Blockchain for IoT: The Challenges and A Way Forward

Imran Makhdoom[1], Mehran Abolhasan[2] and Wei Ni[3]
[1,2]*University of Technology Sydney, Australia,*[3]*Data61-CSIRO*
Imran.Makhdoom@student.uts.edu.au[1], *mehran.abolhasan@uts.edu.au*[2], *Wei.Ni@data61.csiro*[3],

Keywords: Blockchain for IoT, Blockchain Consensus Protocols, Transaction Validation Rules, Blockchain Challenges, IoT Security.

Abstract: Bitcoin has revolutionized the decentralized payment system by excluding the need for a trusted third party, reducing the transaction (TX) fee and time involved in TX confirmation as compared to a conventional banking system. The underlying technology of Bitcoin is Blockchain, which was initially designed for financial TXs only. However, due to its decentralized architecture, fault tolerance and cryptographic security benefits such as user anonymity, data integrity and authentication, researchers and security analysts around the world are focusing on the Blockchain to resolve security and privacy issues of IoT. But at the same time, default limitations of Blockchain, such as latency in transaction confirmation, scalability concerning Blockchain size and network expansion, lack of IoT-centric transaction validation rules, the absence of IoT-focused consensus protocols and insecure device integration are required to be addressed before it can be used securely and efficiently in an IoT environment. Therefore, in this paper we analyze some of the existing consensus protocols used in various Blockchain-based applications, with a focus on investigating significant limitations in TX (Transaction) validation and consensus mechanism that make them inappropriate to be implemented in Blockchain-based IoT systems. We also propose a way forward to address these issues.

1 INTRODUCTION

Millions of embedded devices are being used today in safety and security critical applications such as ICS (Industrial Control Systems), VANET (Vehicular Adhoc Network), disaster management and critical infrastructure (Cam-Winget et al., 2016). A massive number of these devices have been interconnected to each other and further connected to the internet to form an Internet of Things (IoT). It is estimated that by 2020, the number of IoT connected devices will exceed to 30 billion (Lund et al., 2014) and M2M traffic flows are also expected to constitute up to 45% of the whole internet traffic (Evans, 2011). However, due to interconnection with the internet, IoT devices are vulnerable to various security and privacy threats (Cam-Winget et al., 2016; Poulsen, 2003; Greenberg, 2015; Kumar et al., 2016; Sadeghi et al., 2015; Borgohain et al., 2015).

It is also expected that by the end of 2020, more than 25% of corporate attacks would be because of compromised IoT devices (AT&T, 2016). Similarly, the successful launch of sophisticated cyber-attacks like Mirai (Ducklin, 2016), Ransomware (Brewer, 2016), Xafekopy (Kaspersky-Lab, 2017), Night Dragon (Miller and Rowe, 2012), Havex (FSecure-Labs, 2014), and Stuxnet (Langner, 2013) in recent past have rendered existing IoT protocols ineffective.

Moreover, despite centralization and controlled access to data, even the cloud-supported IoT is vulnerable to security and privacy issues (Puthal et al., 2016). Security flaws in IoT are thus leading to attacks on device integrity, data secrecy and privacy, attacks on the availability of network and attacks on the availability and integrity of services, e.g., DoS and DDoS Attacks (Borgohain et al., 2015). The current security issues in IoT can be attributed to centralized network architecture, lack of application layer security, inadequate standardization on IoT products concerning security, i.e., hardware and software, and the wide gap between manufacturers and security analysts. 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, data integrity and redundancy against network attacks.

Although Blockchain-based Bitcoin is a financial transaction protocol, but due to its decentralized architecture and cryptographic security benefits such as user anonymity, fault tolerance, data integrity and authentication, researchers and security analysts around the world are focusing on Blockchain to resolve security and privacy issues of IoT. No doubt there has been a surge in the development of new Blockchain platforms including Ethereum, Hyperledger, Multichain, NEO, and IOTA. However, current Blockchain platforms have some distinct weaknesses that forbid an impromptu implementation of the Blockchain in an IoT environment.

The main contribution of this paper is to highlight peculiar weaknesses in current Blockchain technologies, which are critical for the design and development of a secure Blockchain-based IoT System. These challenges include lack of IoT-centric transaction validation rules, the absence of IoT-oriented consensus protocol and secure integration of IoT devices with the Blockchain. To reach the desired conclusions, we have carried out a comprehensive analysis of some of the prominent Blockchain consensus protocols and have implemented a test scenario of a Blockchain-supported IoT-based environment monitoring system. A prudent solution to these issues would benefit the entire IoT ecosystem.

The rest of the paper is organized as follows. In Section-2, we describe the impact of progression in the Blockchain technology and its impact on IoT. Challenges to the Blockchain's adoption in IoT are explained in Section-3. A way forward to solve some of the challenges is proposed in Section-4, and the paper is concluded with a hint of future work in Section-5.

2 PROGRESSION OF BLOCKCHAIN TECHNOLOGY AND ITS IMPACT ON IOT

Bitcoin-Blockchain has revolutionized the distributed ledger technology with its significant cryptographic security and immutability. As a result, a new Blockchain platform is being introduced every other day, claiming to be better than the others. IoT being a Blockchain use case is not astonishing at all. IoT can leverage the key benefits of Blockchain to resolve its ever-growing security and privacy issues. Blockchain with its decentralized architecture and an unforgeable attribute, provides an ideal solution for IoT systems that are most of the time deployed in a hostile environment without any physical security. IoT systems can use Blockchain technology as a secure, unforgeable and auditable log of data. It can also be used to set policies, control and monitor access rights to user/sensor data and execute various actions based on certain conditions using smart contracts.

