Journal of Network and Computer Applications 125 (2019) 251–279

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

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

Review

Blockchain's adoption in IoT: The challenges, and a way forward

Imran Makhdoom [a,*], Mehran Abolhasan [a], Haider Abbas [b,c], Wei Ni [d]

[a] University of Technology, Sydney, Australia
[b] National University of Sciences and Technology, Pakistan
[c] Florida Institute of Technology, USA
[d] Data61-CSIRO, Australia

ARTICLE INFO

Keywords:
Blockchain
Internet of things
Consensus protocols
IoT security
Decentralized IoT
Blockchain challenges

ABSTRACT

The underlying technology of Bitcoin is blockchain, which was initially designed for financial value transfer only. Nonetheless, due to its decentralized architecture, fault tolerance and cryptographic security benefits such as pseudonymous identities, data integrity and authentication, researchers and security analysts around the world are focusing on the blockchain to resolve security and privacy issues of IoT. However, presently, not much work has been done to assess blockchain's viability for IoT and the associated challenges. Hence, to arrive at intelligible conclusions, this paper carries out a systematic study of the peculiarities of the IoT environment including its security and performance requirements and progression in blockchain technologies. We have identified the gaps by mapping the security and performance benefits inferred by the blockchain technologies and some of the blockchain-based IoT applications against the IoT requirements. We also discovered some practical issues involved in the integration of IoT devices with the blockchain. In the end, we propose a way forward to resolve some of the significant challenges to the blockchain's adoption in IoT.

1. Introduction

There has been an exponential growth in the Internet of Things (IoT) based services in the world, especially in telehealth, manufacturing and in urban areas to form smart cities. IoT is expected to connect 30 billion devices by 2020 (Lund et al., 2014). Use of IoT technology will not only improve the quality of life of people but also contribute to the world economy. IoT is predicted to create about USD 7.1 trillion contributions to the global economy by 2020 (Lund et al., 2014). However, at the same time, IoT devices are vulnerable to a vast number of security and privacy issues, which are known to the manufacturers but security in IoT devices is either neglected or treated as an afterthought (Wurm et al., 2016). According to IBM Institute for Business value (Brody and Pureswaran, 2014), it is critical for the future of IoT that its operational model is revived from costly, trusted and over-arched centralized architecture to a self-regulating and self-managed decentralized model. Such a transformation will provide scalability, reduced cost of infrastructure, autonomy, secure operations in a trustless environment, user-driven privacy, access control and redundancy against network attacks. In this regard, blockchain is being considered as one of the possible mechanisms to realize desired decentralization and resultant trustless networks (Christidis and Devetsikiotis, 2016).

Although blockchain was initially conceived as a financial transaction (TX) protocol in the form of Bitcoin, but due to its cryptographic security benefits such as pseudonymous identities (IDs), decentralization, fault tolerance, TX integrity and authentication, researchers and security analysts around the world are focusing on the blockchain to resolve security and privacy issues of IoT. However, default limitations of Bitcoin blockchain, such as scalability, latency in TX confirmation, large storage, intensive computation and energy requirements, and privacy leakage infer that blockchain technology has to be assessed deeply before it can be used securely and efficiently in an IoT environment.

Related Work. Till date, numerous surveys and some research on blockchain-based IoT technology (Christidis and Devetsikiotis, 2016; Yli-Huumo et al., 2016; Survey on blockchain, 2015; Pilkington, 2016; Tschorsch and Scheuermann, 2015; Dorri et al., 2016; Huh et al., 2017; Conoscenti et al., 2016; Bonneau et al., 2015) has been published but either these papers focus on general applications of the blockchain or discuss technical aspects concerning digital currencies.

* Corresponding author.
E-mail addresses: imran.makhdoom@student.uts.edu.au (I. Makhdoom), mehran.abolhasan@uts.edu.au (M. Abolhasan), haiderabbas-mcs@nust.edu.pk (H. Abbas), Wei.Ni@data61.csiro.au (W. Ni).

https://doi.org/10.1016/j.jnca.2018.10.019
Received 5 March 2018; Received in revised form 22 September 2018; Accepted 29 October 2018
Available online 2 November 2018
1084-8045/

They do not give an insight into blockchain challenges related to IoT. For instance (Yli-Huumo et al., 2016), highlights various security, privacy and performance issues such as DDoS attacks, 51% attack, data malleability, authentication, cryptographic, energy consumption, and usability problems. However, these issues have been discussed concerning cryptocurrencies such as Bitcoin, Ripple and Bitcoin exchanges. The paper also identifies some of the research areas such as scalability, smart contracts, licensing, IoT, security, and privacy, which have been neglected in current research. For most of the part (Yli-Huumo et al., 2016), presents the methodology of its research and broadly highlights the current research topics. Moreover, if we look from IoT perspective (Yli-Huumo et al., 2016), does not focus on this issue. Similarly (Survey on blockchain, 2015), carries out a detailed survey of blockchain technologies and their impact on society and economy. It discusses the problems associated with Bitcoin blockchain. It also draws attention to the wide utilization of blockchain technologies, but IoT is just a point in the long list of potential use cases of the blockchain. Finally, it addresses the issues related to administration and policy guidelines.

In another work (Pilkington, 2016), authors give an overview of blockchain technology, discuss its variants such as Ethereum (Buterin et al., 2014), Ripple (Xrp, 2013), Gridcoin (Gridcoin white paper, 2018), etc., and present a gist of some non-financial applications of the blockchain. It also does not address issues concerning blockchain's adoption in IoT. Similarly (Tschorsch and Scheuermann, 2015), presents a wholesome survey on technical aspects of digital currencies. It discusses the Bitcoin characteristics and related concepts especially the consensus protocols in much detail but with respect to digital currencies. Although the papers mentioned above have covered various aspects of digital currencies and blockchain in detail, but they are not focused on IoT. Moreover, authors in (Dorri et al., 2016) present a lightweight architecture of a smart home. However, the paper just focuses on the limitations of Bitcoin blockchain and propose a solution to avoid Bitcoin's issues of computation intensiveness, latency in TX confirmation and scalability. Correspondingly, the authors compare the security and performance efficiency of their solution with Bitcoin blockchain only.

In yet another work, authors in (Huh et al., 2017) propose one of the use cases of the blockchain for IoT, i.e., configuring and managing IoT devices using blockchain smart contracts. By doing so, authors aim to avoid the security and synchronization issues involved in a client-server model. Where, if a server gets malicious then all the connected devices will be vulnerable to security issues. Therefore, taking advantage of blockchain's trust-free distributed architecture the IoT devices are proposed to be configured and managed through Ethereum smart contracts (Buterin et al., 2014). Moreover (Conoscenti et al., 2016) carries out a literature review of blockchain applications beyond cryptocurrencies and their suitability to IoT. The review also aims at finding a solution to Bitcoin blockchain related vulnerabilities, such as integrity attacks, de-anonymization techniques, and adaptability of Bitcoin blockchain in IoT concerning high TX input in IoT. Whereas (Christidis and Devetsikiotis, 2016), gives an insight into the working of blockchain and smart contracts (Buterin et al., 2014). The authors prudently highlight the blockchain-IoT use cases such as a marketplace for sharing services and resources between IoT devices, P-2-P (Peer-to-Peer) market for renewable energy and supply chain management (SCM). The paper also highlights some issues about the use of blockchain in IoT. These issues include low TX throughput, high latency in PoW-based blockchains, the privacy of users and TX contents, legal matters associated with smart contracts and the need for changes. Similarly, authors in (Bonneau et al., 2015) have also made a valuable contribution to the Bitcoin research. They have carried out an in-depth analysis of numerous Bitcoin properties, stability issues, and Bitcoin forks. Authors also gave an overview of alternatives to Bitcoin consensus and user anonymity/privacy techniques.

Therefore, to cover the gaps in the literature concerning blockchain's adoption in IoT, there is a requirement of carrying out a comprehensive survey to find out that how does existing blockchain technologies impact IoT? Similarly, how can IoT leverage blockchain to resolve its security issues? and what are the impediments in doing so? This paper thus carries out a methodical review of the IoT threat environment, resultant IoT security and performance requirements and the impact of progression in blockchain technologies on IoT. The benefits afforded by the blockchain technologies and some of the blockchain-based IoT applications are pitched against the IoT security and performance requirements to identify the voids. We also carried out a comparison of some of the notable blockchain consensus protocols based on certain security and efficiency factors to determine a suitable technology for the IoT. It is presumed that Hyperledger-Fabric meets the most of the IoT requirements such as user authentication and authorization, identity management, data confidentiality, low latency in TX confirmation and means to achieve autonomous IoT operations using smart contracts also known as "Chaincodes". To discover some practical issues involved in the integration of IoT devices with the blockchain, we implemented an Ethereum blockchain-based IoT supported supply chain monitoring system in an experimental setting. We discovered that there are some challenges in securely sending sensor data from the IoT devices to the blockchain. It is also noticed that currently there is no mechanism to perform a device integrity check, to ascertain the validity of IoT devices. Whereas, it is an important security requirement, since, IoT devices mostly operate in an unprotected environment and are vulnerable to physical compromise, which can result into malicious device operation. We also establish that there is a requirement for IoT-oriented TX validation rules and IoT-focused consensus protocol to meet the specific needs of IoT environment. In the end, a way forward is recommended to address some of the significant blockchain issues. Hence, there are many factors that make our work distinguished from our predecessors.

Contributions of the Paper. The primary objective of this paper is to identify unscaled challenges that hamper the total adoption of blockchain in an IoT environment. The major contributions of the paper are:

1. Detailed analysis of progression in blockchain technology and its impact on IoT in view of security and performance requirements of IoT.
2. Identification of some unique and practical challenges to the blockchain's adoption in IoT.
3. Analysis of few existing blockchain applications and related voids.
4. A way forward to address some of the critical IoT related blockchain issues.

Organization. The rest of the paper is organized as follows: Section 2 provides a background on IoT architecture, introduces IoT threat environment and some security and performance requirements of IoT systems. In Section 3, some important blockchain concepts especially the consensus protocols are illustrated. Progression in blockchain technology and its impact on IoT is highlighted in Section 4. Whereas, Section 5 presents current challenges to the blockchain's adoption in IoT. Latest trends in blockchain-based IoT applications and related issues have been covered in Section 6. Gap analysis and a way forward to address some of the significant challenges is presented in Section 7 and Section 8 respectively. Finally, the paper is concluded with a hint of future work in Section 9.

2. IoT background

This section presents a brief background on IoT including IoT architecture, the difference between IoT and traditional networks, threat environment and some security and performance requirements of IoT systems.

2.1. IoT architecture

Due to the lack of standardization of IoT products the world has not yet been able to agree on a single IoT reference model (Al-Fuqaha et al., 2015). Correspondingly, as shown in Fig. 1, layered architectures and their tasks/functions or purpose discussed in the different literature (Al-Fuqaha et al., 2015; Kumar et al., 2016; Khari et al., 2016; Khan et al., 2012; Qiu et al., 2018) have slight variations. For instance (Al-Fuqaha et al., 2015) presents a 5-layered IoT architecture comprising Objects or Perception Layer, Object Abstraction Layer, Service Management also called as Middleware Layer, Application Layer, and The Business Layer. The tasks/functions or purpose of each layer are shown in the respective colored box in Fig. 1. The Objects or Perception Layer is responsible for querying and collecting sensor data and then forward it to the Object Abstraction Layer. The Object Abstraction Layer acts almost like the Network Layers depicted in all other models, i.e., transfer the data received from the objects (devices) to the next higher layer, i.e., Service Management or Middleware Layer, through various communication protocols such as RFID, 3G/4G, WiFi, BLE (Bluetooth Low Energy), infrared, ZigBee, etc. It can also perform other functions such as cloud computing and handling of data management processes (Al–Fuqaha et al., 2015). The Application Layer performs the typical tasks such as service delivery to the customers/users, provision of an interface to the business layer for high-level data analysis and management of controlled access to data. Lastly, the Business Layer manages all the activities and services, builds a business model, performs Big data analysis for strategic decision making (Al-Fuqaha et al., 2015). However (Kumar et al., 2016), deliberates upon a 4-layered architecture with a distinction between Physical and the Perception Layer. The Physical Layer comprising basic hardware including smart appliances and power supplies acts as a backbone for networking the smart objects. The perception Layer performs the usual task of collecting sensor data, and the Network Layer provides the means to transfer data between devices. Finally, the application layer performs the task of service delivery.

Contrary to the previously discussed IoT architectures (Khari et al., 2016), introduces a 3-layered model comprising the Sensor, Network, and the Application Layer. However, all these layers perform the same tasks as their equivalent, discussed in the previous two models. There is another 5-layered IoT architecture discussed in (Khan et al., 2012), which has almost the same layers as highlighted in (Al-Fuqaha et al., 2015) with a slight variation in the naming convention of layer two. However, the tasks/functions of the layers are nearly similar. Lastly, the authors in (Qiu et al., 2018) introduce a 4-layered IoT architecture comprising Sensing, Networking, Cloud and Application Layer. The notable thing here is the Cloud Computing Layer instead of Service management or a Middleware Layer. The authors propose that the Cloud servers having more computing power, better data analytic features and storage capacity, can better handle the huge data coming from the heterogeneous IoT devices and respond quickly based on emergency event-aware strategies. Whereas, the Middleware has certain issues such as though it can mask the differences in operating systems and network protocols, however, most of the Middleware services use proprietary protocols, which affect the interoperability. Moreover, Middleware services also suffer from time delay and memory overhead amid incompatible protocols of subsystems (Qiu et al., 2018). The authors claim that the cloud servers provide an abstract layer and can flawlessly realize the communication for heterogeneous systems.

2.2. Difference between IoT and traditional networks

The above-mentioned peculiarities make IoT different from traditional IT networks. These differences are important to be highlighted as they influence the development of requisite security and privacy solutions for IoT systems. The significant difference between conventional networks and IoT is the level of resources available at the end devices (Jing et al., 2014). IoT usually comprises resource constraint embedded devices such as RFID and sensor nodes. Low memory, low computing power, and small battery life are the hallmarks of typical IoT devices. Whereas, the traditional networks comprise powerful computers, servers, and smartphones that have ample resources. The traditional networks can, therefore, be secured by complex and multi-factor security protocols without any resource consideration. Contrary to this, IoT systems require lightweight security algorithms that should maintain a balance between security and resource consumption such as battery life, memory and processor usage.

IoT devices mostly connect to the internet or gateway devices through low bandwidth and low power wireless communication media such as 802.15.4, 802.11a/b/g/n/p, LoRa, ZigBee, NB-IoT and SigFox. Whereas, end devices in the traditional IT networks communicate through more secure and faster wired/wireless media such as fiber optics, DSL/ADSL, WiFi, 4G, and LTE. Another difference is that the traditional network devices have almost the same OS and data format, but in the case of IoT because of application-specific functionality and lack of OS, there are different data contents and formats. Hence, because of this diversity, it is difficult to develop a standard security protocol that fits all types of IoT devices and systems. As a result, a wide range of IoT threats are still at loose and threaten the security and privacy of the users.

If we look at the security design, traditional networks are secured by a blend of static network perimeter defense based on firewalls, IDS/IPS, and the end devices are secured by host-based approaches such as antivirus and security/software patches. Whereas, the host-based security approach cannot be applied to the resource constraint IoT devices (Yu et al., 2015). Similarly, because of the IoT devices' vulnerabilities such as lack of physical security, the absence of host-based defense mechanisms (e.g., anti-virus), lack of software updates and security patches, lack of access control measures, cross-device dependencies (e.g., a light sensor is triggered by a light bulb), and lack of IoT-focused attack signatures, the conventional perimeter defense mechanism cannot protect the IoT devices from insider attacks and physical compromise by unauthorized employs/personnel. Correspondingly, the low bandwidth, low power and less secure IoT wireless communication protocols (Koushanfar et al., 2012; Lough, 2001; Vanhoef and Piessens, 2014), weak application security, and vulnerable web applications and APIs (OWASP, 2018; Sivaraman et al., 2015) make IoT devices an ideal target for the attackers. Moreover, there is a lack of consistency and standardization in IoT solutions across the globe due to which there are issues related to interoperability, compatibility, and manageability (Banafa, 2016).

2.3. IoT threat environment

It is estimated that with the rise in the number of things connected to IoT systems to swarming billions of devices by 2020, the potential vulnerabilities will also increase (Ahlmeyer and Chircu, 2016). Hence, the increase in vulnerabilities due to non-standardization of IoT technologies may give rise to security incidents in IoT systems. Correspondingly, the successful launch of sophisticated cyber-attacks like Mirai (Ducklin, 2016), Ransomware (Brewer, 2016), Shamoon-2 (Kovacs, 2017) and DuQu-2 (Infosec-Institute, 2015) on Industrial Control Systems (ICS) and IoT in recent past have rendered existing IoT protocols ineffective and have proved that IoT systems/devices are vulnerable to cyber-attacks resulting into ransom payment, data theft, data forgery and other spurious behavior such as botnet attacks. In addition, mostly being deployed in a hostile or unprotected environment, IoT devices are vulnerable to physical compromise. In a practical manifestation of such an attack, researchers in (Wurm et al., 2016) compromised a smart controller of a house automation system through an open UART interface. Once the researchers gained access to the device, they were able to view the start-up sequence. They modified the boot parameters and gained low-level access to the device. They also brute forced the root password and launched network layer attacks such as port scanning and network traffic analysis.

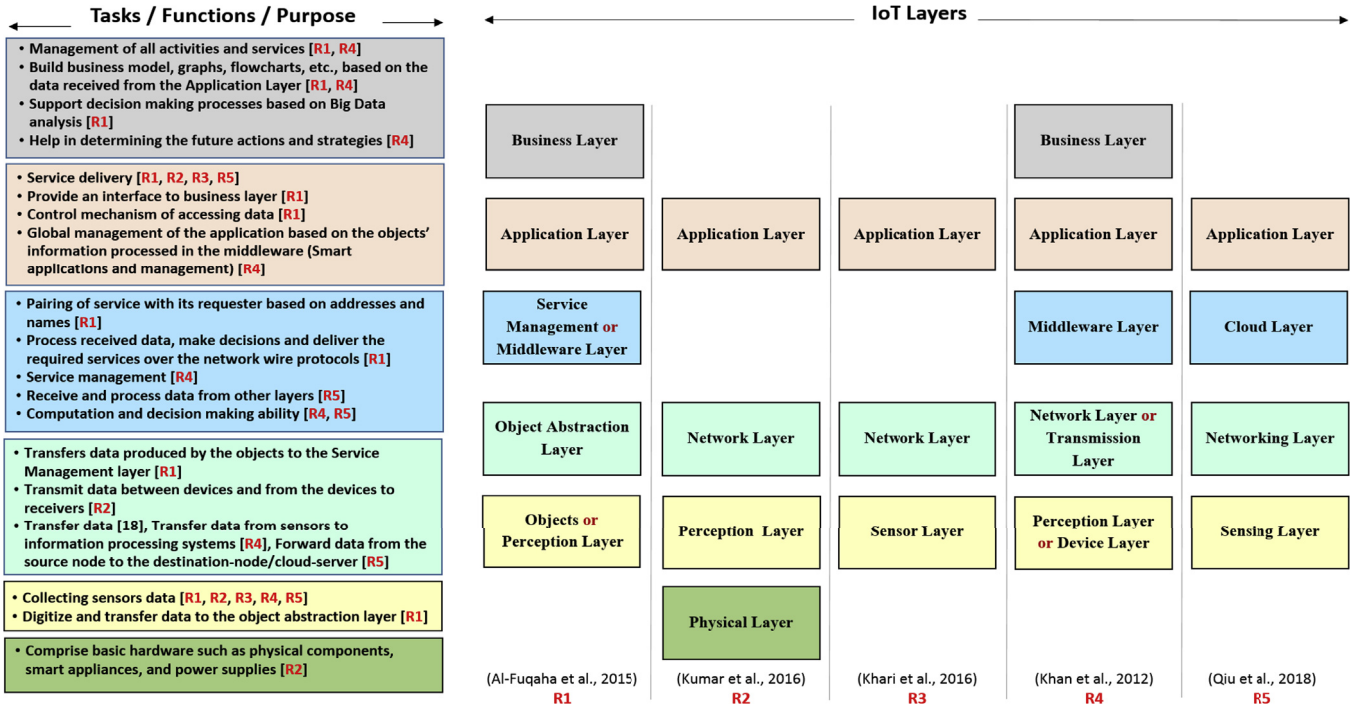

Fig. 1. Variations in the layered architecture of IoT.

Moreover, despite centralization and controlled access to data, even the cloud-supported IoT is vulnerable to security and privacy issues (Puthal et al., 2016). It is estimated that at least one-fifth of the documents uploaded to file-sharing services contains sensitive information and 82% of cloud service providers ensure data security during transmission. However, only 10% encrypt data, once it is stored in the cloud (The CEO's Guide to Data Security, 2016). Cloud being the trusted party is vulnerable to a single point of failure, data privacy breach including unauthorized data sharing and unauthorized data analytics (Kshetri, 2017). The disclosure of personal data leakage concerning 87 million users by Facebook Inc. in April 2018 is a candid example of one of the cloud vulnerabilities (Sara and Michael, 2018). Hence, Security flaws in IoT are thus leading to attacks on device integrity, data integrity, secrecy and privacy, attacks on the availability of network and attacks on the availability and integrity of services, e.g., DoS (Denial of Service) and DDoS (Distributed Denial of Service) Attacks (Borgohain et al., 2015). The current security issues in IoT can be attributed to the poor security-aware design of devices, scarcity of memory, power and computational resources, and trust in cloud-based applications.

Based upon above-discussed resource constraint peculiarities of IoT devices and IoT threat environment, we have deduced some security and performance requirements for future IoT systems. Hierarchical model of these requirements is reflected in Figs. 2 and 3 respectively.

2.4. Security requirements

The design and development of future IoT systems and devices is envisaged to be somewhat standardized as per the security requirements depicted in Fig. 2. The essential security requirement of an IoT system is to be able to operate in a trustless environment. Moreover, most of the IoT applications rely on sensors' data. Hence, unforgeable storage and security against data manipulation and unauthorized sharing is also required. Furthermore, most of the IoT devices, such as smart city environmental sensors (temperature, humidity, gas, etc.), surveillance cameras and intelligent traffic system sensors being deployed in public places without much protection are vulnerable to physical compromise (Arias et al., 2015; Balamurugan and Dyutimoy, 2017). Hence, no operation in an IoT system can be termed safe unless the integrity of the code installed on the IoT device and the integrity of the data being shared between devices is ensured (Sadeghi et al., 2015). Therefore, device security is another important aspect that needs attention by the manufacturers and the security researchers. To protect the network against node compromise and malware attacks, the IoT systems need to authenticate devices before adding them to the network. Similarly, there should be frequent checks to attest the integrity of the code installed on the devices. In case of any suspicion about the device software, the respective node should be revoked temporarily until the secure software update is performed.

IoT devices should also be tamper-resistant concerning both hardware and software modifications. Another vulnerable issue is that due to the scarcity of memory, power and computation resources, redundant cryptographic security measures cannot be implemented in IoT devices (Jing et al., 2014). However, still, IoT devices need some lightweight cryptographic security along with efficient key management system, in which compromised keys should be revoked and updated as and when required. Another important requirement is user security including enrolment, ID management, authentication, and authorization. In addition, a secure IoT system requires protection against unauthorized access to the network and user data.

