Mobile Networks and Applications (2022) 27:1408–1420
https://doi.org/10.1007/s11036-022-02015-4

Consensus Algorithms on Appendable-Block Blockchains: Impact and Security Analysis

Roben C. Lunardi[1,2] · Regio A. Michelin[3] · Henry C. Nunes[2] · Charles V. Neu[4] · Avelino F. Zorzo[2] · Salil S. Kanhere[3]

Accepted: 3 September 2021 / Published online: 3 August 2022

Abstract

The Internet of Things (IoT) has been making people's lives more efficient and more comfortable in the past years, and it is expected to get even better. This improvement may benefit from the use of blockchain to enhance security, scalability, reliability and auditability. Recently, different blockchain architectures were proposed to provide a solution that is better suited for IoT scenarios. One of them, called appendable-block blockchains, proposed a data structure that allows to include transactions in blocks that were already inserted in the blockchain. This approach allows appendable-block blockchains to manage large amounts of data produced by IoT devices through decoupled and appendable data structures. Nevertheless, consensus algorithms can impact throughput and latency in scenarios with large amount of produced transactions, since IoT devices can produce data very quickly (milliseconds) while these data might take some time to be included in a block (seconds). Consequently, it is important to understand the behaviour of different consensus algorithms over appendabble-block blockchain in these type of scenarios. Therefore, we adapted the appendable-block blockchain to use and compare the impact of different consensus algorithms: Practical Byzantine Fault Tolerance (PBFT), witness-based, delegated Byzantine Fault Tolerance (dBFT) and Proof-of-Work (PoW). The results show that both dBFT and PBFT can achieve fast consensus (< 150ms) in the context of appendable-block blockchains. We also present a discussion regarding attacks in each consensus algorithm to help one to choose the best solution (considering performance and security issues) for each scenario.

Keywords Distributed ledgers · Blockchain · Consensus algorithms · Internet of Things · IoT

1 Introduction

The Internet of Things (IoT) is already ubiquitous in our lives. It is present in many different domains in our daily lives, for example, in smart homes or smart buildings [14], public services in smart cities [17], healthcare systems [44], in the production of consumer goods in smart industries [50], public transportation systems [33], supply chains [41], or smart vehicles [54]. Hence, it is expected that IoT will witness even wider adoption, thus generating many benefits in productivity, safety, and efficiency.

An IoT solution in general is composed of a myriad of devices, both in quantity and diversity. As a consequence, despite the benefits, there are several concerns about performance, safety, and security risks in these heterogeneous networks. Also, the fact that critical infrastructure, such as, energy grids and even human lives in the context of healthcare, can rely upon IoT devices make it even more important to guarantee the correct operation of such devices. Thereby, new challenges arise in this large, ever-increasing, and sensitive domain. Common challenges include overheads in computation, data management, and security [11].

Therefore, several researchers have proposed different ways to handle those challenges. To tackle the security issues different proposals investigate the use of the blockchain technology [9, 15, 27, 39, 45, 47]. However, by using blockchain other concerns emerge, among them the most important is regarding performance issues.

Some recent works have addressed this by proposing novel blockchain architectures [9, 15] or innovative

✉ Roben C. Lunardi
roben.lunardi@restinga.ifrs.edu.br

[1] IFRS, Porto Alegre, Brazil

[2] PUCRS, Porto Alegre, Brazil

[3] UNSW, Sydney, Australia

[4] Newcastle University, Newcastle Upon Tyne, UK

blockchain data management solutions [39, 45]. However, few blockchain proposals for IoT present a comparison of different consensus algorithms in a modular solution (that allows to choose the best alternative). The consensus algorithms are a key component in the blockchain definition. These algorithms are responsible for establishing trust among untrusted peers, thus ensuring that the ledger has a consistent state among all nodes. Both modular approach and discussion about different consensus algorithms are crucial to help the analysis between performance and security tailored for a specific scenario.

Appendable-block blockchain [37, 38, 43, 46] - a modular and layer-based blockchain - was proposed to tackle some important IoT challenges, such as: handling high rate of transactions and providing resilience to the application. However, the preliminary presented evaluation [38] did not consider some important metrics, such as the total latency to insert a transaction (from the production of the information to its insertion in the nodes' ledger). Additionally, the prior work did not discuss the impact of other consensus algorithms that are used in other blockchains, such as Proof-of-Work (PoW) and delegated Byzantine Fault Tolerance (dBFT). This discussion evaluates the performance of different consensus algorithms.

This paper expands our previous work [38] to: (*i*) implement modular support for two new consensus algorithms for appendable-block blockchain; (*ii*) perform an extended evaluation of the improved version of appendable-block blockchain - using four different consensus algorithms to evaluate the performance, e.g., time required for consensus and transaction latency of each consensus algorithm; and (*iii*) extend the discussion about main security issues for each consensus algorithm and their impact on appendable-block blockchains.

The remainder of this work is organised as follows. Section 2 presents a background on recent works about blockchains and consensus algorithms used in IoT. Section 3 presents modularisation for appendable-block blockchains and how different consensus algorithms can be used in this kind of blockchain. A performance evaluation is presented in Section 4 with four consensus algorithms and four different metrics. Section 5 discusses the main attacks on blockchains and how they could impact the consensus algorithms. Section 6 discusses threats to validity and limitations of the presented evaluation. Finally, Section 7 concludes this work and indicates some future work directions.

2 Background

In the last few years, several blockchain frameworks have been proposed for different IoT application domains, such as video surveillance [42], supply chains [7], vehicular networks [53], and smart grids [22]. To be used in different domains, different cryptography algorithms, consensus algorithms, data management and block structures can be chosen.

2.1 Blockchain in IoT

IoT networks are a challenging environment in which new technologies have to handle heterogeneous devices, data, scalability, and hardware limitation. Typically these networks are composed of devices provided by multiple manufacturers, with different hardware and communication protocols [49]. Additionally, these devices are constrained in terms of hardware capacity, as they are designed to perform specific tasks. In some domains, such as, smart city [28], a very large number of the devices are deployed and are required to perform multiple different tasks. Such an environment would require a scalable solution to accommodate a large number of heterogeneous devices and efficiently handle the produced data. Moreover, addressing the after-mentioned challenges and still ensure data security is a vital task in any new solution.