Due to a large number of Blockchain platforms currently available, we have carried out a comparison of some of the most prominent ones including Bitcoin, Ethereum, Hyperledger-Fabric and IOTA. They are compared for the suitability of their functionality and security and performance features concerning IoT environment. As shown in Table-1, 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 transactions via a couple of home miners/validators. Currently, Ethereum (Buterin et al., 2014) and Hyperledger (The-Linux-Foundation, 2017) provide such a hybrid technology. Whereas, Bitcoin (Nakamoto, 2008) and IOTA (Popov, 2016) support public participation. It is also imperative to mention here that as you deviate from the public Blockchain the more you go away from the decentralization. Hence, the factor of trust will come into play, the more private a Blockchain network becomes. IoT systems are deployed for multiple applications varying from smart watches to ICS, and again its the Ethereum and Hyperledger that support multiple Blockchain applications beyond fintech (financial technology). An important factor for an IoT system is the speedy transaction confirmation which leads to the requirement of instant consensus agreement without Blockchain forks. It is evident from Table-1 that BFT-based consensus protocols address this issue with greater reliability. Another vital consideration is that IoT systems especially the sensors operating in a smart city environment would be generating millions of transactions per day. Therefore, an ideal IoT-oriented Blockchain platform should not have a transaction fee or gas requirement, such as we have in Ethereum. However, Hyperledger-Fabric has kept this factor optional.

Modern-day IoT systems not only require M-2-M payment methods only but also need to share data, control access rights, set sensor policies and a lot more. In this regard, IOTA has been designed purely for M-2-M micro or even nano pay-

Table 1: 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	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 (Currently)
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	34% attack
17.	TX throughput	7 TPS	8-9 TPS	Can achieve thousands TPS (depending upon number of endorsers, orderers and committers)	Currently, the Coordinator being the bottleneck, the throughput varies between 7-12 TPS
18.	Latency in single confirmation for a TX	10 mins (60 mins for a confirmed TX)	15-20 secs	Less than Bitcoin and Ethereum	Being in transition phase the TX confirmation time varies from minutes to hours
19.	Is it Scalable?	X	X	X	Yes (Scalability improves with the increase in the size of the network)
20.		(Nakamoto, 2008; Bitcoin-Developer, 2017; Bitcoin-Org, 2017)	(James, 2018a; Wood, 2014; Buterin et al., 2014; Etherscan, 2018)	(The-Linux-Foundation, 2017; Hyperledger, 2016; Cachin, 2016; Hyperledger-Docs, 2018; Hyperledger-Fabric, 2018; Hyperledger-Gas, 2016)	(IOTA, 2017; Popov, 2016)

ments without any TX fee. However, it does not support execution of smart contracts, which facilitate the user-driven policy setting and access control rights. The smart contract feature is supported by Ethereum and Hyperledger-Fabric. IoT systems do require TX integrity and authentication, which is ensured by most of the Blockchain platforms. Out of the four, only Hyperledger provides data confidentiality through in-band encryption and ensures the privacy of user data by allowing the creation of private channels (Hyperledger-Fabric, 2018). It also supports ID (Identity) management, anonymity, audit-ability, TX integrity and authorization through public-key certificates (from a trusted CA (Certificate Authority)). Both the factors, i.e., confidentiality and ID management, are important requirements from IoT perspective.

From performance and efficiency point of view Hyperledger provides a high TX throughput without any risk of Blockchain forks, thus, minimizing latency in TX confirmation. Hyperledger uses PBFT (Practical Byzantine Fault Tolerance) for validation of TXs utilizing minimal resources, i.e., low energy and computation cost (Hyperledger, 2016). Unlike Ethereum-Blockchain, it does not require any gas to process the TXs. However, there are some limitations in permissioned Blockchains. Being partially-decentralized, the trust is placed in some known miner/validator nodes. Hence, in case of a successful malware attack such as Mirai (Ducklin, 2016) which can infect and compromise a large number of nodes for malicious purposes, the integrity of TX and Block validation process would be questionable. Moreover, the user enrolment, authentication, and authorization based on public-key certificates is dependent on a trusted CA, which brings some degree of centralization. Also, most of the private/permissioned Blockchains incorporate BFT-based consensus algorithms. Such algorithms are prone to DoS attacks. They can usually tolerate only $f = (n-1)/3$ faulty nodes. BFT-based algorithms 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 from 6000 TPS to less than 1500 TPS (Mattias, 2017). With the advancements in the Blockchain technology and development of some prominent Blockchain platforms (as shown in Table-1), there is a hope that some issues concerning IoT security and performance can be easily resolved. These issues include security of data in storage, transparency in logging events and transactions, ability to operate in a trustless environment, fault tolerance, data confidentiality, user enrolment, ID management, audit-ability, key management and access control. However, there is still a number of performance and security challenges in public and permissioned Blockchains that need considerable research. Therefore, currently it is not possible to just copy and paste a particular type of Blockchain platform for IoT.

3 CHALLENGES TO BLOCKCHAIN'S ADOPTION IN IOT ENVIRONMENT

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 of a frozen food company. The customer orders frozen food products and also decides a temperature threshold that has to be maintained during shipment by the seller. For this test scenario, we used (Raspberry Pi) Rpi-3 with a DS18B20 temperature sensor. The sensor monitors the temperature of the product and periodically sends the sensor data to a private Ethereum Blockchain through a smart contract. An alert is generated for the customer, whenever the temperature threshold policy is violated during shipment. We would describe every aspect in chronological order, as labelled from 1 to 14 in Figure-1;