2.5. Performance requirements

Due to reliance on real-time data sharing by most of the IoT systems like VANETS, Wireless Sensors Networks (WSN), ICS, smart grids, smart homes and SCM, the performance efficiency of the IoT system is as important as its security. Some of the performance requirements desired in IoT systems are shown in Fig. 3. To protect future IoT systems against human errors, they need to be self-regulated and self-managed. An efficient IoT system must cater for the constraint resources of end devices including low memory, low power consumption, and low computational ability. However, an increase in performance efficiency should not be on the pretext of compromising the security of the system. Moreover, an increase in the number of users/IoT devices in future, will result in the generation of more data.

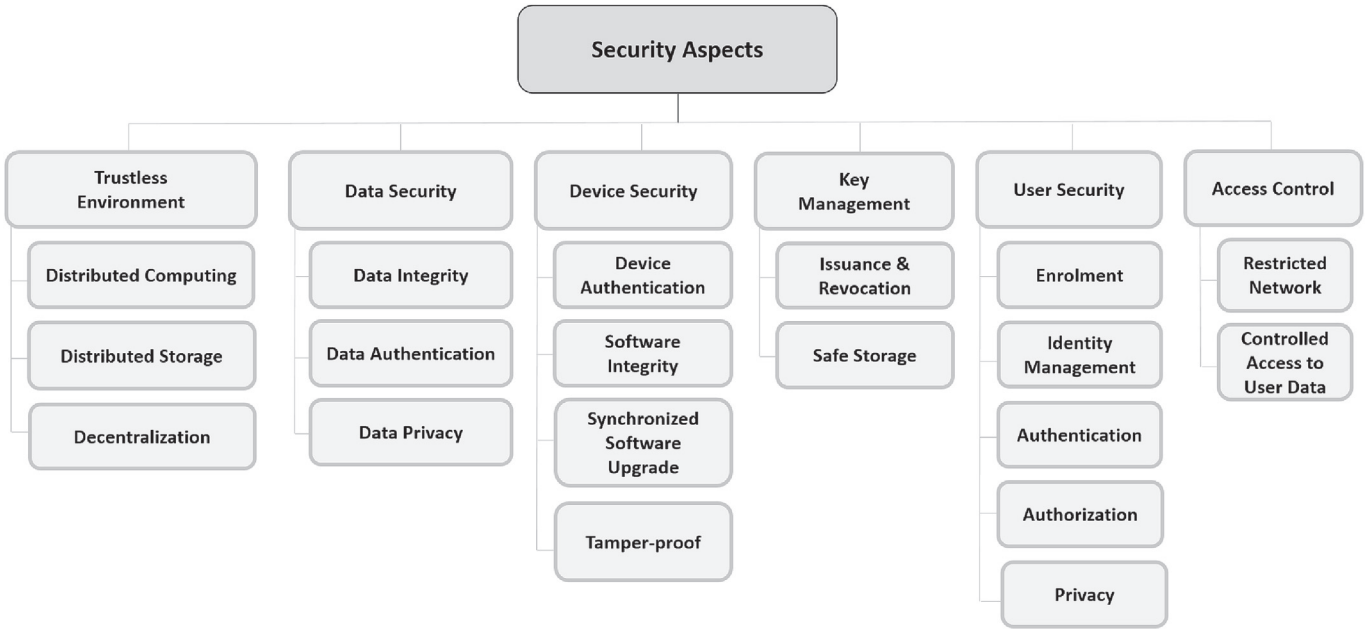

Fig. 2. Security requirements for IoT systems.

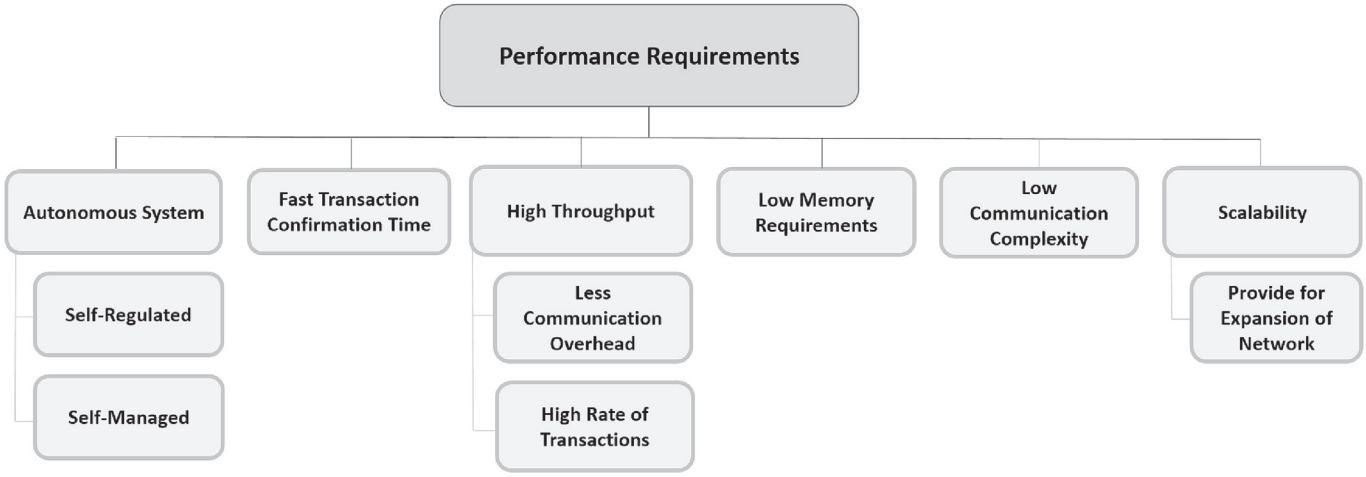

Fig. 3. Performance requirements for IoT systems.

Therefore, it is imperative that the respective IoT system should be able to accommodate future network expansion and handle a large number of messages with high throughput.

The existing threat spectrum coerces the need for a sophisticated security mechanism for IoT. Many security researchers visualize blockchain as the silver bullet to augment IoT security. Therefore, before proceeding further, it is essential to get familiarized with the blockchain technology.

3. Blockchain: an overview

The Bitcoin (Nakamoto, 2008) has very innovatively transformed the method of financial value transfer without any trusted third party. The underlying technology of Bitcoin is blockchain. In simple terms, blockchain comprises a series of blocks in such a way that every new block is cryptographically connected to the previous block. In the case of Bitcoin, the blocks contain a record of financial TXs between Bitcoin users. Due to its inherent benefits, such as immutability, auditability, TX integrity and authentication, fault tolerance, and above all trust-free operation, blockchain is being envisaged to play a vital role in the security of IoT ecosystem. Various benefits of Bitcoin blockchain and how they are achieved are enumerated in Table 1, and some important concepts concerning blockchain technology are illustrated below.

3.1. Key concepts

Transaction (TX). A process that results in the change of state of the blockchain. Depending upon the blockchain platform, a TX ranges from the transfer of a financial value to the execution of an arbitrary code in the form of a smart contract (How does bitcoin work?, 2017). Moreover, in the case of an IoT environment, a TX may be a means of sharing user or environment sensors' data.

Block. It is a set of TXs that happened in the recent past and have not been confirmed yet. The block also has a block header that contains, blockchain version number, hash of the previous block, a random nonce, time stamp and Merkle Root Hash of all the TXs included in the block.

Blockchain. It is a distributed public ledger that keeps a record

Table 1
Benefits of bitcoin blockchain.

Ser	Benefit	Achieved by
1.	Avoids a single point of failure	Distributed public ledger and decentralization
2.	No central authority or third party mediation	Validating the TXs with the consensus of network nodes
3.	No central database	Distributed public ledger
4.	Resilience to node compromise	Network consensus and state machine replication
5.	Auditable and immutable TXs	The recording of validated TXs in an unforgeable blockchain with a timestamp makes them always available for the audit. However, if an attacker acquires 51% or more hash power then he can change the history of the blockchain and double-spend the TXs
6.	Transparency	TXs are publicly announced to enable all nodes of the blockchain network to maintain a same copy of the order of TXs. Moreover, the TXs are published on the blockchain in clear text.
7.	Pseudo-anonymity	Hash of Public Keys
8.	Trust-free operation	Validation of each TX by network nodes
9.	TX authentication and non-repudiation	Signing of TXs by the user's private key using Elliptic Curve Digital Signatures Algorithm (ECDSA) (Zheng et al., 2016)
10.	TX integrity	Taking SHA-256 hash of a TX
11.	Protection against replay attack	Use of timestamps

of all the TXs/blocks (Nakamoto, 2008). Vitalik Buterin in (Buterin, 2015a) gives another perspective that the essence of the blockchain is informational and processual, and does not relate directly to the monetary sphere.

Mining. It is the process of adding validated TXs to a block and then broadcasting that block on the blockchain network, to be known by all the nodes. The mining is done by miner nodes, and the selection of a node to mine a new block is done based on certain lottery schemes. In the case of Bitcoin, miners compete to solve a cryptographic hash puzzle and whosoever finds the solution (also known as proof of work) first, is eligible to mine the next block. When a block is mined and added to the blockchain, then the TXs in that block are confirmed (Bitcoin developer guide, 2017). Irrespective of the type of blockchain platform, usually some lottery scheme is required to randomly select a miner to propose or mine a new block.

Simple/Normal Node. There may be different types of nodes in a blockchain network depending upon their capabilities and resources such as computation capability and memory size. A node may be a simple node, which can only send and receive a TX and does not store the complete copy of the blockchain. In case of an IoT environment, a simple node can be an Arduino-based sensor node, that can only send a sensor reading to the gateway device or receive some commands.

Full Nodes. These nodes maintain a complete copy of the blockchain, but they do not mine a block. However, full nodes validate TXs based upon the consensus rules of the respective blockchain and contribute in accepting or forking out a block (Bitcoin-Forum, 2016). A double-spending or a malicious TX may not even be routed or relayed by a full-node. This implies that full nodes are capable of TX and block propagation. Hence, full nodes are essential for the security of the blockchain. In an IoT environment a Raspberry Pi (Rpi) with more computational and memory resources as compared to an Arduino, can be a full node (EthEmbedded, 2017). It was also tested by running a Go Ethereum version geth-linux-arm7-1.8.3 on a Rpi-3 based sensor node.

Miner/Validator Nodes. These are the full nodes that have the additional capability to mine or validate a new block thus extending the blockchain (Bitcoin-Forum, 2016). Moreover, mining nodes are selected as per specific criteria based upon the type of consensus protocol being used in the blockchain. E.g., In Bitcoin, the mining nodes have to solve a cryptographic puzzle, and the node that does it first is eligible to mine the block. The miner node has to submit a Proof of Work (PoW) along with the mined block so that the rest of the nodes can validate that the puzzle has been correctly solved. If the block is accepted by the rest of the network, the miner node then earns a block reward and TX fee in the form of respective cryptocurrency. Whereas, in Proof of Stake (PoS) consensus protocol, miner nodes are selected randomly based on the coinage, i.e., the number of coins they own and the time since they have those coins. However, in most of the (Byzantine Fault Tolerance) BFT-based consensus protocols, the validator is elected in a round robin fashion to propose a new block. The rest of the member nodes of the quorum, vote on the validity of the block and its TXs. In most of the cases, the block is validated and included in the blockchain upon getting 2/3 majority votes in its favor.

TX/Block Finality. It is related to the final confirmation or approval of a particular TX or a block by the consensus protocol of respective blockchain. It is an important aspect as it infers delay in TX confirmation and ultimately affects the TX throughput of the blockchain. E.g., In Bitcoin, a TX gets one confirmation/approval after 10 min, i.e., once the block containing that TX is mined. However, to get a final confirmation, the TX has to wait until additional five blocks are mined and appended to the block containing that particular TX. Hence, it takes 60 min to finally declare a TX confirmed/approved in Bitcoin blockchain. Whereas, in other blockchains such as Hyperledger (The-Linux-Foundation, 2018) and Tendermint (Tendermint Core, 2018) the TX gets instant confirmation.

Permissioned vs. Permissionless Blockchains. Before defining Public and Private blockchain types, it is imperative to highlight that a blockchain can be a permissioned or a permissionless blockchain based on the restrictions to process the TXs, i.e., creating new blocks of TXs. In a permissionless blockchain, any node can create new blocks of TXs, whereas, in a permisisoned blockchain, TX processing is performed by selected nodes only. As far as the terminology of a Public and a Private blockchain is concerned, it relates to the access to the blockchain data (Garzik, 2015).

Public Blockchain. It may be a permissionless digital ledger that allows free and unconditional participation by any node (Garzik, 2015). Mining in public blockchains is mostly incentive-based, so that miners are encouraged to mine a block. Hence, public ledgers bear more TX cost than private ledgers (Buterin, 2015a). Whereas, the connectivity between nodes in public blockchain is less than in private blockchain, therefore, it takes a longer time to finalize the TXs (Pilkington, 2016). Moreover, to achieve transparency in permissionless blockchains, all the TXs are visible to the public. Hence, issues related to user anonymity and data privacy emerge. Moreover, public blockchains have low TX throughput because of poor TX finality, especially in PoW based blockchains (Lukas, 2018). Real world examples of public blockchains are; Bitcoin (Nakamoto, 2008), Ethereum (Wood, 2014), IOTA (What is iota?, 2017), Litecoin (Litecoin, 2011), Lisk (Lisk documentation, 2018), etc.

Private Blockchain. It can be a permissioned ledger, in which the number of the miner nodes is limited, and their IDs are known. Hence, TX processing is restricted to the selected/pre-defined miner or validator nodes only. Moreover, a user may have access only to those TXs that are directly related to him (Garzik, 2015). E.g., Hyperledger-Fabric enables competing businesses and groups to maintain the privacy and confidentiality of their TXs, using "Private Channels" (Hyperledger-fabric documentation, 2018). Private channels can be termed as restricted

Table 2
Public vs Private Blockchains.

Public (may be Permissionless) Blockchain	Private (may be Permissioned) Blockchain
Permissionless participation	Permissioned participation
IDs of nodes are not known (Use of pseudonymous IDs)	IDs of nodes are known (Garzik, 2015)
Unlimited number of nodes	A limited number of nodes
Less data privacy	Options available for data security
Poor consensus finality (Buterin, 2016)	Instant consensus finality (In BFT-based blockchains) (Buterin, 2016)
Low TX throughput (Lukas, 2018)	High TX throughput (Lukas, 2018)
Good scalability (concerning the number of miner nodes) (Lukas, 2018)	Poor scalability (In BFT-based blockchains) (Lukas, 2018)
Vulnerable to 51% attack (In case of PoW and PoS blockchains)	Vulnerable to node collusion (In BFT-based blockchains) (Garzik, 2015)

messaging paths that can be used to provide TX privacy and confidentiality for specific subsets of network members. All data, including TX, member and channel information, on a channel, are invisible and inaccessible to any network members not explicitly granted access to that channel. Hence, comparing to the public ledgers, there can be more privacy of user information in the private blockchains. Another difference between public and private blockchains is the extent to which they are centralized or ensure anonymity (Pilkington, 2016). TX costs in private ledgers are also low amid less number of nodes (Buterin, 2015a). Due to immediate TX finality permissioned blockchains have high TX throughput (Lukas, 2018). Therefore, it can be attributed that private blockchains are faster than public blockchains. However, private blockchains with BFT-based consensus protocols suffer from poor scalability issues in terms of the number of validator nodes. In addition, according to (Zheng et al., 2016) the TX record in these types of blockchains can be tampered with due to its partial centralization (known and less number of mining nodes). Concerning IoT systems, which are mostly private, a permissioned blockchain is the appropriate ledger technology. Some of the examples of real-world implementation of private ledgers include; Hyperledger (Hyperledger Fabric, 2018), Multichain (Gideon, 2015), Quorum (Quorum - white paper, 2016), etc. The key differences between public and private blockchains are shown in Table 2.

Hybrid Blockchain. Being a balance between public and private blockchain, it is also called as “Partially Decentralized” or “Consortium Blockchain”, (Pilkington, 2016). E.g., In a consortium of ten industrial organizations, every organization maintains a mining/validating node in the blockchain network. In this case, a block may be valid only if it has been signed by minimum seven nodes. All the nodes may have open read access to the blockchain, or it can be restricted to specific nodes only (Buterin, 2015b). However, there is a possibility of tampering with blockchain record due to reduced decentralization (Zheng et al., 2016).

Blockchain Forks. Most of the public blockchains are prone to forks, i.e., if a miner node mines a block and the rest of the network rejects that block due to consensus rules violation, then the small chain extending from the rejected block onwards is forked, and the other longest chain extending from the correct block will be accepted as the valid chain. One of the main reason of forks in public blockchains is due to the consensus mechanism such as PoW, PoS, PoET, and PoA, in which there is no consensus finality once a block is mined. The consensus is achieved subsequently once succeeding blocks keep on extending the chain leading from the older block. The forks can be soft and hard depending upon acceptance and removal by the upgraded (following new consensus rules) and non-upgraded nodes (following old consensus rules) (Bitcoin developer guide, 2017).

A hard fork is created intentionally once a system is upgraded or an important change in consensus rules is deemed necessary. Hence, the latest version of consensus rules is not compatible with the older version. Therefore, a block following the new consensus rules is accepted by upgraded nodes but rejected by the non-upgraded nodes and when the mining software gets blockchain data from the non-upgraded nodes, it refuses to build on the same chain and accepts data only from the upgraded nodes. This creates permanently divergent chains, one for non-upgraded nodes and one for upgraded nodes.

The soft fork is formed when a block violating new consensus rules is rejected by the upgraded nodes but accepted by non-upgraded nodes. It is possible to keep the blockchain from permanently diverging if upgraded nodes control the majority of the hash rate (Bitcoin developer guide, 2017).

From IoT perspective, blockchain forks are not desired as they cause a delay in TX confirmation. E.g., In Bitcoin, due to the blockchain forks, a TX has to wait for six additional blocks to be mined over its respective block, to be considered confirmed. This wait time of six blocks infers a delay of 60 min in a TX confirmation. Whereas, in the case of near-realtime IoT systems such as smart cars, intelligent traffic monitoring systems, drones, health monitors, a delay in TX confirmation can lead to a substantial physical and financial damage.

Smart Contracts. Exploiting the Bitcoin’s ability to execute autonomous scripts, developers have created new versions of the blockchain that can perform arbitrary computations other than transferring coins. E.g., Ethereum blockchain (Buterin et al., 2014) implements scripts called smart contracts (Buterin et al., 2014) that can run any algorithm encoded in them as a part of the TX (Sebastián, 2017). Being deployed on the blockchain, the smart contracts are also called as “Decentralized Applications or DApps”. Since smart contracts reside on the blockchain, they have a unique address. A smart contract can be triggered by addressing a TX to it under some rules that govern the contract. Smart contracts can be used in applications like auto-pay (shopping, parking, route management, tolls, fuel payment), digital rights management, financial services including loan, inheritances, escrow, cryptocurrency wallet controls, capital markets, mortgage, automatic payment of insurance claims (Tuesta et al., 2015), SCM and smart grid (Huckle et al., 2016; Christidis and Devetsikiotis, 2016).

The key idea behind smart contracts is the development of autonomous objects or IoT devices that cannot only rent or sell their data but also maintain their operability by paying for the maintenance services. Such an autonomous system is likely to contribute to the development of an overall “Economy of Things” with the goal of providing efficient and consistent services without any intermediary.

Consensus Protocol. It is the mechanism or set of rules that enables all the full nodes to reach an agreement over the order of TXs. There are many types of consensus protocols being used in different blockchain applications. E.g., PoW, PoS, Practical Byzantine Fault Tolerance (PBFT), etc. Some of the notable consensus protocols are being discussed in succeeding paras.

Consensus Finality. It means, the convergence of the blockchain consensus process on a particular block/order of TXs. However, in reality, a consensus process may result into a permanent block or a stale block that may be forked out later. This aspect is further illustrated by Vitalik Buterin in (Buterin, 2016), that the finality of a TX is always probabilistic. However, it may stand true for a PoW, PoS or PoET consensus protocols (Baliga, 2017), but other consensus protocols may have different finality guarantees. Such as Casper (Buterin, 2016) offers stronger finality guarantees as compared to PoW consensus and similarly, BFT-based consensus protocols provide immediate consensus finality (Vukolić, 2015; Baliga, 2017), and the TXs once confirmed are

not forked out later. From IoT point of view consensus finality is an essential requirement in most of the IoT systems as it also influences latency in TX confirmation.

Proof of work (PoW). It is the computation of a cryptographic hash function with some degree of difficulty (Nakamoto, 2008), i.e., selecting a nonce such that the computed cryptographic hash has a specific number of zeros in the start as defined by the level of difficulty. PoW forms the basis of consensus tactics in Bitcoin and other cryptocurrencies. When a miner node solves the PoW, it is eligible to mine a new block. Whereas, other full nodes in the network mutually confirm its correctness (Zheng et al., 2016). PoW protects against double-spending attacks. Since it is computationally intensive, it is challenging for a single attacker to solve the difficulty for all the modified blocks before the honest nodes in the network (Nakamoto, 2008). It is a common perception that if a malicious miner or a pool of miners gain 51% of the total network hash power, they can control the network (Zheng et al., 2017). However, authors in (Eyal and Sirer, 2018) prove that the malicious/dishonest miners resorting to selfish mining strategy can gain more revenue by only 25% of the total hashing power. Therefore, minimum 2/3 of the network nodes need to be honest to protect against selfish mining; a simple majority is not enough. Moreover, public networks with pseudonymous user IDs are prone to Sybil attack. Therefore, Satoshi Nakamoto conceived PoW-based consensus for Bitcoin blockchain to make Sybil attacks more expensive to be launched (Vukolić, 2015; Miller et al., 2016).

Proof of Stake (PoS). It was conceived based on an idea described in (Szabo, 2004) to improve upon PoW's high latency, high computation, and high energy costs. PoS implies that people with high stakes are less likely to attack the respective network. Hence, an entity with the highest coinage, i.e., number of coins times the days, will be eligible to mine a new block. Moreover, the mining difficulty is inversely proportional to the coinage (Tschorsch and Scheuermann, 2015). However, once the miners claim the reward, the coinage is reset so that other miners/stakeholders also get the chance to mine a block. Therefore, if an attacker wants to launch an attack similar to 51% attack, he must own enough coins so that even when the coinage is reset, he can still gain more than half of the odds (Tschorsch and Scheuermann, 2015). In addition, Nicolas Houy in (Houy, 2014) proves that PoS is vulnerable to a 51% attack, as the few rich stakeholders can collude to manipulate the state of the ledger. Nevertheless, the probability of a 51% attack in PoS is considered to be lower as compared to the PoW (Gao and Nobuhara, 2017). Moreover, the maximum TX rate a PoS protocol has achieved is a few hundred TPS (Transactions Per Second) as compared to Visa's peak capacity of 56000 TPS (EconoTimes, 2017; Bitcoinwiki, 2017). Due to the lack of consensus finality, PoS-based consensus can also lead to blockchain forks (EconoTimes, 2017). A variation of PoS named "Delegated Proof of Stake" (DPoS) (Larimer, 2014; Kwon, 2014) implemented in Bitshare, a digital currency, is considered to be more efficient than PoS in terms of TX confirmation time. Moreover, it can tolerate up to 50% malicious nodes (Zheng et al., 2016; Zheng et al., 2017).