A recent technology that aims to address some of the challenges in IoT networks is the blockchain technology. Despite it's initial concept applied to the financial domain, where it keeps a public ledger, many researchers have studied this technology and proposed changes, making it suitable for different domains. Hence, the blockchain implementation requires adaptations to fulfill the requirements of such domains. To achieve such customisation level, a blockchain should be designed in a modular way, which would allow quick and easy module interchange. Some researchers [32, 52, 55] have defined different modules and layers for blockchains, however, there is no standard available at this moment.

A key blockchain module is the way consensus is achieved to include blocks in the blockchain, i.e., consensus is responsible for validating candidate blocks before inserting them into the ledger and broadcasting that to other peers. Thus, the algorithm choice is directly related to the domain that the devices are deployed in, blockchain type (public, private, permissioned or permissionless) and time to insert a new block.

Hyperledger is one of the most popular blockchain instances that supports customisation, in particular for consensus algorithms: Kafka, Redundant Byzantine Fault Tolerant (RBFT), Sumeragi and Proof-of-Elapsed Time (PoET) [36]. Despite introducing customisation of multiple consensuses, the Hyperledger solution was not designed for IoT scenarios. Feng et al. [20] proposed a hierarchical byzantine fault tolerant consensus algorithm in order to solve the scale issues presented by Practical Byzantine Fault Tolerant (PBFT). The idea consists of clustering nodes and setting a leader for each cluster. Only these leaders will perform the consensus. This approach is similar to what is proposed by gateway-based architectures [15, 39]. However, they do not present performance evaluation of their architecture nor how they implemented their solution.

2.2 Consensus for blockchain in IoT

Due to the high amount of data generated in an IoT network, transactions processing throughput is one of the main concerns in IoT. This is a challenge for blockchains, which delivers poor performance when compared to other traditional solutions, such as, cloud computing, as stated by Christidis et al. [12]. In their view, the problem is greater in networks using the PoW consensus algorithm. The consensus algorithm is at the core of the blockchain, and controls the new data that is appended and the rules that dictated how the peers should operate, ensuring that the data is trusty.

The PoW algorithm was the first consensus algorithm created. After its introduction with Bitcoin, academia and industry developed several other proposals. These algorithms greatly differ in operation, proposing innovations or/and a trade-off between security, privacy, and performance to better fit in a specific scenario. Christidis et al. [12] consider that both the network requirements in which it will be used and the possible attack vectors that can be exploited are the most important factors to decide which blockchain design should be adopted. Consequently, the number of nodes and the processing overhead are important issues to be considered. Christidis et al. [12] also discuss PoW, Proof-of-stake(PoS), PBFT, Tangaroa, Sieve and Ripple's consensus algorithms. It is important to note that none of these consensus algorithms were tailored specifically to be used with IoT and have a more general use approach.

Han et al. [23] expands the discussion by evaluating the performance of different blockchains and their consensus algorithms with a focus on IoT performance. In their work, the Hyperledger Fabric (HLF) v0.6 with the PBFT consensus protocol, HLF v1.0 with the Byzantine Fault-Tolerant State Machine Replication (BFT-SMaRt) consensus protocol, and Ripple with the Ripple consensus protocol are analysed. The used metric was throughput in distinct scenarios, and each scenario had a different number of nodes and data load. Their work [23] focuses on PBFT algorithm variations. These algorithms do not require a high amount of power to operate, in contrast with PoW and variations, which is a positive feature for IoT scenarios. However, as observed, the performance greatly degrades as more nodes are added to the network, which greatly hinders the solution's scalability. Similarly, Sukhwani et al. [51] modeled PBFT using Stochastic Reward Nets and achieve similar results. The initial results showed that PBFT performance can be problematic in large scenarios (their study included up to 100 nodes) and is impacted by the number of nodes and the number of transactions appended to the blockchain.

As previously stated, differing from evaluations of existing consensus algorithms, researchers proposed multiple works with different consensus algorithms to attend IoT scenarios. One of such works that try to tackle the PBFT scalability limitations is the work of Feng et al. [20]. In their work, they propose the Hierarchical Byzantine Fault Tolerant consensus algorithm. It is an interesting approach, although it can be also expressed by Gateways of networks performing a BFT based consensus algorithm (that was also proposed by other works [14, 39]).

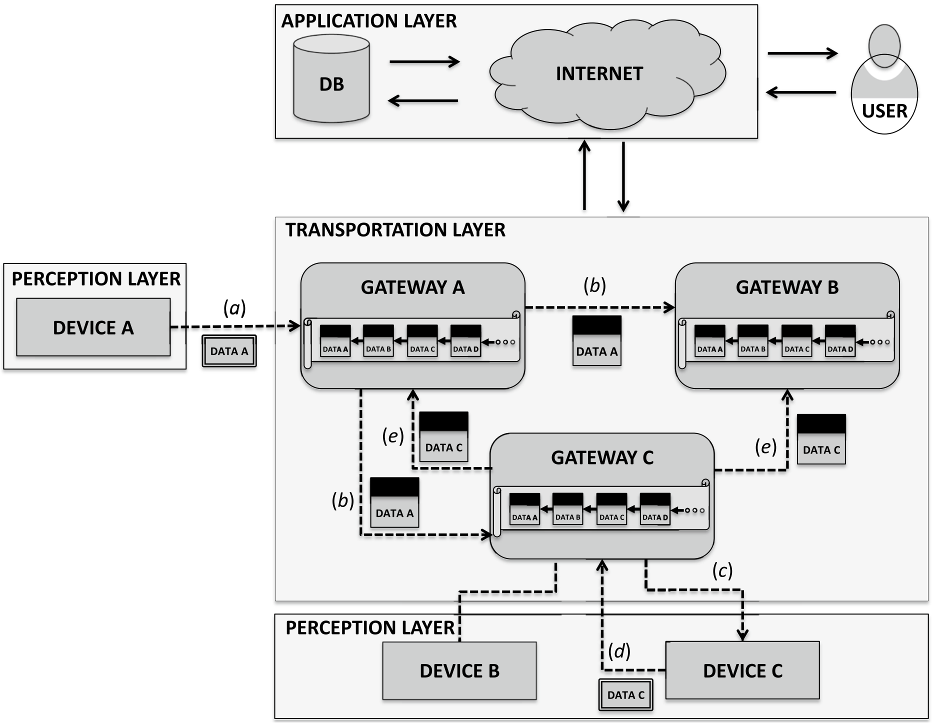
Modifications to other consensus algorithms are also proposed by Fan et al. [19]. Their work consists of an adapted version of the Distributed Proof-of-Stake (DPoS), tailored for IoT environments. In their algorithm, for each block generation, a node is responsible to produce the block, and send it to be validated by the other nodes that are participating in the consensus (similar to the PBFT operation). After the block is validated (or not), the chosen node can be reelected to continue producing blocks, or it can be changed if it is not being fair. Although the algorithm is presented, few discussion is provided about performance, security issues, or its applicability in dynamic scenarios.