1. As discussed in Section-2, the Blockchain is a distributed ledger that provides state machine replication (SMR), ensures data integrity, avoids a single point of failure and tolerates faults in a trustless environment.
2. In scenario-1, the Rpi-3 with a temperature sensor can be a full node or a lite Blockchain client. A full node can validate all TXs but does not mine. Whereas, a lite client can only keep track of its TXs. Hence, we can run a Geth (GO Ethereum) full node in Rpi-3.
3. The temperature sensor senses the environment and its value is extracted via JavaScript in a Web UI (User Interface) or a mobile App (Application).
4. The Web UI or the Mobile App needs a Web3.js library to get and push data to the Blockchain.
5. The Web UI connects with the Blockchain Node running on the Rpi-3 via Web3 provider which can be an HTTP, WS (web socket) or an IPC

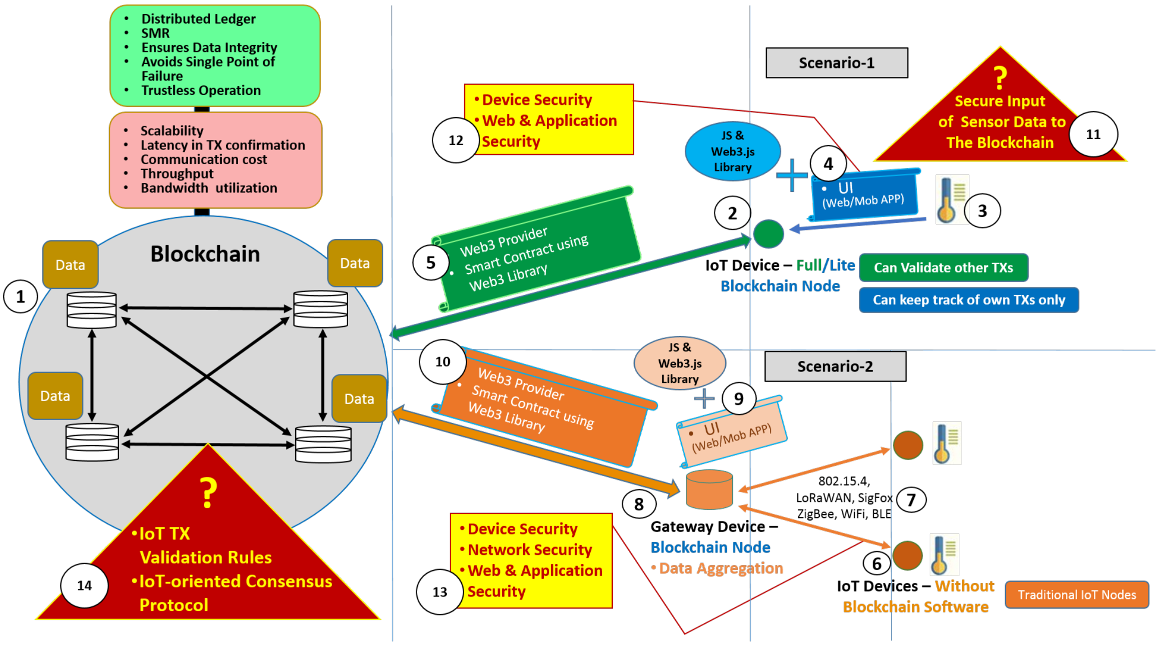

Distributed Ledger
SMR
Ensures Data Integrity
Avoids Single Point of Failure
Trustless Operation
Scalability
Latency in TX confirmation
Communication cost
Throughput
Bandwidth utilization
Blockchain
Data
Data
Data
Data
1
?
•IoT TX Validation Rules
•IoT-oriented Consensus Protocol
14
Scenario-1
•Device Security
•Web & Application Security
12
? Secure Input of Sensor Data to The Blockchain
11
JS & Web3.js Library
4
• UI (Web/Mob APP)
3
2
Web3 Provider
• Smart Contract using Web3 Library
5
IoT Device – Full/Lite Blockchain Node
Can Validate other TXs
Can keep track of own TXs only
10
Web3 Provider
• Smart Contract using Web3 Library
JS & Web3.js Library
Scenario-2
9
• UI (Web/Mob APP)
802.15.4, LoRaWAN, SigFox ZigBee, WiFi, BLE
7
8
Gateway Device – Blockchain Node
• Data Aggregation
6
IoT Devices – Without Blockchain Software
Traditional IoT Nodes
•Device Security
•Network Security
•Web & Application Security
13

Figure 1: Challenges for a Blockchain-based IoT System.

