorithms, simplified payment verification and editable Blockchain, as discussed in Section 7.

3. Limitations of IoT security

IoT network prevails with its ability to interconnect numerous devices possessing various sensing and computing abilities with little human interventions [34]. Sensing and actuating devices form heterogeneous IoT networks to provide various applications. Typical IoT applications include smart home, smart transport, eHealth and smart grid [35].

A typical IoT architecture consists of *Perception, Networking, Service, and Interface Layers*, from bottom to top [36]. The Perception layer, also known as the sensor layer in other IoT architectures summarized in [37], consists of sensors and actuators collecting and processing environmental information to perform functions, such as querying temperature, location, motion, acceleration. The perception layer is an indispensable part of a variety of IoT applications [35]. Various types of end devices can be adopted in the perception layer to bridge the physical world and digital world. Typical end devices include Radio-Frequency IDentification (RFID), wireless sensors and actuators, Near Field Communications (NFC), and mobile phones. For example, RFID tag is a small microchip attached to an antenna. By attaching RFID tags to objects, the object can be identified, tracked, and monitored during logistics, retailing, and supply chain. The Networking layer is responsible for connecting other smart things, network devices, and servers. The Service layer creates and manages specific services to meet the IoT application requirements. The Interface layer facilitates data use interactions with objects for specific applications [36].

3.1. Characteristics of IoT

IoT applications have the potential to affect every aspect of the human daily life. They can be classified into the following four domains: transportation and logistics, healthcare, smart environment (including smart home), and personal and social applications [38]. The end-devices, communication and networking technologies differ to meet targets and demands of various applications. The following are two main aspects that differ among applications.

- Mobility versus stable topology: The topology of IoT applications can vary with different speed. The typical applications with stable and mobile topologies are smart home and vehicular ad hoc networks (VANETs) for transportation application, respectively. Most devices in smart home are stable and consist a stable network topology, while vehicles move rapidly and lead to time-varying topologies. The mobility of the end devices makes the network connectivity unpredictable and entities management challenging [39,40].
- Low-cost versus high-capacity performance: IoT devices are heterogeneous with different hardware platforms and abilities. One type of IoT devices is sensors with tiny size and limited resources for processing, communication and storage. Such devices are typically low-cost and thus can be widely deployed in large scales to measure temperature, pressure, humidity, medical parameters

of human bodies, and chemical and biochemical substances [41]. They typically communicate in wireless ad hoc or mesh networks such as ZigBee [42]. Such sensors are often powered by limited battery, making limited energy a major concern. Recently, new communication technologies, e.g., NB-IoT [43], have been proposed to extend the lifetime of sensors, but sensors are still limited in process, communication and storage abilities. Another type of IoT devices can be more expensive and more powerful, such as mobile phones and vehicles. They have large battery and stronger capabilities of computing and storage. Hence, such kind of devices can contribute to higher capacity.

Implemented with heterogeneous end devices and different protocols, IoT networks have some common IoT-specific characteristics as follows,

- Enormous number of nodes and big IoT data: The number of IoT devices will continuously increase. The number of connected devices in IoT is expected to increase up to 20.4 billion by 2020 [44]. IoT faces not only a large number of nodes but also growing demand for capacity, as numerous end devices sense and collect mass data.
- Decentralization: Decentralization and heterogeneity are the two major characteristics of IoT [45,46]. Decentralization is essential given the large number of IoT nodes, such as in the smart city, because the data to be processed at the same time is considerably huge [47]. IoT devices collect, process and storage data in a decentralized manner. Decentralized algorithms in IoT, e.g., clustering algorithms in wireless sensor network (WSN) and decentralized computing, can contribute to the capacity and scalability of IoT [47].
- Unstable and unpredictable connections: The unstable and unpredictable connections of IoT devices are not only caused by the mobility and the sleep/idle mode of IoT devices, but also typical unreliable wireless links to IoT devices [48]. As a result, a IoT network may divide into disconnected partitions and the partitions can vary with time.

3.2. Security analysis on IoT

Specific characteristics of IoT make data security a severe problem in IoT [38,49]. Firstly, many IoT devices are deployed in human unfriendly and unattended areas, and it can be impossible to keep an eye on the huge number of devices all the time. This makes devices vulnerable to multi-dimensional harms [50]. For example, adversaries may physically capture and control these devices to invade IoT networks [51]. Traditional security mechanisms [52], such as the asymmetric encryption, are computationally demanding for IoT devices with limited abilities. Data from sensors can be stored, forwarded and processed by many different intermediate systems, which increases the risk of being tampered and forged. The unreliable and open wireless channels with broadcast nature bring additional risks to data security. The complexity of the IoT system further increases the above vulnerabilities [53].

The following summarizes the typical attacks on IoT networks from the bottom layer to the top [54].

3.2.1. Attacks to end devices

Adversaries physically capture and control the nodes via node capture attacks. The secret information stored in the captured nodes, such as keys and certificates, become visible to the adversaries [51]. The adversaries can further utilize the captured information to pretend as legitimate nodes and perform other attacks, such as the false data injection attack [55].

3.2.2. Attacks to communication channels

Adversaries may eavesdrop on and interfere with transmitting channels, exploiting the broadcast nature of radio. If signals are not encrypted, the adversaries can readily obtain the information. Even if the signals are encrypted, the adversaries are still able to analyze the streams of signals and infer private information, such as the locations of the sources or destinations [56]. The adversaries can also interfere and even jam the wireless channels by sending noisy signals [57].

3.2.3. Attacks to network protocols

By exploiting the vulnerabilities of network protocols, the adversaries can launch sybil attack, reply attack, man-in-middle, blackhole, wormhole attacks and so on [58]. For example, a sybil device impersonates several legitimate identities in IoT systems. Such attacks would compromise the efficiency and accuracy of voting mechanism and multi-path routing protocols [58].

3.2.4. Attacks to sensory data

IoT networks can communicate by using ad hoc protocols, i.e., messages are transmitted hop-by-hop till reaching their destination. This provides the adversaries opportunities to tamper data or inject false data. An adversary, as a forwarder, can tamper the messages and forward the messages to other nodes, known as data tampering [20]. Authentication algorithms are deployed to prevent the data tampering. False data injection attack refers that adversaries send false data across the targeted network with legitimate identities [55]. Once the false data are accepted, IoT applications may return erroneous instructions or provide wrong services, compromising the reliability of IoT applications and networks. For example, the traffic congestion may aggravate if vehicles accept false road assistant messages. False data injection attacks can hardly be prevented by authentication algorithms.

3.2.5. Denial of service (DoS) attack

The DoS attack represents a category of attacks, which exhaust resources and congest services of IoT systems [57]. For example, a sleep deprivation attack [59] is to break the programmed sleep routines and keep devices or nodes awake all the time until they are out of battery power supply. IoT devices have limited network and communication resources, and thus the DoS attacks can be catastrophic. Such attacks exhaust the limited energy of sensory nodes, reduce the network connectivity, paralyze the entire network, and reduce network lifetime [59].

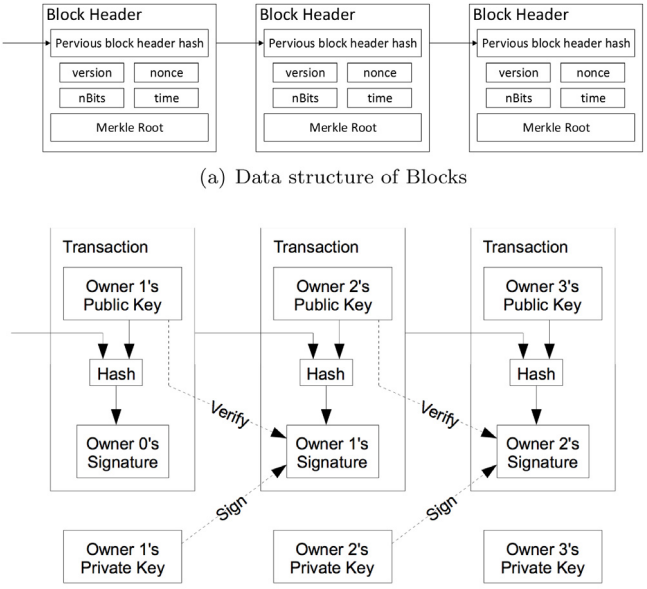
3.2.6. Software attacks

Software attacks refer to a series of attacks which utilize backdoors of software to modify software and control operations [60]. Typical software attacks include malicious virus/worm/scripts [61,60]. Intrusion detect system (IDS) and other traditional Internet security mechanisms are used to tackle the software attacks [62].

Security is a critical concern to IoT applications. Particularly, the integrity of IoT data and devices, e.g., sensor readings and actuator commands, is the basic guarantee for securing IoT operations. Effective mechanisms need to be designed to protect IoT communications for confidentiality, integrity, authentication and non-repudiation of information flows [63]. The IoT devices need to be identified to ensure the data integrity from the origin, which conventionally relies on trusted third parties, e.g., Identity Provider [64]. The authentication and encryption algorithms are used to protect the confidentiality and integrity of IoT data [65]. After the sensory data are sent to the data storage, the data security relies on the data storage service [66].

4. Existing Blockchain technologies

Blockchain provides decentralized data storage service with a tamper-resistant ledger consisting of blocks chained in serial in distributed networks. It can record and secure transactions or transactional events using cryptography [67]. The first Blockchain was proposed



(a) Data structure of Blocks

(b) Data structure of Transactions [68]

Fig. 1. The data structure of the Bitcoin Blockchains.

by Satoshi Nakamoto in 2008 [68] and implemented in 2009 as the enabling technique for the proliferating cryptocurrency — Bitcoin [69].

Blockchain records data in a secure and distributed manner. The basic unit of records in Blockchain is the transaction. Each time a new transaction is generated, it is broadcast to the entire Blockchain network. Nodes receiving the transaction can verify the transaction by validating the signature attached to the transaction, and mine verified transactions into cryptographically secured blocks. Such nodes are known as block miners (or miners for short). To allow a miner to create a block, a consensus problem needs to be solved in a distributed manner. The miners that manage to solve the consensus problem broadcast their new blocks throughout the network [70].

Upon the receipt of a new block, the miners yet to be able to solve the consensus problem append the block to their own chains of blocks locally maintained at the miners, after all the transactions enclosed in the block are verified and the block is also proven to provide the correct answer to the consensus problem. The new block contains a link to the previous block in the chains, by exploiting cryptographic means. All miners can synchronize their chains on a regular basis, and specific terms are defined to ensure the consistent ledger shared across the distributed network, e.g., Bitcoin Blockchain only keeps the longest chain, in the case where there is discrepancy among the chains.

In the following, more detailed descriptions are provided on these key components of Blockchain, i.e., the data structure, the consensus protocol, smart contracts and the security analysis on Blockchain.

4.1. General data structure

As the basic units in Blockchain, transactions are the records of events observed by the miners in the network. A cryptographic private key is used to sign a transaction. The resultant signature is attached to, as an integral part of, the transaction, providing a mathematical proof that the transaction comes from the owner of the private key. The public key, corresponding to the private key, is known to miners for verifying the genuineness of the transaction. It can be achieved via using the public key as the source address in the transaction, preloaded the public key at all miners, or attached the public key and the digital certificate of the public key to the signatures for transmission. Powered by cryptography, the transaction binds an event and its initiator without

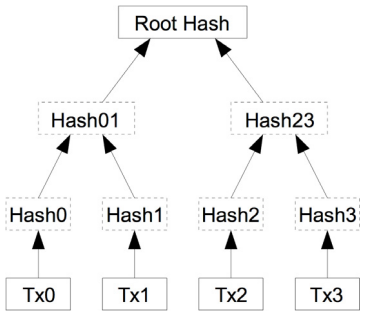


Fig. 2. Transactions are hashed in a Merkle Tree [68].

doubt. Transactions were first used in Bitcoin to capture the financial interactions between two financial parties [68]. Transactions have also been used to elaborately assign the ownership rights and realize programmable events [71,72].

An ordered, backward-linked list of blocks is maintained, as a local record of transactions, at every miner of a network [31]. Being the element of the ledger, every block encapsulates a batch of verified transactions. Every block also has a header containing a link to the parent (previous) block (which is the hashed value of the parent block, e.g., in Bitcoin Blockchain), and an answer in response to the consensus problem, as will be described shortly. The block header may contain other fields, such as timestamp, depending on specific demands. Each block is uniquely identified by a hash value, generated using the cryptographic hash algorithm on the header of the block.

The sequence of hash operations, which link each block to its parent block, creates a tamper-resistant chain which can trace back all the way to the first block ever created. In this way, blocks are chained together to act as the ledger at every individual node, as shown in Fig. 1(a). Note that the link to the parent block is inside the block header and thereby affects the current block's hash value. To modify one block in an available chain, the following blocks, including the child and grandchild blocks, would all need to be recalculated to meet all relevant consensus problems. However, such recalculation is meant to be prohibitive, e.g., requiring intractable computations in the Bitcoin Blockchain. Moreover, the existence of long chains of blocks further secures the intractability of tampering in practice, and constructs tamper-resistant ledger. The locally maintained chains of blocks are regularly compared and updated across the network [30]. Only one chain, e.g., the longest chain in the Bitcoin Blockchain, is publicly accepted to be the ledger of the entire system, and all the locally maintained chains are updated accordingly.

The block header also includes a field that contains information of all transactions in the current block, e.g. the Merkle Root in the Bitcoin Blockchain [73] in Fig. 1(a). Typically, a Merkle tree [74] is built with transactions as leaves, to improve storage efficiency in a block. The Merkle tree has the tree structure in which every leaf node is a transaction and every non-leaf node is the hash of its child nodes, as shown in Fig. 2. The root of the tree is named as "Merkle root". By using the Merkle tree, peers in the Blockchain network can confirm whether a transaction has been mined into a block by verifying the hash of the corresponding branches rather than the transactions mined in the block; or in other words, the entire Merkle tree. By this means, the requirement of storage, memory and network capacity can be highly reduced.

Transactions and blocks are spread and verified across the network (in a peer-to-peer manner) to form distributed consensus. Take Bitcoin for example. When a node generates a valid transaction, the node sends an inventory (inv) message containing the hash of the transaction (TXID), instead of actual transaction data, to all of its neighbors. Neighbors who do not have this transaction respond to the sender. Accordingly, the transaction is transmitted to those neighbors. Once the transaction has been successfully verified, it is further spread to their

subsequent neighbors. This progress continues until the entire network receives the transaction.

Note that only the nodes which generate transactions in the first place are responsible for the propagation of the transactions and can rebroadcast the transactions whenever required. The propagation of blocks resembles that of transactions [30]. The miners, which solve the consensus problem and generate new blocks, take the responsibility of spreading the blocks across the network.

4.2. Byzantine generals' problem and consensus protocol

A fundamental theory that Blockchain exploits extensively is Byzantine Generals' Problem [75]. The Byzantine Generals' Problem is an agreement problem first generalized in [76]. The problem describes the case that peers try to reach a consensus while traitors among the peers may betray the others and prevent them from reaching the consensus. Possible strategies of the betrayers include ignoring messages, providing fake messages, forging messages of others and "two-face" behavior [77], i.e., a node sends conflicting opinions to different nodes. These strategies can lead to Byzantine failures in networks that require consensus [76].

The Byzantine fail mode is the worst failure mode that distributed servers can fail [77]. The failure modes include authentication-detectable Byzantine failures [78] that Byzantine faulty servers forging are detectable with authentication mechanism; performance failures [79] that servers have to deliver correct results but may be early or late; omission failures [80] that service requests are subject to late service responses; crash failures [76] that a server does not respond to any requests; and fail-stop failures [81] that the state of the server exhibiting crash failures can be detected by other correct servers. Some of the Byzantine failures, such as omission failure, two-face attack, and crash failure, are particularly important to Blockchain and can cause inconsistency in Blockchain. For example, the Bitcoin network allows peers to join and leave freely. The peers leaving can be considered as crash failures or fail-stop failures. Omission failures would result in forks because omission failures can stop mined blocks from broadcasting to the rest of the network. Likewise, double-spending attack, a type of "two-faced" attack [77], also belongs to the Byzantine failure. The attackers of double-spending attacks are betrayed commanders in the Byzantine Generals' Problem [76].