Huang et al. [26] propose a credit-based proof-of-work consensus for IoT. It is based on decreasing the mining difficulty for honest nodes and increasing for dishonest nodes through the use of credits generated when new blocks are created. Also, the data structure is modified to use a directed-acyclic graph (DAG) instead of a chain. To assert performance, an evaluation is performed in a smart factory scenario. The results show a better performance than traditional PoW without compromising security.

Another work based on PoW is the Proof-of-Authentication (PoAh) consensus algorithm [40, 48]. PoAH uses media access control as addresses in the blockchain network for each node to reach consensus. The nodes are selected dynamically to verify transactions based on the address. In the work of Maitra et al. [40], a performance evaluation shows energy consumption and latency. However, a comparison with other consensus algorithms is necessary to assert the algorithm performance.

Biswas et al. [5] propose the Proof of Block and Trade (PoBT) consensus algorithm. This algorithm validates transactions (trades) and blocks while still maintain a lightweight algorithm suitable for IoT. One of the approaches to attain its lightweight is limiting the number of peers participating in a session to reduce the latency and increase throughput. This number depends on the total number of nodes in that session. Also, the ledger is split and distributed between nodes, which reduces the memory needs for IoT devices. The consensus algorithm

Fig. 1 Gateway-based Architecture for IoT (extracted from [39])



was implemented in the Hyperledger Fabric and showed a performance improvement compared to traditional Hyperledger Fabric operation.

Li et al. [34] propose an adapted PBFT consensus algorithm for blockchains in IoT. The proposal consists in adapting a reward/punishment system to the traditional PBFT consensus algorithm. This mechanism can help to identify and avoid malicious peers to propose new blocks. Additionally, the authors propose a mechanism to allow that only full-nodes store the entire block, while light-nodes store only the header of each block. Consequently, this mechanism (called as RS erasure) can help to use their proposal over resource-constrained IoT devices.

In general, the consensus for blockchains in IoT are designed for private/ permissioned scenarios. In particular, many proposals adopt a hierarchical architectures (with full-nodes and light-nodes) to allow the adoption of blockchains in IoT environments. The main design challenges for consensus algorithms to be used in blockchains for IoT can be defined in: security, trust, overhead (or performance) and scalability [10]. However some applications can have different demands and requirements. For example, some applications can demand more on the scalability than performance, e.g., vehicles tracking based on GPS have a small rate of updating but have a large number of users; while others can be the opposite, e.g., a limited number of smoke sensors in a smart building requires a lower latency as possible. To tackle that, we present, in the next section, the concept of Appendable-block blockchains and improvements performed to allow the usage (and the evaluation) of different consensus algorithms.

3 Appendable-block blockchain in IoT

This section presents the fundamental concepts of a blockchain architecture that underpins our appendable-block blockchain framework. In the first design of the appendable-block blockchain framework (formerly called R2AC and later on called SpeedyChain), Lunardi et al. [39] presented a lightweight permissioned blockchain that creates blocks on demand. It was designed to focus on IoT designed for Smart Homes/Smart Offices scenarios, using a layer-based architecture [29]. As can be observed in Fig. 1, devices and gateways are separated in different layers, and thus they have different roles in the blockchain.

This layer-based IoT architecture is composed by: (*i*) devices (*D*) in the Perception Layer; (*ii*) Gateways (*G*) in the Transportation Layer; and (*iii*) other nodes, such as Service Providers (*SP*) in the Application Layer. In this architecture, each device can produce and send information to the gateways. That gateways will process the information and append the produced data to a specific block for that device. An interesting feature of how blocks are organised in this blockchain is to allow devices to keep producing and appending information into blockchain independently to the other devices operations (in a high degree of parallelism of insertion in the blockchain).

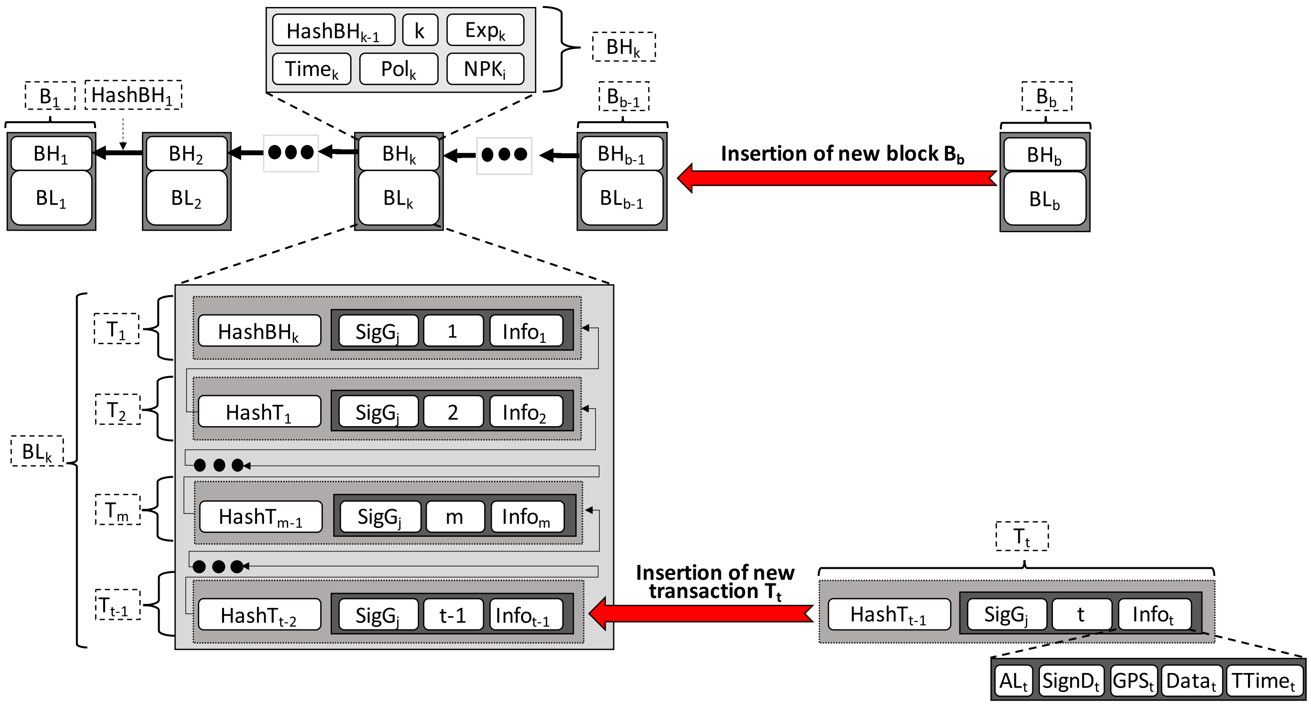


Fig. 2 Appendable-block blockchain data structure (adapted from [38])