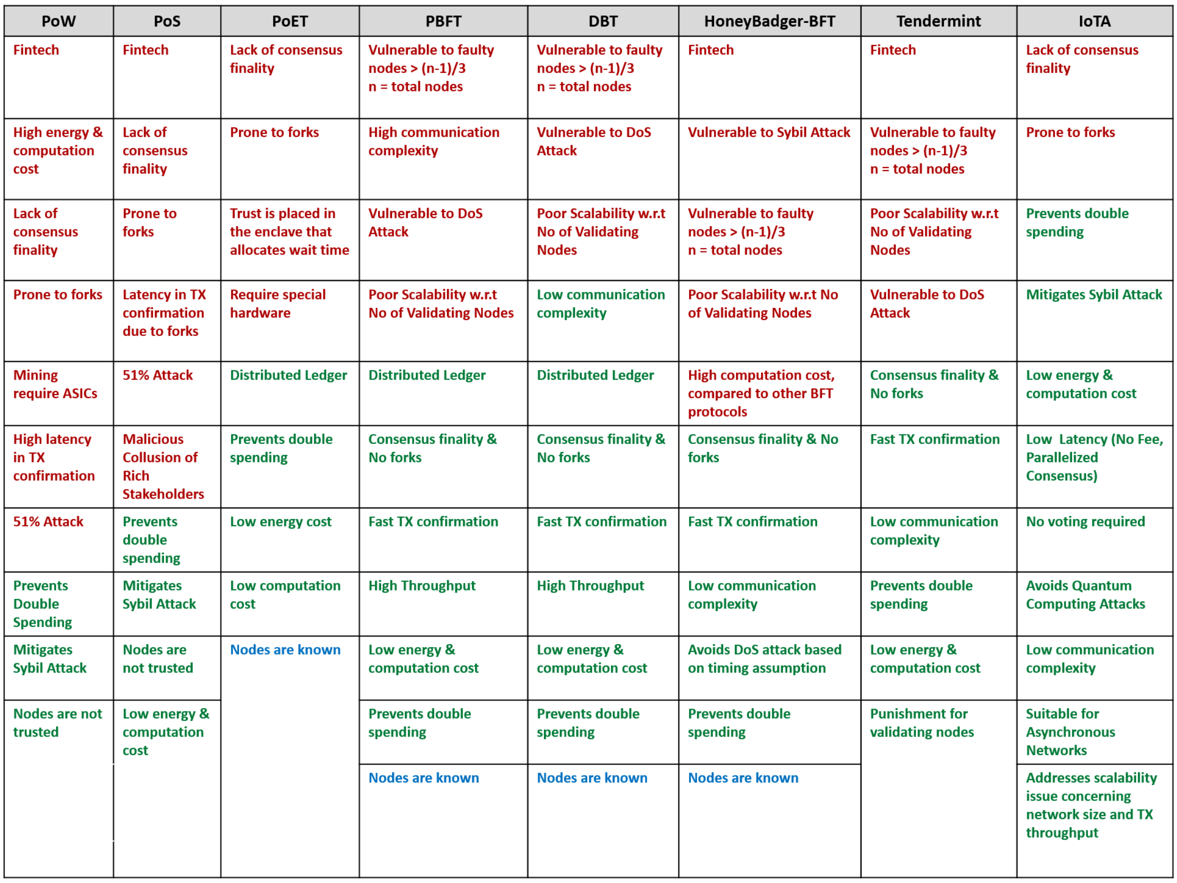
provider. The Web UI (User Interface) posts the sensor reading to the Blockchain through smart contract methods called via the Web3.js library. Hence, a mobile or a Web App is the interface between IoT devices and the Blockchain.

6. In scenario-2 an IoT device can be a resource constrained Arduino device or other embedded systems capable of just sensing and transmitting the temperature sensor readings to a gateway device.
7. The Arduino sensor node communicates with the gateway device through slower and less secure wireless communication media such as 802.15.4, 802.11a/b/g/n/p, LoRa, ZigBee, NB-IoT, and SigFox. Resultantly, IoT systems are prone to data leakage and other privacy attacks (Jing et al., 2014).
8. The gateway device is capable of deploying a Geth full node.
9. It pushes sensor data to the Blockchain through a Web or a Mobile App and Web3.js library.
10. Just like in scenario-1, the gateway also connects to the Geth node through a Web3 provider.
11. However, there is a question of how to ensure the secure input of sensor data to the Blockchain?
12. The 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.
13. The point discussed in Ser.12 is also applicable to the gateway device in scenario-2.
14. Another major weakness observed is that currently, none of the Blockchain platforms implements IoT-focused TX validation rules and IoT-oriented consensus protocol.

The primary challenge is non-availability of 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.

3.1 Lack of IoT-Centric Consensus Protocol

A detailed comparison of some noteworthy consensus protocols is shown in Figure-2. The points shown in red color are not suitable for IoT environment, whereas, points shown in green color are beneficial for an IoT system. The current consensus protocols such as PoW (Nakamoto, 2008), PoS (Szabo, 2004), PoET (Kastelein, 2016), and IOTA (Popov, 2016) are designed for public blockchains with a focus on financial value transfer. These consensus protocols share a common issue that consensus process does not end in a permanently committed block instead, blocks are



PoW	PoS	PoET	PBFT	DBT	HoneyBadger-BFT	Tendermint	IoTA
Fintech	Fintech	Lack of consensus finality	Vulnerable to faulty nodes > (n-1)/3 n = total nodes	Vulnerable to faulty nodes > (n-1)/3 n = total nodes	Fintech	Fintech	Lack of consensus finality
High energy & computation cost	Lack of consensus finality	Prone to forks	High communication complexity	Vulnerable to DoS Attack	Vulnerable to Sybil Attack	Vulnerable to faulty nodes > (n-1)/3 n = total nodes	Prone to forks
Lack of consensus finality	Prone to forks	Trust is placed in the enclave that allocates wait time	Vulnerable to DoS Attack	Poor Scalability w.r.t No of Validating Nodes	Vulnerable to faulty nodes > (n-1)/3 n = total nodes	Poor Scalability w.r.t No of Validating Nodes	Prevents double spending
Prone to forks	Latency in TX confirmation due to forks	Require special hardware	Poor Scalability w.r.t No of Validating Nodes	Low communication complexity	Poor Scalability w.r.t No of Validating Nodes	Vulnerable to DoS Attack	Mitigates Sybil Attack
Mining require ASICs	51% Attack	Distributed Ledger	Distributed Ledger	Distributed Ledger	High computation cost, compared to other BFT protocols	Consensus finality & No forks	Low energy & computation cost
High latency in TX confirmation	Malicious Collusion of Rich Stakeholders	Prevents double spending	Consensus finality & No forks	Consensus finality & No forks	Consensus finality & No forks	Fast TX confirmation	Low Latency (No Fee, Parallelized Consensus)
51% Attack	Prevents double spending	Low energy cost	Fast TX confirmation	Fast TX confirmation	Fast TX confirmation	Low communication complexity	No voting required
Prevents Double Spending	Mitigates Sybil Attack	Low computation cost	High Throughput	High Throughput	Low communication complexity	Prevents double spending	Avoids Quantum Computing Attacks
Mitigates Sybil Attack	Nodes are not trusted	Nodes are known	Low energy & computation cost	Low energy & computation cost	Avoids DoS attack based on timing assumption	Low energy & computation cost	Low communication complexity
Nodes are not trusted	Low energy & computation cost		Prevents double spending	Prevents double spending	Prevents double spending	Punishment for validating nodes	Suitable for Asynchronous Networks
			Nodes are known	Nodes are known	Nodes are known		Addresses scalability issue concerning network size and TX throughput

Figure 2: Comparison of Consensus Protocols.

finalized over a period based upon next few blocks extending the chain and then the longest chain is accepted as the valid chain. 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. 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 Will it also support smart contracts like in the Ethereum and Hyperledger-Fabric blockchains? Secondly, does it provide confidentiality of data? and lastly, what is its faulty node tolerance level ?