There is a large body of research on replication techniques to tolerate Byzantine failures and implement highly available systems, where, however, most research on replication was focused on techniques that tolerate benign faults [82–84]. Earlier Byzantine agreement protocols [85,76] employed signaling expensive recursive confirmations to gain a whole picture of systems before solving the Byzantine Generals' Problem. The communication overhead of the protocols is so high, and typically exponential to the number of peers [86]. Without assumptions about the behavior of faulty processes, techniques that tolerate Byzantine faults, such as Byzantine Fault Tolerance in [76], can provide a potential solution to Blockchain. A popular technique is state machine replication which is a general method for implementing a fault-tolerant service by replicating servers and coordinating client interactions with server replicas [87].

Consensus protocols, the key of Blockchain to maintain a distributed and consistent ledger without centralized coordination, provide solutions to Byzantine Generals' Problem in Blockchain [33]. Consensus protocols define the law of block generations and block selections. Miners in a Blockchain network mine blocks by solving the consensus problem, which prevents any of potentially adversarial participants or compromised miners from hijacking the block generation process. The consensus problem can be announced by Blockchain service providers, or also be generated in a distributed manner following a globally agreed criterion. For any miner, a consensus problem can be locally developed based on the last publicly accepted block in the Blockchain, the block/transaction that the miners trying to mine, and the complexity requirement of the problem specified within the last consistent block of accepted Blockchain. Moreover, the miners are also able to verify each other's blocks based on their blocks and the predefined criterion.

Consensus protocols in open access networks allows unverified and untrustworthy miners to mine blocks without the requirement of verifying their identities. Such kind of Blockchain is known as public Blockchain. The typical consensus protocols of public Blockchain, i.e., Blockchain in open access networks, include Proof of Work (PoW) adopted by Bitcoin, and Proof of Stake (PoS) adopted by Peercoin [88], as will be described in detail in Section 6. However, independent miners can still produce different blocks at the same time, causing disruptions in the growth of Blockchain. These disruptions are known as fork, i.e., the locally maintained chains of blocks become inconsistent between different nodes [89]. Moreover, a large number of miners expend their resources for mining over the same transactions, leading to considerable energy waste and delay.

The other kinds of Blockchains are private Blockchain or permissioned Blockchain, i.e., Blockchain in permissioned networks [90], where authenticated participating miners notify each other in a peer-to-peer fashion of their observations of transactions. Byzantine Fault Tolerance (BFT) algorithms [76] can be exploited at every miner to synthesize their own observations and those of the others, producing consistent blocks in a distributed manner; as will be discussed in detail in Section 6.

4.3. Security analysis on Blockchain

Blockchain attracts attentions for its highly anti-tampering property in decentralized networks. Specifically, Blockchain does not require peers to trust each other. However, Blockchain still exhibits vulnerabilities [91]. Typical security threats to Blockchain are as follows,

- Double spending: adversaries attempt to mislead the transaction receivers with conflicting transactions, e.g., spending the same coin in Bitcoin. Possible attack methods include sending conflicting transactions [92] and pre-mining one or more blocks to get conflicting transactions accepted by the Blockchain [93].
- Attacks on consensus protocols: attackers could break the security assumption of consensus protocols by possessing a considerably large chunk partition of the computing power of the entire network. Such attackers can control and reconstruct the chain. An example is the 51% attack in PoW Blockchains, e.g., Bitcoin [94]. The attackers, owning more than a half of the hash power can make Blockchain accept illegitimate blocks, by solving the consensus problem (e.g., PoW in Bitcoin) faster than the rest of peers. Currently, it has proved that 33% hash power is sufficient to overpower PoW [95].
- Eclipse attacks: Eclipse attacks refer to the attacks in P2P networks where adversaries monopolize all connections to the legitimate nodes and prevent the legitimate nodes from connecting to any honest peers. Eclipse attack to Blockchain is first rised in Bitcoin [96,97] through the randomized protocol, which defines that a node in Bitcoin connects to a certain number of selected neighbors to maintain the peer-to-peer communications and Blockchain related functions. Ethereum was recently reported to have been exposed to Eclipse attacks as well, through the Kademlia peer-to-peer protocol adopted in Ethereum [98].
- Vulnerability of smart contracts [99]: smart contracts are susceptible due to the openness and the irreversibility of Blockchain. Bugs and frauds are transparent to the public including adversaries. Also, it is challenging to make up bugs in the deployed smart contracts due to the irreversibility of Blockchain. An outstanding example is the attack to the Decentralised Autonomous Organisation (DAO) in 2016, known as the DAO attack, which resulted in a forked Ethereum Blockchain [99].

- Programming fraud: the attackers can exploit frauds in programming codes to extract properties of Blockchain, such as the piracy attack reported in 2018 [100,101].
- Distributed denial-of-service attack (DDoS) [102]: the adversaries exhaust the Blockchain resources (such as exhausting the whole network processing capability) by launching a collaborative attack. In 2016, adversaries took underprice EVM instructions to slow down the processing of blocks [103]. The huge number of accounts with low balance produced by adversaries led to a DDoS attack.
- Leakage of private key [104]: the attackers can steal the private key of an account to take over the account. This can be achieved via traditional network attacks [104] or capturing physical nodes [105].

5. Blockchain for IoT: Applications

IoT networks are data-centric, where data are uploaded by a large number of end devices. This makes both data and devices be the targets of potential attacks on IoT. Sensory data in an IoT system can be personal or sensitive [38], e.g., medical IoT [106]; or from national applications, e.g., the IoT-based smart grid [107] and nuclear factory [108]. The integrity and privacy of the data are significant. Blockchain is believed to hold the key to settle security, data integrity and reliability concerns in IoT networks [14]. Provided guaranteed data integrity, Blockchain has drawn a lot of attentions for various IoT applications (e.g., supply chain management [109] and smart city [110]), beyond the cryptocurrency. Blockchain technologies tackle security risks on both the aspects of sensory data and end-devices.

Correctness of sensory data. The data in Blockchain powered IoT networks can be divided into Blockchain-related data, e.g., account, balance and transaction fee, and IoT-related data, e.g., sensory data. The Blockchain-related data can be verified based on previous transactions, e.g., the expense must be less than the balance of an account, as done in other typical Blockchain applications. The IoT-related data are protected by signatures in a transactional fashion, which guarantees that only the messages sent by the authorized IoT devices are recorded and exploited. On the other hand, the correctness of IoT-related data can be guaranteed by the Oracle service which provides an authenticated data feed [111,112]. The backward-linked hashed structure enhances the trustworthiness of sensory data recorded in IoT-Blockchain ledgers.

Malicious behaviors of IoT devices. The malicious behaviors of end devices in IoT-Blockchain can be summarized as the following three types: (1) sending transactions with false signatures, which can be detected, punished and rejected by the Blockchain system; or (2) sending transactions with false data but correct signatures, which can be eliminated by false data detection algorithms and punishing the transactions source nodes [113]; or (3) consuming resources, e.g., DoS, which can be prevented by transaction fee mechanisms [114].

5.1. Blockchain based IoT applications and projects

Most, if not all, existing Blockchain technologies have focused on the application layer, where the networks are typically abstracted to be peer-to-peer without physical restrictions of the network, devices and bandwidth. For instance, Enigma is a recently developed peer-to-peer Blockchain network for decentralized personal data management [21,22].

IOTA targets at providing Blockchain solutions for IoT networks [23]. Launched in 2016, IOTA is built based on the technology "Tangle" with no chains, no blocks and no fees. Tangle inherits the anti-tampering distributed ledger of Blockchain, using a directed acyclic graph (DAG) structure, instead of chains structures as in Bitcoin. Transactions are the only storage units in IOTA. Each transaction confirms another two previously published transactions. Such flexible structure saves energy and work on mining transactions to blocks. Transactions are verified in parallel and accepted by Tangle almost instantly, which provides IOTA high capacity in terms of transaction rate. IOTA abandons transaction fees, as IoT participants could get discouraged in the case that the consumed transaction fee is comparably high with the recorded value in the transactions.

IOTA supports four types of nodes, i.e., Full node, headless node (specifically full nodes running in the local console), Light wallet and Android wallet [115]. IOTA released a Beta version to support light wallet in 2017 [116]. Light wallets connect to a Server on which the IOTA Reference Implementation (IRI) is running to get network status and broadcast transactions. However, even light wallets still need to do computation required to produce legal transactions, required by the DAG structure [117]. Ability-limited IoT devices, e.g., battery powered nodes, are restricted to run light wallets in IOTA.

There are other Blockchain based IoT platforms for specific targets. Partnering up with Samsung Electronics, IBM proposes a Blockchain-based project, named Autonomous Decentralized Peer-to-Peer Telemetry (ADEPT) [24], and advocated device democracy to be the future of IoT [25]. Furthermore, a Blockchain-based data sharing service for businesses and industries has also been launched by IBM [26,27], where IoT data can be shared through private Blockchain ledgers to prevent disputes among business partners.

5.2. The structure of Blockchain based IoT applications

Two different structures can be applied in IoT-Blockchain applications depending on the various abilities of IoT devices.

5.2.1. IoT involved Blockchain

IoT devices would join the Blockchain network and be part of the core functions of Blockchain [101,16], such as generating transactions of raw sensory data, verifying transactions, and even mining blocks. Three virtual roles, i.e., light node, full node and miner [118], needed to be supported in Blockchain-IoT networks. The vehicle ad hoc network demonstrated on the left-hand side of Fig. 3 is a potential application running on this structure [17]. The miners mine transactions into blocks and store all blocks, and have the highest demand for storage and computation. The full nodes store all the blocks, including the block headers and block bodies, but do not play block mining. The full nodes require massive storage and a certain level of computation. The IoT end devices run as light nodes in Blockchain networks. The IoT devices can generate private keys independently, or register with the certificate authority (CA) for access control and audit. The light nodes store the block headers and generate transactions but not mine blocks, they can be supported by the Simplified Payment Verification (SPV) technology [68], as will be introduced later. The light nodes can require less storage and computing power, as compared with the full nodes and miners. Wallet [119] is a special type of light nodes, requiring the minimum storage and computing power. Wallet only has the basic function of transactions and has to be served by full nodes to retrieve data mined in blocks. Take Hyperledger Fabric for example [120], new clients, e.g., IoT devices, would need to register and enroll in the CA first and then maintain their private keys. Here, private keys possessed by clients (light nodes) are applied to generate signatures of transactions to valid the owners of transactions. The clients only generate and broadcast transactions.

5.2.2. Blockchain as a service for IoT

Blockchain provides a service layer [121,120,122] to integrate with the typical IoT architecture, such as the four-level architecture introduced in Section 3. Typically, this structure consists of three virtual roles, i.e., sensor, agent and miner [121]. The smart home demonstrated on the right-hand side of Fig. 3 is a typical IoT application running on this structure [123]. IoT sensors collect sensory data and interact

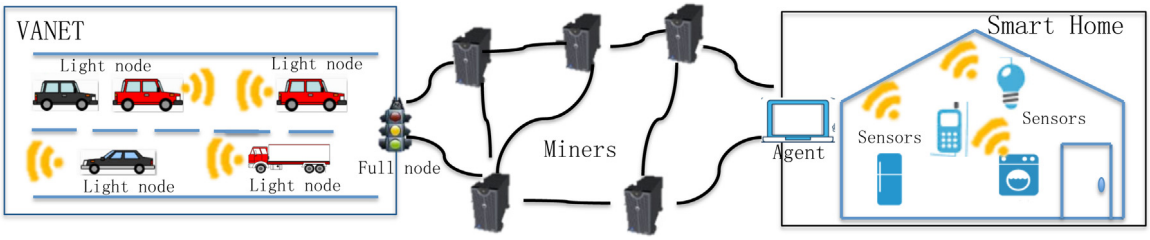


Fig. 3. Illustration of the structure of Blockchain-based IoT networks, where the VANET on the left-hand side is an example of the IoT involved Blockchain structure in Section 5.2.1), and smart home on the right-hand side of the figure is an example of the Blockchain as the service for IoT in Section 5.2.2). IoT data collected by light and full nodes in VANET is sent to miners in the form of transactions, while agents generate and send transactions to miners for sensors in smart home, based on the data collected by sensors. Miners of both structures are computationally capable devices forming peer-to-peer networks to generate blocks and implement Blockchains; see the middle of the figure.

with Blockchain services through Blockchain agents. The sensors do not take part in Blockchain functions. The agents can interpret the collected sensory data as transactions and broadcast the transactions into the Blockchain network [121]. The agents can also take responsibilities of transactions securities using the private keys of the agents, while the IoT devices do not have the keys and are not involved in the Blockchain. Miners, forming a peer-to-peer network, implement the core function of the Blockchain, i.e., verifying transactions and mining transactions into blocks.

5.2.3. Comparison

The IoT involved Blockchain structure achieves security and data integrity by deploying the Blockchain directly on end devices. IoT devices running light node can generate and verify messages in the form of transaction with the help of the SPV technology. On the contrary, the data integrity in the case of Blockchain as the service relies on the security and trustworthiness of the agents. Due to the fact that the agents act as proxies between the IoT devices and the Blockchain network, the agent can carry out the man-in-the-middle attacks, e.g., injection, tampering and forging. In the meantime, the agents increase the risk of single-point failure.

The structure of "Blockchain as services" is easy and flexible to deploy. With the assistance of agents, the IoT module maintains its own characteristics to some extent and, therefore, requires limited modifications on the current system to partner with the Blockchain. For example, the redundancy of sensory data can be solved by using traditional aggregation algorithms [124] at the agents. The aggregated results can reduce the volume of sensory data and relieve the high requirement of IoT applications on the transaction capacity. In contrast, in a IoT involved Blockchain, the IoT devices have to be reprogrammed to run Blockchain applications. The Blockchain applications can be resource consuming, e.g., computation and connection, and can only be deployed on specific devices.

5.3. Challenges of applying Blockchain in IoT applications

Current Blockchains are designed to run in P2P homogeneous networks. However, the characteristics of IoT, for example, limited resource of end devices as compared to high-performance servers or desktop computing devices, prevent directly deploying Blockchain for IoT. The application of Blockchain on IoT devices faces the following challenges.

Computation. The Blockchain activity is unaffordable for the lightweight IoT devices. Some advanced cryptography algorithms, e.g., zero-knowledge [125] and Attribute-based encryption (ABE) [122], used in the privacy-preserving Blockchains are too heavy for IoT devices. A full node in Blockchain has to verify and search every block and transaction, which can also be a heavy load for the resource-limited IoT devices [126]. The PoW-like consensus protocols are unable to run on IoT devices. In the case of Bitcoin, the whole network can conduct around 10^{19} hashes per second [127]. Modern Graphics Processing Unit (GPU) can achieve about 10^7 hashes per second [128]. However, even a powerful IoT device, e.g., Raspberry pi 3 [129], can only achieve about 10^4 hashes per second [130]. As a result, the IoT devices cannot contribute enough computational resources and afford the PoW tasks.

Storage. A massive storage required by Blockchain can be prohibitive for IoT devices. There are about 5×10^5 blocks in Bitcoin in about 9 years. The size of the whole Bitcoin Blockchain is around 150 gigabytes [127]. There are about 5×10^6 blocks in Ethereum. The size of the whole Ethereum Blockchain is around 400 gigabytes [131]. The storage of all blocks is necessary. Without this massive data, the IoT devices are unable to verify the transactions generated by others. Also, a transaction sender needs historical data, e.g., balance and transaction index, to generate new transactions. As a result, the IoT devices should either trust itself by taking the storage load, or trust remote servers, which impose extra communication overhead and secured communication between the IoT devices and the trusted servers. Although the storage demands can be relieved by running IoT devices as light nodes in the Blockchain system, which, however, still need to store the block headers. All headers are about 38 megabytes and 2 GB in Bitcoin Blockchain and Ethereum Blockchain, respectively. Even with advanced Blockchain technology, e.g., SPV technology, the header size can be reduced to about 80 bytes for a Bitcoin block [73] and 500 bytes for an Ethereum block [114]. Moreover, it is expensive to store data on Blockchain. For example, the cost per gigabyte data storage in Ethereum is about 2×10^5 US Dollars [132]. Specifically, a single non-zero 32 bytes data costs 20k gwei/gas, and 1 ether is roughly 12.90 US Dollars [132]. The price is too expensive to be practical in IoT applications. IoT generates big data. The total size of data could be explosive in Blockchain powered IoT because every block would be duplicated n times in an n-node Blockchain network.