Therefore, each device can produce information and send to the gateways to append data to its own block. Gateways will maintain and control the blockchain, which is composed mainly by two parts: block ledger and block header (as shown in Fig. 2). Additionally, gateways are able to maintain only the Block Header - which contains important information about devices (especially their public keys) - without having every device's block ledger. The block ledger is composed by the information digitally signed (both by the device and gateway), and chained through the hash of the previous information (or to the block header if it is the first information of the block ledger).

It is important to note that this work focuses on the blockchain aspects, in particular, on the consensus algorithms performed by the gateways (in the Transportation Layer) and uses concepts that were presented in previous works [38, 39, 43]. Thus, previous framework was extended, providing modularisation, and allowing the blockchain to support different consensus algorithms. Consequently, consensus algorithms can be used based on different IoT environment/requirements. Moreover, the proposed solution was designed to maintain the integrity and availability of the data collected from different sensors/devices for both audition and control (by an application or based on predefined rules), based on predefined policies for each device (in the Perception Layer). As consequence, this modular solution can help to adapt the blockchain to be used in different IoT environments and applications.

3.1 Communication and protocols

As presented previously, appendable-block blockchain is composed by different type of nodes. Full-nodes (also called gateways in IoT environments) manage and handle data produced by light-nodes (also described as devices). It is important to note that appendable-block blockchain was designed as a Private and Permissioned Blockchain, i.e., a set of specific nodes control the access to the blockchain network and they control the insertion the information (through a consensus algorithm).

We assume, in appendable-block blockchains, that all nodes (full-nodes and light-nodes) are capable to use crypthographic algorithms (symmetric and asymmetric encryption, hash and digital signatures). Additionally, we assume every node is identified by (at least) a public key. Every public key should be different and accessible by any participant. Also, each device (light-node) has to be connected to a gateway (full-node) to participate in the IoT network (and interact in this environment). Additionally, the gateways are responsible to manage the device access and provide an API that allows to manage the blockchain.

Before any device performs its first transaction, it should authenticate through a gateway. For example, in Fig. 1, Device *a* is authenticated in the blockchain through Gateway

x. After that, the device has to perform a Key Exchange procedure with the gateway to build a secure channel. This procedure is presented as follows:

1. Device a (represented as D_a) sends a Hello message with its own Public Key DPK_a (e.g., for encryption using the RSA algorithm) to Gateway x (represented as G_x);
2. Gateway x perform the key exchange (e.g., to build an Advanced Encryption Scheme (AES) secure channel) using the received DPK_a;
3. Device a sends a first transaction through an encrypted channel using the AES key generated by the gateway;
4. Gateway x starts the consensus with the other gateways to insert a new block B_a with DPK_a in the block header BH_a and the first transaction T_1 in the block ledger BL_a;
5. After the consensus, if the block is considered valid, the block B_a is inserted in the blockchain;

3.2 Block data

As presented previously in Fig. 2, a block in appendable-block blockchain has two main parts: Block Header (representend as BH_k and *Block Ledger* (BL_k). Therefore, BH_k is composed by some important fields: $HashBH_{k-1}$ contains the hash digest of previous block header; k that is the index of the block in the blockchain ledger; $Time_k$ that represents the block timestamp; Exp_k that presents the threshold time to insert a new transaction in its block ledger (e.g., no information can produce that defined time); Pol_k presents the access policy that the device has to attend; and NPK_j is the node public key. It is important to mention that every node - independent of its type - should have a block in B, composed of at least a block header and the first transaction (that will be discussed next).

Block ledger (BL_k) is composed by the set of linked (by the previous hash) t transactions of the block B_k. A transaction T_m is composed by: $HashT_{m-1}$ that contains the hash of the previous transaction (or the hash of its block header when it is the first transaction of the block ledger); m representing the index of the transaction $\mathrm{T_m}$ in the block ledger BL_k; the digital signature $SigG_m$ (generated using the GPK_h to sign $Info_m$); the $Info_m$ that can be different for each type of node.

Devices provide a set of information, where AL_m is the access level required to access the information from outside of the blockchain that is defined by the device D_j, while the $SigD_m$ represents the signature of (AL_m, GPS_m, $Data_m$, and $TTime_m$) using DPK_j, where GPS_m represents the global position of the device (when it is available), while $\mathrm{Data_m}$ is the data collected/set from/to device D_j and $TTime_m$ is the timestamp when the $Data_m$ was generated/set. It is important to note that $Data_m$ could be formatted differently depending on the device. For example, it could store a single read of a sensor (an integer type) or a set of information, encrypted or not, depending on the configuration established in the IoT Application Layer.

3.3 Smart contracts

The appendable-block blockchain supports the use of smart contracts. This feature uses a unique model proposed in the work of Nunes et al. [46], called Context-based model for smart contracts. This model allows the execution of groups of smart contracts in parallel to process a high number of transactions while still maintaining low latency. These two qualities are important in IoT Domain. Also, the smart contract feature can help in the management and maintenance of IoT Devices as discussed by Christides [12].

However, despite the benefits of this model, there are limitations to its use that may negate its benefits. The most important is that Smart Contracts exist in a context, which is a structure that isolates a group of smart contracts from others. Therefore, a smart contract in a context can not interact with another smart contract in a different context. Thus, a group of smart contracts in one context will have sequential processing and the parallelism feature is attained by processing different contexts in parallel. Therefore, it is important to properly select a program that will execute on top of this model. Programs that can be split into smaller not interacting parts are desirable because these parts can be inserted in different contexts to attain parallelism.