On the other hand PBFT (Castro and Liskov, 2002; Decker and Wattenhofer, 2013), DBFT (NEO, 2017), HoneyBadger-BFT (Miller et al., 2016) and Tendermint (Tendermint, 2017) 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 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). The issue of scalability is there even in the crash-tolerant replication protocols such as Paxos (Lamport, 1998), Zab (Junqueira et al., 2011) and Raft (Ongaro and Ousterhout, 2014), which are used in many large-scale systems but practically never across more than a handful of replicas. Additionally, BFT 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. Hence, there is a little benefit in using the BFT library or any other replication technique when there is a strong positive correlation between the failure probabilities of the replicas.

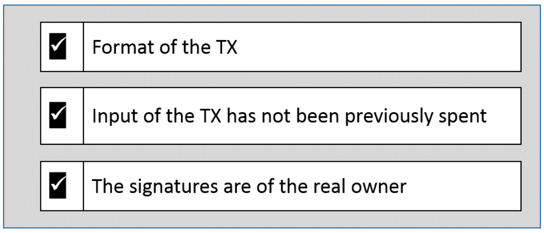
As far as TX latency is confirmed, all full and miner nodes in a blockchain network validate every TX. The block size and the interval between blocks impacts the computation power required to validate all the TXs with minimum latency. Hence, consensus protocols should have high throughput. In this regard, BFT-based protocols can sustain tens of thousands of TXs with practically network-speed latencies (Bessani et al., 2014). Nakamoto Consensus, i.e., PoW is one of the better solutions in a network of untrusted peers that may not be reliable with Byzantine fault tolerance consensus problems. On the contrary, having selected servers or nodes make a decision is sufficient if all the nodes can be trusted. But it is a dangerous principle to rely on, even for in-house “trusted” nodes as they can be compromised. Another major difference between PoW and BFT-based protocols is the notion of availability, which is a critical requirement in realtime 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. However, this is usually not the case with BFT-based protocols.

PBFT is considered to be an expensive protocol concerning message complexity (Luu et al., 2015). Whereas, bandwidth efficiency and 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 hundred thousand sensors. Therefore, any current 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. E.g., In Europe for LoRaWAN protocol that operates on the 868 MHz frequency band, the allowable duty cycle is 1% for any user (Adelantado et al., 2016). 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 summarize, 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.

3.2 TX Validation

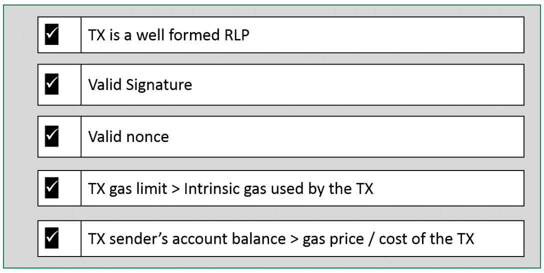
The TX validation process in Bitcoin (Figure-3) validates a TX by checking its format, signatures and the fact that the input to the TX has not been previously spent (Buterin et al., 2014; Bitcoin-Developer, 2018). Whereas, Ethereum-Blockchain checks the format, signatures, nonce, gas, and account balance of the sender's account (Buterin et al., 2014). Ethereum TX validation rules are shown in Figure-4. However, there is a question to the applicability of the same TX validation mechanism to the IoT systems that usually comprise heterogeneous devices, thus sending sensory values or data in distinct formats and different range of values. Also, a targeted or even a generic malware attack can compromise a lot of IoT devices. Subsequently, these devices may be turned into bots and used for further attacks. Therefore, TX validation rules of cryptocurrency centred Bitcoin protocol and general purpose Ethereum-Blockchain may not be appropriate for IoT systems.


☑ Format of the TX

☑ Input of the TX has not been previously spent

☑ The signatures are of the real owner

Figure 3: Bitcoin TX Validation Rules.


☑ TX is a well formed RLP

☑ Valid Signature

☑ Valid nonce

☑ TX gas limit > Intrinsic gas used by the TX

☑ TX sender’s account balance > gas price / cost of the TX

Figure 4: Ethereum TX Validation Rules.

3.3 The Requirement of Huge Resources

According to (Bitcoin.Org, 2017), currently the minimum requirements for a Bitcoin full-node include 177 GB hard disk space, 1 GB RAM, 1 GHz of processing speed and enough network bandwidth to upload 5 GB data per day. Although, the 177 GB size of Bitcoin-Blockchain has reached to its present state over a period of nine years, but this aspect has to be kept in mind while designing a Blockchain-based IoT system. IoT systems comprise resource constraint embedded devices. Therefore, it is a challenge to design a Blockchain-based solution which does not put substantial memory and processing requirements on IoT nodes.

3.4 IoT Device Integration

In the experimental scenario shown in Figure-1, the IoT device pushes sensor data to the Blockchain through a smart contract called in a Web UI or a Mobile App or simply by running a JavaScript in the shell. Currently, Bitcoin-Blockchain and IOTA do not support execution of smart contracts. Hence, the only reliable option is Ethereum Blockchain and Hyperledger-Fabric. Although Ethereum Blockchain is currently the most tested and a reliable platform for multiple DApps (Distributed Applications), however, it has a major limitation, i.e., the smart contracts execute in EVM (Ethereum Virtual Machine) and do not communicate directly with the outside world. Therefore, Web3.js library is used as an interface.