Communication. Nodes in Blockchain require frequent transmissions and data exchanges. This is because Blockchain runs on a P2P network and keeps on exchanging data to maintain consistent records, e.g., for the latest transactions and blocks. Wireless communication technologies, widely used to connect IoT devices, suffer from shadowing, fading, and interference, and more unreliable than wired connections [133] in typical Blockchain projects, e.g., BitCoin. The capacity of wireless technologies is far lower than the requirement of Blockchain. For example, Bluetooth (IEEE 802.15.1) can provide 720 kbps data rate; ZigBee (IEEE 802.15.4) can provide 250 kbps data rate; Ultra-wideband (UWB, IEEE802.15.3) can provide 110 Mbps data rate; Wi-Fi (802.11 a/b/g) can provide 54 Mbps data rate [42]. NB-IoT [43] can provide around 100 kbps signal rate [134].

Energy. Some IoT devices are designed to operate for a long time with battery energy supple. For example, an IoT device is designed to consume 0.3mWh per day, and operate at least 5 years using a CR2032 battery with the capacity of 600mWh [134]. IoT devices adopt energy saving strategies, e.g., sleep mode [135], and high-efficiency communication technologies, e.g., NB-IoT [134]. However, the computation and communication required by Blockchain operations are typically energy hungry. For example, SHA-256 requires around 90 nJ/B [136]. The normalized communication energy cost of Bluetooth is around 140 mJ/Mb; ZigBee is around 300 mJ/Mb; UWB is around 7 mJ/Mb, and Wi-Fi is around 13 mJ/Mb [42]. As a result, the aforementioned energy budget of 0.3 mWh per day can only support about 0.5 MB data (half of a Bitcoin block) processing and transmission using the ZigBee protocol.

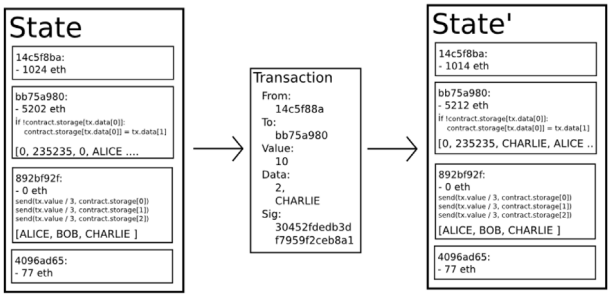


Fig. 4. Ethereum transaction format.

Mobility and partition of IoT. The wireless network can be divided into an infrastructure mode, in which all packets are forwarded by network infrastructures (base stations); and an ad-hoc mode, where the network does not rely on pre-existing infrastructures and each node forwards data for other nodes [137]. The mobility of IoT devices can undermine Blockchain performance. In the infrastructure-based wireless network, the mobility of devices can lead to the growth of signaling and control messages [138]. In contrast, in wireless ad-hoc networks, network partitioning divides the networks into disconnected parts when mobile nodes move with diverse patterns [139].

Latency and capacity. High latency of Blockchain is used to ensure consistency in the decentralized Blockchain networks. The latency which is typically tolerant to Blockchain is unacceptable for many IoT applications. For example, the block confirmation time of 10 min in Bitcoin is too long for delay-sensitive IoT applications, such as vehicle networks. As a matter of fact, high latency of Blockchain leads to the limited Blockchain capacity. The capacity of Blockchains, e.g., 1 MB per 10 min of Bitcoin, is far lower than the requirement of IoT applications. The capacity requirement of IoT varies with different applications. For example, in the application of IoT based smart city [140], the vehicular traces of 700 cars in 24 h is 4.03 GB, around 0.24 MB per hour per car. Meanwhile, the parking lot data from 55 points is 294 KB in around 5 months, i.e., 36 B per day per point. The capacity requirement of IoT applications would continuously proliferate with the increasing number of IoT devices.

5.4. Potential Blockchain designs in IoT applications

5.4.1. Format of transactions

Different from transactions in Bitcoin, the transactions in IoT applications need to support user-defined data structures [141]. A practical example is the transaction in IoT applications built on Ethereum [142,143,141], as shown in Fig. 4. Different from a Bitcoin transaction, an Ethereum transaction has a *data* field indicating the data to be transferred. The data field has variable length, and a sender can pay a higher transaction fee for a longer data field. Note that the transaction fee should be less than the gas limit per block in Ethereum. In other words, the data field cannot enlarge unlimitedly.

The transaction confirmation delay can be affected by transaction size, especially in the IoT networks with unreliable wireless channels. Small transactions can achieve a high transmission success rate and low transmission delay. The User Datagram Protocol (UDP), as a light-weight protocol, is widely used in IoT [144]. Due to the fact that UDP does not provide error-correction, it is better to keep the transaction size less than the payloads of network protocols, e.g., UDP and IP, to avoid fragmentation and improve the transmission success rate. As a result, smaller transactions are expected to be observed by a large number of miners with higher probabilities to be mined into blocks, than large transactions.

The delay can be mitigated with agents which wirelessly connect IoT devices and connect the miners with wire. The agents equally broadcast the transactions with different sizes to the miners.

5.4.2. Incentive and token

Transaction fee is important to balance the transaction cost and adjust the Blockchain resource consumption. For example, transaction fee is used to measure the complexity of transactions in Ethereum [114]. The transactions consuming more resources incur higher transaction fees. On the other hand, the transaction fee also provides a way to reallocate resources, especially in capacity-limited public Blockchains, e.g., Bitcoin with the capacity of 7 tps. In the case of large number of transactions at a moment, transactions can suffer from long confirmation time, and transaction senders can pay more transaction fees to the miners to be given priority (e.g., shorter confirmation time).

The incentive of transaction fee (token) is also attractive and non-negligible in IoT networks. A token system in Blockchain can be used as a reliable reputation or trust system [145]. The transaction fee can increase the cost of attacks in comparison with traditional IoT attacks, e.g., the forged message and DoS attack, and hence discouraging the malicious behaviors [114].

IoT devices may not be able to mine blocks to earn tokens for transaction fees due to their limited resources and typically poor (wireless) backbone links. The IoT devices can "sell" its service, e.g., the renewable energy [146], for tokens. As a return, the service users, e.g., the IoT administrator or cluster header, recharge the IoT devices. The IoT devices are expected to actively take part in Blockchain and obey benign behavioral patterns, although they are prone to act selfish attack [147] with limited bandwidth, energy, and computation resources. With smart contract technologies, the IoT devices can purchase resources, e.g., power or data pack. This can motivate the IoT devices to earn tokens.

5.4.3. Smart contract

A smart contract is a piece of "cryptoeconomically secured execution of code" that runs on the basis of Blockchain [148–150]. Without any assistance of third parties, the smart contract self-executes the corresponding contractual clause once the defined condition is triggered. In addition, it also provides real-time auditing, since all actions are recorded and verified as transactions in a decentralized Blockchain ledger. These transactions are trackable and undeniable, hence enhancing the machine-execution security [151]. Smart contract translates various assets, such as IoT devices and digital assets, into virtual identities in Blockchain, and enables them to interact with other assets [12]. Smart contract is appealing to replace normal contracts as an efficient and secure method. The code of smart contract is stored in Blockchain and identified by a unique address. A smart contract can be called in two ways: one is by validated transactions with a smart contract address in the receiver field; the other way is the internal execution of code [114]. Therefore, all execution records can be traced using the Blockchain ledger. The smart contract is executed independently and automatically on every node in the Blockchain network. Several Blockchain projects, including Ethereum and Bitcoin, have implemented smart contract [152–159,112]. As IoT expects sensors in unmanned areas running and acting automatically with defined rules in decentralized manner, the smart contract has the potential to improve the efficiency and security of IoT applications. IoT devices can carry out autonomous transactions through smart contracts [10]. With smart contract, Blockchain is used to replace the Intelligent Transportation Structure (ITS) and realize reliable firmware update of IoT devices [160].

5.4.4. Off-chain storage

Off-chain storage is a possible solution to reducing the storage cost. Data can be stored separately at another place and use a pointer to index to Blockchain. In [22], to achieve the off-chain storage, two new types of transactions are proposed, namely, transaction for access control management, and transaction for data storage and retrieval. Off-chain key–value store is an implementation of Kademilia [161], a distributed hash table (DHT). The DHT is maintained by a network of nodes which are independent of Blockchain. Data are randomized across the nodes and replicated to ensure availability.

5.5. Security discussions on Blockchain-based IoT applications

Although the Blockchain technology is known to be tolerant to the Byzantine Problem, Blockchain has unsolved security issues which would continue to exist in Blockchain based IoT networks.

5.5.1. Privacy

Blockchain can suffer from privacy issues, including user's privacy and data confidentiality, due to the fact that transactions are designed to be publicly viewed and verified by all the peers.

(i) Users privacy: Although a user can create multiple virtual identities independently in Blockchain, the one-to-many mapping between a physical user and virtual identities can be constructed based on a transaction graph [162–164], and the identity of a physical user can be conjectured [165–167].

A fully anonymous electronic cash should achieve untraceability (or in other words, for each incoming transaction all possible senders are equiprobable) and achieve unlinkability (or in other words, for any two outgoing transactions, it is impossible to prove they are sent to the same person) [168]. Bitcoin is not anonymous but pseudo-anonymous [169,168]. That is achieved by three means, the mapping of a physical user to the virtual identity is maintained by the user only; virtual identities are allowed to be independently generated as many as required; mixing service is provided to mix the funds of a number of virtual identities to confuse and prevent backtracking the original sources of funds [170].

The user's privacy is protected by advanced cryptography technologies in recent Blockchains. Hawk [171] attempted to solve the privacy issue of smart contracts in public Blockchain, which automatically generates an efficient cryptographic protocol using cryptographic primitives, namely, zero-knowledge proofs [172]. Zero-knowledge proof enables a statement to be verified without any information except the statement itself [173]. Zero-knowledge proof has also been used in Zerocoin [174], Zerocash [175], Provisions [176], etc., to achieve anonymous proof of ownership instead of the public-key based signatures. Although privacy-preserving, the zero-knowledge based cryptocurrencies require more resources which highly restrict their applications. For example, a Zerocoin transaction is longer than 45 kB, and needs 450 ms to be verified [175]. Generating a Zerocash transaction consumes around 3.2 GB of memory and around 50 s of computing time [125]. Another key technology to preserve privacy of users is ring signature [177,178], which is performed by any member of a group of users with its private key and others' public keys. In ring signature, a statement is endorsed by members in a particular group of people. For example, Monero [179] is an untraceable Blockchain based on ring signature which breaks the link between sender and transaction. The ring signature does not guarantee the unlinkability of the transaction and receiver, as the transactions do need the address of the receiver to be delivered. Cryptonote [168] achieves the unlinkability with a single address by performing Diffie–Hellman exchanges to get a shared secret between the sender and the receiver. One-time destination key is then generated by the sender and used as the temporary address of the receiver of the transaction. Once the transaction is identified by checking every passing transaction, the real receiver can recover the corresponding one-time key and spend the fund. Note that there is a trade-off between privacy and capacity because the size of a transaction would grow with an increasing size of the group.

(ii) Data privacy: The aforementioned untraceability and unlinkability do not interact with or support data confidentiality. IoT-Blockchain also needs to keep data confidential. The confidentiality of Blockchain can be preserved by confidential transaction technologies. For example, Elements project [180] and Monero [181] keep the content of transactions, i.e., the amount to be transferred, only visible to intended participants. Meanwhile, the content can be verified such that no more coins than available ones can be spent in a cryptographic means. Confidential transactions utilize several cryptographic technologies, including Borromean ring signatures [182] and Pedersen commitment schemes [182].

Another possible solution for privacy is the attributed-based encryption (ABE) [183], where secret keys are generated based on the attributes of peers. By applying ABE, sensory data in transactions can be encrypted and decrypted by the miners and users, using decryption credentials from attribute authorities, if and only if attributes of the miners or users satisfy the access structure of the ciphertext [122]. Fully homomorphic encryption (FHE) [184] which allows computations on the encrypted data provides another solution. Although FHE achieves higher confidentiality as the data is processed without data decryption [21], it is inefficient, and thus has not been implemented in practice [22].

5.5.2. Identity and device management

In IoT applications, the owners should know the identities of their devices and vice versa [53]. However, in current public Blockchains, e.g., Bitcoin and Ethereum, peers are defined by their public addresses that can be created independently without prior notification to the others. A query–answer model based name service is proposed in [185], where virtual identities of IoT devices are verified according to their latest activities. It is considered in [185] that a physical node can be interpreted as multiple virtual nodes in Blockchains. In the case of private Blockchains, the peers need to be authorized to enter the Blockchain network. As a result, the identity management is the fundamental requirement of private Blockchains. For example, Hyperledger Fabric provides identity management to implement the enrollment and transaction certificates [120].

5.5.3. Access control

As a distributed system, Blockchain enables IoT devices to formulate their own access control policies and take full control of their own data, achieving device democracy [25]. One technology to implement access control is programmable smart contracts [186]. The smart contracts, implementing access control policies, can be either deployed upon data, subject to the identity of the data controller or specific data; or upon the data controller for multiple data subjects. The other way to implement access control is to use the Blockchain as a database to store all access control policies for each pair of resource and requester in the form of transactions [187,188]. If an access request is admitted, the access grant transaction can be recorded in the Blockchain and broadcast to the Blockchain network. Otherwise the access request transaction is rejected, and a notification is sent to its sender.

6. Blockchain for IoT: Technologies

In this section, we discuss typical technologies of Blockchains which can be used in IoT applications. We first present three categories of current Blockchain networks and map IoT applications into suitable Blockchain categories. Further the core function of Blockchain, namely, the consensus protocol, is analyzed from two key points, followed by represent Blockchain projects compared in the suitability in IoT applications.

Based on access controls of the Blockchain networks, the state-of-the-art Blockchains can be categorized into public Blockchain, private Blockchain, and hybrid Blockchain which mixes of the former two.

(1) Public Blockchain: The dominant class of Blockchain is public Blockchain in which, with no access control, any uncertified, untrustworthy node can read and record transactions, and take part in mining blocks and contributing to Blockchain [189]. Designed for open-access public distributed networks, public Blockchains can provide strong scalability. However, preserving the consistent records of public Blockchain becomes increasingly difficult, as the network scales up, and would compromise the block generation rate of public Blockchain consequently. This is due to the fact that, without access control, public networks do not have strict control policy on the identification and certification of any participants [90], and therefore the implemented

consensus protocols have to scarify the block generation rate for security. Specifically, PoW and PoX are normally used in public Blockchain as consensus protocols, achieving lower block generation rate compared with PBFT algorithm used in private Blockchain, which will be analyzed in detail later in this section.

Current public Blockchain projects, including Bitcoin and Ethereum, also demonstrate the openness and capacity-limited characteristics. Public Blockchain is suitable for the IoT applications with open access or flexible peers at a large scale, such as VANET and supply chain.

(2) Private Blockchain: Another popular class of Blockchain is private Blockchain which resides in closed proprietary networks with stringent access control and read/write permission, as well as participant identification and certification [190,191]. Private Blockchains can meet the privacy requirement and has been increasingly drawing attention from financial institutions [192]. The proprietary networks, on which private Blockchains operate, can be optimized for high speed and low latency [193]. For example, a high speed of up to tens of thousands of transactions per second can be achieved in private Blockchains [194].

Private Blockchain adopts BFT protocols, i.e., PBFT and its variability, as consensus protocols, which provide higher capacity with restricted access control. The access control provided by private Blockchain further protects IoT applications from external adversaries [195]. In general, private Blockchain is suitable for IoT applications with small scale of miners, because of the high communication complexity and overhead of BFT protocols. When the network size goes beyond twenty, the capacity of private Blockchain dramatically slows down [196].

Apart from various BFT consensus protocols, private Blockchain can use other efficient consensus protocols, e.g., Paxos [82] and Raft [197], in response to specific types of failures, e.g., crash failures [76] and fail-stop failures [81].