Proof of Activity. Proof of Activity is a combination of PoW and PoS (Bentov et al., 2014). It has been developed in the wake of an assumption based on an economic phenomenon called "Tragedy of the Commons". Which implies that over the period the block reward in PoW-based cryptocurrencies will subside, hence, the miner nodes will have less interest in ensuring the security of the network, thereby making it vulnerable to various attacks. Therefore, the proposed Proof of Activity protocol aims to increase the cost of an attack for a malicious user by forcing it to achieve eight times faster hash rate than the honest miners in the network. In addition, it reduces the computation complexity to 1/10th of the Bitcoin PoW, hence, minimizing the energy consumption as well. However, Proof of Activity also aims to secure only cryptocurrency applications.

Proof of Authority (PoA). Based on PoS, PoA is developed as an alternative to PoW in private blockchains. It has been implemented by Parity (Ethcore, 2018). In this protocol, the authorities are predetermined and each authority is assigned a fixed time slot within which it can generate blocks. Each authority is known based on its true ID, therefore, instead of monetary value at stake, PoA implies validator's ID at stake. Hence, any misconducting validator will be publicly known (Proof of authority, 2017). PoA makes a strong assumption that the authorities are trusted, and therefore, it is only suitable for permissioned ledgers. However, PoA is also being used by Ethereum test network Kovan (Dinh et al., 2017).

Proof of Elapsed Time (PoET). To address the problems of high power consumption and latency in the PoW-based consensus protocols, Intel developed a lottery-based consensus protocol named "PoET" for Sawtooth Lake, a blockchain-based distributed application platform. According to this protocol, the miner node which presents the least waiting time is selected to mine the next block. The PoET leader election protocol meets the criteria for a good lottery algorithm, i.e., fairness, investment and verification. It randomly distributes leadership election across the entire population of the validators. PoET is secured by the Trusted Execution Environment (TEE) through Intel's Software Guard Extension (Kastelein, 2016a). Except leader election based on PoET for which specialized hardware is required, the rest of the protocol works like Bitcoin protocol. The trust is also placed in the hardware that generates the random wait time.

The Proof of Burn (PoB). It implies that the users send coins to a verifiable but an unspendable address, thus burning the coins, to be eligible to mine a block (Iain, 2018). The difference between PoW and PoB is that PoB has no energy costs and its economic implications add towards the stability of the network. PoB has been adopted by a cryptocurrency Slimcoin (Slimcoin, 2014).

BFT-based Consensus. BFT is a family of state machine replication protocols (Lamport, 1978; Schneider, 1990) that protects against arbitrary faults by replicating the services on multiple nodes. The safety and liveness property of BFT protocols can tolerate no more than $(n-1)/3$ faulty replicas over the lifetime of the system (Castro and Liskov, 2002), where n is the total number of replicating nodes. However, in reality, any number of nodes can get malicious or show abnormal behavior (Ducklin, 2016). In contrast to PoW, BFT-based protocols require the IDs of the consensus nodes to be known, hence making it suitable for permissioned blockchains (Vukolić, 2015). BFT-based state-machine replication protocols are considered to have poor scalability as they have never really been tested for the scalability beyond 10–20 nodes (Brewer, 2000). Similarly, authors in (Hardjono and Smith, 2016) state that BFT-based protocols are not considered suitable for a network with more than 100 nodes. The leading cause of the scalability issue seems to be the network communication which often involves $O(n)^2$ messages per consensus request (Castro and Liskov, 2002). Some of the variations of BFT-based protocols, which are currently being used in various blockchain platforms are mentioned in succeeding paras.

Practical Byzantine Fault Tolerance (PBFT). It is designated to be more efficient than a PoW concerning latency and energy costs, but it can only tolerate up to 33% malicious nodes. PBFT (Castro and Liskov, 2002) is considered to be an expensive protocol concerning the number of messages required for consensus. The client's request is processed through 5 different stages, i.e., initially broadcast from client to all the replicas, then processed through pre-prepare, prepare, commit and execution stage. Hence, in a network with four replicas, a single request requires 32 messages between client and replicas, i.e., 4 in stage-1, 3 in stage-2, 9 in stage-3, 12 in stage-4 and 4 in stage-5 respectively.

Moreover, in every stage of PBFT protocol, the decision is based upon no of certificates received for the previous stage. The number of certificates required to make a decision depends upon the estimated number of faulty nodes, e.g., to commit a message/request the replicas have to receive at least $2f$ prepared certificates for that request, and to finally execute the request, the replicas need at least $2f+1$ commit messages. This means that the number of faulty nodes has to be predetermined. PBFT protocol guarantees liveness based on weak timing

assumptions. It operates in a primary-backup mechanism, and replicas move through a succession of configurations called views. Replicas initiate a change-view request, i.e., elect a new primary, when a respective primary fails or does not respond in a set timeout period (Castro and Liskov, 2002; Decker and Wattenhofer, 2013). Such weakly synchronous protocols are expected to degrade significantly when the underlying network behaves in an unpredictable manner. Therefore, the asymptotic communication complexity of PBFT in worst conditions can rise to ∞ (Miller et al., 2016). Moreover, such a mechanism is expected to be vulnerable to less throughput in case of frequent network failures, and even DoS attacks, where a persistent adversary causes network interruptions.

In a demonstration of such a DoS attack, authors in (Miller et al., 2016) implemented a malicious network scheduler to intercept and delay all view change messages of a PBFT protocol. They concluded that due to network interruptions and weak synchrony property of PBFT, the replicas remained stuck in view changes and never moved forward. They also deduced that such behavior is not restricted to PBFT. Instead, all protocols that rely on weak timing assumptions to tackle crashes can be affected by DoS attacks.

DBFT (Delegated Byzantine Fault Tolerance). DBFT has been implemented by NEO (NEO.org, 2017), an open source blockchain project. NEO aims to realize the goal of the smart economy by employing the triad of digital assets, digital ID and smart contracts. DBFT consensus protocol is based on proxy voting. The NEO holders select the delegates/bookkeeper nodes that maintain the digital ledger. A speaker is selected amongst all the bookkeeping nodes, and together these nodes reach an agreement and generate new blocks. The protocol is tolerant to $f = (n-1)/3$ faults (NEO.org, 2017; Erik, 2017), where, n is the total number of delegates/bookkeeper nodes and f is the number of faulty nodes in a consensus process. NEO provides efficiency by generating a block in 15–20 s with a throughput of 1000 TPS (NEO.org, 2017; Neo.org, 2017; Steemit, 2017). A new block is generated at the end of each round based on at least $n-f$ signatures by the bookkeeping nodes (Erik, 2017; Neo.org Consensus, 2017). During the consensus process, DBFT also depends upon weak-synchrony (weak timing assumption). Hence, a view change is requested by the nodes if consensus does not take place in a particular view (Erik, 2017). Therefore, DBFT is also vulnerable to DoS attacks based on network failures/interruptions. However, DBFT provides consensus finality without any risk of blockchain forks (EconoTimes, 2017). As far as communication complexity is concerned, for one client and four validator nodes, DBFT consensus require ten messages to process a TX.

Honeybadger-BFT. It is designed and optimized for a cryptocurrency scenario with restricted bandwidth but significant computing power (Miller et al., 2016). It employs a BFT atomic broadcast protocol that provides optimal asymptotic communication complexity of $O(n)$ in the asynchronous network setting. Therefore, it does not rely on timing assumptions to make progress whenever messages are delivered regardless of actual clock time. As per experimental results, Honeybadger-BFT provides better throughput in terms of TPS, than PBFT. However, it has been tested for the tolerance of up to $f=n/4$ faulty nodes only. Moreover, the latency in TX confirmation also increases with the rise in the number of validating nodes. Hence, while expanding the network, there is a need to maintain a balance between the number of nodes, bandwidth utilization, and latency tolerance level of the users/applications.

Tendermint. Based on BFT, Tendermint employs a consensus protocol without mining. A block is initiated by a proposer, which is selected in a round-robin fashion from dedicated validators (with voting power equal to their bond deposit). TX validation is done based on majority voting, i.e., honest validators should have a majority vote of $>= 2/3$ of the total votes. There are three standard and two special steps in each validation round. The consensus process in deciding the next block can be extended to many rounds (no bound on maximum rounds is given) depending upon certain conditions (Tendermint Core, 2018). Some of these conditions include: the designated proposer is not online, block proposed by the designated proposer did not propagate in time and even if there is a valid block, but $> 2/3$ pre-votes or $> 2/3$ pre-commits were not received by enough validators in time. This dependence on time can be exploited by any MITM (Man in the Middle) adversary who can simply delay the messages from the proposers, thus forcing the protocol to go for so many rounds that the system experiences delays in computing new block heights. To curb false block propagation by the proposers, Tendermint employs a concept of punishment by confiscating the bond deposit of the faulty proposers. For one client and four validator nodes, Tendermint consensus protocol needs to share 21 messages to process a single TX. It can also tolerate at the most $< 1/3$ faulty/malicious nodes.

Algorand. Algorand is a new cryptocurrency developed to overcome the issues of TX latency and blockchain forks in PoW, and PoS cryptocurrencies (Gilad et al., 2017). By using a Byzantine agreement protocol, a block is finalized at the end of the consensus process. Hence, TX confirmation time is brought down to an order of a minute. It also protects against Sybil attack by randomly selecting committee members for the consensus agreement based on their weight. Where weights are derived from the amount of money/cryptocurrency, one owns. Thus, as long as more than some fraction (over 2/3) of the money is owned by the honest users, Algorand can avoid forks and double-spending. It addresses the issue of scalability concerning BFT protocols such as PBFT, which are considered to be communication intensive and can scale merely to a dozen nodes/servers. It achieves this by randomly selecting a small set of committee members for each step of the consensus protocol.

Algorand avoids targeted attacks against the committee members by not using a fixed set of members. It selects the members in a private and non-interactive way. The users compute a Verifiable Random Function (VRF) on their public and private keys. The result of the function indicates to the users that whether they are selected to participate in the consensus process or not. In this non-interactive way of selection, an adversary does not know exactly who the committee members are. Algorand, makes it further secure, by selecting new committee members for each step of the consensus process. In this way, even if the attacker comes to know about a committee member once he starts participating in the consensus process, his attack efforts are futile, as that member will not participate in the next step. Algorand is claimed to be resilient to DoS attacks and it can continue to operate even in the absence of some of the users/nodes. As far as TX throughput is concerned, Alogrand commits a 2 MB block in 22 s and on the average commits about 750 MB of TXs in an hour, which is approximately 125 x Bitcoin's throughput.

IOTA. It is a blockless distributed ledger developed to enable micropayments in IoT industry (Popov, 2016). It employs tangle, a Directed Acyclic Graph (DAG) to store TXs instead of a blockchain. It is believed to be a successor of blockchain technology, as it addresses the issues of scalability and high TX fee. Latency in TX confirmation is reduced by making consensus (TX validation) parallelized, and an integral part of the TX generation process. IOTA does not require a miner to mine a block of valid TXs, rather, every node approves/validates randomly selected two previous TXs, before initiating its own TX. However, for the TX to be valid, the node must solve a PoW-based puzzle (similar to Bitcoin). IOTA is believed to be suitable for asynchronous networks, as all the nodes may not see the same set of TXs. Therefore, nodes do not have to achieve consensus on which valid TXs have to be included in the ledger. Instead, a specific node just decides between two conflicting TXs by running a tip selection algorithm based on Markov Chain Monte Carlo (MCMC) method, which selects a TX based on acceptance probability. E.g., A user runs MCMC 100 times for a particular TX, and if that TX is accepted 51 times, then it means that the TX was approved with 51% confidence. For high-valued TXs, the threshold can be set as high as 99% acceptance probability. However, IOTA does not have consensus finality. Hence, it is also prone to forks which cause latency in TX confirmation. It is also not clear yet that after how many direct or indirect approvals a TX is safe to be declared confirmed?

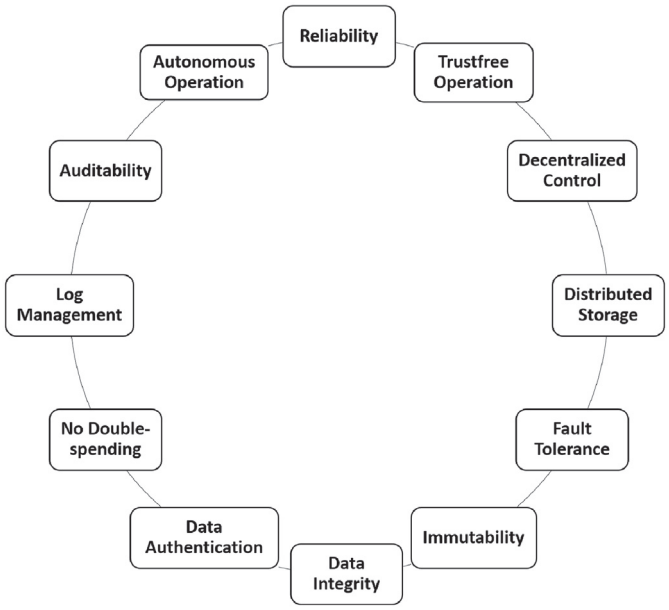

Fig. 4. Blockchain for IoT.

For better performance efficiency, even if a node does not initiate any TX, it still has to work by relaying new TXs to other nodes, as each node maintains a record of TXs received from its neighboring nodes. As far as security is concerned, to protect against spamming attacks, every TX is weighted based upon the amount of work done during PoW by the issuing node. Authors of IOTA claim that it protects against double spending and quantum computing attacks by capping maximum own weight that can be assigned to a TX by the issuing node. Secure and authenticated data sharing between multiple nodes is also one of the core features of IOTA (Larimer, 2014). In spite of all these features, IOTA's security is questionable as some security researchers from MIT Media Lab were able to break into IOTA's customized hash function "Curl" (Iota vulnerability report, 2017).

4. Progression of blockchain technology and its impact on IoT

Bitcoin blockchain has revolutionized the distributed ledger technology with its significant cryptographic security and immutability. IoT can leverage the key benefits of the blockchain (as shown in Fig. 4) to resolve its ever-growing security and privacy issues. E.g., The challenge of secure data sharing between heterogeneous IoT devices and guarantee of the trustworthiness of their data, can be met by the common blockchain platform that guarantees the immutability of data. Therefore, the blockchain, with its decentralized architecture and unforgeability, provides an ideal solution for IoT systems mostly operating in an untrusted environment.

IoT systems can also leverage blockchain technology as a secure, unforgeable and auditable log of events and TXs, as per type of the application. It can also be used to set policies, control and monitor access rights to user/sensor data and execute various actions autonomously based on pre-defined conditions using smart contracts (Buterin et al., 2014). However, in the past few years, due to IoT devices' constrained resources; storage, processing and limited power, use of cloud services has been on the rise to leverage cloud's computational and storage capabilities. But as discussed in Section 2, the cloud has its weaknesses. Therefore, it is imperative to highlight the major differences between a cloud and the blockchain.

As shown in Table 3, cloud services are provided under the centralized control of one trusted entity. Hence, the cloud is vulnerable to the single point of failure concerning security and privacy issues (Puthal et al., 2016) including data manipulation (Kshetri, 2017; Gaetani et al., 2017), and the availability of cloud services. In regard to data manipulation, the cloud service provider has to be a trusted party as it has control over the data stored in the cloud and related services. Therefore, the cloud provider can manipulate user data (Gaetani et al., 2017). Whereas, blockchain is orchestrated in a way that all the miner and full nodes in the blockchain network maintain a same copy of the blockchain state and the trust is distributed among all the network nodes. Hence, if one device's blockchain data is altered, the system will reject it, and the blockchain state will remain un-tampered. Correspondingly, the single point of failure also concerns the non-availability of the services when the cloud servers are down because of software bugs, cyber-attacks, power problems, cooling and other issues (Kshetri, 2017). Whereas in the case of the blockchain, data is replicated on many computers/nodes and problems with few nodes do not disrupt the blockchain services. The blockchain is therefore good for data security and availability. However, blockchain has a limitation that with every passing day the size of the blockchain increases, e.g., the current size of Bitcoin Blockchain is 182.8 GB (Blockchain size, 2017), and all the miner and full nodes are required to store the complete blockchain. In case of IoT this challenge is more pronounced, e.g., in a smart city IoT scenario, the sensor data coming from hundreds of thousands of IoT nodes will result in a rapid increase in the blockchain size, and the constraint resources of IoT devices concerning data storage make it difficult to handle large volumes of data. Hence, this limitation affects the utility of IoT devices as full or validating nodes in a blockchain network.

Cloud is also vulnerable to unauthorized data sharing. E.g., in the recent past, private data of 87 million users was provided by Facebook to a British political consulting firm "Cambridge Analytica" without users' permission (Sara and Michael, 2018; Granville, 2018). Such a data breach results in irreversible data security and privacy issues. Whereas, blockchain with its smart contract technology gives users the freedom to restrict access to their data to authorized entities only, without placing trust in any third party or a cloud service provider (Khan and Salah, 2018). Here a question arises how the data is stored and managed by the miners without compromising its confidentiality? In

Table 3
Cloud vs. Blockchain.

Cloud	Blockchain
Centralized architecture	Decentralized control
Trust is placed in the cloud provider	Trust is distributed in the network
Single point of failure (due to the possibility of data manipulation by the cloud provider)	Distributed architecture with blockchain state replicated on all the miner and full nodes
Vulnerable to data manipulation	Immutable
Prone to un-authorized data sharing	User-defined access control based on smart contracts
User data under control of cloud provider	Offers autonomous data sharing between users/devices through smart contracts
Users are never clear about intracloud TXs	Complete transparency by maintaining an unforgeable log of events and TXs
Not ideal for high data availability and low latency requirements of IoT	Provides edge storage and computing in terms of miner nodes that store the full copy of the blockchain
Costly infrastructure	Less expensive

this regard, a blockchain technology "Hyperledger-Fabric" follows a unique execute-order-validate architecture. To support this architecture, there are three types of nodes in the Hyperledger-Fabric based on their roles; i.e., clients, peers, and orderers. The clients submit TXs in the form of chaincodes for execution. Whereas, peers execute TX proposals for the validation and endorsement as defined by the endorsement policy. An endorsement policy states that which, and how many peers are required to endorse the correct execution of a smart contract. Finally, the ordering service nodes (orderers) establish the total order of all the TXs and output a block containing TXs. Orderers are entirely unaware of the application state, and they neither execute the TXs nor participate in the TX validation process (Androulaki et al., 2018). Hence, the execution of chaincodes by limited peers defined through endorsement policy restricts the exposure of TX payload and client ID to selected peers only. Moreover, to keep private data completely confidential from all unauthorized users, the data values within chaincode/smart contract can be encrypted, before sending TXs to the ordering service and appending blocks to the ledger (Hyperledger Fabric Model, 2017). The encrypted data written to the ledger can be decrypted only by a user in possession of the corresponding decryption key. E.g., if a user wants that his financial or health-related data should not appear in plaintext, he can encrypt the data with the public key of the user who is entitled to view that data. The user can then decrypt the ciphertext using his private key. Data can also be encrypted/decrypted using Symmetric-key algorithm such as AES. In addition to data encryption, role-based access control can also be built into the chaincode logic (Security and Access Control, 2017).

As far as issues concerning bandwidth are concerned, due to the full replication mechanism in the blockchain, every node must store a copy of all the blocks (Min et al., 2016). Moreover, the decentralized nature of the consensus process infers that nodes in the Blockchain network interact with other nodes to exchange information about the blockchain to participate in the consensus process, validate TXs, and create new blocks (Ramachandran and Krishnamachari, 2018). Therefore, Bitcoin-derived blockchain employs a gossip protocol so that all state modifications to the distributed ledger must be broadcast to all the nodes participating in the consensus process. Bitcoin blockchain being public and permissionless, any node can join the network and participate in the consensus process. Hence, there is a great likelihood that the node with the smallest available bandwidth will become the network bottleneck. Moreover, as the size of the blockchain grows, the requirements for storage, bandwidth, and compute power required for participating in the consensus process increases. Hence at some point in time, it may not be feasible for all the nodes to process a block thus leading to the risk of centralization. In a traditional cloud-based system, such a situation can be addressed, simply by adding more servers, using load balancing techniques or by increasing the bandwidth to handle the added TXs. Additionally, in the decentralized public blockchains, it is very difficult to control the public nodes (Preethi, 2017). However, in the case of private blockchains, which are mostly permissioned, only some selected nodes participate in the consensus process. Hence you have the ability to ensure that every node on the network has high computation power along with high bandwidth internet connection. (Preethi, 2017).

Moreover, due to the imminent increase in IoT devices connected to the internet, there would be an explosion in the volume of data produced by smart devices. Whereas, the existing cloud-based storage and computing solutions cannot handle such a large scale data due to the IoT requirements of high availability, real-time data delivery, scalability, security, resilience, and low latency (Sharma et al., 2018). Therefore, it is believed that blockchain due to its P-2-P distributed network architecture and state replication on all the nodes can augment the security and real-time data availability of fog nodes as an alternative to centralized cloud storage and computing (Sharma et al., 2018). However, still blockchain's scalability issue concerning the ever-increasing size need to be resolved.

Coming over to the progression in blockchain technology and the suitability of a blockchain platform for an IoT environment, we carried out a comparison (Makhdoom et al., 2018) of some of the most prominent blockchain platforms, including Bitcoin (Nakamoto, 2008), Ethereum (Buterin et al., 2014), Hyperledger-Fabric (The-Linux-Foundation, 2018) and IOTA (Popov, 2016). Although, IOTA is not as mature at the moment as compared to Ethereum and Hyperledger-Fabric but we have included it because its architecture is different than blockchain, it offers fee-less TXs, and is designed for M-2-M interactions. It also has the potential to resolve blockchain's scalability issue concerning low TX throughput with an increase in the number of network users. As shown in Table 4, the main security and performance considerations to ascertain the most suitable blockchain platform for an IoT system are as follows; the blockchain platform should provide a hybrid network concerning validating nodes' participation. As some IoT networks such as smart cities may have a large number of stakeholders willing to contribute to the security of the public blockchain network and on the other side, there may be a private network such as a smart home, where the owner would be validating the TXs via a couple of home miner/validator nodes. Currently, only Ethereum (Buterin et al., 2014) provide such a hybrid technology, whereas, Bitcoin (Nakamoto, 2008) and IOTA (Popov, 2016) support public participation. It is also imperative to mention that the level of decentralization in permissioned ledgers is affected by the lack of public access to TX validation process, as it is currently done by limited miner/validator nodes. Whereas, the limited number of validating nodes is vulnerable to malicious collusion (Garzik, 2015).

IoT systems are deployed for multiple applications, varying from smart watches to Industrial Control Systems (ICS), and again its the Ethereum and Hyperledger-Fabric that support multiple blockchain applications beyond fintech. Another important factor for an IoT system is low latency in TX confirmation which leads to the requirement of instant consensus agreement without blockchain forks. It is evident from Table 4 that Hyperledger-Fabric based on PBFT/SIEVE consensus protocols (Castro and Liskov, 2002) addresses this issue with greater reliability. Another essential aspect is that IoT systems especially the sensors operating in a smart city environment would be generating millions of TXs per day. Therefore, an ideal IoT-oriented blockchain platform should not have a TX fee or gas requirement, e.g., Hyperledger-Fabric has the option to set Tx fee or not.