3.4 Consensus

We adapted appendable-block blockchain to allow the adoption of different consensus algorithms. Before discussing different consensus algorithms, first we need to present what is a valid block or transaction. For a transaction to be considered valid, it should have a NPK_i that is already in the blockchain, a valid signature (based on the data transmitted and NPK_i), and a $TTime_m$ lower than its Exp_k (present in the block header) to ensure that no transactions are inserted in an expired block. Moreover, to ensure that a block header is valid: (*i*) the gateways should agree that a new node NPK_i can be part of the blockchain B; (*ii*) the access policy Pol_k for this node NPK_i should be defined; (*iii*) the Exp_k should be calculated to avoid a large block in size. In this work we assume that this validation is performed by the gateways through predefined rules.

Four different consensus algorithms were incorporated to appendable-block blockchains: (*i*) validation based on the authority of gateways and using a specific number of witness, where every block should be signed by at least a predefined number of witness (2 witness were adopted in this work); (*ii*) adapted PBFT algorithm, where more than 2/3 of the active gateways should validate and sign the block; (*iii*) adapted dBFT algorithm, where more than 2/3 of delegated gateways should validate and sign the block; (*iv*) a simplified PoW algorithm, used for comparison since it is adopted in many different blockchains, where a gateway achieves a hash with a certain characteristic (in this work, first 12 bits should be equal to 0). All consensus algorithms, except PoW, could be summarised in Algorithm 1.

```
Require: receive a $NPK_i$ to perform consensus
 1: $b \leftarrow$ lastIndex($B$)
 2: $HashBH_{k-1} \leftarrow$ hash($BH_b$)
 3: $k \leftarrow b + 1$
 4: $Time_k \leftarrow$ getTime()
 5: $Exp_k \leftarrow$ defineExp()
 6: $Pol_k \leftarrow$ setPolicy()
 7: $BH_k \leftarrow \{HashBH_{k-1}, k, Time_k, Exp_k, Pol_k, NPK_i\}$
 8: $consensusResponses \leftarrow$ performConsensus($BH_k$)
 9: if $consensusResponses > minimumResponses$ then
10:     broadcast($BH_k$)
11: end if
```

In order to encapsulate the new block B_k, every information from the block header BH_k must be set, such as the hash of the previous block header BH_b (line 2), block index k (line 3), the timestamp using the time of block creation $Time_k$ (line 4), an expiration time Exp_k to control the validity of the block (line 5), and the access policy Pol_k that the new node is submitted to (line 6 in Algorithm 1). It is important to note that both Exp_k and Pol_k are defined at IoT Application Layer. After the block header is created, the consensus is performed (line 7). The consensus is performed only by gateway nodes. After the consensus is performed and it receives more than the minimum responses for each consensus algorithm, the new block is broadcast to the peers (line 10).

3.5 Transaction insertion

Every time a node produces information $Info_m$, it has has to communicate to a gateway to append the transaction to its block ledger BL_i. This operation is performed only if the node public key (NPK_i) was already inserted in a block header BH_i, as we presented previously. When a gateway receives a new information $Info_m$, the digital signature $SigD_m$ present in $Info_m$ should be validated using the device's public key NPK_i.

After the validation of the signature, the gateway performs the encapsulation of the new transaction, setting: the hash of the previous transaction $HashT_{m-1}$, the index of the transaction (based on the last transaction) m, and the digital signature from the gateway that is processing the transaction $SigG_m$ using its secret key GSK_h.

After that, the gateway creates the new transaction and broadcast it to the other gateways. It is important to note that this procedure is independent from the consensus to append blocks. In this work we focus the analysis on the consensus to append new blocks. A deep discussion about consensus for transactions was performed in a previous work [37].

4 Experimental evaluation

We evaluated four different consensus algorithms using the SpeedyChain framework (an appendable-block blockchain) [43]: PBFT, dBFT, PoW and Witness-based consensus. We used a PoW with 12 bits for the hash difficulty (number of bits zero required in the first bits of the block hash). This difficulty was chosen due to present performance close to others consensus. Also, we used the same emulated IoT environment presented in the previous work [38]. The IoT environment was emulated using the Core emulator [1] to create a container-based network composed by network equipment, gateways and devices. The experiments were performed on a Virtual Machine (VM) with 6-core processor, 16GB of memory and 64MB of graphics memory running Ubuntu 18.04 operating system using a Virtual Box hypervisor over a Macbook Pro with 2.3 GHz 8-Core Intel Core i9 processor, 32GB DDR4 memory.

For all consensus experiments, a network with ten (10) gateways and one thousand (1,000) devices were used. We generated one million (1,000,000) transactions for each experiment. A transaction on the experiments represents sensors readings (temperature, CO2, etc) signed by a device and signed by a gateway. This type of scenario is similar to the largest and most demanding scenarios evaluated in our previous work [38].

4.1 Metrics

In order to perform an evaluation of the four different consensus on appendable-block blockchain, we adopted 4 different metrics:

- **T1**: Time to perform consensus and insert a new block (first time that device is connected) in the leader gateway. This metric represents the time of the consensus not considering the time that other gateways take to insert the block;
- **T2**: Time to perform consensus and replicate it to all gateways (after consensus). It can be understood as the overall time spent for each block insertion procedure;
- **T3**: Time to insert a transaction in the blockchain after a gateway receives it. This metric represents the overhead of the transaction insertion procedure;
- **T4**: Average time to insert a transaction in the blockchain for all gateways (from when it is created to its insertion in the ledger of each gateway). This is important to measure