(3) Hybrid Blockchain: Another class of Blockchain is hybrid Blockchain which was proposed to leverage the advantages of public and private Blockchains, to be more specific, the block generate rate of private Blockchain and the scalability of public Blockchain [193].

For instance, Luu et al. [198,199] developed a Computationally-scalable Byzantine Consensus Protocol for Blockchains, where the capacity of Blockchain can scale nearly linearly (i.e., $O(\frac{n}{\log n})$ or $O(\frac{n}{\log \log n})$) with the computation capability. In this design, a permission-less distributed network is uniformly clustered into smaller committees. Firstly, the peers in network need to solve the PoW puzzle to prove their identities and avoid Sybil attack. Then peers are uniformly clustered into committees based on their computational power revealed through the required time to solve the PoW puzzle. Each committee processes a disjoint set of transactions. The intra-committee consensus is achieved by using Byzantine consensus protocols, i.e., PBFT. The final consensus among committees, achieved by the Byzantine consensus protocols, is broadcast across the network. This hybrid design exhibits strong scalability to large-scale networks with, e.g., 1600 nodes.

Another recent example of hybrid Blockchain is ByzCoin [89] which dynamically forms hash power-proportionate consensus groups to collect recently-successful block miners. Communication trees can be employed to optimize transaction commitment and verification under normal operation.

More examples of hybrid Blockchain include a resilience optimal Byzantine consensus algorithm that Crain et al. [200] proposed for consortium Blockchain which relies on neither a leader, nor signatures or randomization. The proposed consensus protocol involves reducing multivariate Byzantine consensus to binary Byzantine consensus satisfying a validity property. The property is that, if all non-faulty processes propose the same value, no other value can be decided.

The hybrid Blockchain is attractive to IoT applications due to the complexity and heterogeneity of IoT networks. A hierarchical Blockchain structure was proposed for the smart home applications, where a private Blockchain, maintained by resourceful "miners", runs at every home and public Blockchain runs on the "miner" network [123].

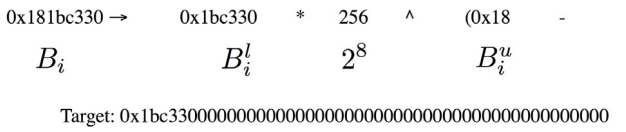


Fig. 5. The conversion of the "nBits" field into a target [73].

```
02000000 ........................... Block version: 2

b6ff0b1b1680a2862a30ca44d346d9e8
910d334beb48ca0c0000000000000000 ... Hash of previous block's header
9d10aa52ee949386ca9385695f04ede2
70dda20810decd12bc9b048aaab31471 ... Merkle root

24d95a54 ........................... Unix time: 1415239972
30c31b18 ........................... nBits
fe9f0864 ........................... Nonce
```

Fig. 6. An example of block header including version, previous block header hash, Merkle root hash, time, nBits and nonce [73].

The above three kinds of Blockchain are suitable for different applications. The consensus protocol is the core to ensure the function of Blockchain. In the Blockchain network, nodes broadcast transactions throughout the whole network and reach consensus on the accepted transactions by following the consensus protocol. Consensus protocol addresses two major problems: *What is the principle to validate unit data?* and *What is the structure of unit data in the Blockchain ledger?*

6.1. The principle of unit data validation

6.1.1. Proof of work (PoW)

PoW provides a practical means to achieve consensus among the chains of blocks generated in a distributed fashion, meanwhile preventing untrustworthy participants from tampering or corrupting the chains. PoW produces problems that are hard to accomplish but easy to be verified, e.g., using hash functions which are one-way functions easy to compute with a given input, but hard to derive the input from the output. Take the Bitcoin Blockchain for example. Every block in Bitcoin takes around 10 min to be mined across the entire network. On the other hand, the answer for PoW can be easily verified with a hash operation. In this way, Bitcoin can implement a one-CPU-one-vote strategy [68] to prevent Sybil attack [201] where a single entity can pretend to be multiple identities in a consensus process.

Bitcoin PoW is set by a global target at the ith epoch, denoted by T_i, which is a 256-bit unsigned integer, as shown the "Target" in Fig. 5. By adjusting the 32 bit "nonce" field, the hash of a valid block header, concatenating all the fields in the header, including version, previous block header hash, Merkle root hash, time, nBits and nonce, demonstrated as Fig. 6, needs to be equal to or less than the target [202]. A smaller T_i is a stricter target and it is hard to find a hash outcome equal to or less than a small T_i through adjusting the "nonce" field. Bitcoin uses SHA-256 [203] as the hash function, which maps data of arbitrary size to 256 bits. The miners have to try exhaustively different values for the "nonce" field until a valid hash value is achieved. Miners have to follow a set of common rules about transactions. For example, none of the new transactions to be mined shall conflict or repeat any transactions already mined in blocks and accepted publicly. Follows the rules, miners can select transactions to be mined in new blocks.

The target of PoW in Bitcoin is adjusted every 2016 blocks [126] to stabilize the block generating rate. Miners have to follow the target renewal process, otherwise, the new mined blocks will not be accepted by Bitcoin network. The target T_i for new blocks to be mined in the ith epoch is compacted to 32 bits, denoted by B_i, and saved in a 32 bit "nBits" field in the block header [73]. The conversion of B_i to target T_i is given by

$$T_i = B_i^l \times 2^{8\times(B_i^u-3)}, \quad (1)$$

where B_i^l is the value of the low 24 bits of B_i and B_i^u is the value of the high 8 bits of B_i. The process of the conversion is shown in Fig. 5.

The difficulty [202] to generate a valid hash value, has been quantitatively characterized by the ratio between the maximum target and the current target, as given by

$$D_i = \frac{T_{max}}{T_i}, \tag{2}$$

where T_{max} is the maximum target and approximately is 2^{256-32}, i.e., 2^{224}. For example, the difficulty of Bitcoin is 678760110082 on 12th June 2017 [127]. With a given difficulty D_i, the average time of mining a block, denoted by $E(t)$, can be approximated to

$$E(t) = \frac{D_i \times 2^{32}}{r}, \tag{3}$$

where r is the hash rate of the miner, i.e., the number of hash operations that can be accomplished per second.

Once a new block is generated, it is sent to the whole network using flooding algorithms [204] i.e., every incoming packet is sent through every outgoing link. When a peer in Bitcoin network receives a new block, it checks whether the "nBits" value matches the target renewal process and calculates the hash of the block header to check whether the hash of the header meets the claimed target in the "nBits" field. The receiver also checks other content of the block for validation [126].

Peers can work co-operatively to eliminate conflicts in-between. This is known as pooled mining [205–207]. However, the pooled mining cannot remarkably increase the Blockchain extension rate, because the difficulty of finding the nonce can be dynamically adjusted in response to the changing block generation rate [202]. Further, pooled mining tends to transfer the distributed Blockchain into a centralized system. This would be detrimental to the tamper resistance of Blockchain. To prevent the peers from working cooperatively, Miller et al. [208] proposed non-outsourceable scratch-off puzzles. The pooled mining is achieved based on the fact that the members of a mining pool do not trust one another, and they submit cryptographic proofs to other pool members in order to demonstrate that they contribute to benefiting the pool. In the pooled mining, an employer can hire miners to mine blocks. The employer will obtain the reward from mined blocks and share the reward with miners according to their cryptographic proofs. The non-outsourceable puzzle is designed to disable the relationship by letting the real miner steal the reward from the employer without trace [208].

In general, PoW is only used to find the nonce and does not contribute useful services. An exception is that Permacoin [209] uses PoW to provide data preservation service. Permacoin requires peers to invest storage to store files, and computational resources to carry out the proof process and provide services.

6.1.2. Proof of X

Participating peers can also be validated via other proofs, instead of finding the nonce, i.e., PoW. Another popular proof is Proof of Stake (PoS) [88], which is an energy-saving alternative to PoW. Instead of demanding users to find a nonce, PoS requires the peers to prove the ownership of the amount of currency under the assumption that the peers owning more currencies would be less likely to attack the network's integrity. Originating from [210], an account balance based selection has been developed to approve blocks. However, such selection is inherently unfair because a single richest participant is bound to dominate the network.

Proof-of-Activity (PoA) [211] incorporates PoW and PoS. First, the miners try to generate empty block headers, i.e., header data that consists of the hash of the previous block, the miner's public address, the index for the block, and a nonce, by solving a hash puzzle like PoW. After that, the empty block headers are broadcast to the network. N "lucky" stakeholders are selected to sign the block header. The Nth stakeholder combines the empty block header, which has been approved by $(N-1)$ stakeholders, and transactions into a block. The reward is shared among the N stakeholders and the miner. Unlike PoW, the attacks with more than 50% hash power in PoA are unable to dominate the existing block chain or determine the chain extension. However, PoA requires the empty block header to be signed and broadcast N times, hence increasing communication complexity and reducing system capacity.

Many other solutions have also been proposed in coupling with the stake size to decide which one to generate the next block. In particular, Blackcoin [212] uses randomization to predict the next generator, and Peercoin favors coin age based selection [88]. Compared to PoW, PoS is more energy-efficient. Unfortunately, since the cost of mining blocks in PoS is low, and nearly zero, PoS is vulnerable to attacks, e.g., long-range attack, nothing at stake attack, initial distribution attack, bride attack, coin age accumulation attack, and precomputing attack [213]. For example, an attacker having enough stake can attempt to overwrite the Blockchain from some existing block. Even adversaries with a minority set of stakes in PoS based Blockchain can produce a valid alternative Blockchain starting from the genesis block (or any sufficiently old block), known as the long-range attack. The nodes newly joining the Blockchain network are not able to reliably distinguish the actual Blockchain and the alternative Blockchain. On the contrary, such attacks are prevented by the enormous amount of computing power/time needed to reconstruct the Blockchain in PoW.

Other proposed PoX approaches used in public Blockchain include Proof-of-Deposit (PoD) [214], Proof-of-Burn (PoB) [215], Proof of Elapsed Time (PoET) [216]. In PoD, the participation in mining requires depositing coins in a time-locked bond account, during which the coins cannot be transferred. Each miner has a voting power corresponding to the amount of the locked coins. A block is valid, as long as it receives $\frac{2}{3}$ of the total voting power. The voting process resembles to PBFT, and is a round-based consensus protocol. The voting process consists of three steps: *Propose*, *Pre-vote* and *Pre-commit*. After a peer has received more than $\frac{2}{3}$ of pre-commits, it proceeds to extend its chain. PoD can destroy the bonded coins of a participator who signs conflicting transactions, so as to avoid double-spending attack [214]. In PoB, a miner sends coins to an un-spendable address, i.e., burn coin, to mine blocks. The coins, from un-spendable addresses, can be shared between miners who mine blocks as rewards. However, the coin burn is uncontrollable, and the total coin can decrease [215]. Contributed by Intel, Sawtooth uses PoET [216] as the consensus protocol. In PoET, every node is given a trusted random time. After the time expires, the corresponding node can generate a block. PoET is based on the Intel trust platform Software Guard Extensions (SGX) [217].

6.1.3. PBFT

Byzantine Fault Tolerance (BFT) [76] is typically used in private Blockchain to formulate consensus protocols and guarantees consistency by exploiting the solutions to the Byzantine Generals' Problems — agreement problems, as described in Section 4.2. Particularly, the PBFT algorithm [18] has been extensively used to eliminate the Byzantine failures. In 1999, Castro and Liskov proposed the first Byzantine-fault-tolerant, state machine replication algorithm, named "practical Byzantine Fault Tolerance (PBFT)" [18], which yields a communication overhead of $O(n^2)$ in a network of n peers. As a leader-based BFT algorithm, PBFT has one *primary* and $(n-1)$ *backups* in an n-node network, where the backups can be corrupted. The *primary* is responsible for receiving the requests from clients and initializing the algorithm. Inspired by Viewstamped Replication [218] and illustrated in Fig. 7, PBFT consists of four stages: (a) a client sends a request to invoke a service operation to the primary; (b) the primary multicasts the operation to the backups; in specific, the *primary* (replica 0) assigns the sequence number to the mth request from the client and multicasts a *PRE-PREPARE* message with the assignment; (c) replicas execute the request and reply to the client; If a *backup* agrees on the assignment, i.e., correct and validated parameters, it multicasts a *PREPARE* message. When a *backup* receives messages that agree on the assignment from a quorum, i.e., $2f$ validated and consistent *PREPARE* messages from

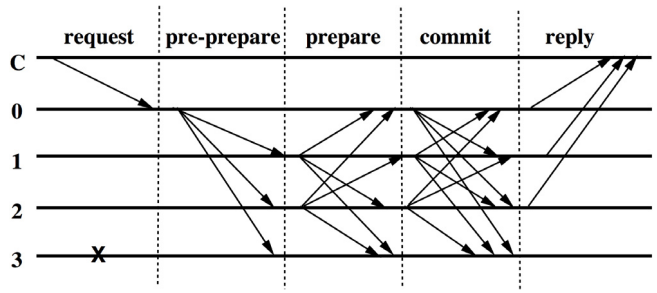


Fig. 7. The PBFT operation in the case of no primary faults [18], where C is the client, replica 0 is the *primary*, replica 1 to replica 3 are *backups* and replica 3 is faulty.

different *backups*, it multicasts a *COMMIT* message. A *backup* executes the request m and sends a reply to the client after receiving $2f$ validated and consistent *COMMIT* messages; and (d) the client waits for $(f+1)$ replies from different replicas with the same result which is the result of the operation tolerant to up to f failures.

The PBFT algorithm is resilient. It has been proved that the PBFT algorithm can ensure n peers within a synchronous and reliable network to reach consensus, as long as there are no more than $\frac{n-1}{3}$ betrayed peers [76]. Specifically, the algorithm only requires $n \geq 3f+1$ replicas to tolerate up to f faulty replicas and guarantee the consistent, fault-free output to the client [219]. This is because $(3f+1)$ *PREPARE* messages at any backup node, including its own, are sufficient at the second stage for a credible, uncorrupted backup to generate a genuine *COMMIT* message. The third and fourth stages can both guarantee the received consistent replies to outnumber the up to f faulty replies at any backup and the client.

The PBFT algorithm is efficient and is able to process thousands of requests per second with processing latency in sub-milliseconds [220]. However, apart from the $O(n^2)$ overhead, PBFT also necessitates all participating nodes to be adequately identified, certificated and authorized. For these reasons, the PBFT algorithm is suitable for private Blockchain in a relatively small and controllable scale.

The PBFT algorithm is susceptive to partitioning in networks. A partition cannot extend its own chain unless the number of trustworthy nodes in the partition, denoted by n_{pb}, meets $n_{pb} \geq n - \lfloor \frac{n-1}{3} \rfloor$; where n is the total number of nodes in a network. This is because PBFT can tolerate at most $\lfloor \frac{n-1}{3} \rfloor$ faulty replicas out of a total of n replicas [18].

6.1.4. Variability of PBFT

Miller et al. [221] developed an asynchronous BFT protocol, named "HoneyBadgerBFT", which can guarantee availability in the absence of time synchronization. HoneyBadgerBFT reduces the atomic broadcast protocol [222] for an asynchronous common subset (ACS) which provides better efficiency. The ACS primitive allows each node to propose a value and guarantees that every node outputs a common vector containing the input values of at least $(n-2f)$ correct nodes in a network of n peers with f failures. HoneyBadgerBFT requires $O(n)$ communication cost in a network of n peers. Therefore, HoneyBadgerBFT can support large network applications, and achieve more than 1500 transactions per second in a network of 104 peers.

A recent international umbrella project on Blockchain, named Hyperledger, is focused on practical Blockchain techniques and implements BFT algorithms in Blockchain [223]. Hyperledger is an open source collaborative effort created to advance cross-industry Blockchain applications. Hosted by the Linux Foundation, the project is participated by leading organizations in finance, banking, IoT, supply chain, manufacturing and technology. The Hyperledger contains a series of independent Blockchain projects, e.g., Fabric [120], Burrow [224], Iroha [225] and Sawtooth [226].

Fabric is the flagship project contributed by Digital Asset and IBM. Fabric has a modular architecture and supports loading modules dynamically, e.g., consensus protocols and membership services. Fabric uses PBFT as its default consensus protocol. To keep Fabric programmable, smart contracts are specially designed to be hosted by container technologies, named as "chaincode". Burrow is an extension of Ethereum and focuses on the permissioned smart contract service. It uses Tendermint [214], a BFT type middleware for Blockchain, as the consensus protocol. Iroha aims to provide encapsulated C++ components for other projects. Iroha also applies PBFT as its consensus protocol.