In such a situation, the Blockchain is only useful as a secure distributed database once the sensor data is stored in it. However, before that data integrity is dependent on the security of the device, Web UI and Mobile App itself. Keeping in view the current threat scenario, where, IoT devices can easily be hacked, and malicious code can be executed remotely, the integrity of IoT devices unless validated frequently, would always be questionable. Currently, the only solution available is "Oraclize" (Oraclize, 2018). Which 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, it provides proof of authentication along with the 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 cater for the IoT devices.

4 A WAY FORWARD

4.1 IoT-centric Consensus Protocol

Development of an ideal consensus algorithm for an IoT environment demands that firstly, the requirements of a consensus algorithm for a Blockchain-based IoT system (shown in Figure-5) be distinguished from other applications especially, the fintech. The foremost requirement for IoT systems is IoT-centric TX validation rules. It is important as every new TX is independent of the previous TX and an incident or change in environmental conditions can influence the change in the sensor readings. Hence, IoT TX validation rules need to be carefully scripted and must incorporate environmental context, e.g., in a smart home, the fireplace is only ignited if another sensor, e.g., a motion sensor also detects the presence of a human in that room. The consensus protocol should be robust against Sybil Attack and must have consensus finality to avoid forks. It is vital for the minimum possible delay in TX confirmation and the ultimate high TX throughput.

A typical IoT system is prone to physical or cyber attacks. An example of such an attack on IoT devices in recent past is Mirai Attack in which a large number of IoT devices including DVR and CCTV cameras were compromised and turned into bots. These bots were then used to launch a DDoS attack on a DNS service provider DYN by directing 620 Gbps data in the form of millions of DNS lookup requests (Ducklin, 2016). Whereas, most of the BFT-based protocols can tolerate only $f = (n-1)/3$ faulty nodes. Therefore, an IoT-centric consensus protocol must tolerate 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 participate in the consensus process. Along with security, there are some performance requirements as well. These include low computation overhead, low energy consumption and less communication complexity.

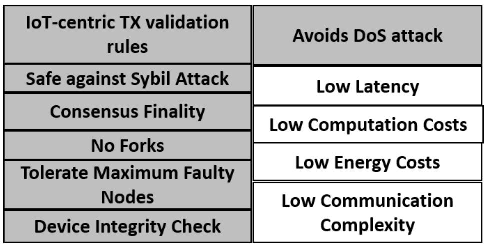

IoT-centric TX validation rules
Safe against Sybil Attack
Consensus Finality
No Forks
Tolerate Maximum Faulty Nodes
Device Integrity Check
Avoids DoS attack
Low Latency
Low Computation Costs
Low Energy Costs
Low Communication Complexity

Figure 5: Considerations for IoT Consensus Protocol.

4.2 Scalability

It not only affects the Blockchain-size but also indirectly influences the consensus process. E.g., the increase 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. To address the issue of scalability concerning the management of everincreasing 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., 2015a; Zyskind et al., 2015b). 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. The Blockchain only contains the pointers/references to data and not all the nodes replicate all TXs.

IBM (IBM-ADEPT, 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 Raspberry Pi based sensor nodes, These nodes only store own Blockchain address and balance. They 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 neighbourhood. 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.

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 (IOTA, 2017). Instead of a conventional Blockchain, IOTA is a distributed architecture based on DAG called Tangle (Popov, 2016). 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 (IOTA-Support, 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 Figure-6 (IOTA-Support, 2017).

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).

4.3 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 store the complete Blockchain and validate every TX in an order. It does improve security but also creates a bottleneck in case of high TX volume. As the Blockchain cannot process more transactions 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 Figure-7). 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 (James, 2018b; REX-Blog, 2017). In its purest form, each shard has its own transaction history and it is effected only by the TXs it contains. E.g., say 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

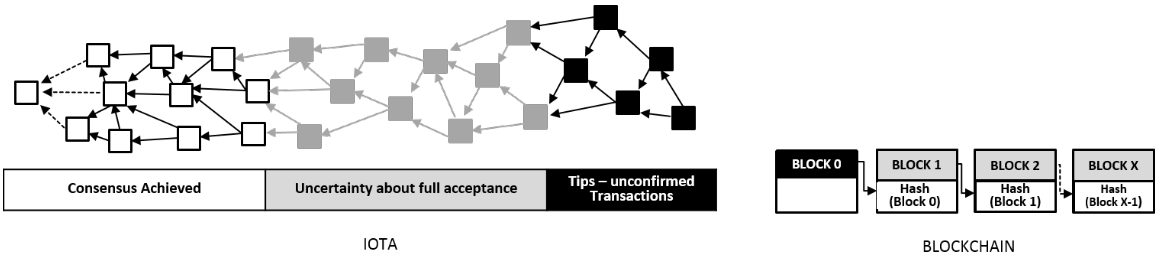


Figure 6: IOTA vs Blockchain

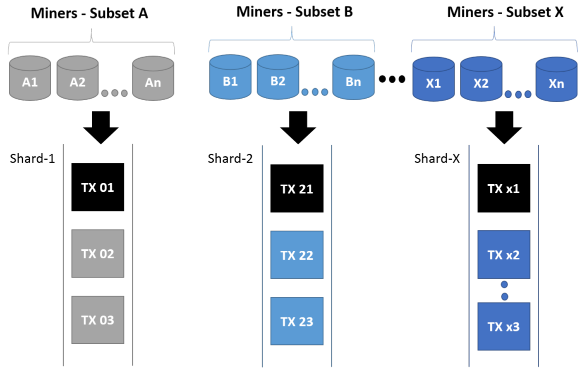


Figure 7: Sharding.