The modern IoT systems not only require M-2-M micropayment methods but also need controlled access to user data, easy management of sensor policies and much more. Correspondingly, IOTA (Popov, 2016) is designed purely for M-2-M micro or even nano payments. However, currently IOTA has not yet implemented smart contracts (Buterin et al., 2014) which are essential for user-driven policy setting and access control rights. Whereas, currently the requirement of smart contracts is met by Ethereum and Hyperledger-Fabric. Another important requirement for many IoT systems sharing private data of the users is the confidentiality of data. In this regard, only Hyperledger-Fabric provides data confidentiality and also ensures the limited privacy of user data by allowing the creation of private channels (Hyperledger-fabric documentation, 2018) and encryption of data values in chaincodes (Hyperledger fabric model, 2017). Private channels are restricted messaging paths that provide TX privacy and confidentiality for specific subsets of network members. All data, including TX, member and channel information, on a channel, are invisible and inaccessible to any network members not part of that channel. Moreover, the execution of users' TXs/chaincodes for validation is not performed by all the peers. Instead, only one or more specific endorsing peers, as defined by the endorsement policy for a particular chaincode execute the TX/chaincode for validation (Androulaki et al., 2018). Hyperledger-Fabric also supports ID management and TX authorization through public-key certificates (from a trusted CA (Certificate Authority)) which are vital requirements for IoT.

Table 4
Comparison of blockchain platforms.

Ser	Features	Bitcoin	Ethereum	Hyperledger-Fabric	IOTA
1.	Fully developed	✓	✓	✓	In Transition
2.	Miner participation	Public	Public, Private, Hybrid	Private	Public
3.	Trustless operation	✓	✓	Trusted validator nodes	✓
4.	Multiple applications	Financial only	✓	✓	Currently, financial only
5.	Consensus	PoW	PoW, PoS (“Casper”)	PBFT (for deterministic TXs), SIEVE (Prototype)	Currently a coordinator approves the TXs through a Tip Selection Algorithm
6.	Consensus finality	X	X	✓	X
7.	Blockchain forks	✓	✓	X	Not exactly forks, but a tangle can be faded out later
8.	Fee less	X	X	Optional	✓
9.	Run smart contracts	X	✓	✓	X (Not presently)
10.	TX integrity and authentication	✓	✓	✓	✓
11.	Data Confidentiality	X	X	✓	X
12.	ID management	X	X	✓	X
13.	Key management	X	X	√ (through CA)	X
14.	User authentication	Digital Signatures	Digital Signatures	Based on enrolment certificates	Digital Signatures
15.	Device authentication	X	X	X	X
16.	Vulnerability to attacks	51%, linking attacks	51%	>1/3 faulty nodes	It is in Beta Testing
17.	TX throughput	7 TPS	8-9 TPS	>3500 TPS (depending upon number of endorsers, orderers and committers)	Currently, the Coordinator being the bottleneck, the throughput varies between 7 and 12 TPS
18.	Latency in single confirmation of a TX	10 min (60 min for a confirmed TX)	15–20 s	Less than Bitcoin, Ethereum & IOTA	Being in transition phase the TX confirmation time varies from minutes to hours
19.	Is it Scalable?	X	X	X	Yes (Scalability concerning unapproved/pending TXs improves with the increase in the size of the network)
20.		(Nakamoto, 2008; Bitcoin developer guide, 2017; Bitcoin.org, 2017)	(Buterin et al., 2014; Wood, 2014; James, 2018a; Etherscan, 2018)	The-Linux-Foundation (2018); Hyperledger-fabric documentation, 2018; Hyperledger Fabric, 2018; Androulaki et al. (2018); Hyperledger whitepaper, 2016; Cachin, 2016; Gas with hyperledger fabric?, 2016)	(What is iota?, 2017; Popov, 2016)

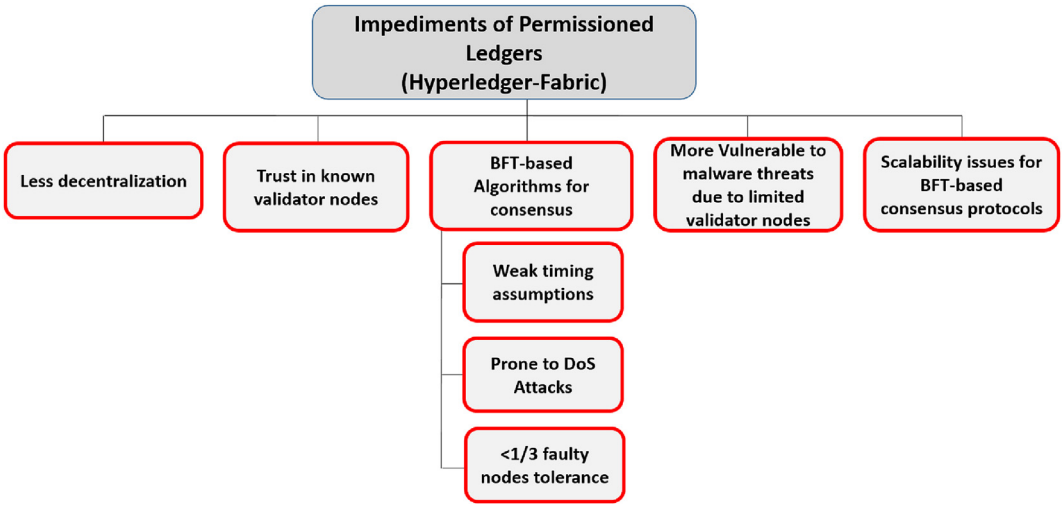

Fig. 5. Impediments of permissioned blockchains.

As far as performance is concerned Hyperledger-Fabric provides higher TX throughput than Bitcoin, Ethereum and IOTA. Hyperledger-Fabric consumes minimal energy and computation resources by using PBFT and SIEVE (a variation of PBFT) for validation of TXs i.e., low energy and computation cost (Hyperledger whitepaper, 2016). Unlike Ethereum blockchain, it does not require any gas to process the TXs. Based on a BFT-based consensus protocol Hyperledger-Fabric is a preferred technology for a permissioned ledger. However, there are some limitations in permissioned blockchains (shown in Fig. 5). Being partially-decentralized, the trust is placed in some known miner/validator nodes. Hence, in the case of a successful malware attack such as Mirai (Sophos-Naked-Security, 2016) which can infect and compromise a large number of nodes for malicious purposes, the chances of TX and block validation process in permissioned ledger to be affected are more than in a permissionless or a public ledger with a huge number of miner nodes. Moreover, the user enrolment, authentication, and authorization based on public-key certificates is currently dependent on a trusted CA, which brings some degree of centralization. However, a DKMS (Decentralized Key Management System) for Hyperledger-Fabric is under testing for release in near future (DKMS, 2018). Moreover, permissioned ledgers mostly use BFT-based consensus protocols. Whereas, such protocols are prone to DoS attacks. They can usually tolerate not more than $f = (n-1)/3$ faulty nodes. BFT-based protocols such as PBFT are believed to have high communication complexity, and they perform very poorly in adverse network conditions. Moreover, BFT-based consensus protocols have poor scalability, as the TX throughput decreases badly with an increase in the number of validator nodes, e.g., if the number of endorser nodes is increased from 1 to 14 in Hyperledger-Fabric, the TX throughput decreases to less than 1500 TPS (Scherer, 2017). However, still BFT-based protocols

Table 5
IoT requirements vs progression in blockchain technologies.

Ser	IoT Requirements	Blockchain Technology
IoT Security Requirements		
1.	Trust-free Operation	√ (All)
2.	Distributed Storage	√ (All)
3.	Decentralized Control	√ (All)
4.	Data Integrity	√ (All)
5.	Data Authentication	√ (All)
6.	Data Confidentiality/Privacy	√ (Hyperledger-Fabric)
7.	Pseudonymous IDs	√ (All - based on Pseudonymous IDs)
8.	Privacy-Preserving Computation	None
9.	User Enrolment	√ (Hyperledger-Fabric)
10.	Identity Management	√ (Hyperledger-Fabric)
11.	User Authentication	√ (All)
12.	User Authorization	√ (Hyperledger-Fabric)
13.	Key Management (Key Issuance & Revokation)	√ (Hyperledger-Fabric)
14.	Restricted Network Access	√ (Ethereum & Hyperledger-Fabric)
15.	Device Authentication	None
16.	Software Integrity Check	None
17.	Runtime/Synchronized Software Update	None
18.	Detection of Compromised Device	None
19.	IoT-centric Consensus Protocol	None
20.	IoT-focused TX Validation Rules	None
21.	Consensus Finality	√ (Hyperledger-Fabric)
22.	No Forks	√ (Hyperledger-Fabric)
IoT Performance Requirements		
1.	Autonomous System	√ (Ethereum & Hyperledger-Fabric based on Smart Contracts)
2.	Low Latency in TX Confirmation	√ (Hyperledger-Fabric)
3.	Low Communication Complexity	√ (Bitcoin, Ethereum, IOTA)
4.	Scalability	√ (IOTA - TX confirmation rate increases with the increase in network size)

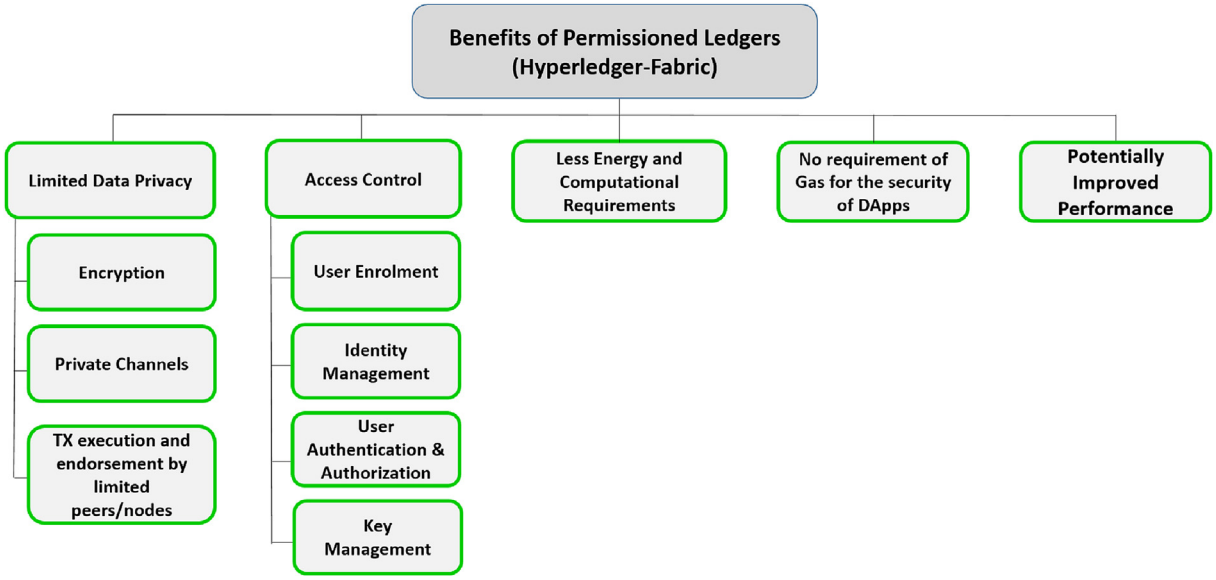

Fig. 6. Benefits of permissioned blockchains.

provide low latency and much higher throughput than permissionless blockchains.

To conclude, Table 5 presents a recap of what all IoT security and performance requirements are met by the advanced blockchain technologies and what are still outstanding. Concerning IoT security requirements, many data and user security aspects have been addressed by the blockchain platforms except privacy-preserving computation on sensitive user data, and most of the issues related to device security including device authentication, software integrity check, run-time/synchronized software update, detection of compromised device, IoT-centric consensus protocol and IoT-focused TX validation rules. As far as IoT performance requirements are concerned, some of these requirements are addressed by Hyperledger-Fabric. However, low communication complexity and scalability should also be kept in view while designing an ideal IoT-oriented consensus protocol.

Concerning suitability of an appropriate blockchain platform for IoT, as discussed in Section 3, BFT-based private/permissioned blockchains due to potentially improved performance and user security are suitable for IoT environment. Moreover, the IDs of the nodes that can control and update the shared state are known in permissioned blockchains (Cachin and Vukolic, 2017). Overall, private/permissioned blockchains offer more security and comparatively better performance than public/permissionless blockchains. The benefits of the permissioned ledger (Hyperledger-Fabric) are shown in Fig. 6. It is imperative to mention here that unlike other permissioned and even permissionless blockchains such as Ethereum, Tendermint, Quorum and Chain, Hyperledger-Fabric has a unique TX lifecycle of execute-order-validate. In which, although all peers validate the TXs to update the ledger, but not every peer executes the smart contract TXs. Hyperledger-Fabric uses endorsement policies to define which peers need to execute which TXs. This means that a given chaincode can be kept private from peers that are not part of the endorsement policy (Androulaki et al., 2018). However, it is recommended that any proposed solution should meet IoT security and performance requirements already illustrated in Section 2 and the challenges (Section 5) to blockchain's adoption in IoT.

5. Challenges to Blockchain's adoption in IoT

To identify some real issues concerning blockchain's adoption in IoT, we implemented a test case scenario of an IoT-based supply chain monitoring system (Makhdoom et al., 2018). The customer orders frozen food products and also decides a temperature threshold that has to be maintained during the shipment by the seller. An alert is generated for the customer, whenever the temperature threshold policy is violated during shipment. The test scenario and the challenges discovered while integrating IoT devices with the blockchain are explained in chronological order as labeled from 1 to 6 in Fig. 7.

1. A Rpi-3 based sensor node (scenario-1) can be connected directly to the blockchain as a full node (EthEmbedded, 2017) or a lite blockchain client (Light client protocol, 2018). A full node can validate other TXs, but a lite client can only keep a track of its own TXs.
2. The temperature sensor senses the environment and its value is extracted via a web UI (User Interface) or a mobile app (application). The web UI or mobile app connected to the blockchain node push the sensor reading to the blockchain through smart contract. Hence, a mobile or a web app is the interface between IoT devices and the blockchain.
3. In scenario-2 an IoT device can be a resource-constrained Arduino device or any other embedded system capable of just sensing and transmitting the temperature sensor readings to a gateway device.
4. The Arduino-based sensor node communicates with the gateway device through slower and less secure wireless communication media such as 802.15.4 (Gutierrez et al., 2001), 802.11 (WLAN standards) (Chen et al., 2017a), LoRa (Sinha et al., 2017), ZigBee (Ergen, 2004), NB-IoT (Sinha et al., 2017) and SigFox (Sigfox services, 2018). Resultantly, IoT systems are prone to data leakage and other privacy attacks (Jing et al., 2014). Moreover, this arrangement also limits the blockchain-based device-to-device interaction, as now only the gateway device can access the blockchain or smart contracts.
5. Just like in scenario-1, the gateway also connects to the Geth node through a web3 provider and pushes sensor data to the blockchain through a smart contract using a web or a mobile app.
6. However, there were certain challenges observed during this setup. Firstly, there is a question of how to ensure the secure input of sensor data to the blockchain? Secondly, currently, none of the blockchain platforms implement IoT-focused TX validation rules and IoT-oriented consensus protocol. Lastly, an intermediary between the sensor node and the blockchain is the UI, which cannot leverage the cryptographic security provided by the blockchain. Instead, additional device, web, and application security measures have to be taken.

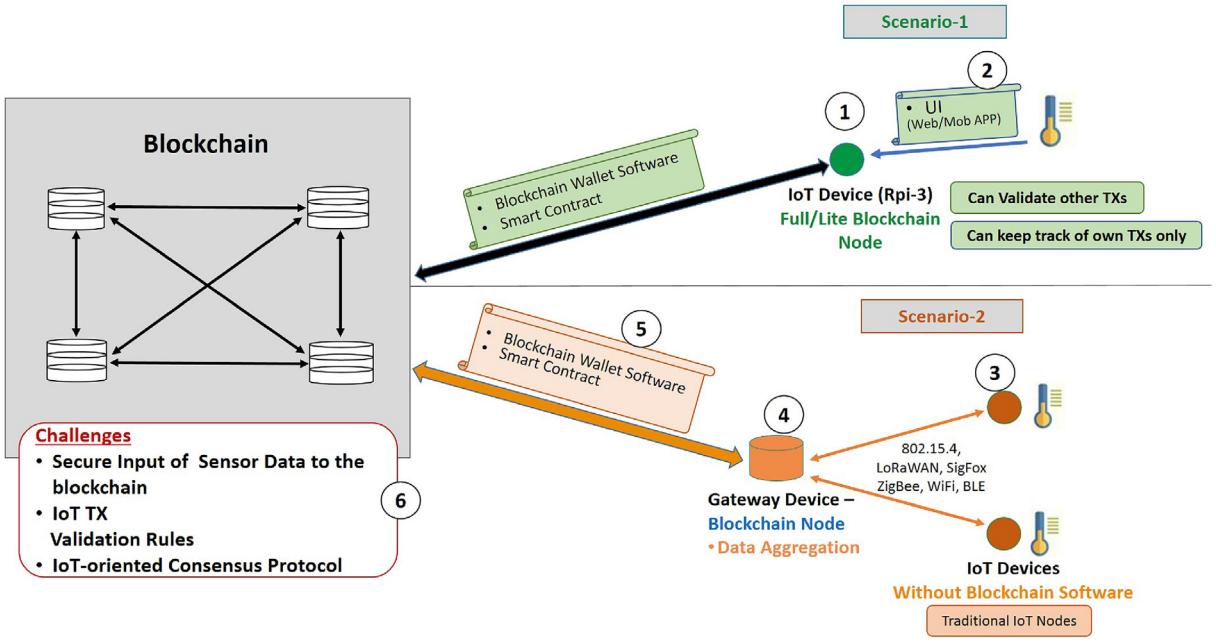

Fig. 7. Challenges for a blockchain-based IoT system.

As mentioned above, the primary challenge observed is the non-availability of an IoT centric consensus protocol. It also has some embedded issues such as TX/block validation rules, consensus finality, resistance to DoS attacks, low fault tolerance, and scalability concerning high TX volume, protection against Sybil Attack, and communication complexity. Another related issue is the secure integration of IoT devices with the blockchain. These issues are being discussed in detail in succeeding paras.

5.1. Lack of IoT-Centric consensus protocol

Fig. 8 presents a comprehensive comparison of some noteworthy blockchain consensus protocols. The points shown in green color are suitable for an IoT system whereas, points shown in red color are not appropriate for IoT. The current consensus protocols such as PoW (Nakamoto, 2008), PoS (Szabo, 2004), PoET (Kastelein, 2016a), and IOTA (Popov, 2016) are designed for permissionless blockchains, with a focus on financial value transfer. However, PoS and PoET can also be used in permissioned blockchains (Baliga, 2017). These consensus protocols share a common issue that the consensus process is probabilistic and does not end in a permanently committed block. Hence, they are prone to blockchain forks (EconoTimes, 2017). The lack of consensus finality results into delayed TX confirmation, which is not suitable for most of the real/near real-time IoT systems requiring instant TX confirmation. Moreover, PoET requires special hardware and the enclave that allocates wait time has to be the trusted entity. PoET is also proved to be vulnerable to node compromise (Chen et al., 2017b). In addition, as IOTA is currently in open-beta testing phase, it is assumed that some questions related to its security and performance efficiency will be answered in due course of time. E.g., Firstly, will it be an efficient IoT micro-payment system only? or It will also support smart contracts like in the Ethereum and Hyperledger-Fabric blockchains. Secondly, does it provide confidentiality of data? and lastly, what is the faulty node tolerance level of IOTA?

On the other hand PBFT (Castro and Liskov, 2002; Decker and Wattenhofer, 2013), DBFT (NEO.org, 2017), HoneyBadger-BFT (Miller et al., 2016) and Tendermint (Tendermint Core, 2018) are BFT-based protocols. BFT is considered to be the desired protocol for permissioned blockchains, in which ID of nodes is required to be known (Vukolić, 2015), but it also has certain drawbacks. Except for HoneyBadger-BFT, rest of the BFT-based protocols are prone to DoS attacks due to weak timing assumptions (Miller et al., 2016). Whereas, the protocols based on timing assumptions are not suitable for unreliable networks, as liveness property of weakly synchronous protocols can fail when the weak timing assumptions are violated due to malicious network adversary capable of launching DoS attacks (Miller et al., 2016).

The weak synchrony also adversely affects the throughput of such systems (Miller et al., 2016). Another major issue with BFT protocols is scalability concerning the number of validator nodes since they are not usually tested thoroughly beyond 20 nodes (Vukolić, 2015). It can be attributed to the intensive network communication which often involves as many as $O(n^2)$ messages per block (Castro and Liskov, 2002). However, Algorand (Gilad et al., 2017) claims to address the issue of scalability by randomly selecting a small set of committee members for each step of the consensus protocol. It uses Verifiable Random Functions (VRFs) for random selection of the users. It is also imperative to mention that in Algorand, the committee size is dynamic and is dependent upon two conditions, i.e., $\frac{1}{2}g + b \leq T_{step}.\tau_{step}$ and $g > T_{step}.\tau_{step}$, where, g and b is the number of honest and malicious committee members respectively, T is the number of votes needed to reach consensus and τ is the expected committee size. Concerning fault tolerance, BFT-based protocols are only capable of masking non-deterministic faults occurring on at the most $f = (n-1)/3$ replicas (Castro and Liskov, 2002). Where f is the number of faulty nodes and n is the number of total nodes.

As far as TX throughput is confirmed, BFT-based protocols can sustain tens of thousands of TXs with practically network-speed latencies (Bessani et al., 2014). Another major difference between PoW and BFT-based protocols is the notion of availability, which is a critical requirement in real-time IoT systems, i.e., PoW being an incentive-based protocol, does not guarantee that a pending TX will be included in the next block, as it is mostly at the discretion of the miners to select TXs based on their fee. Additionally, bandwidth efficiency and low communication-complexity are also critical requirements, because most of the devices in an IoT system use wireless communication protocols and a typical smart city IoT network may comprise thousands of sensors. In this regard, PBFT is considered to be an expensive protocol concerning message complexity (Luu et al., 2015). Therefore, any cur-

Consensus Protocol → / Features ↓	PoW	PoS	PoET	PBFT	DBFT	HoneyBadger-BFT	Tendermint	Algorand	IoTA
Area of use	Fintech	Multiple Applications	Multiple Applications	Multiple Applications	Multiple Applications	Fintech	Multiple Applications	Fintech	Currently for Financial value transfer
Energy costs	High	Low (as compared to PoW)	Low (as compared to PoW)	Low	Low	Low	Low	Low	Yes
Computation costs	High	Low (as compared to PoW)	Low (as compared to PoW)	High communication complexity	Low	High (As compared to other BFT protocols)	Low	Low	Low
Consensus Finality	Probabilistic	Probabilistic	Probabilistic	Instant	Instant	Instant	Instant	Instant	Probabilistic
Prone to Forks	Yes	Yes	Yes *(Baliga,2017)*	No	No	No	No	No	Yes
Latency in TX Confirmation	High	Low (as compared to PoW)	Low (as compared to PoW)	Low (Fast TX confirmation and high throughput)	Low (Fast TX confirmation and high throughput)	Low (Fast TX confirmation and high throughput)	Low (Fast TX confirmation and high throughput)	Low	Low Latency (No Fee, Parallelized Consensus)
Vulnerabilities	Prone to 51% attack	• Prone to 51% attack • Prone to malicious collusion of rich stakeholders	Node compromise *(Chen et al., 2017b)*	• Vulnerable to faulty nodes > (n-1)/3 (n = total nodes) • Vulnerable to DoS Attack • Poor Scalability concerning number of validating nodes	• Vulnerable to faulty nodes > (n-1)/3 (n = total nodes) • Vulnerable to DoS Attack • Poor Scalability concerning number of validating nodes	• Vulnerable to faulty nodes > (n-1)/3 (n = total nodes) • Poor Scalability concerning number of validating nodes	• Vulnerable to faulty nodes > (n-1)/3 (n = total nodes) • Vulnerable to DoS Attack • Poor Scalability concerning number of validating nodes	Vulnerable to dishonest nodes holding more than 2/3 of the total money	Still in B testing.
Type of Blockchain	Permissionless	Permissionless and permissioned (both)	Permissionless and Permissioned (both)	Permissioned	Permissioned	Permissioned	Permissioned and permissionless (both)	Permissionless	Currently Permissionless
Requirement of special hardware	Not essential	No	Yes, Trusted Execution Environment e.g., Intel SGX	No	No	No	No	No	No
Additional Features						Avoids DoS attack (based on timing assumption) faced by other BFT-based consensus protocols	Punishment for dishonest validating nodes	More scalable than other Byzantine agreement protocols	• Avoids Quantum Computing Attacks • Suitable for Asynchronous Networks • Improved TX throughput with the increase in network size

Fig. 8. Comparison of consensus protocols.

rent or future blockchain-based solution must be able to sustain a large number of IoT devices and comply with the regulations of wireless communications as per respective country's law (Adelantado et al., 2017). Moreover, despite reduced communication complexity and suitability for asynchronous networks, Honeybadger-BFT is not considered appropriate for IoT systems because of its cryptocurrency centric approach and low fault tolerance of $f = n/4$ faulty nodes only.