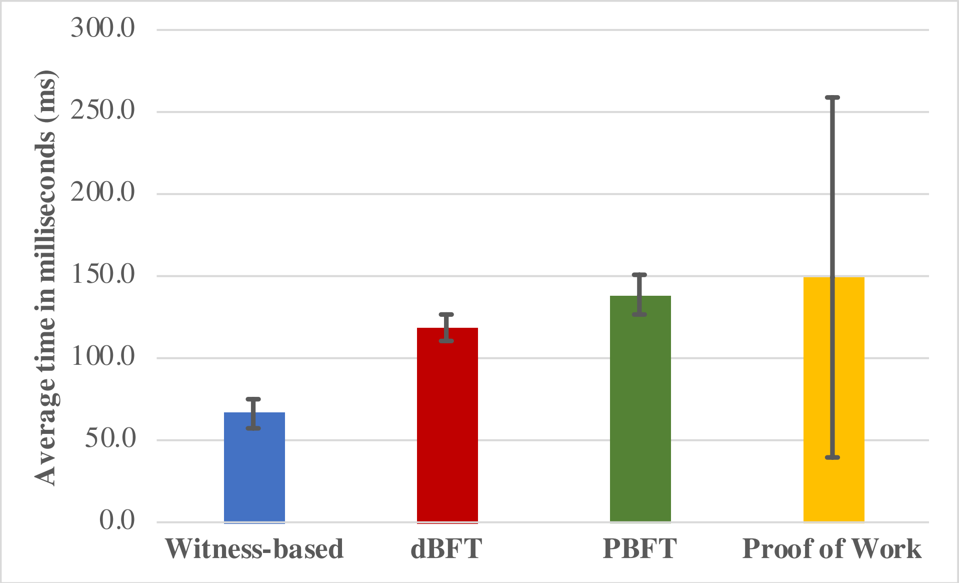


Fig. 3 **T1**: Average time to perform the consensus

the gateway performance and average latency for each transaction insertion.

All metrics represent the average time in milliseconds (ms) of ten repetitions for each scenario, and using a confidence interval of 95%.

4.2 Results

As expected, witness-based consensus presented better performance for all metrics and PoW presented the worst results. However, witness-based consensus was used as a baseline for the results and it is more likely to be affected by different attacks (e.g., Sybil attacks) in comparison to PBFT and dBFT. Additionally, PoW with 12 bits is not well suited to protect against malicious gateways. As a comparison, Bitcoin's PoW started with a difficulty of 32 bits and, currently, the difficulty is over 70 bits [6]). As shown in Fig. 3, the consensus procedure (metric **T1**) using witness-based approach was performed in 66.94±8.47ms (first bar, in blue). It is nearly the half of the time expend by the dBFT (second bar, in red) with 118.96±7.69ms, the second best evaluated consensus algorithm. PBFT (third bar, in green) achieved consensus in 138.61±12.47ms and PoW (fourth bar, in yellow) achieved consensus in 149.23±108.84ms. Even using the same number of gateways in dBFT (delegates) and PBFT to perform the consensus procedure, dBFT reduced in 15% the time compared to PBFT. Moreover, average results in PoW present a high deviation due to lottery characteristics of this consensus algorithm (finding a hash with a specific characteristic).

Considering the total time to perform consensus and propagate the block to all gateways (metric **T2**), witness-based approach had similar results to dBFT, an average of 162.69±65.51ms and 196.33±16.06ms respectively (Fig. 4). In this case, both PBFT and PoW increased in nearly twice the time required to perform **T2**, with an average of 370.71±56.51ms and 390.05±171.65ms. This shows that, considering an overall view of the blockchain network, dBFT can perform the consensus close to witness-based approach, but with much better results than PBFT. Also, it is important to note that, again, PoW presents a high deviation.

We also analysed the impact of each consensus algorithm on the performance of transactions insertion. The overhead to insert a transaction in the gateway (metric **T3**) presents very similar results in all consensus algorithms (as can be observed in Fig. 5). For all consensus algorithms, it takes between 4.10±0.24ms (obtained using witness-based) and 4.60±0.35ms (obtained using PoW).

An important aspect that should be considered in IoT environments is the latency to insert information. This can impact in the processing of a sensor reading, for example. Consequently, the time that takes to all gateways to insert produced information is crucial. In relation to this metric (T4), good results were obtained in all consensus, varying from 68.31±2.17ms (in witness-based) to 148.71±3.51ms (in PoW)). Also, dBFT (77.60±0.69ms) and PBFT (77.61±1.55ms) presented nearly the same results considering transaction latency (Fig. 6).

The evaluation presented good results in the emulated IoT scenarios with the four used consensus algorithms. However, it is important to note that witness-based approach was used only as a baseline to other results and it has some important security issues. Also, using PoW with a difficulty of 12 bits is not suitable in heterogeneous environment (where