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 transaction capacity only but cannot scale state-storage. Moreover, they are vulnerable to DoS attacks (James, 2018b).

4.4 IoT Device Integration

In addition to Web and Mobile App security, device security can be augmented to ensure the integrity of sensor data, to be pushed to the Blockchain. The first in device security measures is device enrolment, in which only approved devices be allowed to communicate with the Blockchain and call smart contract methods. Smart contracts can restrict calling of various methods to a specific node only. Secondly, all the unnecessary ports on the device should be blocked such as JTAG and UART. Since any open port can be used by an adversary to access the device and make malicious changes. Thirdly, most of the commercially available IoT devices such as sensing devices do not have secure execution environment to keep the cost low. Therefore, the device integrity check should frequently be performed to ensure its legitimacy.

5 CONCLUSIONS

Blockchain has revolutionized the technological world with its distributed network architecture, decentralized control and ability to sustain autonomous, self-regulating, self-managed and fault tolerant IoT systems. However, still there exist some open research challenges that need to be resolved to leverage Blockchain's benefits at the optimum. These challenges include, IoT-centric TX and block validation rules, IoT-oriented consensus protocol, fast TX confirmation for real-time IoT systems, scalability, and secure device integration to the Blockchain. In future, we plan to develop a secure IoT-focused Blockchain consensus protocol with some embedded features such as device integrity check, maximum faulty nodes tolerance level and IoT-oriented TX validation rules. A judicious research in this regard will benefit entire IoT ecosystem.

REFERENCES

Adelantado, F., Vilajosana, X., Tuset-Peiro, P., Martinez, B., and Melia, J. (2016). Understanding the limits of lorawan. *arXiv preprint arXiv:1607.08011*.

AT&T (2016). The CEO's Guide to Data Security. Protect your data through innovation - AT&T Cybersecurity Insights (Vol 5). Available at `https://www.business.att.com/cybersecurity/docs/vol5-datasecurity.pdf`, Viewed 8 January 2018.

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

Bitcoin-Developer (2017). Bitcoin developer guide. Available at `https://bitcoin.org/en/developer-guide\#block-chain`, Viewed 17 December 2017.

Bitcoin-Developer (2018). Transactions. Available at `https://bitcoin.org/en/developer-guide#transactions`, Viewed 17 March 2018.

Bitcoin-Org (2017). Warning: Better security has costs. Available at `https://bitcoin.org/en/bitcoin-`

core/features/requirements, Viewed 28 December 2017.

Bitcoin.Org (2017). System reqirements. Available at https://bitcoin.org/en/bitcoin-core/features/requirements, Viewed 17 December 2017.

Borgohain, T., Kumar, U., and Sanyal, S. (2015). Survey of security and privacy issues of internet of things. *arXiv preprint arXiv:1501.02211*.

Brewer, R. (2016). Ransomware attacks: detection, prevention and cure. *Network Security*, (9):5–9.

Brody, P. and Pureswaran, V. (2014). Device democracy: Saving the future of the internet of things. Available at https://www-935.ibm.com/services/multimedia/GBE03620USEN.pdf, Viewed 14 December 2017.

Buterin, V. et al. (2014). A next-generation smart contract and decentralized application platform. *Whitepaper*.

Cachin, C. (2016). Architecture of the hyperledger blockchain fabric. In *Workshop on Distributed Cryptocurrencies and Consensus Ledgers*.

Cam-Winget, N., Sadeghi, A. R., and Jin, Y. (2016). Can IoT be secured: Emerging challenges in connecting the unconnected. In *Design Automation Conference (DAC), 2016 53nd ACM/EDAC/IEEE*, pages 1–6. IEEE.

Castro, M. and Liskov, B. (2002). Practical byzantine fault tolerance and proactive recovery. *ACM Transactions on Computer Systems (TOCS)*, 20(4):398–461.

Decker, C. and Wattenhofer, R. (2013). Information propagation in the bitcoin network. In *Thirteenth International Conference on Peer-to-Peer Computing (P2P)*, pages 1–10. IEEE.

Ducklin, P. (2016). Mirai "Internet of Things" malware from krebs ddos attack goes open source. *Naked Security*. Available at https://nakedsecurity.sophos.com/2016/10/05/mirai/, Viewed 12 February 2018.

EconoTimes (2017). Blockchain project antshares explains reasons for choosing dbft over pow and pos. Available at http://www.econotimes.com/Blockchain-project-Antshares-explains-reasons-for-choosing-dBFT-over-PoW-and-PoS-659275, Viewed 18 December 2017.

Etherscan (2018). Ethereum transaction chart 2018. Available at https://etherscan.io/chart/tx, Viewed 5 February 2018.

Evans, D. (2011). The Internet of Things: How the next evolution of the internet is changing everything. *CISCO Whitepaper*, 1(2011):1–11.

FSecure-Labs (2014). Havex Hunts For ICS/SCADA Systems. Available at https://www.f-secure.com/weblog/archives/00002718.html, Viewed 9 December 2017.

Greenberg, A. (2015). Hackers remotely kill a jeep on the highway—with me in it. *Wired*, 7:21.

Hyperledger (2016). Hyperledger whitepaper. Available at https://github.com/hyperledger/hyperledger/wiki/Whitepaper-WG, Viewed 19 February 2018.

Hyperledger-Docs (2018). Introduction to hyperledger fabric. Available at https://hyperledger-fabric.readthedocs.io/en/latest/blockchain.html, Viewed 13 February 2018.

Hyperledger-Fabric (2018). Hyperledger-fabric documentation. Available at https://media.readthedocs.org/pdf/hyperledger-fabric/latest/hyperledger-fabric.pdf, Viewed 18 February 2018.