To conclude, certain aspects concerning the blockchain consensus protocols are required to be improved for its application in IoT. These aspects include IoT centric TX/block validation rules, resistance to DoS attacks (exploiting timing assumptions), increased fault tolerance (> 1/3 faulty nodes), and low communication complexity.

5.2. TX validation rules

The TX validation process in Bitcoin (shown in Fig. 9) validates a TX based on certain rules including correct TX format, valid signatures and the fact that the TX has not been previously spent (Buterin et al., 2014; Bitcoin-Developer-Guide, 2018). On the other hand (as shown in Fig. 10), Ethereum blockchain validates the format, signatures, nonce, gas, and account balance of the sender's account (Buterin et al., 2014). However, there emerges a question that can the existing TX validation rules of blockchain platforms be applied to the IoT systems? That usually comprise heterogeneous devices, thus sending sensory values or data in distinct formats and different range of values. Moreover, IoT devices are also vulnerable to cyber-attacks. Hence, a targeted or

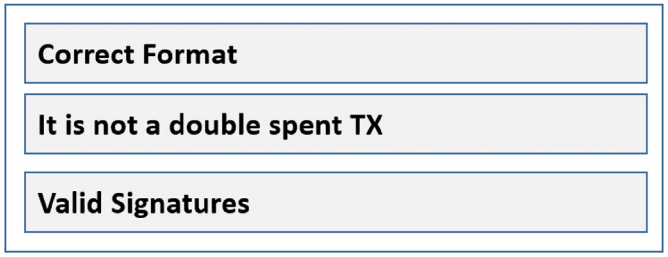

Fig. 9. Bitcoin Tx validation rules.

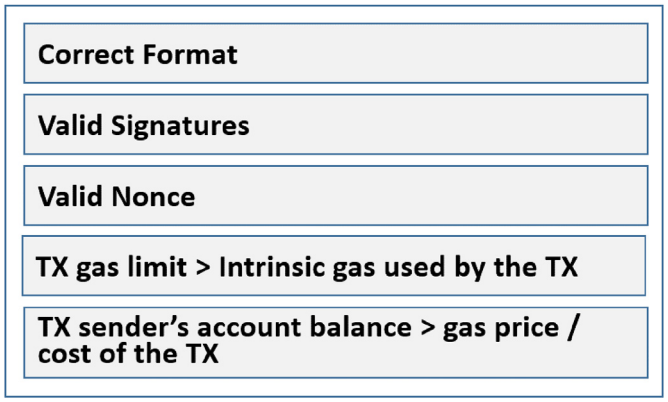

Fig. 10. Ethereum Tx validation rules.

even a generic malware attack can infect a lot of IoT devices. Subsequently, these devices may be turned into a botnet and used for further attacks. Therefore, TX validation rules of fintech-oriented Bitcoin and general purpose Ethereum blockchain may not be suitable for IoT systems (Makhdoom et al., 2018).

5.3. Scalability

It not only affects the blockchain size but also indirectly influences the consensus process. E.g., Rise in the number of users will also increase the number of TXs. Hence, if the consensus protocol has less throughput, then the latency in TX confirmation will be increased. Both the issues are being discussed separately in the succeeding paras.

Storage Capacity. A typical smart city IoT system with thousands of end nodes can generate a huge amount of data in no time. This data is then analyzed to extract information for various applications. Whereas, blockchain is not designed to store such a large amount of data. Moreover, the requirement of storing the complete blockchain by the full and miner nodes limits the integration of resource constraint IoT devices directly with the blockchain. In addition, with the continuous increase in the size of the blockchain, the storage requirements also increase thus putting more limitations on resource constraint devices to act as full or validator nodes. The increased blockchain size also takes longer to synchronize once new users/devices join the network. Therefore, it is a challenge to design a secure blockchain-based IoT solution which on one side caters for the constraint resources of IoT devices and on the other inherit maximum benefits of the blockchain.

Inherent Latency of Blockchain. The real-time data sharing requirements of most of the IoT systems like WSN, ICS, smart vehicles, intelligent transport system and smart grids, demand improvement in TX confirmation time, without compromising on the security and performance of the system.

E.g., In a PoW-based blockchain, reducing the block generation time does lessen the TX confirmation time but to achieve the same level of security as with 10 min Block time; a TX has to wait for more confirmations because of less difficulty in mining a block. Moreover, with less block time there would be more stale blocks, hence, an increase in the waste of computing and energy resources. Another factor associated with TX latency is the block size. There is a belief that by increasing the block size, say from 1 MB to 2 MB in Bitcoin blockchain, the throughput can be increased. But in reality, a bigger block will take longer to propagate in the network. Therefore, nodes with low bandwidth internet connections will suffer, and resourceful miners with more bandwidth will be at an advantage (Eyal et al., 2016). In addition, an increased block size will also result into the faster growth of blockchain size, that will affect the number of full nodes in the network, as more resources would be required to store the complete blockchain. Accordingly, Fig. 11 shows the disadvantages of having bigger blocks.

It is therefore concluded that to achieve security in a fully decentralized blockchain, there has to be a trade-off between performance efficiency and level of security, to prevent the system from bending towards centralization. As a blockchain system with a certain degree of centralized control may have some security and trust issues.

5.4. IoT device integration

In the test scenario shown in Fig. 7, the IoT devices send sensor data to the blockchain through a web UI. Same can also be done by running a JavaScript code in the shell or a mobile App. Presently, smart contracts are only supported by some of the blockchain technologies including Ethereum and Hyperledger-Fabric. Though Ethereum blockchain is currently the most tested and a reliable platform for multiple DApps (Distributed Applications), however, it has a major weakness, i.e., the smart contracts execute in EVM (Ethereum Virtual Machine) and do not communicate directly with the outside world. Therefore, the web3.js library is used as an interface.

In such a situation, the blockchain is only useful as a secure distributed database. However, before the data goes in the blockchain its integrity is dependent on the security of the device, web UI or mobile app. Keeping in view the current IoT threat scenario, in which IoT devices can easily be compromised, and malicious code can be executed remotely, the integrity of IoT devices would always be doubtful. Moreover, IoT data can also be corrupted due to some hardware/software failure or human error. Such an anomaly in sensor data cannot be detected unless the devices are tested for any hardware failure, software misconfiguration or other malicious modifications. At the moment, the only available solution is "Oraclize" (Oraclize, 2018). It extracts data from various sources including web pages, WolframAlpha, IPFS, and any secure application running on Ledger Nano S. To prove the legitimacy of data, a "Proof of Authentication" is provided along with the

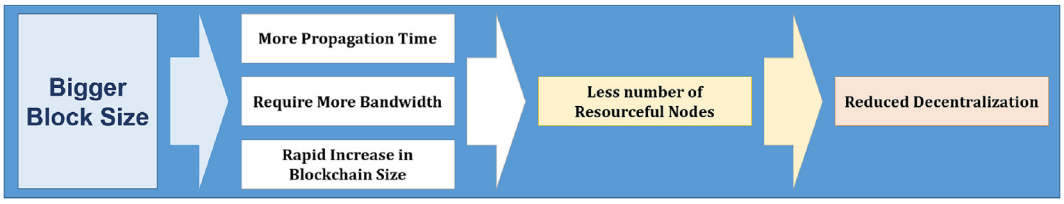

Fig. 11. Disadvantages of bigger blocks.

requested data, i.e., the proof that data has not been changed and is in its original form as obtained from the source. However, it does not support IoT devices.

Another aspect of IoT device integration with the blockchain is lack of resources to be a full node or a miner node. Full and miner nodes are required to store the complete copy of the blockchain. Hence, a direct interaction of the IoT device with the blockchain through a blockchain client software will have additional memory and computational costs. Therefore, due diligence is required for enabling IoT devices to have a wide range of interactions with the blockchain (Makhdoom et al., 2018).

5.5. Protection of IoT devices against malware/remote code execution attacks

This issue has two aspects, first is related to ransomware attacks, which has an insignificant effect in the case of a distributed ledger. Until even few nodes are unaffected, the network still has the accurate replica of the distributed ledger. However, the second aspect is that a node compromised due to malware can introduce fake/malicious data in the network. As in sensors-based IoT systems, each sensor has its unique data which is event-based and is difficult to be linked to old TXs, unlike in Bitcoin. Therefore, it would be very challenging for other nodes to validate a particular sensor data/TX. Hence, there is a requirement of malware-detection/software-attestation in a blockchain-based IoT system that can detect malicious nodes. This aspect is further linked to the availability of a runtime software/firmware update mechanism. For example, an IoT system is hit by a wiper or a ransomware attack that wipes or encrypts all data including the OS/firmware files on end devices, thus making the devices non-functional. One of the recovery mechanism would be to initiate a firmware update procedure.

5.6. Secure and synchronized software upgrade

Because of their critical functionalities, most of the IoT devices remain in continuous operation without any firmware or software updates. Hence, they are more vulnerable to cyber-attacks. Therefore, there is a need for a runtime firmware/software upgrading/updating mechanism. However, due to the decentralized architecture of the blockchain, currently, there is no mechanism to ensure synchronized software upgrade in the end devices.

5.7. Additional issues

In addition to the challenges discussed above, there are some more issues which have been identified from the literature review.

User Privacy and Data Security. As discussed in Section 4, most of the blockchain platforms keep on-chain data in plain text, where every TX can be checked, audited and traced back to the genesis block. Although, this level of transparency does help to operate in a trustless environment yet at the same time it affects users' privacy and data secrecy. Moreover, the pseudonymous IDs used by the Bitcoin blockchain do not guarantee total anonymity and thus are vulnerable to linking attacks (Conoscenti et al., 2016). Therefore, the applications running on public blockchains need additional cryptographic security, once dealing with sensitive or private user data along with some additional de-anonymization measures to de-link user ID.

Concerning, user privacy/anonymity, currently, there are many variations of Bitcoin blockchain that claim to provide anonymous TXs. For instance, Monero (Monero, 2017) ensures user anonymity by using a ring signature scheme to make the TXs untraceable. Similarly, Zerocash (Zerocoin project, 2018) let its users to convert Bitcoins into Zerocoins (anonymous coins) and thus make obscure TXs. However, it is to be well thought out that, how to ensure user anonymity on a blockchain while guaranteeing user authentication and accountability. Whereas, to ensure data privacy on the blockchain, the data can be encrypted. Correspondingly, a blockchain-based smart contract system named "Hawk" (Kosba et al., 2016) stores encrypted TXs on the blockchain. Similarly, for private blockchains, Hyperledger-Fabric (Hyperledger-fabric documentation, 2018) addresses this issue by providing support for data encryption and sharing of data using private channels. In the same way, Quorum (Quorum-white paper, 2016) makes use of cryptography and segmentation to ensure the security of sensitive data. However, still, there is a lack of blockchain-technologies that can ensure privacy-preserving computations and data analytics.

Integration of IoT Communication Protocols. There is an essential requirement for integration of IoT communication protocols such as BTLE, Bluetooth, 6LoWPAN, 802.15.4, Zigbee, LoRaWAN, etc., with blockchain for TX record, future verification and possible monetization (IBM, 2015).

6. Latest trends in blockchain-based IoT applications and related voids

Researchers and innovators around the world are developing and investigating ingenious ways to implement blockchain in IoT environment. These use cases aim to take advantage of the inherent benefits of the blockchain such as decentralized control, immutability, cryptographic security, fault tolerance, data integrity and authentication, and capability to run smart contracts. Table 6 shows some of these applications, the purpose of their development and respective blockchain platform. It is evident that not all the applications use open source blockchain platforms such as Ethereum and Hyperledger. Out of eight applications mentioned here, three applications use proprietary blockchains designed to their specific needs. Additionally, the main

Table 6
Blockchain applications.

Application	Purpose	Blockchain Platform
ADEPT (IBM, 2015)	An autonomous, robust, scalable and secure framework for IoT devices	Ethereum
Security framework for smart cities (Biswas and Muthukkumarasamy, 2016)	Blockchain-based security framework for secure communication between smart city entities	Not mentioned
Secure firmware update (Lee and Lee, 2016)	Blockchain-based IoT device secure firmware update and integrity check	Proprietary blockchain with PoW consensus
Smart home architecture (Dorri et al., 2016; Dorri et al., 2017)	Lightweight architecture of a blockchain-based smart home to control access to devices' data	Proprietary with no PoW
VANETS (Leiding et al., 2016)	Decentralized and self-managed VANET	Ethereum
eBusiness model (Zhang and Wen, 2016)	Blockchain-based autonomous sharing of data and properties	Ethereum
Transparency of SCM (Underwood, 2016; Kastelein, 2016b)	Object tracking and record of ownership	IBM blockchain based on Hyperledger-Fabric
Slock.it (Christoph, 2015)	Managing things' services through smart contracts	Ethereum
Enigma (Zyskind et al., 2015a)	Privacy-preserving data computation	Proprietary

Table 7
Main characteristics of blockchain-based IoT applications.

Characteristic	Applications								
	ADEPT	Smart Cities	Firmware Update	Smart Home	VANETS	IoT eBusiness	SCM	Slock.it	Enigma
Why is blockchain used?	Take advantage of smart contracts and network consensus	For improved reliability and better fault tolerance	To ensure data integrity, data authentication and non-repudiation during firmware verification	For distributed trust and a common platform for controlled access to IoT devices and their data	For decentralized control	To achieve a transparent self-managed and self-regulating system based-on smart contracts	Due to its unforgeability	Due to its decentralized control and ability to execute smart contracts	For decentralized control
What blockchain platform is used?	Ethereum	Not mentioned	Proprietary blockchain platform with PoW consensus	Proprietary	Ethereum	Ethereum	IBM blockchain platform	Ethereum	Proprietary
How is TX validation done?	As in Ethereum	Not mentioned	Not mentioned	Not mentioned	Not mentioned	Not mentioned	Not mentioned	Not mentioned	Not mentioned
What conventional issues are resolved?	Trust in a centralized authority/entity, single point of failure, user and data privacy issues, errors induced through human interactions.	Difficulty in sharing data received from heterogeneous devices	Mitigating the effects of cyber-attacks, avoids network congestion issues	It provides controlled access to IoT data and also ensures data confidentiality, integrity, and availability along with protection against DDoS attacks	Centralized control and privacy issues	Centralized control and issues in transparent data sharing/services	Vulnerabilities of a centralized database	Centralized control and human intervention for access control and manual handing over of the products	Data privacy during sharing and distributed computation
What blockchain issues are resolved?	Data privacy, user privacy, ID management, user-defined access control for data, and scalability	None	Scalability (related to blockchain size)	Computational intensiveness, latency in TX confirmation and energy consumption by forgoing the use of PoW in block mining	Not mentioned	Not mentioned	Not mentioned	Scalability, by reducing the number of TXs to be mined in a block	Scalability, by storing actual data on the off-chain DHT

characteristics of these applications are shown in Table 7. We have tried to highlight the answers to certain questions concerning these applications such as Why is blockchain used? What blockchain platform is used? How is TX validation done? What conventional issues are resolved? and What blockchain issues are resolved? These applications are further discussed in detail with an objective to highlight their functionality, special features, voids and any innovation or cutting-edge feature that aims to resolve some of the challenges discussed in Section 5.

6.1. Autonomous Decentralized Peer-to-Peer Telemetry

To take advantage of blockchain's ability to run smart contracts and network consensus on the validation of TXs IBM disclosed a Proof of Concept (PoC) for a blockchain-based Autonomous Decentralized Peer-to-Peer Telemetry system (ADEPT) (IBM, 2015) in 2015. Based on Ethereum blockchain, ADEPT aims to implement a decentralized, autonomous, robust, scalable and secure framework for IoT which has no single point of failure. The proposed framework uses TeleHash protocol for peer-to-peer messaging, and BitTorrent for distributed file sharing. As shown in Table 7, the proposed system aims to resolve the issues in conventional IoT networks concerning trust in a centralized authority/entity, single point of failure, user and data privacy issues, errors induced through human interactions. It also endeavors to provide data privacy, user privacy, ID management, user-defined access control for data, and scalability. Certain voids regarding its employment in IoT are:

Voids. It is a PoC and requires further testing to ensure its reliability concerning security and performance efficiency.

6.2. Blockchain-based security for smart cities

Key Features. In a conventional setting, due to non-availability of a universal standard for smart devices, there are issues related to difficulty in sharing data received from heterogeneous devices and integration of these devices to provide cross functionality. Hence, Biswas and Muthukkumarasamy in (Biswas and Muthukkumarasamy, 2016) present an overview of a blockchain-based security framework for secure communication between smart city entities. Authors claim that the integration of the blockchain with devices in the smart city will provide a shared platform where all the devices would be able to communicate securely. Moreover, the use of blockchain will prevent against data availability and data integrity attacks. It also provides an unforgeable log of TXs, that can be later used for audit purposes.

Voids. There is no qualitative or quantitative analysis of the proposed framework including computation and transmission overheads. Moreover, it is not clear that what blockchain platform, consensus protocol, and TX/block validation technique is implemented in the smart city application?

6.3. Secure firmware update

Key features. It is a blockchain-based IoT device firmware update scheme that lets the devices to securely check the firmware version and its integrity and then download the latest firmware. (Lee and Lee, 2016). This scheme vows to mitigate the effects of cyber-attacks targeting known firmware vulnerabilities. It also avoids network congestion issues, that may arise due to simultaneous firmware update/download requests by a large number of IoT devices in an IoT network with thousands of devices, deployed in a client-server model. It also aims to contain the size of the blockchain by avoiding the storage of updated firmware on the blockchain. Instead, it is done by implementing a P-2-P firmware sharing network using BitTorrent. However, it is not clear that what all messages are logged on the blockchain for auditing. If all the messages related to firmware verification are logged, then the proposed scheme does not mention that how it will manage the ever-increasing size of the blockchain?

Voids. The proposed scheme has not been evaluated for the communication complexity and energy consumption. Moreover, it is assumed that all the nodes work correctly, whereas in the actual setting any number of nodes can be compromised. It is also not stated that how does the request node extracts and pushes the model number and firmware version to a blockchain TX? Another issue is that the nodes do some PoW to reach a consensus on the firmware verification. But it is not mentioned that what measures have been taken to avoid blockchain forks?, what is the latency in TX confirmation? and how much time does a single firmware verification/update takes? It is also not mentioned that which nodes can perform PoW and which cannot? The distribution of normal nodes (resource constraint devices) and the miner nodes is also not given.

6.4. Blockchain-based smart home architecture

Key features. Ali Dorri and Raja Jurdak in (Dorri et al., 2016) and (Dorri et al., 2017) propose a secure, private and lightweight architecture of a blockchain-based smart home application. Application of blockchain in a smart home differs from a conventional Bitcoin blockchain in many ways. Unlike Bitcoin blockchain, the local blockchain in the smart home is centrally managed by its owner. It has a policy header, which also acts as an access control list that allows the owner to control all the TXs happening in his home. For device-to-device communication, the miner issues a shared key between respective devices as per policy defined by the owner. The proposed scheme provides controlled access to IoT data. It also ensures data confidentiality, integrity, and availability along with protection against DDoS attacks. It aims to solve certain blockchain issues such as computational intensiveness, latency in TX confirmation and energy consumption by forgoing the use of PoW in block mining. To reduce computational overhead, and energy consumption each block is mined without any PoW. Moreover, the latency in TX confirmation is reduced by considering a TX, true, whether it is mined in a block or not. In addition, the proposed scheme utilizes cloud storage to ease up the memory requirements for smart home devices. However, certain voids observed in this scheme are as under:

Voids. Few aspects need further explanation with reasoning. Firstly, the hallmark of blockchain is the decentralized network, whereas, in this scheme the Home-Miner, CHs (Cluster Heads) and the cloud storage are providing a single point of failure at the respective layer. Secondly, most of the blockchain platforms validate TXs and blocks on a consensus decision by all the network nodes. However, in this case, it is at the discretion of the CH, whether to retain a block or reject it. Thirdly, it is only the Home Miner that mines a block without any PoW, whereas, it is the difficulty level in PoW that protects the blockchain against double spending and data forgery attacks. Lastly, in contrary to consensus-based TX validation in usual blockchain platforms, the Home Miner checks all the incoming and outgoing TXs. Therefore, keeping in view the possibility of Byzantine General's Problem (Castro and Liskov, 1999), if the Home Miner gets corrupted or malicious, the integrity of the blockchain TXs cannot be guaranteed. The nodes use The Onion Router (TOR) for connection to the overlying network to achieve more anonymity/privacy at IP Layer. The overlay network maintains Cluster Heads (CH), that store Public Keys of the requesters, requestees and the list of TXs forwarded to other CHs. It is up to the CH, whether to keep a new block or not, whereas in Bitcoin blockchain it is a consensus decision.

6.5. Blockchain-based self-managed vehicle ad-hoc networks (VANETS)

Key features. The conventional VANETS have a centralized managing authority. This arrangement has many drawbacks from a single point of failure to present a lucrative target to the attacker. Moreover, due to centralized management, it has less user privacy. To avoid such issues, Leiding. et al. (Leiding et al., 2016) propose an

Ethereum blockchain based decentralized, self-managing VANET with a challenge-response based authentication. The complete VANET is regulated by Ethereum-based applications (smart contracts), which are used to enforce certain rules or provide different services. Each node/user is registered and identified by its Ethereum address, i.e., a hash of its public key. To access services provided by Ethereum-based applications, every node has to pay in the form of Ethers. Thereby the users fund the network infrastructure. The payment made by the users serves as the incentive for the vendors providing Ethereum-based applications and associated services. In a real-world scenario, the Ethereum account of a user can be used to make automated payments of car insurance, registration, additional services like real-time traffic update and payment of traffic violation fines.