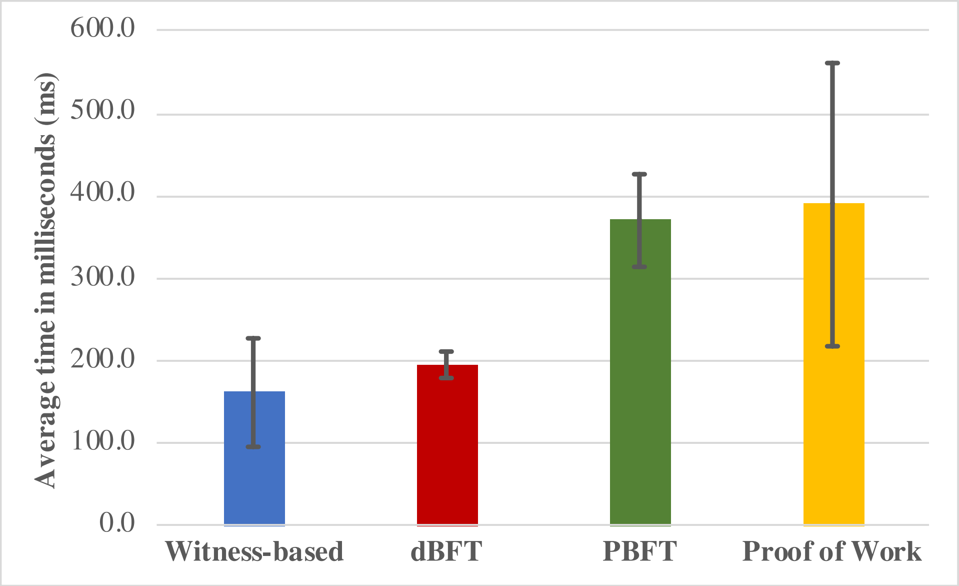


Fig. 4 **T2**: Average time to perform the consensus and propagate the block

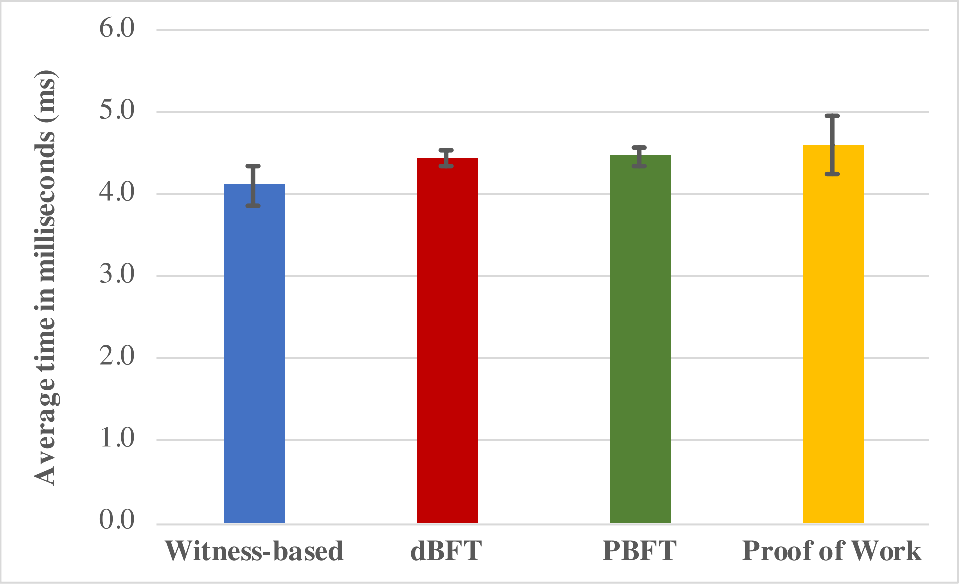


Fig. 5 **T3**: Overhead of the transaction insertion procedure

devices with high computing power can take the power of the mining procedure). Additionally, it is important to mention that the code that implements the proposed blockchain was developed using the Python programming language and is available at GitHub.[1]

In our previous work [38], the witness-based and PBFT were evaluated in a different hardware configuration. Consequently, metrics T1 and T3 present slightly different results. Furthermore, in this work we performed evaluation in four different consensus and we used two new metrics: T2 and T4.

5 Security discussion regarding consensus algorithms

In this section, we present a discussion about known attacks that could affect appendable-block blockchains and the four evaluated consensus algorithms. In order to analyse these attacks, we classified them using the stack model proposed by Zorzo et al. [55]. Even though we mention different attacks, in this paper we focus on the main attacks that

[1] https://github.com/conseg/speedychain/tree/Multilevel

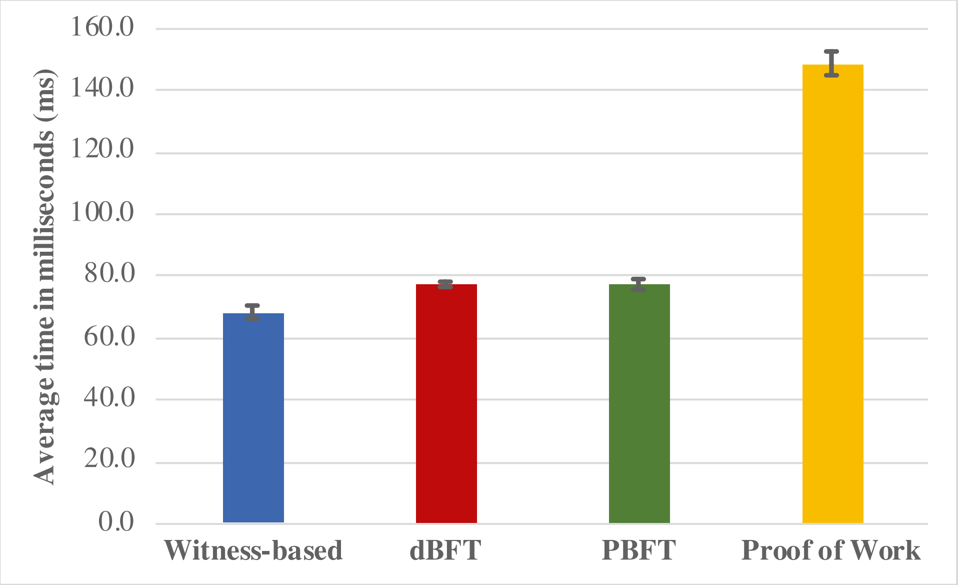


Fig. 6 **T4**: Latency to insert a transactions in all gateways

compromise the consensus layer, i.e., 51% Attack, Block-withholding, Bribery Attack, Double Spending, Finney Attack, Fork-after-withhold, Selfish Mining, Sybil Attack and Vector76 Attack. We briefly describe those attacks next.

Double Spending, Finney, Vector 76%, and Transaction Malleability attacks are aimed at spending coins in multiple transactions. In **Double Spending attack** [13], a malicious user sends multiple transactions to reachable peers in order to spend the same coin more than once. Alternatively, **Finney attack** [13] consists of a dishonest miner holding a pre-mined block, and spending the same coin that is used in a transaction of the pre-mined block. Combining these two attacks, **Vector 76% attack** [13] consists of requesting to withdraw the value of a transaction that was confirmed and sending the same value to another transaction, exploring the fork resolution algorithm (generating conflicts in the longest chain).

Many proposals that adopt blockchain in IoT scenarios [9, 15, 35, 39, 43] do not use cryptocurrencies. Consequently, Double spending, Finney, and Vector 76 and Transaction Malleability attacks are not attractive for malicious users. However, some of blockchains proposals for IoT support tokens for M2M (machine-to-machine) payments. Considering appendable-block blockchains, these attacks do not represent a threat, in particular when using dBFT and PBFT due to the voting procedure. In both consensus algorithms, sending multiple transactions with the same timestamp, signature, and information will be discarded in case of collision. Also, in case of incorrect order, the transaction will be discarded. In the case of PoW and witness-based approach, these attacks can be effective if a token structure is created to appendable-block blockchains. However, tokens were not introduced or discussed in appendable-block blockchains.

There are different attacks that explore vulnerabilities in the mining mechanism of PoW, such as 51%, Selfish Mining, Block-Withholding, Fork-After-Withholding, and Bribery attacks. The **51% attack** consists of a malicious user controlling more than 50% of network processing power, thus this user could rewrite the blockchain blocks and define the blockchain behaviour [21]. Similarly, **Selfish Mining** attack consists of a malicious user (or a pool) keeping own mined blocks private until its chain reach a length longer than the main blockchain. As per the fork rule, the attacker chain will now become the main chain [18]. **Block-Withholding** happens when a malicious miner - which is participating in a mining pool - finds a valid hash value and sends it directly to the blockchain network, thus avoiding division of the reward for mining the block [3]. Similarly, in **Fork-After-Withhold (FAW)** a malicious miner holds the block until another miner (from the same pool) identifies a block. Then, the malicious miner sends its block, forcing the pool to generate a fork (this block could be sent to multiple pools in order to increase its reward) [31]. **Bribery attack** [8] consists of a malicious user exploring the mining power of different nodes (through financial incentives) to include conflicting transactions in the blockchain (e.g., can be used to force a Double Spending). **Sybil attack** relies on a malicious node assuming multiple identities in the network with the ultimate goal of influencing the network [16]. The **Eclipse attack** consists of a malicious user aiming to monopolise the incoming and outgoing connections of a victim, thus isolating the victim from the main blockchain network [24].