Hyperledger-Gas (2016). Gas with hyperledger fabric? Available at https://stackoverflow.com/questions/38635778/gas-with-hyperledger-fabric, Viewed 15 February 2018.

IBM-ADEPT (2015). An internet of things practitioner perspective. Technical report, IBM. Available at https://archive.org/details/pdfy-esMcC00dKmdo53-_, Viewed 21 March 2018.

IOTA (2017). What is iota? Available at https://iota.readme.io/v1.2.0/docs, Viewed 24 March 2018.

IOTA-Support (2017). An introduction to iota. Available at http://www.iotasupport.com/whatisiota.shtml, Viewed 27 March 2018.

James, R. (2018a). Ethereum/wiki - mining. Available at https://github.com/ethereum/wiki/wiki/Mining, Viewed 5 February 2018.

James, R. (2018b). On sharding blockchains. Available at https://github.com/ethereum/wiki/wiki/Sharding-FAQs, Viewed 27 April 2018.

Jing, Q., Vasilakos, A. V., Wan, J., Lu, J., and Qiu, D. (2014). Security of the internet of things: perspectives and challenges. *Wireless Networks*, 20(8):2481–2501.

Junqueira, F. P., Reed, B. C., and Serafini, M. (2011). Zab: High-performance broadcast for primary-backup systems. In *41st International Conference on Dependable Systems & Networks (DSN)*, pages 245–256. IEEE.

Kaspersky-Lab (2017). Trojans exploit WAP subscriptions to steal money 2017. Available at https://www.kaspersky.com/blog/wap-billing-trojans/18080/, Viewed 15 February 2018.

Kastelein, R. (2016). Intel jumps into blockchain technology storm with 'sawtooth lake' distributed ledger. Available at http://www.the-blockchain.com/2016/04/09/, Viewed 21 March 2018.

Konstantinos, K., Angelos, S., Irena, B., Jeff, V., and Grance, T. (2016). Leveraging blockchain-based protocols in iot systems. *NIST-Computer Security Resource Center*.

Kumar, S. A., Vealey, T., and Srivastava, H. (2016). Security in internet of things: Challenges, solutions and future directions. In *49th Hawaii International Conference on System Sciences (HICSS)*, pages 5772–5781. IEEE.

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

Langner, R. (2013). To kill a centrifuge: A technical analysis of what stuxnet's creators tried to achieve. Available at https://www.langner.com/wp-content/

uploads/2017/03/to-kill-a-centrifuge.pdf, Viewed 11 December 2017.

Lund, D., MacGillivray, C., Turner, V., and 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., and Saxena, P. (2015). Scp: A computationally-scalable byzantine consensus protocol for blockchains. *IACR Cryptology ePrint Archive*, 2015:1168.

Mattias, S. (2017). Performance and scalability of blockchain networks and smart contracts. Master's thesis, Umea University, Sweden.

Miller, A., Xia, Y., Croman, K., Shi, E., and Song, D. (2016). The honey badger of bft protocols. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 31–42. ACM.

Miller, B. and Rowe, D. (2012). A survey SCADA of and critical infrastructure incidents. In *Proceedings of the 1st Annual conference on Research in information technology*, pages 51–56. ACM.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system.

NEO (2017). Whitepaper. Available at http://docs.neo.org/en-us/, Viewed 18 December 2017.

Ongaro, D. and Ousterhout, J. K. (2014). In search of an understandable consensus algorithm. In *USENIX Annual Technical Conference*, pages 305–319.

Oraclize (2018). How it works. Available at http://www.oraclize.it/, Viewed 6 May 2018.

Poon, J. and Buterin, V. (2017). Plasma: Scalable autonomous smart contracts. *Whitepaper*.

Popov, S. (2016). The tangle. *Whitepaper*. Available at http://iotatoken.com/IOTAWhitepaper.pdf, Viewed 26 March 2018.

Poulsen, K. (2003). Slammer worm crashed ohio nuke plant net. *The Register*, 20.

Puthal, D., Nepal, S., Ranjan, R., and Chen, J. (2016). Threats to networking cloud and edge datacenters in the internet of things. *IEEE Cloud Computing*, 3(3):64–71.

REX-Blog (2017). Sharding, raiden, plasma: The scaling solutions that will unchain ethereum. Available at https://blog.rexmls.com/sharding-raiden-plasma-the-scaling-solutions-that-will-unchain-ethereum-c590e994523b, Viewed 21 April 2018.

Sadeghi, A. R., Wachsmann, C., and Waidner, M. (2015). Security and privacy challenges in industrial Internet of Things. In *Design Automation Conference (DAC), 2015 52nd ACM/EDAC/IEEE*, pages 1–6. IEEE.

Szabo, N. (2004). The idea of smart contracts. *IEEE International Workshop on Electronic Contracting (WEC)*.

Tendermint (2017). Byzantine consensus algorithm. Available at https://tendermint.readthedocs.io/en/master/specification/byzantine-consensus-algorithm.html, Viewed 15 February 2018.

The-Linux-Foundation (2017). Hyperledger business blockchain technologies. Available at https://www.hyperledger.org/projects, Viewed 19 February 2018.

Vukolić, M. (2015). The quest for scalable blockchain fabric: Proof-of-work vs. bft replication. In *International Workshop on Open Problems in Network Security*, pages 112–125. Springer.

Wood, G. (2014). Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Project Yellow Paper*, 151:1–32.

Zyskind, G., Nathan, O., et al. (2015a). Decentralizing privacy: Using blockchain to protect personal data. In *Security and Privacy Workshops (SPW)*, pages 180–184. IEEE.

Zyskind, G., Nathan, O., and Pentland, A. (2015b). Enigma: Decentralized computation platform with guaranteed privacy. *arXiv preprint arXiv:1506.03471*.