Voids. The proposed scheme does not explain how PoW will be performed by the miner nodes to mine a block in the blockchain? There is no discussion about what information about each node will be published on the blockchain? Certain other aspects also need due considerations, like, who will mine the block? How will V-2-V (Vehicle-to-Vehicle) communication take place in the blockchain-based VANET? and what is the latency in communication? Latency is an inherent weakness in the blockchain protocol. Whereas, most of the times, the nodes/cars connected to VANET need real-time information about traffic and road conditions.

6.6. IoT eBusiness model

Key Features. In yet another venture (Zhang and Wen, 2016), Yu Zhang and Jiangtao Wen propose a blockchain-based decentralized electronic business model for the IoT. The proposed model aims to share paid data and smart properties like a car, parking space, house, fuel, e-shopping, commodities, and services, by applying the concept of Decentralized Autonomous Corporations (DAC). The key idea here is that DAC is automated without any intervention by humans and make use of smart contracts for decision making. It enables rapid information exchange among all stakeholders, i.e., sensors, computers, humans, DACs, buyers, sellers, etc. Moreover, each device in IoT can serve as a service provider. The proposed model has been designed and developed by modifying and optimizing basic elements and operating modes of the conventional e-commerce system. The efficiency is increased by removing the third party, working in low trust environment and reducing latency.

The DAC model can be deployed for each smart device/sensor to trade it's paid data for some service like power, additional module and software up-gradation, etc. The authors implemented the test case of the proposed model using Ethereum blockchain and aim to further develop an automated transfer of ownership service for smart properties.

Voids. Although authors gave a detailed overview and insight into their proposed e-business model for IoT, yet it was not clear, how the constraint resources of IoT devices like less computational power, small memory, and low energy consumption will be met? The proposed solution mostly focused on the working of e-business model, so there is a lack of discussion on technical aspects. Hence, details like, which are the miner nodes? What data from the blockchain will be stored on IoT devices? What are the security measures used to protect against device compromise and how devices are integrated with the blockchain? need more deliberation.

6.7. Transparency of supply chain management (SCM)

Key Features. The blockchain is an ideal platform to ensure product authenticity and transparency during its complete supply chain cycle. It will help in tracking the origin and the transformations undergone by a product in the supply chain by maintaining a formal registry. The digital ledger can be connected to a supply chain sensor network connecting cargo trucks, storage coolers, etc., to keep track of product location and its environment parameters like temperature and humidity (Reid, 2015).

In a similar endeavor, Everledger, a UK-based global startup has launched a Global Digital Ledger based on IBM Bluemix (Michael and Buell, 2018) to digitally certify diamonds to assist in the prevention of fraud. The digital ledger stores complete data about diamonds including their ownership and TX history. The immutable ledger will support owners, insurance companies, banks and law enforcement agencies to verify the complete life cycle of a diamond since its discovery in the mine until its sale in the market and subsequent ownership. Till date, Everledger has certified more than 1 million diamonds. The company has not disclosed any technical details about Everledger. However, it claims to use a hybrid blockchain model to take advantage of permissioned controls as in the private blockchains (Wüst and Gervais, 2017). The company is also aiming to apply the same solution for the security of fine arts, vintage cars and wine (Underwood, 2016; Kastelein, 2016b).

Voids. Irrespective of practical manifestation of the blockchain in SCM, there is an inherent issue of interfacing blockchain and different types of physical devices. Moreover, there are questions related to the status update regarding location and condition of a product in transit to a customer. Which is currently done manually by a human or by a sensing device. Now in a distributed environment, no other sensor node knows about the exact condition of this product once it has reached the warehouse, except the node reporting upon it. Therefore, there has to be some element of trust in that sensor node, such that its input data is accepted in the blockchain. Hence, if all the nodes are trusted, then there is no need of a blockchain. Moreover, if there is no trust, then the complete supply chain is compromised, and any malicious node can inject false data (Wüst and Gervais, 2017).

6.8. Managing things' services through smart contracts

Key Features. To exploit blockchain's ability to run smart contracts, "Slock.it" was developed as a commercial product (Prisco, 2015). It is a smart lock called Slock, which is controlled through smart contracts on Ethereum blockchain. In practice, the slock can be any smart device available for rent such as bike, car, computer, etc. Conventional smart devices are controlled by an app (application) for a pre-defined purpose. However, using smart devices through the blockchain gives the users unlimited options and use cases such as renting out rooms, cars, bikes, electronic appliances, and parking facilities. The founder of "Slock.it" in (Christoph, 2015) demonstrates the complete process of renting a slock. The perceived working of Slock.it is shown in Fig. 12. Firstly, the owner registers its slock/item for rent, on the app provided by the blockchain service provider. As soon as the owner registers his device, the device gets a private/public key pair in the smart contract. The owner then sets the deposit amount (same as security) and the cost per minute/hour/day for a particular slock/item.

On the other side, when the client wants to rent a service/slock, the client just selects the desired item/slock and then clicks the rent-it button to sign the contract. The client can also see the amount required to be deposited and the cost per minute/hour for the said service. As soon as the customer clicks rent-it, a TX is initiated on the blockchain. The TX confirmation can take some time equivalent to 1 or 2 Blocks generation period depending upon the settings of the service provider. Once the TX is confirmed, the client can click the open option and access the service. When the customer has used the service, he can terminate the service by clicking the close button on the app. As soon as the service is closed, a TX is initiated on the blockchain, and the client gets his balance money (Balance = Deposit - the cost of service) through smart contract.

The slocks/smart-devices are integrated with a blockchain-based smart contract hosted on a single or distributed blockchain servers, through embedded devices running a blockchain client software. The embedded device can be a Rpi, an Intel Edison, Samsung Artik-5 or any

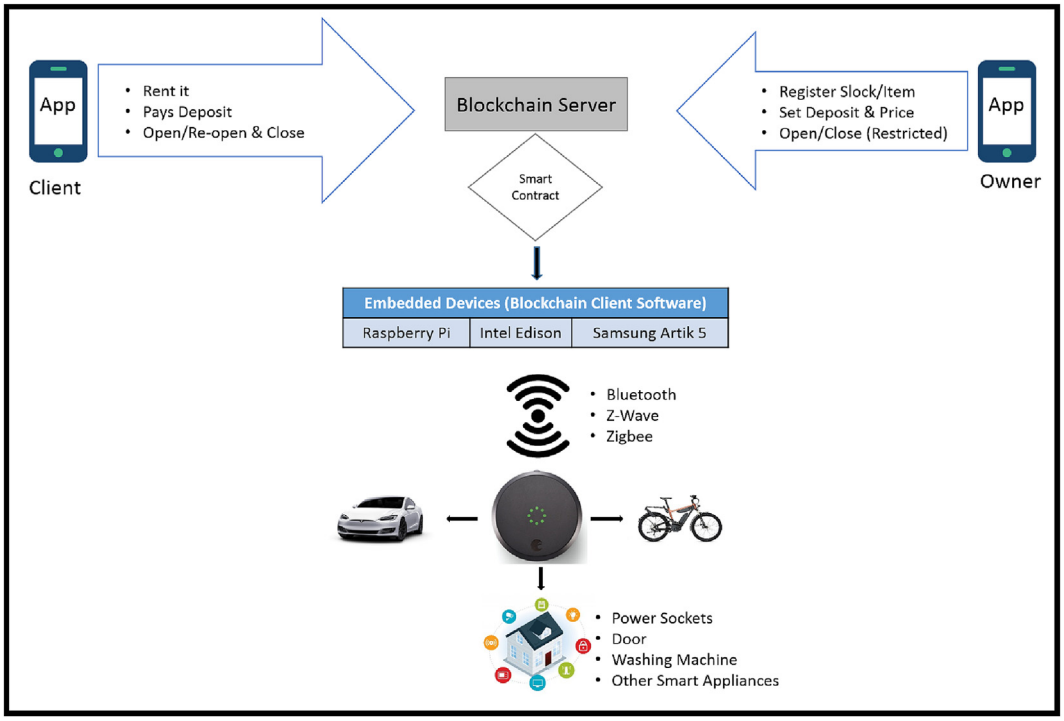

Fig. 12. Managing IoT Device Services using Smart Contracts.

other SoC (System on Chip) solution. The blockchain client communicates with smart devices/slocks through Bluetooth, Z-Wave, ZigBee or any other communication protocol supported by the service provider. Considering the scalability factor, only initial open and last close TXs are recorded in the blockchain. Rest of the open and close TXs during usage of the rented service/slock are termed as whisper messages and are not stored in the blockchain. However, these messages are verified through the private/public key of the client. The scalability issue can be managed differently depending upon the system architecture and the type of devices being used.

Voids. Apart from inherent Ethereum blockchain benefits, Slack.it mostly focuses on the functionality of the product. It is not mentioned that what security measures are taken to ensure device security.

6.9. Security and privacy of data

Considerable work has been done to ensure the privacy of user data on the blockchain-based networks. A data management system for decentralized networks has been proposed in (Zyskind et al., 2015b). It protects against issues related to data transparency and auditability, data ownership and access control. Moreover, Viral Communications, MIT Media Lab has developed Ethos, a Bitcoin-like network for secure sharing of personal data (MIT-Media-Lab, 2014). However, suitability of Ethos for its application in an IoT system still requires deep assessment. In addition to this, a privacy-preserving decentralized computation platform named Enigma (Zyskind et al., 2015a) has been proposed. It ensures confidentiality of data by implementing secure multi-party computation guaranteed by verifiable secret sharing scheme. Enigma restricts access to complete data by all the nodes, i.e., every node has a secret share of data, and it performs computations on that particular share without leaking information to the other nodes. Such an arrangement decreases memory requirement for embedded devices, and the distributed storage enables performance of more intense computations on data.

Voids. Although, the idea of decentralized computation in Enigma seems feasible, yet the computation and communication overhead is required to be analyzed for its efficient implementation in an IoT system. Since most of the IoT end devices like sensor nodes, communicate using wireless media. Any current or future solution for secure data sharing and distributed computing must comply with the regulations of wireless communications as per respective country's law. The distributed computation schemes like multi-party secret sharing schemes (Zyskind et al., 2015a), seems very efficient but their efficacy regarding bandwidth/channel utilization needs to be assessed. E.g., In Europe for LoRaWAN protocol that operates on the 868 MHz frequency band, the allowable duty cycle is 1% for each user/device (Adelantado et al., 2017). Hence, any blockchain-based secure data sharing platform for IoT systems should cater for such limitations.

7. Gap analysis

In spite of inherent benefits of the blockchain, i.e., TX integrity, TX authentication, non-repudiation, an auditable log of events, etc., there are numerous challenges (highlighted in Section 5), that needs due consideration for a secure adoption of blockchain in IoT. Further elaborating on these issues, firstly, the current consensus protocols such as PoW, PoS, PoET, IOTA, PoA, and Proof of Activity are designed for public blockchains (PoS and PoET also support permissioned blockchains) in which the miner is selected based on some lottery scheme. Thus, a block is mined by the lottery winner without network consensus. The previous block is confirmed only, once the next miner and the subsequent other miners extend the chain. Hence, these protocols lack instant consensus finality and are prone to blockchain forks. As far as BFT-based consensus protocols are concerned, although they do provide consensus finality and avoid forks along with low latency in TX confirmation, yet they are prone to DoS attacks. Moreover, with an increase in the number of replicating/validator nodes, the communication complexity also increases. On the other hand, IOTA provides low latency in initial TX approval. However, it is currently not determined that after how much time and indirect approvals the TX stands confirmed. This

is an important aspect in near-realtime IoT service management, such as toll payment by the smart car, payment for gas, parking fee, etc. Hence, IoT-centric consensus protocol is required to be designed and developed duly considering factors such as IoT centric TX/block validation rules, resistance to DoS attacks (exploiting timing assumptions), increased fault tolerance (>1/3 faulty nodes), consensus finality and low communication complexity.

If we look at the blockchain-based IoT applications, discussed in Section 6, Table 8 shows a synthesis matrix, that pitches the challenges identified (Section 5) against the blockchain-based IoT applications. It is evident that most of the challenges are not tackled by any of the blockchain applications. In this regard, the foremost issues are lack of IoT-focused consensus protocol and TX validation rules followed by secure device integration and secure firmware update. Only two applications, i.e., firmware update and smart home mention consensus protocol. In that firmware update application only comments that it uses PoW consensus for firmware verification. However, no further details are given as to how it manages PoW's computation and energy costs and latency in TX processing? It also does not comment about any distinction between the miner and normal nodes. On the other side, the smart home application uses a proprietary blockchain platform and does not use PoW consensus protocol because of its high computation and energy costs and latency in TX confirmation. However, the proposed scheme does not mention that how it selects miners for subsequent block mining? Currently, it seems that only the smart home miner mines the block for all the devices in a particular house, which is against the trust-free and decentralized architecture of the blockchain. Rest of the applications do not discuss any issue related to consensus protocols.

The third hitch is regarding the scalability of the blockchain. Only four applications, i.e., ADEPT, secure firmware update, smart home, and Slock.it address this issue. Generally, scalability can be interpreted in terms of the size of the blockchain and latency in TX confirmation concerning network expansion. A typical IoT system, e.g., smart city environment monitoring system may comprise thousands of embedded devices with limited memory and power resources. The constraint resources cannot store the ever-increasing size of the blockchain, which is required to maintain a full node. Hence, this aspect limits the number of full nodes in the network. However, if there are less full nodes with mining capabilities, then it means the workload of mining TXs will be on limited mining nodes, which may create a bottleneck and result into high latency in TX confirmation. Therefore, due diligence is required in resolving the issue of scalability, as this limitation has a significant impact on the design of blockchain-based IoT systems.

The fourth issue is of secure IoT device integration with the blockchain. None of the applications brace this problem. Therefore, there is a need to design and develop a method to securely interface IoT devices with the blockchain such that the data from heterogeneous IoT devices can be directly sent to the blockchain. It is also essential to ensure the integrity of IoT devices for correct operation in a trustless environment, without the use of any additional hardware, e.g., trusted platform modules. The factor of secure hardware is specifically mentioned here, as in practice manufacturers reduce the cost of IoT devices such as CCTV cameras, embedded sensor modules, smart watches, smart TV, etc., by cutting investment on security hardware/features and just focusing on the application features.

Protection against malware attacks and runtime firmware/software upgrade is another lacking area. Although, authors in (Lee and Lee, 2016) propose a blockchain-based firmware update procedure. However, the proposed scheme does not protect against node compromise attacks in which node hardware configuration is changed to allow for back-door access later. Hence, an attacker can install malicious code in the memory of a node to launch further attacks on the network like espionage and DoS by initiating unnecessary network traffic to target legitimate users/applications.

Table 8
Gap analysis.

Challenges	Applications								
	ADEPT	Smart Cities	Firmware Update	Smart Home	VANETS	IoT eBusiness	SCM	Slock.it	Enigma
IoT centric consensus protocol	X	X	X	X	X	X	X	X	X
IoT focused TX validation rules	X	X	X	X	X	X	X	X	X
Scalability	Yes	X	Yes (By not storing firmware files on the blockchain)	Yes (By storing device data on cloud storage)	X	X	X	Yes (By limiting the number of TXs to be mined in a block.)	X
Secure device integration	X	X	X	X	X	X	X	X	X
Protection against device compromise	X	X	X	Yes (By limiting outward data flow from the devices)	X	X	X	X	X
Secure firmware update	X	X	Yes	X	X	X	X	X	X
Data Security	Yes	X	Yes	Yes	X	X	X		Yes
Privacy-preserving computation	X	X	X	X	X	X	X	X	Yes

Although most of the applications do not consider or need data security in the form of data encryption. However, it is no more an unaddressed issue as the blockchain-platforms such as Hyperledger-Fabric and IBM ADEPT provide data confidentiality and data privacy.

Another important predicament is related to privacy of sensitive data. In a blockchain-based distributed system, preserving the privacy of sensitive user data such as financial information, health data, personal/house security data, during distributed processing is still a big challenge. The distributed computation scheme Enigma (Zyskind et al., 2015a), seems very efficient but its efficacy regarding bandwidth/channel usage needs to be assessed. Hence, any future solution should also cater for computation/transmission overheads and bandwidth utilization.

8. A way forward

8.1. IoT-centric consensus protocol and transaction validation rules

The design and development of an ideal consensus protocol for an IoT environment demands that the requirements of a consensus protocol for a blockchain-based IoT system be distinguished from existing general purpose and cryptocurrency oriented consensus protocols. Some of these requirements are shown in Fig. 13. The points mentioned in blue color are concerning security/consistency and the points shown in the green color pertains to the performance requirements. The foremost requirement for IoT systems is that the TXs should be validated based on IoT-centric TX validation rules. It is an essential requirement since every new TX in IoT is mostly independent of the previous TX and an incident or change in environmental conditions can influence the change in the sensor readings. Therefore, IoT TX validation rules should be carefully drafted and they must incorporate environmental context, e.g., in a smart home, the fireplace is ignited, only if the camera or any other sensor also detects the presence of a human in that room. It means a sensor reading is validated based on the environmental context and not in isolation. The consensus protocol should also be robust against Sybil Attack and must have consensus finality to avoid forks. Other than avoiding forks, consensus finality is equally vital for achieving minimum latency in TX confirmation and the ultimate high TX throughput.

Moreover, IoT systems are also vulnerable to physical or cyber-attacks. In recent past, a cyber-attack named “Mirai” (Sophos-Naked-Security, 2016), infected a large number of IoT devices including DVR and CCTV cameras and turned these devices into bots. The compromised devices were then used to launch a DDoS attack on a DNS service provider “DYN” by directing huge data traffic in the form of millions of DNS lookup requests. Whereas, if we look at the BFT-based protocols, most of them can only tolerate $f = (n-1)/3$ faulty nodes. Therefore, an IoT-centric consensus protocol must have the capability to sustain maximum possible faulty/dishonest nodes. An important consideration to lessen the effect of faulty nodes is to carry out random integrity check of the validator/mining nodes so that no dishonest node participates in the consensus process (Makhdoom et al., 2018). In addition to the security requirements, there are some performance considerations as well. These include low computation overhead, low energy consumption, and less communication complexity.

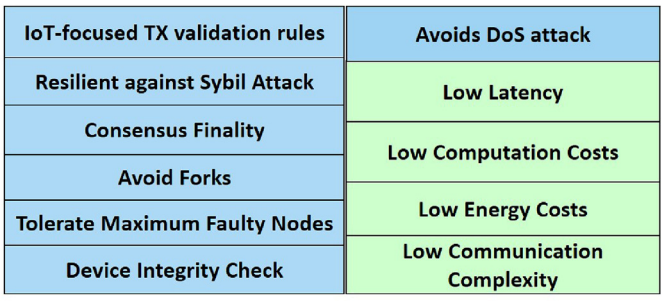

Fig. 13. Considerations for IoT consensus protocol.

8.2. Managing blockchain size

To address the issue of scalability concerning the management of ever-increasing blockchain size on light/embedded IoT devices, various blockchain architectures are being proposed such as sidechains and treechains. An example of a sidechain is a decentralized P-2-P network designed for multi-party privacy-preserving data storage and processing (Zyskind et al., 2015b; Zyskind et al., 2015a). The proposed model implicitly improves the issue of blockchain scalability by storing user data on an off-chain network of private nodes in the form of DHT (Li, 2006). The blockchain only contains the pointers/references to data, and not all the nodes replicate all TXs.

IBM (IBM, 2015) also addresses the issue of blockchain size by introducing a concept of universal and regional blockchains. It is achieved by categorizing the network nodes into light peers, standard peers and peer exchanges depending upon their processing, storage, networking, and power capabilities. The light peers consist of embedded devices, such as Arduino and Rpi-based sensor nodes. These nodes only store own blockchain address and balance and rely on other trusted peers to obtain TXs relevant to them. Whereas, the standard peers have more processing power and storage capacity than the light peers. They can store some of the recent TXs of their own and the light peers in their neighborhood. Finally, the peer exchanges have high storage and computing capabilities, and they can replicate complete blockchain data with an additional feature of data analytic services. In addition, as per NIST (Konstantinos et al., 2016), resource-constrained devices may maintain a compressed ledger containing only their TXs.

Authors in (Gaetani et al., 2017) and (Aniello et al., 2017) also propose a scalable 2-layer blockchain architecture to log distributed database TXs. The first layer represents a permissioned blockchain comprising a miner each from respective federation members. The miners in layer one are selected randomly based on a fast consensus protocol. The hash of the first layer blockchain is periodically stored on the second layer using PoW to ensure the integrity of the hashes. Hence, if a malicious node alters the log in the first blockchain, the hash of the data would be different as in the second layer. Hence, forgery can easily be detected. To achieve scalability in the proposed scheme especially layer-1, these authors propose data sharding, in which every miner maintains a DHT-based ledger on the basis of keyspace partitioning and only handles TXs for specific subsets of keys. Thus tuning TX load on miners and making the system more scalable.

Another solution proposed for the scalability of Ethereum blockchain is called “Plasma” (Poon and Buterin, 2017). It uses a series of smart contracts to create hierarchical trees of sidechains, which can be thought of as “subchains”. The subchains live within a parent blockchain and periodically communicate with the root-chain (Ethereum). The subchains are off-line, hence, theoretically, there can be as many subchains as desired (REX-Blog, 2017). Similarly, BigchainDB (Bigchaindb, 2018) introduces a blockchain database that utilizes the benefits of both, the blockchain and the big data distributed database. It integrates the immutability and decentralization of the blockchain with the high throughput and fast TX settlement time of big data distributed database.

8.3. Improving upon TX confirmation time

TX confirmation time can also be associated with the problem of blockchain scalability. In current public blockchains such as Bitcoin and Ethereum, the miner nodes are required to store the complete blockchain and validate every TX in an order. This arrangement does help in ensuring the security of the system but can also be prone to bottlenecks in case of high TX volume. Since the blockchain cannot process

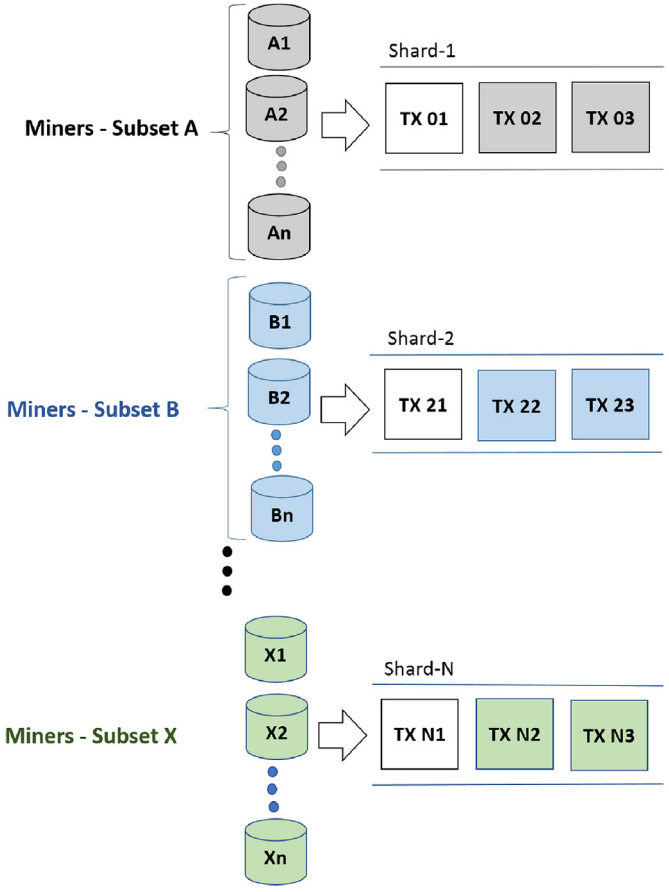

Fig. 14. Sharding.

more TXs than a single node can. One of the methods being researched to reduce TX confirmation time is "Sharding" (James, 2018b). It means a subset of miner nodes process a subset of TXs (as shown in Fig. 14). The subset of miner nodes should be populated in a way that the system is still secure, and at the same time, several TXs can be processed simultaneously (REX-Blog, 2017; James, 2018b). In its purest form, each shard has its own TX history, and it is affected only by the TXs it contains. E.g., In a multi-asset blockchain, there are n shards, and each shard is associated with one particular asset. In more advanced forms of sharding, TXs on one shard can also trigger events on some other shard. This is usually termed as cross-shard communication. However, currently being in a novice state, there are numerous challenges that should be resolved before sharding is adopted publicly. Some of these challenges include; cross-shard communication, fraud detection, single-shard manipulation, and data availability attacks (James, 2018b).