6.2. The structure of unit data

The structure specifies how unit data is stored and how to decide the main ledger. It refers to the case that more than one ledger exists at the same time in a distributed network. If the different ledgers are all accepted, the Blockchain network gains more capacity but also the risks to double spending. Double-spending attack refers to the fault that a coin is successfully spent more than once [68]. In general case, double-spending attack leads to contradictory records in a distributed system. Blockchain can only ensure that all records are consistent at the end. Before eventually accepted, the records can be temporarily accepted and then dropped. This makes the double-spending attack possible.

6.2.1. Chained blocks

In a Blockchain with chained-data structure, such as Bitcoin, only a single chain can be eventually accepted across the system, and the chain is named as the *main chain* [88]. Each block contains a cryptographic hash of the previous block header, using the SHA-256 hash algorithm, which links the current block to the previous block, as shown in Fig. 1. This prevents tampering upon the Blockchain. It is possible that the chains maintained at different parts of the network are inconsistent, due to a limited view of each part on the rest of the network; or in the other words, network partitioning which can prevail in future IoT networks. To address this, Bitcoin takes a simple rule which has the system to only take the longest of the chains as the main chain and discard the rest [68]. For practical implementation, a peer in Blockchain switches to the longer Blockchain if it sees one, or retains its own.

Based on whether blocks are dropped after locally maintained chains are synchronized for consistent records [32], two different types of inconsistency can be defined as shown in Fig. 8.

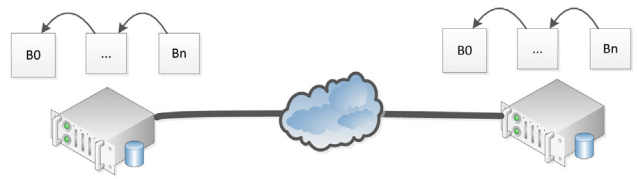
The first type of inconsistency accounts for that only one partition has mined one or more blocks since the last consistent Blockchain. The other partitions reach consensus by accepting the new blocks. No block is dropped during synchronization [73]; see Fig. 8(b);

The second type of inconsistency accounts that more than one partition appends blocks from the last consistent state. Only the longest chain remains, while all the others are recalled. The recalled blocks are named as stale blocks [227]; see Fig. 8(c).

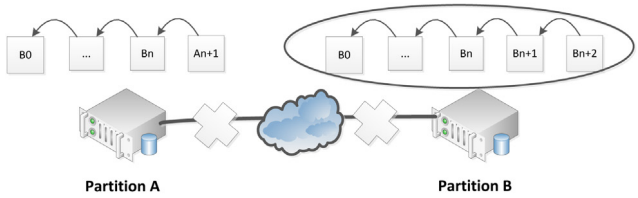
Obviously, a miner can address *the first type of inconsistency* by accepting the latest blocks. In the case of *the second type of inconsistency*, a miner needs to abandon part of its own blocks and switch to the longer main chain.

The second type of inconsistency results in a loss of transactions in stale blocks. Nevertheless, the risk of transaction loss is low for two reasons. The first reason is that, the transactions in a stale block may be in an accepted block, because transactions are visible to many parts of the network and can be observed in different blocks and mined at different nodes. The second reason is that, if a transaction is dropped with the stale block, the transaction returns to the state of an unconfirmed transaction and waits for being put into another new block.

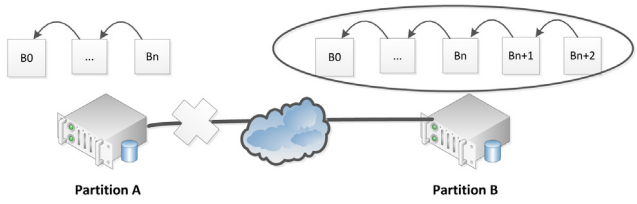
A double-spending attack can be undergone, when conflicting transactions/blocks exist in different partitions in *the second type of inconsistency*. Bitcoin recommends the coin receivers to wait for six block confirmations [68], to prevent double-spending attacks. A double-spending attack also occurs in chained Blockchain by leveraging the consensus protocols. Take 51% attack [94] PoW for example. If a powerful node holds more than 50% of the network computational



(a) The consistent main chain without partitions



(b) The first type of inconsistency



(c) The second type of inconsistency

Fig. 8. Two types of inconsistency in Blockchain with partitions.

resource, the powerful attacker is able to generate blocks so fast and hijack the main chain yielding the longest-chain rule. As a result, the attacker can dominate the chain of arbitrary length. By July 2018, the network hash rate is about 4×10^{19} hash per second [127], and the 51% attack becomes nearly impossible. Moreover, Bitcoin limits the block generation rate to be one block per ten minutes, so as to mitigate inconsistency. This has been achieved by adjusting the difficulty of PoW as described earlier in this section.

6.2.2. DAG

Other solutions for consensus exploit the fact that some abandoned blocks, mined under a consensus protocol but excluded from the main chain because of forks, can be used to improve the capacity. This can be achieved by adjusting the data structure. One of the consensus protocols, named Tangle [228], uses Directed Acyclic Graph (DAG) to organize blocks, instead of chain, where DAG is a finite directed graph with no directed cycles. In Tangle, a transaction must approve (point to) two previous transactions. Finally, one of the conflicting records can win the approval competition and be accepted. Unlike the single-copy in the chain structure, Tangle does not drop conflicting transactions and keeps them in different branches of DAG. The DAG structure can achieve better capacity.

6.2.3. GHOST

Another protocol, named Greedy Heaviest-Observed Sub-Tree (GHOST), arranges blocks in tree structures [229,230]. It takes the path from the genesis block, the first block in the Blockchain, to the heaviest sub-tree which has the maximum number of blocks; or in other words,

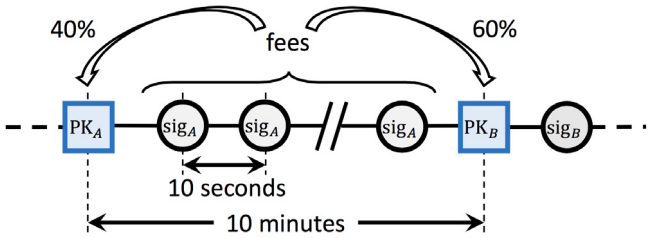


Fig. 9. The structure of the Bitcoin-NG chain. Microblocks (circles) are signed with the private key matching the public key in the last key block (squares). Fee is distributed 40% to the leader and 60% to the next one [232].

contains the heaviest computation quantity as the publicly accepted main chain. GHOST can speed up generating blocks from around 10 min per block in Bitcoin to 12 s per block in Ethereum [231,114]. As a result, the capacity of Blockchain can be improved.

6.2.4. Mix structure

Bitcoin-NG (Next Generation) [232] is a public Blockchain protocol, which puts the task of block generation to computational powerful leaders to accelerate transaction confirmation. Bitcoin-NG decouples Bitcoin's Blockchain operations into two phases of leader election and transaction serialization. The leader election is based on the speed of solving computationally demanding puzzles like PoW. The elected leader is recorded in the *key blocks*. The leader has the responsibility of serializing transactions by generating *microblocks*. A *microblock* contains transactions and a header referring to the previous block. The *microblock* does not contain the nonce and therefore can be generated in a predefined rate which can be much higher than the generation rate of key blocks. The *key blocks* and *microblocks* are chained together like Bitcoin, as shown in Fig. 9. Each block has a header containing the unique reference of its predecessor.

6.3. Comparison of Blockchain for IoT application

A comparative summary of existing Blockchain techniques for IoT application is provided in Table 1 with emphasis on their suitability for IoT networks. All IoT devices can use the Blockchain services. Resourceful IoT devices with powerful computational ability, persistent power supply, sufficient storage and high-speed network connections, such as vehicles [16], can be miners or full nodes in Blockchains. The IoT devices with less powerful computing capability, e.g., smart TV, can be light nodes in Blockchain and obtain Blockchain services via full nodes or miners. The IoT devices with limited storage, computing and communication abilities, can interact with the Blockchain core functions via agents (e.g., the full nodes) [121], as discussed in Section IV-C. The consensus protocols are the core functions which decide the performance of Blockchain-based IoT applications, such as block rates, consistency, scalability and security. PoW based consensus protocols are reported to be the most secure in open networks [234]. However, PoW eliminates the potential of block mining at IoT devices due to its heavy computational requirements. PoS based consensus protocols can significantly reduce the energy consumption, as compared with PoW. PoS provides a chance for IoT devices to take part in block mining. However, the block generation rate of each of PoW and PoS based consensus protocols is limited. PBFT based consensus protocols for private Blockchains can serve IoT systems with high block generation rate but bring constraints on the number of miners that can be involved [86]. Besides the consensus protocols, e.g., PoW, PoS, and PBFT, the capacity and scalability also rely on the running environment and configurations, such as network speed, and block size. The superscripts "-", "*" and "#" are indicative of different sources of the displayed data.

Table 1
Performance comparison of Blockchain in IoT application (N.A.: not available)

Name	Type	Consensus protocol	Capacity	Scale	Application	Merits	Demerits
Bitcoin [68]	Public	PoW + Longest chain	7 tps^{-}	$10^{5\#}$	Cryptocurrency	High partition tolerance Tamper-resistant	Limited capacity High computational complexity
Ethereum [233]	Public	PoW + GHOST	12 seconds/block*	$10^{5\#}$	Cryptocurrency Smart contract Blockchain platform	Programmable High partition tolerance	High computational complexity
IOTA [23]	Public	PoW + TANGLE	>800 tps *	10^{3*}	Blockchain platform IoT	High capacity No transaction fees Partition tolerant	Not programmable
Fabric [120]	Private	PBFT	10^{5} tps*	20*	Smart contract Blockchain platform	High capacity No fork Modular architecture	Low partition tolerance High communication overhead Limited scalability Authentication center required
Burrow [224]	Private	Tendermint	10^{5} tps*	tens*	Smart contract	Smart contract support	Authentication center required
Sawtooth [226]	Public	PoET	N.A.	N.A.	Blockchain platform	Low computational complexity	Only works with Intel CPU
Ppcoin [88]	Public	PoS	0.1 tps$^{\#}$	$10^{3\#}$	Cryptocurrency	Low computational complexity	Risk of attack from the richest
Bitcoin-NG [232]	Public	PoW	tens tps$^{\#}$	10^{3*}	Blockchain	Low computational complexity	Risk of malicious leader
SCOIN [198]	Public	SCP	>22 tps*	80*	Cryptocurrency	Committee structure	High computational complexity
Slimcoin [215]	Public	PoB	N.A.	N.A.	Cryptocurrency	Low computational complexity	Risk of coin loss

- The superscript "-" indicates theoretical analysis. For example, Bitcoin has a theoretical upper bound of 7 tps for the transaction rate which is limited by the block generation rate and the block size.
- The superscript "*" indicates experimentally validated results.
- The superscript "#" indicates historical records. For example, Ethereum network has more than 30,000 nodes across the world in June 2017 [235]. In this sense, those results could be underestimated due to lack of stress-tested.

Among the aforementioned Blockchain projects, Ethereum is appropriate for many IoT applications with large numbers of IoT devices and inhomogeneous network structures. As a public Blockchain, Ethereum exhibits strong scalability by supporting massive heterogeneous devices. The major drawbacks of Ethereum (Homestead launched in 2016 [236]) for IoT applications are high computational complexity and limited capacity. Nevertheless, Ethereum is evolving with efficient PoS consensus protocols in Ethereum Serenity milestone [236], which makes Ethereum more IoT-friendly. On the other hand, Fabric is applicative for the IoT networks with immense data. Fabric has embedded Blockchain into its client-service model, and has achieved high capacity, up to tens thousand of transactions per second. However, Fabric requires a controllable network environment and cannot be as publicly accessible as the Ethereum.

7. Future research directions

This section presents future directions in optimizing security, scalability and capacity of Blockchain for future large-scale high-capacity IoT applications. The design of Blockchain for IoT application would also adapt to the specific properties of IoT networks, such as immense scale, inherent partitioning incomplete network connectivity, non-trivial topology, non-zero propagation delay, heterogeneous data, and finite device memory.

7.1. Sharding

Sharding Blockchain [237] is a novel mechanism to enable transactions to be processed in parallel. By this means, the block generation rate of Blockchain can be significantly improved. The early sharding proposals, e.g., [199], only shard transaction processing and maintain a single public Blockchain. Ubiquitously deployed IoT networks are expected to generate huge amounts of data across large landscapes. On the other hand, the data of IoT may exhibit strong locality and heterogeneity and can be only useful to local regions. This gives an opportunity of developing sharding Blockchains in IoT environments. A *primary chain* can be designed to capture important but less frequent global events of interest across large IoT networks, whilst *secondary chains* can be designed to record frequent local events of interest only to regional networks. The two sets of Blockchains can operate at different time scales. The hash values of the *secondary chains* can be secured in the *primary chain* in a transactional fashion. Particularly, the *primary chain*, recording less frequent global events, can be synchronized at significantly lower paces, thereby reducing the capacity requirement for preserving consistency across a large network scale. The two sets of Blockchains need to be interconnected to guarantee the integrity of all records, both globally and locally. With an emphasis on implementation, some initial research activities have been reported in [238].

7.2. Side chain

Apart from the ubiquity of IoT networks, some IoT devices can have the capability of traveling over large distances, such as those installed on aircrafts, intercontinental trains and ships [239]. The integrity of the data these IoT devices can generate, such as the erosion of aircraft components, is equally, if not more, important to those generated by static IoT devices. However, the data of nomadic IoT devices can be mined in blocks while the devices are away from home networks or network partitions, and embedded in different Blockchains. The migration and integration itself is a form of tampering. The migration of the blocks or segments involving the blocks' back to the home networks, is important to maintain consistent records of nomadic devices, but is challenging due to the tamper-resistant nature of Blockchain. The side chain technology [240,192] provides a solution to transferring assets between multiple Blockchains. With the side chain technology, the tokens can be transferred among different Blockchains in a decentralized way. The asset transfer process is similar to the currency exchange [240]. However, more challenges associated with side chains to be addressed include the proliferation of the chains in home networks and the implantation of the chains into the main chain.

7.3. IoT-specific consensus

Specifically designed consensus protocols for various requirements would be important to benefit IoT-Blockchain applications which are data-centric. The consensus protocol can be designed to reach data consensus by validating transaction data instead of the syntax of the transactions only. Note that sensor observations are highly correlated

in the space domain, due to high density in the network topology. Furthermore, the nature of the physical phenomenon constitutes the temporal correlation between consecutive observations of a sensor node. Spatial and temporal correlations, along with the collaborative nature of IoT, raise potentials to develop content-oriented consensus protocol [241]. The correctness of sensory data can be cross-validated with sensory data from its neighbors and historical data [242].

7.4. Simplified payment verification (SPV)

The task of block mining can be too heavy, and the size of Blockchain data can be too large, to be implanted in IoT devices. A simplified payment verification (SPV) technology [68] makes it possible to verify transactions without running block mining task and storing all historical blocks. The Blockchain nodes powered by SPV only need a small amount of resources and can be deployed on IoT devices. In SPV, a node only needs to keep the chained block headers and a Merkle branch linking to the transaction to the verified. Although the SPV node cannot validate the transaction by itself, it can check whether the Blockchain network has accepted the transaction by comparing the Merkle branch linking to the transaction. For example, the Ethereum SPV nodes have been deployed on smart bicycles [143]. Light node is a SPV implement in Ethereum [118]. The light nodes need to fetch Blockchain data from the nodes owing all the blocks, e.g., the Light Ethereum Subprotocol (LES) server in Ethereum [118].