Another approach to reducing TX processing time is "Raiden". It proposes the use of state channel technology to scale the Ethereum network off-chain and to facilitate micro-TXs between IoT devices (REX-Blog, 2017). The off-chain TXs will allow a set of nodes to establish payment channels between each other, without directly transacting with the Ethereum blockchain. Hence, Off-chain TXs would be faster and cheaper than on-chain TXs because they can be recorded immediately, and there is no need to wait for block confirmations. However, it is believed that Channel-based strategies can scale TX capacity only but cannot scale state-storage. Moreover, they are vulnerable to DoS attacks (James, 2018b).

In another development, to address Bitcoin blockchain's problems of scalability, high TX fee and requirement of substantial hardware resources, a blockless architecture named "IOTA" have been introduced (What is iota?, 2017). IOTA is a distributed architecture based on DAG called Tangle (Popov, 2016), instead of a conventional blockchain. It aims to promote machine economy, in which smart devices can interact with each other by making smallest possible, nano-payments. To ensure fast TXs, IOTA does not require TX fee. Moreover, the consensus (TX validation) and normal TX process are also inter-knitted, i.e., before making a new TX, each user randomly approves/validates previous two TXs. IOTA achieves high throughput by parallelizing the TX validation process. Hence, an increase in the number of new TXs on the Tangle is inversely proportional to the TX settlement time (An introduction to iota, 2017). Therefore, an expanding network contributes well to the overall security and fast TX settlement. The two TXs to be approved by every new TX are randomly selected based on MCMC (Markov Chain Monte Carlo) method. A TX getting more and more direct/indirect approvals is considered to be more accepted by the network. Hence, it would be difficult for anyone to double-spend that particular TX. The difference between IOTA and a typical blockchain architecture is shown in Fig. 15 (An introduction to iota, 2017).

8.4. Secure IoT device integration with the blockchain

In addition to securing the web UI and mobile App, IoT device integration with the blockchain can be augmented by device enrolment, in which only approved devices be allowed to communicate with the blockchain and call smart contract methods. Correspondingly, smart contracts can restrict access to selected methods to a specific node only. Concerning the physical security of IoT devices, all the unnecessary ports such as JTAG and UART should be blocked. Since any open port can be used by an adversary to access the device and make malicious changes. Moreover, most of the commercially available IoT devices such as sensing devices do not have a secure execution environment due to cost effects. Therefore, the device integrity check should frequently be performed to ensure its legitimacy (Makhdoom et al., 2018).

As of today, most of the IoT systems depend on a certain cloud platform due to computational and storage scarcity and because of the same, resource constraints IoT devices cannot be used as a miner or full nodes in a blockchain network. Hence, to ensure a smooth transition from cloud to blockchain based network, IoT systems can leverage Fog Computing components that already follow some degree of distribution and are more resourceful than IoT devices. The Fog nodes can function as blockchain miners and can facilitate direct interaction

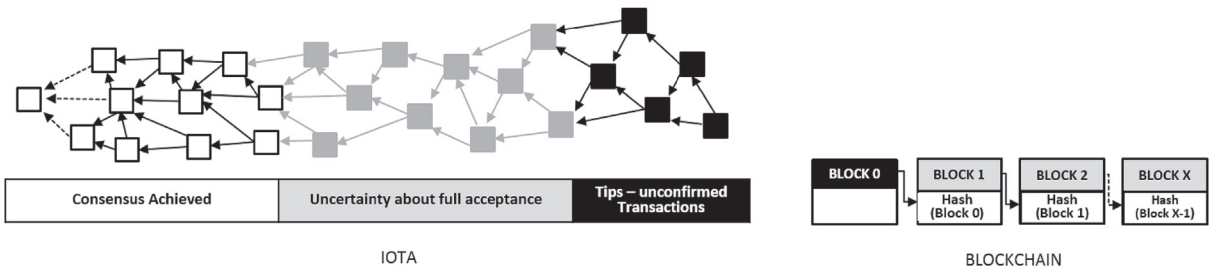

Fig. 15. IOTA vs. Blockchain.

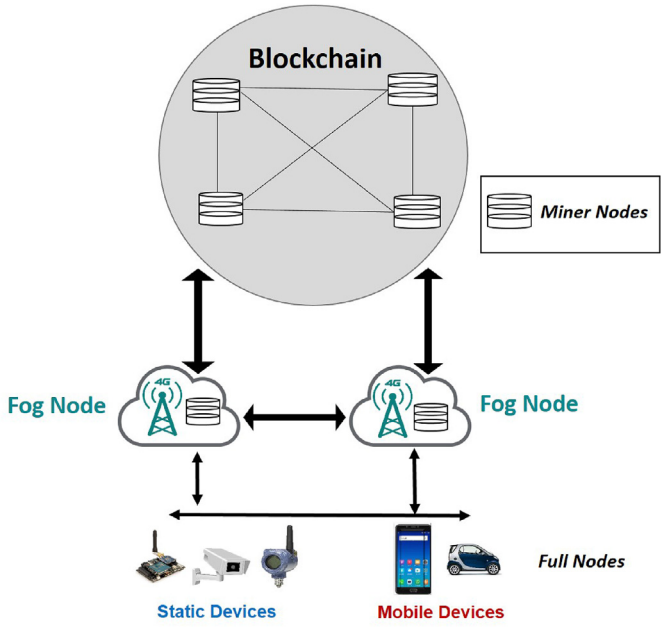

Fig. 16. Blockchain and IoT Integration using Fog Nodes.

between IoT devices and the blockchain. E.g., As shown in Fig. 16, the fog nodes can incorporate blockchain miner nodes to collect and mine the TXs received from the IoT devices in a block. The IoT devices have enough resources to be the full nodes. Hence, they can store the blockchain and also route and validate the TXs. In this way, most of the TXs from the IoT devices would be propagated to both the fog nodes. Hence, IoT can leverage existing fog computing infrastructure to adopt blockchain technology, until IoT devices are manufactured with embedded blockchain mining functionality to gain on the maximum benefits of blockchain's distributed architecture.

8.5. Integration of IoT communication protocols with the blockchain

To integrate blockchain protocols with the communication layer of IoT (IBM, 2015), and (Biswas and Muthukkumarasamy, 2016) recommend the use of TeleHash as the messaging protocol, which is based on Kademlia DHT (Zyskind et al., 2015b). It is a lightweight and a secure P-2-P network protocol that uses encryption for secure mesh communication across multiple platforms (Telehash, 2017). TX records can be converted into blocks and further broadcast into the blockchain network.

8.6. Resolution of Bitcoin Blockchain's limitations

Till now, we have analyzed every aspect of the blockchain, from its basic concepts to the advancements in blockchain platforms, related challenges and latest trends in blockchain-based IoT applications. However, we feel it important to present a consolidated gist of the evolution of blockchain technology that aims at mitigating Bitcoin blockchain's limitations. This summary will help blockchain and IoT researchers to understand related technologies and find their way forward to resolve blockchain-based IoT issues. Hence, Table 9 pitches Bitcoin blockchain's limitations and vulnerabilities against requisite blockchain technologies and applications that promise to abate respective limitations.

9. Conclusion and future work

No doubt, IoT is the future of an autonomous digitized economy of the world by liquefying and personalizing the physical objects (Brody and Pureswaran, 2014). However, to achieve this status, it has to undergo a conceptual transformation both at the design and the development stages. That day is not far off, once machines will interact with machines without human intervention to achieve performance efficiency, durability, operational effectiveness, and financial economy. Therefore, it is imperative to design and develop a secure blockchain-based IoT system that meets the future requirements of an autonomous digital world. The future IoT system should be compatible with existing IoT technologies so that the transformation from a traditional centralized architecture to a self-maintained decentralized system is economically feasible. Moreover, performance aspects should also be given due consideration, in parallel to the security issues. Hence, in this paper, we initially introduced the IoT threat environment, resultant security and performance requirements for IoT systems and key blockchain concepts. Then, we analyzed the impact of blockchain technology on IoT followed by identification of challenges to blockchain's endorsement for IoT. Later, we reviewed various blockchain-based IoT applications to highlight the trends in IoT applications and the blockchain issues resolved by these applications. In the end, we carried out the gap analysis and

Table 9
Resolution of bitcoin blockchain limitations.

Bitcoin Blockchain Limitations	Advancement in Blockchain Platforms/Applications/Technologies
Energy and computation intensive PoW consensus	PoS (Szabo, 2004), PoET (Kastelein, 2016a), PoB (Iain, 2018), PoA (Ethcore, 2018; Proof of authority, 2017), BFT-based consensus protocols (Miller et al., 2016; Lamport, 1978; Schneider, 1990; NEO.org, 2017; Erik, 2017)
Lack of consensus finality and forks	BFT-based consensus protocols (Miller et al., 2016; Lamport, 1978; Schneider, 1990; NEO.org, 2017; Erik, 2017)
Latency in TX confirmation	Ethereum (GHOST, Casper) (Buterin et al., 2014), Hyperledger-Fabric (PBFT, SIEVE) (Hyperledger-fabric documentation, 2018), Bitcoin-NG (Eyal et al., 2016), and BFT-based blockchains (NEO.org, 2017)
Low Throughput	BFT-based blockchains (Multichain (Gideon, 2015), Hyperledger-Fabric (Hyperledger-fabric documentation, 2018))
De-anonymization (Linking attacks) (Conoscenti et al., 2016)	Monero (Monero, 2017), Zerocash (Zerocoin project, 2018),
Scalability (Size of blockchain)	Universal and regional blockchains (IBM) (IBM, 2015), Sidechains (Zyskind et al., 2015b; Poon and Buterin, 2017), Data compression (NIST) (Konstantinos et al., 2016), Scalable blockchain architecture (Gaetani et al., 2017; Aniello et al., 2017), BigchainDB (Bigchaindb, 2018)
51% attack (Yli-Huumo et al., 2016), Double-spending (Nakamoto, 2008; Armknecht et al., 2015)	BFT-based consensus protocols (Miller et al., 2016; Lamport, 1978; Schneider, 1990; NEO.org, 2017; Erik, 2017)
No runtime firmware/software update	Secure firmware upgrade (Lee and Lee, 2016), Gitar (Ruckebusch et al., 2016), RemoWare (Taherkordi et al., 2013)
Data privacy	Multichain (Gideon, 2015), Quorum (Quorum - white paper, 2016), Hyperledger-Fabric (Hyperledger-fabric documentation, 2018), Hawk (Kosba et al., 2016), DHT (Li, 2006)
Privacy-preserving computation	Enigma (Zyskind et al., 2015a), Homomorphic encryption (Carpov et al., 2016)
Limited scripting	Smart contracts supported by Ethereum (Buterin et al., 2014), Hyperledger-Fabric (Hyperledger-fabric documentation, 2018)
Legal issues in smart contracts	Alastria (Alastria, 2017) (Idea of a national regulated blockchain)
Public/Permissionless blockchain	Private/Permissioned blockchains Ethereum (Buterin et al., 2014), Multichain (Gideon, 2015), Quorum (Quorum - white paper, 2016), Hyperledger-Fabric (Hyperledger-fabric documentation, 2018)

recommended a way forward to resolve some of the significant challenges that hinder the adoption of the blockchain in IoT environment.

In future work, the authors of this paper plan to develop a blockchain-based secure IoT architecture with an IoT centric consensus protocol to ensure security as well as performance efficiency of the TX validation process. Moreover, to ensure the integrity of the IoT TXs, we plan to embed device validation in the TX validation process.

References

Adelantado, Ferran, Vilajosana, Xavier, Tuset-Peiro, Pere, Martinez, Borja, Melia-Segui, Joan, Watteyne, Thomas, 2017. Understanding the Limits of Lorawan. IEEE Commun. Mag. 55 (9), 34–40. arXiv preprint arXiv:1607.08011.

Ahlmeyer, M., Chircu, A.M., 2016. Securing the internet of things: a review. Iss. Inf. Syst. 17 (4), 21–28.

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

Alastria, 2017. National Blockchain Ecosystem. Last accessed 11 September 2018. https://alastria.io/#1.

An Introduction to Iota, 2017. Last accessed 16 September 2018. http://www.iotasupport.com/whatisiota.shtml.

Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., et al., 2018. Hyperledger fabric: a distributed operating system for permissioned blockchains. In: Proceedings of the Thirteenth EuroSys Conference. ACM, p. 30.

Aniello, L., Baldoni, R., Gaetani, E., Lombardi, F., Margheri, A., Sassone, V., 2017. A prototype evaluation of a tamper-resistant high performance blockchain-based transaction log for a distributed database. In: Proceedings of the IEEE 13th European Dependable Computing Conference (EDCC), pp. 151–154.

Arias, O., Wurm, J., Hoang, K., Jin, Y., 2015. Privacy and security in internet of things and wearable devices. IEEE Trans. Multi-Scale Comput. Syst. 1 (2), 99–109.

Armknecht, F., Karame, G.O., Mandal, A., Youssef, F., Zenner, E., 2015. Ripple: overview and outlook. In: Proceedings of the International Conference on Trust and Trustworthy Computing. Springer, pp. 163–180.

Balamurugan, B., Dyutimoy, B., 2017. Security in network layer of iot: possible measures to preclude. In: J. N., T. R. (Eds.), Security Breaches and Threat Prevention in the Internet of Things, IGI Global, pp. 46–75 Ch. 3.

Baliga, A., 2017. Understanding Blockchain Consensus Models. White paper. Persistent Systems Ltd.

A. Banafa, IoT Standardization and Implementation Challenges, IEEE Internet of Things. http://iot.ieee.org/newsletter/july-2016/iot-standardization-and-implementation-challenges.html.

Bentov, I., Lee, C., Mizrahi, A., Rosenfeld, M., 2014. Proof of activity: extending bitcoin's proof of work via proof of stake [extended abstract] y. ACM SIGMETRICS Perform. Eval. Rev. 42 (3), 34–37.

Bessani, A., Sousa, J., Alchieri, E.E., 2014. State machine replication for the masses with bft-smart. In: Proceedings of the IEEE/IFIP 44th International Conference on Dependable Systems and Networks (DSN), pp. 355–362.

Bigchaindb: The Blockchain Database, 2018. Last accessed 13 September 2018. https://www.bigchaindb.com/.

Biswas, K., Muthukkumarasamy, V., 2016. Securing smart cities using blockchain technology. In: Proceedings of the IEEE 14th International Conference on Smart City High Performance Computing and Communications, pp. 1392–1393.

Bitcoin developer guide, 2017. Last accessed 17 September 2018. https://bitcoin.org/en/developer-guide#block-chain.

Bitcoin-Developer-Guide, Transactions, Developer Guide, 2018. Last accessed 13 September 2018. https://bitcoin.org/en/developer-guide#transactions.

Bitcoin-Forum, 2016. Difference between Miners and Nodes. Last accessed 21 July 2018. https://bitcointalk.org/index.php?topic=1734235.0.

Bitcoin.org, 2017. Warning: Better Security Has Costs. Last accessed 22 July 2018. https://bitcoin.org/en/bitcoin-core/features/requirements.

Bitcoinwiki, 2017. Scalability. Last accessed 19 September 2018. https://en.bitcoin.it/wiki/Scalability.

Blockchain Size, 2017. Last accessed 16 September 2018. https://www.blockchain.com/charts/blocks-size.

Bonneau, J., Miller, A., Clark, J., Narayanan, A., Kroll, J.A., Felten, E.W., 2015. Sok: research perspectives and challenges for bitcoin and cryptocurrencies. In: Proceedings of the IEEE Symposium on Security and Privacy (SP), pp. 104–121.

Borgohain, T., Kumar, U., Sanyal, S., 2015. Survey of Security and Privacy Issues of Internet of Things. arxiv:arXiv preprint arXiv:1501.02211.

Brewer, E.A., 2000. Towards robust distributed systems. In: PODC, vol. 7, .

Brewer, R., 2016. Ransomware attacks: detection, prevention and cure. Netw. Secur. 9 (2016), 5–9.

Brody, P., Pureswaran, V., 2014. Device Democracy: Saving the Future of the Internet of Things. IBM.

Buterin, V., et al., 2014. A Next-generation Smart Contract and Decentralized Application Platform. (white paper).

Buterin, Vitalik, 2015a. The Value of Blockchain Technology. Last accessed January 2017. https://blog.ethereum.org/2015/04/13/visions-part-1-the-value-of-blockchain-technology/.

Buterin, Vitalik, 2015b. On Public and Private Blockchains. Last accessed 10 January 2018. https://blog.ethereum.org/2015/08/07/on-public-and-private-blockchains/.

Buterin, Vitalik, 2016. On Settlement Finality. Last accessed 18 July 2018. https://blog.ethereum.org/2016/05/09/on-settlement-finality/.

Cachin, C., 2016. Architecture of the hyperledger blockchain fabric. In: Proceedings of the Workshop on Distributed Cryptocurrencies and Consensus Ledgers, vol. 310,.

Cachin, C., Vukolić, M., 2017. Blockchains Consensus Protocols in the Wild. arXiv preprint arXiv:1707.01873.

Carpov, S., Nguyen, T.H., Sirdey, R., Constantino, G., Martinelli, F., 2016. Practical privacy-preserving medical diagnosis using homomorphic encryption. In: Proceedings of the IEEE 9th International Conference on Cloud Computing (CLOUD), pp. 593–599, https://doi.org/10.1109/CLOUD.2016.0084.

Castro, M., Liskov, B., 2002. Practical byzantine fault tolerance and proactive recovery. ACM Trans. Comput. Syst. 20 (4), 398–461.

Castro, M., Liskov, B., 1999. Practical byzantine fault tolerance. In: OSDI, vol. 99, pp. 173–186.

Chen, S., Ma, R., Chen, H.-H., Zhang, H., Meng, W., Liu, J., 2017a. Machine-to-machine communications in ultra-dense networks—a survey. IEEE Commun. Surv. Tutor. 19 (3), 1478–1503.

Chen, L., Xu, L., Shah, N., Gao, Z., Lu, Y., Shi, W., 2017b. On security analysis of proof-of-elapsed-time (poet). In: Proceedings of the International Symposium on Stabilization, Safety, and Security of Distributed Systems. Springer, pp. 282–297.

Christidis, K., Devetsikiotis, M., 2016. Blockchains and smart contracts for the internet of things. IEEE Access 4, 2292–2303.

Christoph, J., 2015. Slock.it 3 Minutes Demo. Last accessed 18 September 2018. https://slock.it/index.html.

Conoscenti, M., Vetrò, A., Martin, J.C.D., 2016. Blockchain for the internet of things: a systematic literature review. In: Proceedings of the IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA), pp. 1–6, https://doi.org/10.1109/AICCSA.2016.7945805.

Decker, C., Wattenhofer, R., 2013. Information propagation in the bitcoin network. In: Proceedings of the IEEE Thirteenth International Conference on Peer-to-peer Computing (P2P), pp. 1–10.

Dinh, T.T.A., Wang, J., Chen, G., Liu, R., Ooi, B.C., Tan, K.-L., 2017. Blockbench: a framework for analyzing private blockchains. In: Proceedings of the ACM International Conference on Management of Data, SIGMOD '17. ACM, New York, NY, USA, pp. 1085–1100. http://doi.acm.org/10.1145/3035918.3064033.

DKMS (Decentralized Key Management System) Design and Architecture V3 (2018. Last accessed 14 September 2018). URL https://github.com/hyperledger/indy-sdk/blob/master/doc/design/005-dkms/DKMS%20Design%20and%20Architecture%20V3.md.

Dorri, A., Kanhere, S.S., Jurdak, R., 2016. Blockchain in Internet of Things: Challenges and Solutions. http://arxiv.org/abs/1608.05187.

Dorri, A., Kanhere, S., Jurdak, R., Gauravaram, P., 2017. Blockchain for iot security and privacy: the case study of a smart home. In: Proceedings of the IEEE 2nd Workshop on Security, Privacy, and Trust in the Internet of Things (PERCOM) Hawaii, USA.

Ducklin, P., 2016. Mirai "internet of Things" Malware from Krebs Ddos Attack Goes Open Source. https://nakedsecurity.sophos.com/2016/10/05/mirai/.

EconoTimes, 2017. Blockchain Project Antshares Explains Reasons for Choosing Dbft over Pow and Pos. Last accessed 18 September 2018, http://www.econotimes.com/Blockchain-project-Antshares-explains-reasons-for-choosing-dBFT-over-PoW-and-PoS-659275.

Ergen, S.C., 2004. Zigbee/ieee 802.15. 4 Summary. UC Berkeley, September 10, p. 17.

Erik, Z., 2017. A Byzantine Fault Tolerance Algorithm for Blockchain. Last accessed 14 September 2018, http://docs.neo.org/en-us/node/whitepaper.html.

Ethcore, 2018. Parity. Last accessed 13 September 2018, https://wiki.parity.io/.

EthEmbedded, 2017. Ethereum Computer Built on Embedded Devices. Last accessed 16 September 2018, http://ethembedded.com/.

Etherscan 2018. Last accessed 11 September 2018. URL https://etherscan.io/chart/tx.

Eyal, I., Sirer, E.G., 2018. Majority is not enough: bitcoin mining is vulnerable. Commun. ACM. 61 (7), 95–102. http://doi.acm.org/10.1145/3212998.

Eyal, I., Gencer, A.E., Sirer, E.G., Van Renesse, R., 2016. Bitcoin-ng: a scalable blockchain protocol. In: NSDI, pp. 45–59.

Gaetani, E., Aniello, L., Baldoni, R., Lombardi, F., Margheri, A., Sassone, V., 2017. Blockchain-based Database to Ensure Data Integrity in Cloud Computing Environments. https://eprints.soton.ac.uk/411996/.

Gao, Y., Nobuhara, H., 2017. A proof of stake sharding protocol for scalable blockchains. Proceed. Asia-Pacific Adv. Netw. 44, 13–16.

Garzik, J., 2015. Public versus Private Blockchains Part 1. Permissioned blockchains.

Gas with hyperledger fabric?, 2016. Last accessed 15 February 2018. https://stackoverflow.com/questions/38635778/gas-with-hyperledger-fabric.

Gideon, G., 2015. Multichain Private Blockchain - White Paper. Last accessed 5 July 2018. https://www.multichain.com/download/MultiChain-White-Paper.pdf.