7.5. Editable Blockchain

The storage of IoT devices can be very limited for the explosively growing size of a Blockchain ledger, as a huge number of IoT devices keep recording a large number of events in the long term. Even in the case of Bitcoin recording financial data, its total size has grown up to 149 GB by December 2017 since the genesis block in 2009 [243]. However, the data of some IoT applications will be meaningless after a constant duration. For example, the record of food is meaningless after the food has been consumed. Hence, such data can be deleted from the Blockchain to decrease the Blockchain storage. Also, fraud actions and records on IoT Blockchains raise demand for editable Blockchain technology without breaking the trust of the stored data. Editable Blockchain enables delete or modify some blocks when satisfying specific conditions. As the "editability" is somewhat contrary to the inherent "immutability" of Blockchain, the editable Blockchain is required to guarantee secure conditions and records for any edit actions. Currently, editable Blockchains have been designed with cryptographic algorithms, such as variations of the chameleon hash function [244].

8. Conclusion

This paper surveyed the use of Blockchain to resolve the myriad of data security concerns in IoT. The impact of massive IoT devices, limited computing power, low communication bandwidth and error-prone radio links on the performance of Blockchain was studied. The state-of-the-art Blockchain technologies were analyzed in detail, followed by comparison of the technologies in terms of applicability to the IoT scenarios. Research directions were pointed out to improve capacity, security and scalability of Blockchains for future effective integration of Blockchain and IoT technologies.

Acknowledgment

This work was supported, in part, by Ultimo Digital Technologies Pty Ltd under UCOT program, and also supported by National Natural Science Foundation of China (NSFC) under Grant NO. 61471129, and National Key R&D Program of China under Grant NO. 2017YFB0802703.

References

[1] M. Katagi, S. Moriai, Lightweight cryptography for the Internet of Things.
[2] B. Fabian, O. Günther, Security challenges of the EPCglobal network, Commun. ACM 52 (7) (2009) 121–125.
[3] K. Rose, S. Eldridge, L. Chapin, The Internet of Things: An overview, The Internet Society, 2015, pp. 1–50.
[4] R.H. Weber, Internet of Things – New security and privacy challenges, Comput. Law Secur. Review 26 (1) (2010) 23–30, http://dx.doi.org/10.1016/j.clsr.2009.11.008.
[5] Y. Liu, B. Dong, B. Guo, J. Yang, W. Peng, Combination of cloud computing and Internet of Things (IOT) in medical monitoring systems, Int. J. Hybrid Inform. Technol. 8 (12) (2015) 367–376.
[6] H.F. Atlam, A. Alenezi, A. Alharthi, R.J. Walters, G.B. Wills, Integration of cloud computing with Internet of Things: Challenges and open issues, in: 2017 Proc. IEEE Int. Conf. Internet of Things (iThings) and IEEE Green Comput. Commun. (GreenCom) and IEEE Cyber, Physical Social Comput. (CPSCom) and IEEE Smart Data (SmartData), pp. 670–675, http://dx.doi.org/10.1109/iThings-GreenCom-CPSCom-SmartData.2017.105.
[7] X. Lyu, W. Ni, H. Tian, R.P. Liu, X. Wang, G.B. Giannakis, A. Paulraj, Optimal schedule of mobile edge computing for Internet of Things using partial information, IEEE J. Sel. Areas Commun. 35 (11) (2017) 2606–2615.
[8] G. Booth, A. Soknacki, A. Somayaji, Cloud security: Attacks and current defenses, in: Pro. 8th Annu. Symp. Informat. Assurance, ASIA13, Citeseer, 2013, pp. 4–5.
[9] N. Chidambaram, P. Raj, K. Thenmozhi, R. Amirtharajan, Enhancing the security of customer data in cloud environments using a novel digital fingerprinting technique, Int. J. Digit. Multimed. Broadcast. 2016 (2016) 1.
[10] N. Kshetri, Can Blockchain strengthen the Internet of Things? IT Prof. 19 (4) (2017) 68–72.
[11] Z. Zheng, S. Xie, H.-N. Dai, H. Wang, Blockchain challenges and opportunities: A survey, Int. J. Web Grid Services.
[12] K. Christidis, M. Devetsikiotis, Blockchains and smart contracts for the Internet of Things, IEEE Access 4 (2016) 2292–2303.
[13] B. Betts, Blockchain and the promise of cooperative cloud storage, URL http://www.computerweekly.com/feature/Blockchain-and-the-promise-of-cooperative-cloud-storage, 2017-05-22.
[14] A. Dorri, S.S. Kanhere, R. Jurdak, Blockchain in internet of things: Challenges and solutions, 2016, arXiv:1608.05187.
[15] X. Zha, X. Wang, W. Ni, R.P. Liu, Y.J. Guo, X. Niu, K. Zheng, Blockchain for IoT: The tradeoff between consistency and capacity, Chin. J. Internet of Things 1 (1) (2017) 21.
[16] P.K. Sharma, S.Y. Moon, J.H. Park, Block-VN: A distributed blockchain based vehicular network architecture in smart city, J. Inf. Process. Syst. 13 (1) (2017) 184–195.
[17] B. Leiding, P. Memarmoshrefi, D. Hogrefe, Self-managed and blockchain-based vehicular ad-hoc networks, in: 2016 Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.: Adjunct, New York, NY, USA, pp. 137–140.
[18] M. Castro, B. Liskov, Practical Byzantine fault tolerance, in: Proc. 3rd Symp. Operating Syst. Des. Implementation, OSDI'99, New Orleans, USA, 1999.
[19] M.C.K. Khalilov, A. Levi, A survey on anonymity and privacy in bitcoin-like digital cash systems, IEEE Commun. Surv. Tutor. 20 (3) (2018) 2543–2585.
[20] X. Zha, K. Zheng, D. Zhang, Anti-pollution source location privacy preserving scheme in wireless sensor networks, in: 2016 13th Annual IEEE International Conference on Sensing, Communication, and Networking, SECON, 2016, pp. 1–8, URL http:/dx.doi.org/10.1109/SAHCN.2016.7732970.
[21] G. Zyskind, O. Nathan, A. Pentland, Enigma: Decentralized computation platform with guaranteed privacy, 2015, arXiv preprint URL arXiv:1506.03471.
[22] G. Zyskind, O. Nathan, A.S. Pentland, Decentralizing privacy: Using blockchain to protect personal data, in: 2015 Proc. IEEE Secur. and Privacy Workshops, SPW'15, IEEE, pp. 180–184.
[23] I.O.T.A, IOTA, 2017, URL https://www.iotatoken.com.
[24] B. Panikkar, S. Nair, P. Brody, V. Pureswaran, ADEPT: An IoT Practitioner Perspective, IBM, 2014, URL http://static1.squarespace.com/static/55f73743e4b051cfcc0b02cf/55f73e5ee4b09b2bff5b2eca/55f73e72e4b09b2bff5b3267/1442266738638/IBM-ADEPT-Practictioner-Perspective-Pre-Publication-Draft-7-Jan-2015.pdf?format=original.
[25] P. Brody, V. Pureswaran, Device Democracy: Saving the Future of the Internet of Things, IBM, 2014, URL https://public.dhe.ibm.com/common/ssi/ecm/gb/en/gbe03620usen/global-business-services-global-business-services-gb-executive-brief-gbe03620usen-20171002.pdf.
[26] I.B.M., Watson Internet of Things, 2017, URL https://www.ibm.com/internet-of-things/.
[27] C. O'Connor, What Blockchain Means for You, and the Internet of Things, 2017, URL https://www.ibm.com/blogs/internet-of-things/watson-iot-blockchain/.
[28] J. Fedak, How can Blockchain Improve Cloud Computing, 2016, URL https://medium.com/iex-ec/how-blockchain-can-improve-cloud-computing-1ca24c270f4f.
[29] M. Chen, S. Mao, Y. Liu, Big data: A survey, Springer Mobile Netw. Appl. 19 (2) (2014) 171–209.

[30] F. Tschorsch, B. Scheuermann, Bitcoin and beyond: A technical survey on decentralized digital currencies, IEEE Commun. Surv. Tutor. 18 (3) (2016) 2084–2123, http://dx.doi.org/10.1109/COMST.2016.2535718.

[31] M. Swan, Blockchain, O'Reilly Media, 2015.

[32] A. Lewis, A Gentle Introduction to Blockchain Technology, 2015, URL https://bitsonblocks.net/2015/09/09/a-gentle-introduction-to-blockchain-technology/.

[33] M. Crosby, P. Pattanayak, et al., BlockChain technology: Beyond bitcoin, Appl. Innov. 2 (2016) 6–10.

[34] M. Pticek, V. Podobnik, G. Jezic, Beyond the internet of things: The social networking of machines, Int. J. Distrib. Sens. Netw. 12 (6) (2016) 8178417.

[35] C. Perera, A. Zaslavsky, P. Christen, D. Georgakopoulos, Context aware computing for the internet of things: A survey, IEEE Commun. Surv. Tutor. 16 (1) (2014) 414–454.

[36] L.D. Xu, W. He, S. Li, Internet of things in industries: A survey, IEEE Trans. Ind. Inf. 10 (4) (2014) 2233–2243.

[37] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, Internet of things: A survey on enabling technologies, protocols, and applications, IEEE Commun. Surv. Tutor. 17 (4) (2015) 2347–2376.

[38] L. Atzori, A. Iera, G. Morabito, The Internet of Things: A survey, Comput. Netw. 54 (15) (2010) 2787–2805.

[39] X. Zha, W. Ni, X. Wang, R.P. Liu, Y.J. Guo, X. Niu, K. Zheng, The impact of link duration on the integrity of distributed mobile networks, IEEE Trans. Inf. Forensics Secur. 13 (9) (2018) 2240–2255.

[40] Q. Cui, Y. Wang, K. Chen, W. Ni, I. Lin, X. Tao, P. Zhang, Big data analytics and network calculus enabling intelligent management of autonomous vehicles in a smart city, IEEE Internet of Things J. (2018) 1–1.

[41] P. Sethi, S.R. Sarangi, Internet of Things: Architectures, protocols, and applications, J. Electr. Comput. Eng. (2017).

[42] J.S. Lee, Y.W. Su, C.C. Shen, A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi, in: Proc. 33rd Annu. Conf. the IEEE Ind. Elect. Soc., IECON'07, 2007, pp. 46–51.

[43] R. Ratasuk, B. Vejlgaard, N. Mangalvedhe, A. Ghosh, NB-IoT system for M2M communication, in: 2016 Proc. IEEE Wireless Commun. and Netw. Conf. Workshops, WCNCW'16, pp. 1–5.

[44] Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025 (in billions), 2016, URL https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/.

[45] O. Vermesan, P. Friess, P. Guillemin, et al., Internet of Things strategic research roadmap, Internet of Things-Global Technol. Soc. Trends 1 (2011) (2011) 9–52.

[46] C. Ren, X. Lyu, W. Ni, H. Tian, R.P. Liu, Distributed online learning of fog computing under non-uniform device cardinality, IEEE Internet of Things J. (2018) 1–1.

[47] C.-W. Tsai, C.-F. Lai, M.-C. Chiang, L.T. Yang, et al., Data mining for Internet of Things: A survey, IEEE Commun. Surv. Tutor. 16 (1) (2014) 77–97.

[48] X. Zha, W. Ni, K. Zheng, R.P. Liu, X. Niu, Collaborative authentication in decentralized dense mobile networks with key predistribution, IEEE Trans. Inf. Forensics Secur. 12 (10) (2017) 2261–2275.

[49] I. Makhdoom, M. Abolhasan, J. Lipman, R.P. Liu, W. Ni, Anatomy of threats to the Internet of Things, IEEE Commun. Surv. Tutor. (2018) 1–1.

[50] G. Gan, Z. Lu, J. Jiang, Internet of Things security analysis, in: 2011 Int. Conf. Internet Technol. and Appl, 2011, pp. 1–4.

[51] E. Alsaadi, A. Tubaishat, Internet of Things: Features, challenges, and vulnerabilities, Int. J. Adv. Comput. Sci. Inf. Technol. 4 (1) (2015) 1–13.

[52] X. Liu, M. Zhao, S. Li, F. Zhang, W. Trappe, A security framework for the Internet of Things in the future internet architecture, Future Internet 9 (3) (2017).

[53] R. Roman, P. Najera, J. Lopez, Securing the Internet of Things, Comput. 44 (9) (2011) 51–58.

[54] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, W. Zhao, A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications, IEEE Internet of Things J. 4 (5) (2017) 1125–1142.

[55] R. Xu, R. Wang, Z. Guan, L. Wu, J. Wu, X. Du, Achieving efficient detection against false data injection attacks in smart grid, IEEE Access 5 (2017) 13787–13798.

[56] K. Mehta, D. Liu, M. Wright, Protecting location privacy in sensor networks against a global eavesdropper, IEEE Trans. Mobile Comput. 11 (2) (2012) 320–336.

[57] N. Namvar, W. Saad, N. Bahadori, B. Kelley, Jamming in the Internet of Things: A game-theoretic perspective, in: 2016 IEEE Global Commun. Conf., GLOBECOM, 2016, pp. 1–6.

[58] K. Zhang, X. Liang, R. Lu, X. Shen, Sybil attacks and their defenses in the Internet of Things, IEEE Internet of Things J. 1 (5) (2014) 372–383.

[59] A. Mosenia, N.K. Jha, A comprehensive study of security of Internet-of-Things, IEEE Trans. Emerg. Top. Comput. 5 (4) (2017) 586–602.

[60] J. Perry, Anatomy of an IoT malware attack, 2017, URL https://www.ibm.com/developerworks/library/iot-anatomy-iot-malware-attack/.

[61] X. Wang, W. Ni, K. Zheng, R.P. Liu, X. Niu, Virus propagation modeling and convergence analysis in large-scale networks, IEEE Trans. Inf. Forensics Secur. 11 (10) (2016) 2241–2254.

[62] X. Wang, K. Zheng, X. Niu, B. Wu, C. Wu, Detection of command and control in advanced persistent threat based on independent access, in: Proc. 2016 IEEE Int. Conf. Commun., ICC' 16, 2016, pp. 1–6.

[63] J.E. Boritz, Is practitioners' views on core concepts of information integrity, Int. J. Account. Inf. Syst. 6 (4) (2005) 260–279.

[64] A. Fongen, Identity management and integrity protection in the Internet of Things, in: Proc. 3rd Int. Conf. Emerg. Secur. Technol., EST'12, 2012, pp. 111–114.

[65] H.C. Pöhls, JSON sensor signatures (JSS): End-to-end integrity protection from constrained device to IoT application, in: Proc. 9th Int. Conf. Innovative Mobile Internet Serv. in Ubiquitous Comput., IMIS'15, 2015, pp. 306–312.

[66] C. Wang, Q. Wang, K. Ren, W. Lou, Privacy-preserving public auditing for data storage security in cloud computing, in: Proc. 29th Annu. IEEE Int. Conf. Comput. Commun., INFOCOM'10, 2010, pp. 1–9.

[67] S. Davidson, P. De Filippi, J. Potts, Economics of Blockchain, 2016, URL https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2744751.

[68] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, 2008, URL https://bitcoin.org/bitcoin.pdf.

[69] Bitcoin, 2017-05-20, URL https://bitcoin.org/en/.

[70] A. Miller, J. Litton, A. Pachulski, et al., Discovering Bitcoin's public topology and influential nodes, 2015, URL https://allquantor.at/blockchainbib/pdf/miller2015topology.pdf.

[71] How do Bitcoin Transactions Work? 2015-03-20, URL http://www.coindesk.com/information/how-do-bitcoin-transactions-work/.

[72] O. Wyman, Blockchain in Capital Markets, 2016, URL http://www.oliverwyman.com/content/dam/oliver-wyman/global/en/2016/feb/BlockChain-In-Capital-Markets.pdf.

[73] Bitcoin developer guide, 2017, URL https://bitcoin.org/en/developer-guide.

[74] R.C. Merkle, Protocols for public key cryptosystems, in: Proc. 1st IEEE Symp. Secur. Privacy, SP' 80, 1980, pp. 122–122, http://dx.dio.org/10.1109/SP.1980.10006.

[75] D. Ghosh, How the byzantine general sacked the castle: A look into blockchain, 2016-04-06, URL https://medium.com/@DebrajG/how-the-byzantine-general-sacked-the-castle-a-look-into-blockchain-370fe637502c.

[76] L. Lamport, R. Shostak, M. Pease, The Byzantine generals problem, ACM Trans. Program. Lang. Syst. 4 (3) (1982) 382–401.

[77] S. Poledna, Fault-Tolerant Real-Time Systems: The Problem of Replica Determinism, vol. 345, Springer Sci. & Business Media, 2007.

[78] D. Dolev, H.R. Strong, Authenticated algorithms for Byzantine agreement, SIAM J. Comput. 12 (4) (1983) 656–666.

[79] F. Cristian, H. Aghili, R. Strong, D. Dolev, Atomic broadcast: From simple message diffusion to Byzantine agreement, Inform. and Comput. 118 (1) (1995) 158–179.

[80] K.J. Perry, S. Toueg, Distributed agreement in the presence of processor and communication faults, IEEE Trans. Softw. Eng. SE-12 (3) (1986) 477–482.

[81] R.D. Schlichting, F.B. Schneider, Fail-stop processors: An approach to designing fault-tolerant computing systems, ACM Trans. Comput. Syst. 1 (3) (1983) 222–238.

[82] L. Lamport, The part-time parliament, ACM Trans. Comput. Syst. 16 (2) (1998) 133–169.

[83] A. Mostéfaoui, S. Rajsbaum, M. Raynal, A Versatile and Modular Consensus Protocol, IRISA, 2001.

[84] B. Charron-Bost, A. Schiper, The heard-of model: Computing in distributed systems with benign faults, Distrib. Comput. 22 (1) (2009) 49–71.

[85] M. Pease, R. Shostak, L. Lamport, Reaching agreement in the presence of faults, J. ACM 27 (2) (1980) 228–234.

[86] M. Fitzi, U. Maurer, Efficient byzantine agreement secure against general adversaries, Distrib. Comput. (1998) 134–148.

[87] P.J. Marandi, M. Primi, F. Pedone, High performance state-machine replication, in: Proc. 41st Int. Conf. Depend. Syst. Netw., DSN' 11, IEEE, 2011, pp. 454–465.

[88] S. King, S. Nadal, Ppcoin: Peer-To-Peer Crypto-Currency with Proof-Of-Stake, 2012, URL https://peercoin.net/assets/paper/peercoin-paper.pdf.

[89] E.K. Kogias, P. Jovanovic, N. Gailly, I. Khoffi, L. Gasser, B. Ford, Enhancing bitcoin security and performance with strong consistency via collective signing, in: Proc. 25th USENIX Secur. Symp., USENIX Secur. 16, USENIX Association, 2016, pp. 279–296.

[90] V. Buterin, On public and private blockchains, Ethereum Blog (2015) URL https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains/.

[91] M. Conti, C. Lal, S. Ruj, et al., A survey on security and privacy issues of bitcoin, 2017, arXiv preprint URL arXiv:1706.00916.

[92] G.O. Karame, E. Androulaki, S. Capkun, Double-spending fast payments in bitcoin, in: Proc. 19th ACM Conf. Comput. Commun. Secur., CCS'12, ACM, 2012, pp. 906–917.

[93] H. Finney, Best Practice for Fast Transaction Acceptance-How High Is the Risk, 2011, URL https://bitcointalk.org/index.php?topic=3441.0.

[94] M. Bastiaan, Preventing the 51%-Attack: A Stochastic Analysis of Two Phase Proof of Work in Bitcoin, 2015, URL http://referaat.cs.utwente.nl/conference/22/paper/7473/preventingthe-51-attack-a-stochastic-analysis-of-two-phase-proof-of-work-in-bitcoin.pdf.

[95] I. Eyal, E.G. Sirer, Majority is not enough: Bitcoin mining is vulnerable, in: 2014 Proc. Int. Conf. Financial Cryptography Data Secur., FC '14, Springer, pp. 436–454.

[96] E. Heilman, A. Kendler, A. Zohar, S. Goldberg, Eclipse attacks on bitcoin's peer-to-peer network, in: Proc. 24th USENIX Secur. Symp., USENIX Secur.' 15, Washington, D.C., 2015, pp. 129–144.

[97] K. Nayak, S. Kumar, A. Miller, E. Shi, Stubborn mining: Generalizing selfish mining and combining with an eclipse attack, in: 2016 Proc. IEEE Eur. Symp. Secur. Privacy, EuroSP' 16, pp. 305–320.

[98] E.H. Yuval Marcus, S. Goldberg, Low-Resource Eclipse Attacks on Ethereum's Peer-To-Peer Network, 2018, p. 15, URL https://www.cs.bu.edu/~goldbe/projects/eclipseEth.pdf.

[99] N. Atzei, M. Bartoletti, T. Cimoli, A survey of attacks on ethereum smart contracts (SoK), in: Proc. 6th Int. Conf. Principles Secur. Trust, 2017, pp. 164–186.

[100] S.S. Team, Billions of Tokens Theft Case cause by ETH Ecological Defects, 2018, URL https://paper.tuisec.win/detail/eb44c15d3627fe2.

[101] X. Wang, X. Zha, G. Yu, W. Ni, R.P. Liu, Y.J. Guo, X. Niu, K. Zheng, Attack and defence of ethereum remote APIs, in: 2018 Proc. IEEE Globecom Workshops, GC Wkshps'18, 2018.

[102] M. Vasek, M. Thornton, T. Moore, Empirical analysis of denial-of-service attacks in the bitcoin ecosystem, in: 2014 Proc. Int. Conf. Financial Cryptography Data Secur., FC '14, Springer, pp. 57–71.

[103] D. Meegan, Ethereum Continues to Suffer From DDoS Attacks, 2016, URL https://www.ethnews.com/ethereum-continues-to-suffer-from-ddos-attacks.

[104] S. Verbücheln, How perfect offline wallets can still leak bitcoin private keys, 2015, arXiv preprint URL arXiv:1501.00447.

[105] M. Smache, N.E. Mrabet, J.J. Gilquijano, A. Tria, E. Riou, C. Gregory, Modeling a node capture attack in a secure wireless sensor networks, in: 2016 Proc IEEE 3rd World Forum on Internet of Things, WF-IoT'16, 2016, pp. 188–193.

[106] Y. Yang, X. Liu, R.H. Deng, Lightweight break-glass access control system for healthcare Internet-of-Things, IEEE Trans. Ind. Inf. 14 (8) (2017) 3610–3617.

[107] W.L. Chin, W. Li, H.H. Chen, Energy big data security threats in IoT-based smart grid communications, IEEE Commun. Mag. 55 (10) (2017) 70–75.

[108] R. Langner, Stuxnet: Dissecting a cyberwarfare weapon, IEEE Secur. Privacy 9 (3) (2011) 49–51.

[109] K. Korpela, J. Hallikas, T. Dahlberg, Digital supply chain transformation toward Blockchain integration, in: Proc. 50th Hawaii Int. Conf. Syst. Sci, 2017.

[110] K. Biswas, V. Muthukkumarasamy, Securing smart cities using blockchain technology, in: Proc. 18th IEEE Int. Conf. High Performance Comput. Commun.; 14th IEEE Int. Conf. Smart City; 2nd IEEE Int. Conf. Data Sci. Syst., HPCC/SmartCity/DSS'16, 2016, pp. 1392–1393.

[111] F. Zhang, E. Cecchetti, K. Croman, A. Juels, E. Shi, Town crier: an authenticated data feed for smart contracts, in: Proc. 23rd ACM Conf. Comput. Commun. Secur., CCS' 16, ACM, New York, NY, USA, 2016, pp. 270–282.

[112] Oraclize, 2018, URL http://www.oraclize.it.

[113] P. Golle, D. Greene, J. Staddon, Detecting and correcting malicious data in vanets, in: Proc.1st ACM Int. Workshop Vehicular Ad Hoc Netw., VANET '04, ACM, New York, NY, USA, 2004, pp. 29–37.

[114] G. Wood, Ethereum: A secure decentralised generalised transaction ledger, Ethereum Project Yellow Paper, 2014, URL https://ethereum.github.io/yellowpaper/paper.pdf.

[115] Wallet knowledge base, 2018, URL https://iotasupport.com/walletknowledgebase.shtml.

[116] D. Schiener, The IOTA Lightwallet, 2017, URL https://blog.iota.org/the-iota-lightwallet-c8c738d7192b.

[117] Why 0chain, not IOTA, is ideal for IOT apps, Dec, 2017, URL https://medium.com/0chain/why-0chain-not-iota-is-ideal-for-iot-web-enterprise-apps-bdd1154d148f.

[118] J. McKinney, Light Client Protocol, 2017, URL https://github.com/ethereum/wiki/wiki/Light-client-protocol.

[119] I.O.T.A, Light vs. Full Node, URL https://iota.readme.io/v1.2.0/docs/light-vs-full-node.

[120] Fabric, 2017, URL https://github.com/hyperledger/fabric.

[121] A. Dorri, S.S. Kanhere, R. Jurdak, Towards an optimized blockchain for IoT, in: Proc. 2rd Int. Conf. Internet-of-Things Design Implementation, ACM, 2017, pp. 173–178.

[122] Y. Rahulamathavan, R.C.W. Phan, S. Misra, M. Rajarajan, Privacy-preserving Blockchain based IoT Ecosystem using attribute-based encryption, in: 2017 Proc. IEEE Int. Conf. Advanced Netw. Telecommun. Syst, Odisha, India.

[123] A. Dorri, S.S. Kanhere, R. Jurdak, P. Gauravaram, Blockchain for IoT security and privacy: The case study of a smart home, in: 2017 Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops, PerCom Workshops, pp. 618–623.

[124] F. Al-Doghman, Z. Chaczko, J. Jiang, A review of aggregation algorithms for the internet of things, in: Proc. 25th Int. Conf. Syst. Eng., ICSEng'17, 2017, pp. 480–487.

[125] Elements project, 2017, URL https://z.cash/support/zig.html.

[126] Protocol rules, 2016, URL https://en.bitcoin.it/wiki/Protocol_rules.

[127] Blockchain, 2017, URL https://blockchain.info.

[128] Ethereum mining hardware, 2017, URL https://www.buybitcoinworldwide.com/ethereum/mining-hardware/.

[129] Raspberry pi, 2017, URL https://www.raspberrypi.org.

[130] Mining bitcoin only with raspberry pi, 2016, URL https://bitcointalk.org/index.php?topic=1535364.0.

[131] Etherscan, 2017, URL https://etherscan.io.

[132] Y.Y. Goland, Going Off Chain for Storage, 2017, URL http://www.goland.org/off_chain_storage_and_the_enterprise/.

[133] I.H. Hou, P.R. Kumar, Real-time communication over unreliable wireless links: A theory and its applications, IEEE Wirel. Commun. 19 (1) (2012) 48–59.

[134] M. Lauridsen, I.Z. Kovacs, P. Mogensen, M. Sorensen, S. Holst, Coverage and capacity analysis of LTE-M and NB-IoT in a rural area, in: Proc. 84th IEEE Vehicular Technol. Conf., VTC-Fall'16, 2016, pp. 1–5.

[135] J.M. Liang, J.J. Chen, H.H. Cheng, Y.C. Tseng, An energy-efficient sleep scheduling with QoS consideration in 3GPP LTE-advanced networks for Internet of Things, IEEE J. Emerg. Sec. Top. Circuits Syst. 3 (1) (2013) 13–22.

[136] B. Westermann, D. Gligoroski, S. Knapskog, Comparison of the power consumption of the 2nd round SHA-3 candidates, in: in: M. Gusev, P. Mitrevski (Eds.) Proc. 2nd Int. Conf. ICT Innovations, Berlin, Heidelberg, 2010, pp. 102–113.

[137] J. Chen, S.H.G. Chan, S.C. Liew, Mixed-mode WLAN: The integration of ad hoc mode with wireless LAN infrastructure, in: 2003 Proc. IEEE Global Telecommun. Conf., GLOBECOM '03, Vol. 1, pp. 231–235.

[138] M. Zorzi, A. Gluhak, S. Lange, A. Bassi, From today's INTRAnet of Things to a future INTERnet of Things: A wireless- and mobility-related view, IEEE Wirel. Commun. 17 (6) (2010) 44–51, http://dx.doi.org/10.1109/MWC.2010.5675777.

[139] K.H. Wang, B. Li, Group mobility and partition prediction in wireless ad-hoc networks, in: 2002 Proc. IEEE Int. Conf. Commun., ICC' 02, Vol. 2, pp. 1017–1021.

[140] M.M. Rathore, A. Ahmad, A. Paul, S. Rho, Urban planning and building smart cities based on the internet of things using big data analytics, Comput. Netw. 101 (Supplement C) (2016) 63–80.

[141] S. Huh, S. Cho, S. Kim, Managing IoT devices using Blockchain platform, in: Proc. 19th Int. Conf. Advanced Commun. Technol., ICACT' 17, 2017, pp. 464–467.

[142] V. Buterin, Ethereum: A next-generation smart contract and decentralized application platform, 2014, URL https://github.com/ethereum/wiki/wiki/White-Paper.

[143] C. Jaffe, C. Mata, S. Kamvar, Motivating urban cycling through a blockchain-based financial incentives system, in: 2017 Proc. ACM Int. Joint Conf. Pervasive and Ubiquitous Comput. and 2017 Proc ACM Int. Symp. on Wearable Computers, UbiComp '17, ACM, New York, NY, USA, 2017, pp. 81–84.

[144] S. Raza, D. Trabalza, T. Voigt, 6LoWPAN Compressed DTLS for CoAP, in: Proc. 8th IEEE Int. Conf. Distributed Computing in Sensor Syst., DCOSS'12, 2012, pp. 287–289.

[145] D. Chen, G. Chang, D. Sun, J. Li, J. Jia, X. Wang, TRM-IoT: A trust management model based on fuzzy reputation for Internet of Things, Comput. Sci. Inf. Syst. 8 (4) (2011) 1207–1228.

[146] M. Mihaylov, S. Jurado, N. Avellana, K.V. Moffaert, I.M. de Abril, A. Nowé, Nrgcoin: virtual currency for trading of renewable energy in smart grids, in: Proc. 11th Int. Conf. Eur. Energy Market, EEM14, 2014, pp. 1–6.

[147] Q. He, D. Wu, P. Khosla, SORI: A secure and objective reputation-based incentive scheme for ad-hoc networks, in: 2004 IEEE Wireless Commun. and Networking Conf., IEEE Cat. No.04TH8733, vol. 2, 2004, pp. 825–830.

[148] S. Raval, Decentralized Applications: Harnessing Bitcoin's Blockchain Technology, O'Reilly Media, Inc., 2016.

[149] N. Szabo, Smart contracts: Building blocks for digital markets, EXTROPY: J. Transhumanist Thought (1996).

[150] Universa, How Smart Contracts Will Kill Bureaucracy, 2017, URL https://medium.com/universablockchain/how-smart-contracts-will-kill-bureaucracy-c22a48e2e60.

[151] J. Dai, M.A. Vasarhelyi, Toward blockchain-based accounting and assurance, J. Inf. Syst. 31 (3) (2017) 5–21, http://dx.doi.org/10.2308/isys-51804.

[152] F. Al Khalil, M. Ceci, L. O'Brien, T. Butler, A Solution for the Problems of Translation and Transparency in Smart Contracts, 2017, URL http://www.grctc.com/wp-content/uploads/2017/06/GRCTC-Smart-Contracts-White-Paper-2017.pdf.

[153] I. Kaiser, Yes, Bitcoin Can Do Smart Contracts and Particl Demonstrates How, 2017, URL https://bitcoinmagazine.com/articles/yes-bitcoin-can-do-smart-contracts-and-particl-demonstrates-how/.

[154] A. Hertig, How Do Ethereum Smart Contracts Work? 2018, URL https://www.coindesk.com/information/ethereum-smart-contracts-work/.

[155] O. Asor, ABOUT τ-CHAIN, Feb, 2015, URL http://tauchain.org/tauchain.pdf.

[156] Nxt, 2018, URL https://nxtplatform.org/.

[157] RSK, 2018, URL https://www.rsk.co/.

[158] This is how smart contracts and ethereum work, 2017, URL https://medium.com/startup-grind/gentle-intro-to-blockchain-and-smart-contracts-part-2-30a6c9a40946.

[159] Blockchain-oracles, 2018, URL https://blockchainhub.net/blockchain-oracles.

[160] A. Boudguiga, N. Bouzerna, L. Granboulan, et al., Towards better availability and accountability for IoT updates by means of a Blockchain, in: Proc. 2017 IEEE Eur. Symp. Secur. Privacy Workshops, EuroSPW, Paris, France, 2017.