Gilad, Y., Hemo, R., Micali, S., Vlachos, G., Zeldovich, N., 2017. Algorand: scaling byzantine agreements for cryptocurrencies. In: Proceedings of the 26th Symposium on Operating Systems Principles. ACM, pp. 51–68.

K. Granville, Facebook and Cambridge Analytica: What You Need to Know as Fallout Widens, New York Times. https://www.nytimes.com/2018/03/19/technology/facebook-cambridge-analytica-explained.html.

Gridcoin white paper, 2018. Last accessed 5 July 2018. https://www.gridcoin.us/assets/img/whitepaper.pdf.

Gutierrez, J.A., Naeve, M., Callaway, E., Bourgeois, M., Mitter, V., Heile, B., 2001. Ieee 802.15. 4: a developing standard for low-power low-cost wireless personal area networks. IEEE Netw. 15 (5), 12–19.

Hardjono, T., Smith, N., 2016. Cloud-based commissioning of constrained devices using permissioned blockchains. In: Proceedings of the 2nd ACM International Workshop on IoT Privacy, Trust, and Security. ACM, pp. 29–36.
Houy, N., 2014. It will cost you nothing to'kill'a proof-of-stake crypto-currency. Econ. Bull. 34 (2), 1038–1044.
How Does Bitcoin Work, 2017. Last accessed 19 July 2018. https://bitcoin.org/en/how-it-works.
Huckle, S., Bhattacharya, R., White, M., Beloff, N., 2016. Internet of things, blockchain and shared economy applications. Proced. Comput. Sci. 98, 461–466.
Huh, S., Cho, S., Kim, S., 2017. Managing iot devices using blockchain platform. In: Proceedings of the IEEE 19th International Conference on Advanced Communication Technology (ICACT), pp. 464–467.
Hyperledger Fabric Model - Privacy 2017. Last accessed 14 September 2018. URL https://hyperledger-fabric.readthedocs.io/en/release-1.2/fabric_model.html#privacy.
Security and Access Control 2017. Last accessed 14 September 2018. URL https://hyperledger-fabric.readthedocs.io/en/release-1.2/Fabric-FAQ.html.
Hyperledger Whitepaper, 2016. Last accessed 13 September 2018. https://github.com/hyperledger/hyperledger/wiki/Whitepaper-WG.
Hyperledger-fabric documentation, 2018. Last accessed 10 September 2018. https://media.readthedocs.org/pdf/hyperledger-fabric/latest/hyperledger-fabric.pdf.
Hyperledger Fabric, 2018. Last accessed 13 September 2018, https://hyperledger-fabric.readthedocs.io/en/latest/blockchain.html.
Iain, S., 2018. Proof of Burn. Last accessed 11 September 2018. https://en.bitcoin.it/wiki/Proof_of_burn#cite_note-1.
IBM, 2015. Adept: an Internet of Things Practitioner Perspective. Tech. rep. https://archive.org/details/pdfy-esMcC00dKmdo53-.
Infosec-Institute, 2015. Duqu2.0: the Most Sophisticated Malware Ever Seen. Last accessed 16 July 2018 http://resources.infosecinstitute.com/duqu-2-0-the-most-sophisticated-malware-ever-seen/#gref.
Iota vulnerability report: Cryptanalysis of the Curl Hash Function Enabling Practical Signature Forgery Attacks on the Iota Cryptocurrency, 2017. Last accessed 10 September 2018. https://github.com/mit-dci/tangled-curl/blob/master/vuln-iota.md.
James, R., 2018a. Ethereum/wiki - Mining. Last accessed 28 July 2018. https://github.com/ethereum/wiki/wiki/Mining.
James, R., 2018b. On Sharding Blockchains. Last accessed 15 September 2018. https://github.com/ethereum/wiki/wiki/Sharding-FAQ.
Jing, Q., Vasilakos, A.V., Wan, J., Lu, J., Qiu, D., 2014. Security of the internet of things: perspectives and challenges. Wireless Network 20 (8), 2481–2501.
Kastelein, R., 2016a. Intel Jumps into Blockchain Technology Storm with 'sawtooth Lake' Distributed Ledger. Last accessed 18 September 2018. http://www.the-blockchain.com/2016/04/09/.
Kastelein, R., 2016b. Everledger Rolls Out Blockchain Technology to Digitally Certify Kimberley Diamonds. Last accessed 28 July 2018. http://www.the-blockchain.com/2016/09/20/everledger/.
Khan, Minhaj Ahmad, Salah, Khaled, 2018. Iot security: review, blockchain solutions, and open challenges. Future Generat. Comput. Syst. 82, 395–411, Elsevier.
Khan, R., Khan, S.U., Zaheer, R., Khan, S., 2012. Future Internet: the internet of things architecture, possible applications and key challenges. In: Proceedings of the IEEE 10th International Conference on Frontiers of Information Technology (FIT), pp. 257–260.
Khari, M., Kumar, M., Vij, S., Pandey, P., Vaishali, 2016. Internet of Things: proposed security aspects for digitizing the world. In: Proceedings of the 3rd International Conference on Computing for Sustainable Global Development (INDIACom), pp. 2165–2170.
Konstantinos, K., Angelos, S., Irena, B., Jeff, V., Grance, T., 2016. Leveraging Blockchain-based Protocols in Iot Systems. NIST-Computer Security Resource Center, https://csrc.nist.gov/CSRC/media/Presentations/Leveraging-Blockchain-based-Protocols-in-IoT-Syste/images-media/1_iot_stavrous.pdf.
Kosba, A., Miller, A., Shi, E., Wen, Z., Papamanthou, C., Hawk, 2016. The blockchain model of cryptography and privacy-preserving smart contracts. In: Proceedings of the IEEE Symposium on Security and Privacy (SP), pp. 839–858, https://doi.org/10.1109/SP.2016.55.
Koushanfar, F., Sadeghi, A.-R., Seudie, H., 2012. EDA for secure and dependable cybercars: challenges and opportunities. In: Proceedings of the 49th ACM Annual Design Automation Conference, pp. 220–228.
Kovacs, E., 2017. Shamoon attacks possibly aided by greenbug group. Last accessed 16 July 2018, https://www.securityweek.com/shamoon-attacks-possibly-aided-greenbug-group .
Kshetri, N., 2017. Can blockchain strengthen the internet of things? IT Profes. 19 (4), 68–72, https://doi.org/10.1109/MITP.2017.3051335.
Kumar, S.A., Vealey, T., Srivastava, H., 2016. Security in internet of things: challenges, solutions and future directions. In: Proceedings of the IEEE 49th Hawaii International Conference on System Sciences (HICSS), pp. 5772–5781.
Kwon, J., 2014. Tendermint: Consensus without Mining, Draft V. 0.6, Fall.
Lamport, L., 1978. Time, clocks, and the ordering of events in a distributed system. Commun. ACM 21 (7), 558–565.
Larimer, D., 2014. Delegated Proof-of-stake (Dpos). Bitshare whitepaper.
Lee, B., Lee, J.-H., 2016. Blockchain-based secure firmware update for embedded devices in an internet of things environment. J. Supercomput. 1–16.
Leiding, B., Memarmoshrefi, P., Hogrefe, D., 2016. Self-managed and blockchain-based vehicular ad-hoc networks. In: Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct, pp. 137–140.
Li, R., 2006. Distributed hash table. College of Computer Science, HUST, http://idc.hust.edu.cn/rxli/teaching/p2p/2%20DHT.pdf.
Light Client Protocol, 2018. Last accessed 28 July 2018. https://github.com/ethereum/wiki/wiki/Light-client-protocol.
Lisk Documentation, 2018. Last accessed 5 July 2018. https://lisk.io/documentation.
Litecoin, 2011. Last accessed 5 July 2018. https://litecoin.org/.
Lough, D.L., 2001. A Taxonomy of Computer Attacks with Applications to Wireless Networks. Ph.D. thesis. Virginia Tech.
Lukas, K., 2018. In-depth on Differences between Public, Private and Permissioned Blockchains. Last accessed 4 July 2018 https://medium.com/@lkolisko/in-depth-on-differences-between-public-private-and-permissioned-blockchains-aff762f0ca24.
Lund, D., MacGillivray, C., Turner, V., Morales, M., 2014. Worldwide and Regional Internet of Things (Iot) 2014–2020 Forecast: a Virtuous Circle of Proven Value and Demand. International Data Corporation (IDC). Tech. Rep.
Luu, L., Narayanan, V., Baweja, K., Zheng, C., Gilbert, S., Saxena, P., 2015. Scp: a computationally-scalable byzantine consensus protocol for blockchains. IACR Cryptol. 1168 ePrint Archive 2015.
Makhdoom, I., Abolhasan, M., Ni, W., 2018. Blockchain for iot: the challenges and a way forward. In: Proceedings of the 15th International Joint Conference on e-Business and Telecommunications - Volume 2: SECRYPT, INSTICC, SciTePress, pp. 428–439, https://doi.org/10.5220/0006905605940605.
Michael, M., Buell, D., 2018. Bluemix Is Now Ibm Cloud. Last accessed 19 September 2018. https://www.ibm.com/blogs/bluemix/2017/10/bluemix-is-now-ibm-cloud/.
Miller, A., Xia, Y., Croman, K., Shi, E., Song, D., 2016. The honey badger of bft protocols. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security. ACM, pp. 31–42.
Min, X., Li, Q., Liu, L., Cui, L., 2016. A permissioned blockchain framework for supporting instant transaction and dynamic block size. In: Proceedings of the IEEE Trustcom/BigDataSE/I SPA, pp. 90–96.
MIT-Media-Lab, 2014. Ethos. Last accessed 26 July 2018. http://viral.media.mit.edu/projects/ethos/.
Monero: Private Digital Currency, 2017. Last accessed 25 July 2018. https://getmonero.org/.
Nakamoto, S., 2008. Bitcoin: a Peer-to-peer Electronic Cash System.
NEO.org, 2017. Neo- Whitepaper. Last accessed 18 September 2018. http://docs.neo.org/en-us/.
Neo.org, 2017. Consensus. Last accessed 17 September 2018. http://docs.neo.org/en-us/node/consensus.html.
Oraclize, 2018. How it Works. Last accessed 11 September 2018. http://www.oraclize.it/.
OWASP, 2018. Top 10 Application Security Risks - 2017. Last accessed 17 September 2018. https://www.owasp.org/index.php/Top_10-2017_Top_10.
Pilkington, M., 2016. Blockchain Technology: Principles and Applications, Research Handbook on Digital Transformations, p. 225.
Poon, J., Buterin, V., 2017. Plasma: Scalable Autonomous Smart Contracts. (White paper).
Popov, S., 2016. The Tangle, https://iota.org/IOTA_Whitepaper.pdf.
Preethi, K., 2017. Blockchains Don't Scale. Not Today, at Least. But There's Hope. Last accessed 16 September 2018. https://hackernoon.com/blockchains-dont-scale-not-today-at-least-but-there-s-hope-2cb43946551a.
Prisco, G., Nov-2015. Slock.it to Introduce Smart Locks Linked to Smart Ethereum Contracts, Decentralize the Sharing Economy. Bitcoin Magazine. [Online]. Available: https://bitcoinmagazine.com/articles/slock-it-to-introduce-smart-locks-linked-to-smart-ethereum-contracts-decentralize-the-sharingeconomy-1446746719. (Accessed 20 May 2016).
Proof of authority: Consensus Model with Identity at Stake, 2017. Last accessed 17 September 2018. https://wiki.parity.io/.
Puthal, D., Nepal, S., Ranjan, R., Chen, J., 2016. Threats to networking cloud and edge datacenters in the internet of things. IEEE Cloud Comput. 3 (3), 64–71.
Qiu, T., Chen, N., Li, K., Atiquzzaman, M., Zhao, W., thirdquarter 2018. How can heterogeneous Internet of things build our future: a survey. IEEE Commun. Surv. Tutorials 20 (3), 2011–2027. https://doi.org/10.1109/COMST.2018.2803740.
Quorum-white paper, 2016. Last accessed 5 July 2018. https://github.com/jpmorganchase/quorum-docs/blob/master/Quorum%20Whitepaper%20v0.1.pdf.
Ramachandran, G.S., Krishnamachari, B., 2018. Blockchain for the Iot: Opportunities and Challenges, CoRR Abs/1805.02818. arXiv:1805.02818, http://arxiv.org/abs/1805.02818.
Reid, W., 2015. How Bitcoin's Technology Could Make Supply Chains More Transparent. Last accessed 19 September 2018. http://www.coindesk.com/how-bitcoins-technology-could-make-supply-chains/.
REX-Blog, 2017. Sharding, Raiden, Plasma: the Scaling Solutions that Will Unchain Ethereum. Last accessed 13 September 2018. https://blog.rexmls.com/sharding-raiden-plasma-the-scaling-solutions-that-will-unchain-ethereum-c590e994523b.
Ruckebusch, P., De Poorter, E., Fortuna, C., Moerman, I., 2016. Gitar: generic extension for internet-of-things architectures enabling dynamic updates of network and application modules. Ad Hoc Netw. 36, 127–151.
Sadeghi, A.-R., Wachsmann, C., Waidner, M., 2015. Security and privacy challenges in industrial internet of things. In: Proceedings of the ACM/EDAC/IEEE 52nd Design Automation Conference (DAC). IEEE, pp. 1–6.
Sara, S., Michael, N., 2018. Facebook Has Been Worried about Data Leaks like This since it Went Public in 2012. Last accessed 11 September 2018. https://www.cnbc.com/2018/04/12/facebook-warned-of-data-breaches-years-ago-when-it-went-public-in-2012.html.
Scherer, M., 2017. Performance and Scalability of Blockchain Networks and Smart Contracts. Master's thesis. Umea University, Sweden.
Schneider, F.B., 1990. Implementing fault-tolerant services using the state machine approach: a tutorial. ACM Comput. Surv. 22 (4), 299–319.

Sebastián, P., 2017. An Introduction to Ethereum and Smart Contracts: a Programmable Blockchain. Last accessed 24 July 2018. https://auth0.com/blog/an-introduction-to-ethereum-and-smart-contracts-part-2/.
Sharma, P.K., Chen, M.Y., Park, J.H., 2018. A software defined fog node based distributed blockchain cloud architecture for iot. IEEE Access 6, 115–124, https://doi.org/10.1109/ACCESS.2017.2757955.
Sigfox services, 2018. Last accessed 12 September 2018. https://www.sigfox.com/en.
Sinha, R.S., Wei, Y., Hwang, S.-H., 2017. A survey on lpwa technology: lora and nb-iot. ICT Expr. 3 (1), 14–21.
Sivaraman, V., Gharakheili, H.H., Vishwanath, A., Boreli, R., Mehani, O., 2015. Network-level security and privacy control for smart-home iot devices. In: Proceedings of the IEEE 11th International Conference OnWireless and Mobile Computing, Networking and Communications (WiMob), pp. 163–167.
Slimcoin, 2014. The Next Generation of Cryptocurrencies. Last accessed 12 September 2018. http://slimco.in/.
Sophos-Naked-Security, 2016. Mirai "internet of Things" Malware from Krebs Ddos Attack Goes Open Source. Last accessed 17 July 2018. https://nakedsecurity.sophos.com/2016/10/05/mirai-internet-of-things-malware.
Steemit, 2017. Neo's Consensus Protocol: How Delegated Byzantine Fault Tolerance Works. Last accessed 15 September 2018. https://steemit.com/neo/@basiccrypto/neo-s-consensus-protocol-how-delegated-byzantine-fault-tolerance-works.
Survey on blockchain Technologies and Related Services, 2015. http://www.meti.go.jp/english/press/2016/pdf/0531_01f.pdf.
Szabo, N., 2004. The idea of smart contracts. IEEE International Workshop on Electronic Contracting (WEC).
Taherkordi, A., Loiret, F., Rouvoy, R., Eliassen, F., 2013. Optimizing sensor network reprogramming via in situ reconfigurable components. ACM Trans. Sens. Netw. 9 (2), 14.
Tendermint Core 2018. Last accessed 15 September 2018. https://tendermint.com/docs/introduction/introduction.html#consensus-overview.
Telehash, 2017. Telehash Encrypted Mesh Protocol. Last accessed 13 September 2018. http://telehash.org/.
The CEO's Guide to Data Security, 2016. Protect Your Data through Innovation - AT&T Cybersecurity Insights, vol. 5, https://www.business.att.com/cybersecurity/docs/vol5-datasecurity.pdf.
The-Linux-Foundation, 2018. Hyperledger Business Blockchain Technologies. Last accessed 15 September 2018. https://www.hyperledger.org/projects.
Tschorsch, F., Scheuermann, B., 2015. Bitcoin and beyond: a technical survey on decentralized digital currencies. IEEE Commun. Surv. Tutor. 18 (3), 2084–2123.
Tuesta, D., Alonso, J., Cámara, N., et al., 2015. Smart Contracts: the Ultimate Automation of Trust, Digital Economy Outlook, https://www.bbva.com/wp-content/uploads/en/2016/11/Digital_Economy_Outlook_Oct15_Cap1.pdf.
Underwood, S., 2016. Blockchain beyond bitcoin. Commun. ACM 59 (11), 15–17.
Vanhoef, M., Piessens, F., 2014. Advanced wi-fi attacks using commodity hardware. In: Proceedings of the 30th ACM Annual Computer Security Applications Conference, pp. 256–265.
Vukolić, M., 2015. The quest for scalable blockchain fabric: proof-of-work vs. bft replication. In: Proceedings of the International Workshop on Open Problems in Network Security. Springer, pp. 112–125.
What is iota?, 2017. Last accessed 18 September 2018. https://iota.readme.io/v1.5.0/docs.
Wood, G., 2014. Ethereum: a Secure Decentralised Generalised Transaction Ledger, Ethereum Project Yellow Paper, vol. 151, pp. 1–32.
Wurm, J., Hoang, K., Arias, O., Sadeghi, A.-R., Jin, Y., 2016. Security analysis on consumer and industrial iot devices. In: Proceedings of the IEEE 21st Asia and South Pacific Design Automation Conference (ASP-DAC), pp. 519–524.
Wüst, K., Gervais, A., 2017. Do you need a blockchain? IACR Cryptol. ePrint Arch. 2017, 375.
Xrp: The Digital Asset for Payments, 2013. Last accessed 5 July 2018. https://ripple.com/xrp/.
Yli-Huumo, J., Ko, D., Choi, S., Park, S., Smolander, K., 2016. Where is current research on blockchain technology? - a systematic review. PloS One 11 (10), e0163477.
Yu, T., Sekar, V., Seshan, S., Agarwal, Y., Xu, C., 2015. Handling a trillion (unfixable) flaws on a billion devices: rethinking network security for the internet-of-things. In: Proceedings of the 14th ACM Workshop on Hot Topics in Networks, p. 5.
Zerocoin project, 2018. Last accessed 25 July 2018. http://zerocoin.org/.
Zhang, Y., Wen, J., 2016. The Iot Electric Business Model: Using Blockchain Technology for the Internet of Things, Peer-to-peer Networking and Applications, pp. 1–12.
Zheng, Z., Xie, S., Dai, H.-N., Wang, H., 2016. Blockchain Challenges and Opportunities: a Survey. Work Pap.
Zheng, Z., Xie, S., Dai, H., Chen, X., Wang, H., 2017. An overview of blockchain technology: architecture, consensus, and future trends. In: Proceedings of the IEEE International Congress on Big Data (BigData Congress), pp. 557–564.
Zyskind, G., Nathan, O., Pentland, A., 2015a. Enigma: Decentralized computation platform with guaranteed privacy, CoRR abs/1506.03471. http://arxiv.org/abs/1506.03471.
Zyskind, G., Nathan, O., Pentland, A., 2015b. Decentralizing privacy: using blockchain to protect personal data. In: Proceedings of the IEEE Security and Privacy Workshops, pp. 180–184, https://doi.org/10.1109/SPW.2015.27.

Imran Makhdoom (S′18) received the B.E. degree in telecommunications engineering and the master's degree in information security from the National University of Sciences and Technology, Pakistan, in 2004 and 2015, respectively. He is currently pursuing the Ph.D. degree with the University of Technology Sydney researching on IoT security. Before that, he worked as a Project Manager on various wireless communication and IT projects involving Satellite, OFC and CISCO networks. He has also served in a semi-government organization for various cyber-security auditing tasks from 2014 to 2016. He is an EC-Council Certified Secure Computer User and certified IoT specialist from University of California Irvine, USA. He was a recipient of the President's Gold Medal for securing the first position in his master's degree.

Mehran Abolhasan (S′01–M′03–SM′11) received the B.E. degree in computer engineering and the Ph.D. degree in telecommunications from the University of Wollongong in 1999 and 2003, respectively. He is currently an Associate Professor and the Deputy Head of the School of Electrical and Data Engineering, University of Technology Sydney. He has authored over 120 international publications and has won over $3 million in research funding. His current research interests are software-defined networking, IoT, wireless mesh, wireless body area networks, cooperative networks, 5G networks and beyond, and sensor networks.

Haider Abbas (SM′16) is a Cyber Security Professional, an Academician, a Researcher, and an Industry Consultant who took professional trainings and certifications from the Massachusetts Institute of Technology, USA; Stockholm University, Sweden; the Stockholm School of Entrepreneurship, Sweden; IBM, USA; and the EC Council. He received the M.S. degree in engineering and management of information systems and the Ph.D. degree in information security from the KTH-Royal Institute of Technology, Stockholm, Sweden, in 2006 and 2010, respectively. His professional career consists of activities ranging from research and development and industry consultations (government and private), through multinational research projects, research fellowships, doctoral studies advisory services, international journal editorships, conferences/workshops chair, invited/keynote speaker, technical program committee member, and reviewer for several international journals and conferences. He is also an Adjunct Faculty and Doctoral Studies Advisor at the Florida Institute of Technology, USA and Manchester Metropolitan University, United Kingdom. In recognition of his services to the international research community and excellence in professional standing, he has been awarded one of the youngest Fellows of the Institution of Engineering and Technology, U.K.; a fellow of the British Computer Society, U.K.; and a fellow of the Institute of Science and Technology, U.K. He has also been elected to the grade of Senior Member of Institute of Electrical and Electronics Engineers (IEEE), USA.

Wei Ni (M′09–SM′15) received the B.E. and Ph.D. degrees in electronic engineering from Fudan University, Shanghai, China, in 2000 and 2005, respectively. He is currently a Team Leader with CSIRO, Sydney, Australia, and an Adjunct Professor with the University of Technology Sydney. He was a Post-Doctoral Research Fellow with Shanghai Jiaotong University from 2005 to 2008, the Deputy Project Manager of the Bell Labs R&I Center, Alcatel/Alcatel-Lucent from 2005 to 2008, and a Senior Researcher with Devices Research and Development, Nokia from 2008 to 2009. He also holds adjunct positions with the University of New South Wales and Macquarie University. His research interests include stochastic optimization, game theory, graph theory, as well as their applications to network and security. He has been serving as the Vice Chair of IEEE NSW VTS Chapter and Editor of IEEE Transactions on Wireless Communications since 2018, the Secretary of IEEE NSW VTS Chapter from 2015 to 2018, the Track Chair for VTC-Spring 2017, the Track Co-Chair for IEEE VTC-Spring 2016, and the Publication Chair for BodyNet 2015. He also served as the Student Travel Grant Chair for WPMC 2014, a Program Committee Member of CHINACOM 2014, and a TPC Member of IEEE ICC′14, ICCC′15, EICE′14, and WCNC′10.