[161] P. Maymounkov, D. Mazières, Kademlia: A peer-to-peer information system based on the XOR metric, in: P. Druschel, F. Kaashoek, A. Rowstron (Eds.), Peer-to-Peer Syst.: 1st Int. Workshop, IPTPS' 02, Springer Berlin Heidelberg, Berlin, Heidelberg, 2002, pp. 53–65.

[162] F. Reid, M. Harrigan, An analysis of anonymity in the bitcoin system, in: Y. Altshuler, Y. Elovici, A.B. Cremers, N. Aharony, A. Pentland (Eds.), Secur. and Privacy in Social Networks, Springer New York, New York, NY, 2013, pp. 197–223, http://dx.doi.org/10.1007/978-1-4614-4139-7_10.

[163] E. Androulaki, G.O. Karame, M. Roeschlin, T. Scherer, S. Capkun, Evaluating user privacy in bitcoin, in: A.R. Sadeghi (Ed.), Financial Cryptography and Data Secur.: 17th Int. Conf., FC 2013, Okinawa, Japan, April 1–5, 2013, Revised Selected Papers, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 34–51.

[164] D. Ron, A. Shamir, Quantitative Analysis of the Full Bitcoin Transaction Graph, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 6–24.

[165] D. Ron, A. Shamir, Quantitative analysis of the full bitcoin transaction graph, in: 2013 Proc. Int. Conf. Financial Cryptography Data Secur., FC '13, Springer, pp. 6–24.

[166] S. Meiklejohn, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G.M. Voelker, S. Savage, A fistful of bitcoins: Characterizing payments among men with no names, in: 2013 Proc. Conf. Internet measurement, ACM, pp. 127–140.

[167] A. Biryukov, D. Khovratovich, I. Pustogarov, Deanonymisation of clients in Bitcoin P2P network, in: Proc. 21st ACM Conf. Comput. Commun. Secur., CCS' 14, ACM, 2014, pp. 15–29.

[168] N. van Saberhagen, Cryptonote v 2. 0, 2013.

[169] P. Koshy, D. Koshy, P. McDaniel, An Analysis of Anonymity in Bitcoin Using P2P Network Traffic, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[170] bitcoinwiki, Mixing Service, 2017, URL https://en.bitcoin.it/wiki/Mixing_service.

[171] A. Kosba, A. Miller, E. Shi, Z. Wen, C. Papamanthou, Hawk: The Blockchain model of cryptography and privacy-preserving smart contracts, in: Proc. 37th IEEE Symp. Secur. Privacy, SP' 16, IEEE, 2016, pp. 839–858.

[172] U. Feige, A. Fiat, A. Shamir, Zero-knowledge proofs of identity, J. Cryptology 1 (2) (1988) 77–94.

[173] C. Rackoff, D.R. Simon, Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack, Springer Berlin Heidelberg, Berlin, Heidelberg, 1992, pp. 433–444, http://dx.doi.org/10.1007/3-540-46766-1_35.

[174] I. Miers, C. Garman, M. Green, A. Rubin, Zerocoin: Anonymous distributed e-cash from Bitcoin, in: Proc. 34th IEEE Symp. Secur. Privacy, SP' 13, IEEE, 2013, pp. 397–411.

[175] E.B. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, M. Virza, Zerocash: Decentralized anonymous payments from Bitcoin, in: Proc. 35th IEEE Symp. Secur. Privacy, SP' 14, IEEE, 2014, pp. 459–474.

[176] G.G. Dagher, B. Bünz, J. Bonneau, J. Clark, D. Boneh, Provisions: Privacy-preserving proofs of solvency for bitcoin exchanges, in: Proc. 22nd ACM Conf. Comput. Commun. Secur., CCS' 15, ACM, 2015, pp. 720–731.

[177] J.K. Liu, V.K. Wei, D.S. Wong, Linkable spontaneous anonymous group signature for ad hoc groups, in: ACISP, vol. 4, Springer, 2004, pp. 325–335.

[178] J.K. Liu, D.S. Wong, Linkable ring signatures: Security models and new schemes, in: 2005 Proc. Int. Conf. Computational Sci. and Its Appl, Springer, pp. 614–623.

[179] MONERO, 2017, URL https://getmonero.org.

[180] G. Maxwell, Confidential transaction, the initial investigation, URL https://elementsproject.org/elements/confidential-transactions/investigation.html.

[181] S. Noether, A. Mackenzie, et al., Ring confidential transactions, Ledger 1 (2016) 1–18.

[182] D. Demirel, J. Lancrenon, How to securely prolong the computational bindingness of pedersen commitments, IACR Cryptol. EPrint Archive 2015 (2015) 584.

[183] V. Goyal, O. Pandey, A. Sahai, B. Waters, Attribute-based encryption for fine-grained access control of encrypted data, in: Proc. 13th ACM Conf. Comput. Commun. Secur, in: CCS '06, ACM, New York, NY, USA, 2006, pp. 89–98, http://dx.doi.org/10.1145/1180405.1180418, URL http://doi.acm.org/10.1145/1180405.1180418.

[184] C. Gentry, Fully homomorphic encryption using ideal lattices, in: Proc. of the 41st Annu. ACM Symp. on Theory of Computing, in: STOC '09, ACM, New York, NY, USA, 2009, pp. 169–178, http://dx.doi.org/10.1145/1536414.1536440, URL http://doi.acm.org/10.1145/1536414.1536440.

[185] V. Daza, R.D. Pietro, I. Klimek, M. Signorini, CONNECT: CONtextual name discovery for Blockchain-based services in the IoT, in: 2017 Proc. IEEE Int. Conf. Commun., ICC'17, pp. 1–6.

[186] R. Neisse, G. Steri, I. Nai-Fovino, A blockchain-based approach for data accountability and provenance tracking, 2017, arXiv preprint URL arXiv:1706.04507.

[187] A. Ouaddah, A.A. Elkalam, A.A. Ouahman, Towards a novel privacy-preserving access control model based on blockchain technology in IoT, in: Europe and MENA Cooperation Advances in Inf. and Communication Technologies, Springer, 2017, pp. 523–533.

[188] A. Ouaddah, A. Abou Elkalam, A. Ait Ouahman, FairAccess: A new Blockchain-based access control framework for the Internet of Things, Secur. Commun. Netw. 9 (18) (2016) 5943–5964.

[189] G.W. Peters, E. Panayi, Understanding modern banking ledgers through blockchain technologies: Future of transaction processing and smart contracts on the internet of money, in: Banking Beyond Banks and Money, Springer International Publishing, Cham, 2016, pp. 239–278.

[190] G. Yu, X. Wang, X. Zha, J.A. Zhang, R.P. Liu, An optimized round-robin scheduling of speakers for peers-to-peers-based byzantine faulty tolerance, in: 2018 Proc. IEEE Globecom Workshops, GC Wkshps'18, 2018.

[191] D.W. Kravitz, J. Cooper, Securing user identity and transactions symbiotically: IoT meets Blockchain, in: 2017 Proc. Global Internet Things Summit, GIoTS'17, pp. 1–6.

[192] M. Pilkington, Blockchain technology: Principles and applications, in: Research Handbook on Digital Transformations, Edward Elgar Publishing, 2016, p. 225.

[193] M. Vukolić, The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication, in: Int. Workshop on Open Problems in Network Secur, Springer, 2015, pp. 112–125.

[194] R. Kotla, L. Alvisi, M. Dahlin, A. Clement, E. Wong, Zyzzyva: Speculative byzantine fault tolerance, in: Proc. 21st ACM SIGOPS Symp. Operating Syst. Principles, SOSP '07, Vol. 41, pp. 45–58.

[195] P.K. Sharma, S. Singh, Y.S. Jeong, J.H. Park, DistBlockNet: A Distributed Blockchains-Based Secure SDN Architecture for IoT Networks, IEEE Commun. Mag. 55 (9) (2017) 78–85.

[196] Y. Guo, C. Mate Jr., Crysto: A Scalable and Permission-less Blockchain Platform, URL https://cdecker.github.io/btcresearch/2017/guocrysto.html.

[197] D. Ongaro, J. Ousterhout, In search of an understandable consensus algorithm, in: 2014 Proc. USENIX Annu. Tech. Conf., USENIX ATC 14, Philadelphia, PA, pp. 305–319.

[198] L. Luu, V. Narayanan, K. Baweja, C. Zheng, S. Gilbert, P. Saxena, SCP: A computationally-scalable Byzantine consensus protocol for Blockchains, IACR Cryptol. EPrint Archive 2015 (2015) 1168.

[199] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, P. Saxena, A secure sharding protocol for open blockchains, in: Proc. 23rd ACM Conf. Comput. Commun. Secur., CCS' 16, ACM, 2016, pp. 17–30.

[200] T. Crain, V. Gramoli, M. Larrea, M. Raynal, (Leader/Randomization/Signature)-free Byzantine Consensus for Consortium Blockchains, 2017, arXiv preprint URL arXiv:1702.03068.

[201] J.R. Douceur, The sybil attack, in: 2002 Proc. Int. Workshop Peer-to-Peer Syst, Springer, 2002, pp. 251–260.

[202] Difficulty, 2017, URL https://en.bitcoin.it/wiki/Difficulty.

[203] D. Eastlake 3rd, T. Hansen, US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF), Tech. rep., 2011.

[204] A.S. Tanenbaum, D.J. Wetherall, Computer Networks, Pearson, 2011.

[205] M. Rosenfeld, Analysis of bitcoin pooled mining reward systems, 2011, arXiv preprint URL arXiv:1112.4980.

[206] A. Laszka, B. Johnson, J. Grossklags, When bitcoin mining pools run dry, in: 2015 Proc. Int. Conf. Financial Cryptography Data Secur., FC '15, Springer, 2015, pp. 63–77.

[207] I. Eyal, The miner's dilemma, in: Proc. 36th IEEE Symp. Secur. Privacy, SP' 15, IEEE, 2015, pp. 89–103.

[208] A. Miller, A. Kosba, J. Katz, E. Shi, Nonoutsourceable scratch-off puzzles to discourage bitcoin mining coalitions, in: Proc. 22nd ACM Conf. Comput. Commun. Secur., CCS' 15, ACM, 2015, pp. 680–691.

[209] A. Miller, A. Juels, E. Shi, B. Parno, J. Katz, Permacoin: Repurposing bitcoin work for data preservation, in: Proc. 35th IEEE Symp. Secur. Privacy, SP' 14, IEEE, 2014, pp. 475–490.

[210] N. Szabo, The Idea of Smart Contracts, 1997, URL http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/idea.html.

[211] I. Bentov, C. Lee, A. Mizrahi, M. Rosenfeld, Proof of activity: Extending Bitcoin's proof of work via proof of stake, SIGMETRICS Perform. Eval. Rev. 42 (3) (2014) 34–37.

[212] P. Vasin, Blackcoin as Proof-Of-Stake Protocol v2, 2014, URL https://blackcoin.co/blackcoin-pos-protocol-v2-whitepaper.pdf.

[213] B. Group, Proof of Stake versus Proof of Work, 2015, URL http://bitfury.com/content/5-white-papers-research/pos-vs-pow-1.0.2.pdf.

[214] J. Kwon, Tendermint: Consensus without mining, 2014, URL https://tendermint.com/static/docs/tendermint.pdf.

[215] I. Stewart, et al., Proof of Burn, 2012, URL https://en.bitcoin.it/wiki/Proof_of_burn.

[216] Sawtooth lake documentation, 2017, URL https://intelledger.github.io/introduction.html#proof-of-elapsed-time-poet.

[217] Intel software guard extensions, 2017, URL https://software.intel.com/en-us/sgx.

[218] B.M. Oki, B.H. Liskov, Viewstamped replication: A new primary copy method to support highly-available distributed systems, in: Proc. 7th Annu. ACM Symp. on Principles Distrib. Comput, ACM, 1988, pp. 8–17.

[219] G. Bracha, S. Toueg, Asynchronous consensus and broadcast protocols, J. ACM 32 (4) (1985) 824–840.

[220] R. Guerraoui, N. Knežević, V. Quéma, M. Vukolić, The next 700 BFT protocols, in: Proc. 5th Eur. Conf. Comput. Syst, ACM, 2010, pp. 363–376.

[221] A. Miller, Y. Xia, K. Croman, E. Shi, D. Song, The honey badger of bft protocols, in: Proc. 23rd ACM Conf. Comput. Commun. Secur., CCS' 16, ACM, 2016, pp. 31–42.

[222] C. Cachin, K. Kursawe, F. Petzold, V. Shoup, Secure and efficient asynchronous broadcast protocols, in: 2001 Proc. Annu. Int. Cryptology Conf., CRYPTO '01, Springer, pp. 524–541.

[223] Hyperledger, 2015, URL https://www.hyperledger.org.

[224] Burrow, 2017, URL https://github.com/hyperledger/burrow.

[225] Iroha, 2017, URL https://www.hyperledger.org/projects/iroha.

[226] Sawtooth, 2017, URL https://www.hyperledger.org/projects/sawtooth.

[227] Stale block, 2017, URL https://bitcoin.org/en/glossary/stale-block.

[228] S. Popov, The Tangle, 2016, URL https://www.iotatoken.com.

[229] Y. Sompolinsky, A. Zohar, Accelerating bitcoin's transaction processing. Fast money grows on trees, not chains, IACR Cryptol. EPrint Archive 2013 (881) (2013).

[230] Y. Sompolinsky, A. Zohar, Secure high-rate transaction processing in bitcoin, in: 2015 Proc. Int. Conf. Financial Cryptography Data Secur., FC '15, Springer, pp. 507–527.

[231] V. Buterin, Toward a 12-second block time, 2014, URL https://blog.ethereum.org/2014/07/11/toward-a-12-second-block-time/.

[232] I. Eyal, A.E. Gencer, E.G. Sirer, R. Van Renesse, Bitcoin-ng: A scalable blockchain protocol, in: Proc. 13th USENIX Symp. on Networked Syst. Design Implementation, NSDI'16, 2016, pp. 45–59.
[233] Ethereum, 2017, URL https://www.ethereum.org.
[234] N. Kabessa, PoW vs. PoS, 2017, URL https://medium.com/blockchain-at-columbia/pow-vs-pos-tech-talk-77f9a1bf05d7.
[235] ethernodes.org, Network number 1, 2017, URL https://www.ethernodes.org/network/1.
[236] Ethereum homestead documentation–the homestead release, 2016, URL http://www.ethdocs.org/en/latest/introduction/the-homestead-release.html.
[237] Vbuterin, On sharding blockchains, 2017, URL https://github.com/ethereum/wiki/wiki/Sharding-FAQ?from=groupmessagef.
[238] A.S.d.P. Crespo, L.I.C. García, Stampery Blockchain Timestamping Architecture (BTA)-Version 6, 2017, arXiv preprint URL arXiv:1711.04709.
[239] L. Tan, N. Wang, Future internet: The Internet of Things, in: Proc. 3rd Int. Conf. Advanced Comput. Theory Eng., ICACTE' 10, vol. 5, IEEE, 2010, pp. V5–376.
[240] A. Back, M. Corallo, L. Dashjr, et al., Enabling Blockchain Innovations with Pegged Sidechains, 2014, URL https://blockstream.com/sidechains.pdf.
[241] M.C. Vuran, Ö.B. Akan, I.F. Akyildiz, Spatio-temporal correlation: Theory and applications for wireless sensor networks, Comput. Netw. 45 (3) (2004) 245–259, In Memory of Olga Casals.
[242] D. Romero, V.N. Ioannidis, G.B. Giannakis, Kernel-Based reconstruction of space-time functions on dynamic graphs, IEEE J. Sel. Topics Signal Process. 11 (6) (2017) 856–869.
[243] Size of the bitcoin blockchain from 2010 to 2017, by quarter, Dec, 2017, URL https://www.statista.com/statistics/647523/worldwide-bitcoin-blockchain-size/.
[244] G. Ateniese, B. Magri, D. Venturi, E. Andrade, Redactable blockchain - or-rewriting history in bitcoin and friends, in: 2017 IEEE Eur. Symp. Secur. Privacy, EuroSP' 17, 2017, pp. 111–126.