Selfish Mining, **51%**, **Block-Withholding**, **FAW** and **Bribery** attacks are based on strategies adopted by PoW consensus algorithms. Consequently, choosing a solution for IoT that uses a different consensus algorithm (e.g., dBFT, PBFT, and witness-based approach) can help to avoid these kind of attacks. A key aspect to be considered is related to the hardware constraints in IoT devices, such as computing power, memory, and storage.

Biryukov et al. [4] present a **Deanonymization** technique where it is possible to identify users retrieving a list of Internet clients on different servers and linking them to transactions in the blockchain. However, appendable-block blockchain was proposed to be used in a private/consortium and permissioned environment. Therefore, the access to the information is managed by gateways. Consequently, this attack can be effective if a gateway is tampered to leak information maintained by the gateway.

Johnson et al. [30] presented that **Distributed Denial of Service (DDoS) attack** can be used to reduce the performance of a set of nodes in a blockchain, e.g., mining capability in Bitcoin blockchain. This attack can be effective if a PoW consensus algorithm is adopted in appendable-block blockchain due to the high hardware demand. PBFT, dBFT and witness-based approaches can be affected by DDoS performed against the network. However, this kind of attack requires to that malicious users share the access to the network (which is controlled in a private/consortium environment).

Eclipse attacks occurs when a set of malicious nodes control the communication of a node to the rest of the blockchain network nodes [25]. This attack is effective in appendable-blockchain (in any consensus algorithm adopted), particularly due to the hierarchy of nodes. However, this problem was mitigated in a previous work [37], where each device can connect and send information to multiple gateways at the same time.

6 Threats to validity & limitations

The evaluation tests were performed in a controlled environment. However, there are two main threats to validity of our evaluation. The first internal threat is the instrumentation used in the evaluation. The hardware used to perform the evaluation can impact on the obtained results. For example, this can be observed comparing the results with the previous work [38]. However, the experiments presented in this work were all performed in the same hardware. Also, differences obtained using the four consensus algorithms is expected to be reproduced in any hardware. We intend to consider different hardware in future work. The second internal threat is related to the selection of the scenario (number of gateways, devices and transactions). A different selection can influence the obtained results.

A relevant limitation present in this work is that we performed the evaluation in an emulated scenario. However, in this work we adopted the same libraries and cryptography algorithms that were used in a previous evaluation of real hardware [39]. Consequently, it is expected that IoT hardware is capable to execute the same operations that were presented in this work.

Another important limitation is that we assumed that a device can connect to another gateway. However, we did not discuss this situation in this paper. This discussion is performed in another work [37].

7 Final considerations & future work

Due to the popularisation of smart devices (e.g., IP Cameras, smart TVs, vacuum cleaners), data security in IoT became critical in many different domains (e.g, smart grids, smart cities, smart healthcare). Appendable-block blockchain has the potential to be one of the blockchains solutions used in this kind of environment. Particularly, this blockchain can help to provide data integrity and resilience to the system, at the same time that provides a small data insertion latency (milliseconds). However, different consensus algorithms provide different performance results and they can present particular security issues. To tackle this, we presented a comparison of different consensus algorithms focusing on evaluating its performance and discussing the main security threats.

This work showed that it is possible to use different consensus algorithms in appendable-block blockchains with acceptable (in all cases) performance results. In this work, we evaluated dBFT, PBFT, PoW and a witness-based (as a baseline) consensus algorithms. In particular, dBFT seems to be an interesting approach with good performance and security. The consensus using dBFT can be achieved (in average) under 200ms. Also, the latency to propagate a transaction to all gateways (in average) using dBFT was kept under 80ms.

Additionally, we discussed the most important known attacks to blockchain. We discussed how appendable-block blockchains are affected by them when using the four evaluated consensus algorithms. In particular, many attacks are effective to PoW consensus algorithms. Additionally, it was observed that malicious gateways could interfere or delay the transaction inclusion in the blockchain. This issue is discussed in a different work [37].

As future work, we intend to model the appendable-block blockchain in a blockchain simulator (e.g., BlockSim [2]). The evaluation through simulation can help to increase the scale of the scenarios. Also, further discussion can help to improve reliability and security of insertion of transactions. Some improvements, considering different consensus algorithms for each context [37] and performance evaluation using multiple consensus in different contexts, will be discussed in a future work.

Acknowledgements This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. Avelino F. Zorzo is supported by CNPq (315192/2018-6). This work was supported by the INCT Forensic Sciences through the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq − process # 465450/2014-8). The work was supported by IFRS. The work has been supported by the Cyber Security Research Centre Limited whose activities are partially funded by the Australian Government's Cooperative Research Centres Programme.

References

1. Ahrenholz J, Danilov C, Henderson TR, Kim JH (2008) CORE: A real-time network emulator. In: 27th IEEE military communications conference (MILCOM 2008). pp. 1–7
2. Alharby M, van Moorsel A (2020) Blocksim: An extensible simulation tool for blockchain systems. Frontiers in Blockchain 3:28
3. Bag S, Ruj S, Sakurai K (2017) Bitcoin block withholding attack: Analysis and mitigation. IEEE Transactions on Information Forensics and Security 12(8):1967–1978
4. Biryukov A, Khovratovich D, Pustogarov I (2014) Deanonymisation of clients in bitcoin p2p network. In: Proceedings of the 2014 ACM SIGSAC conference on computer and communications security, CCS '14. ACM, New York, NY, USA, pp 15–29
5. Biswas S, Sharif K, Li F, Maharjan S, Mohanty SP, Wang Y (2020) Pobt: A lightweight consensus algorithm for scalable IoT business blockchain. IEEE Internet of Things Journal 7(3):2343–2355
6. Blockchain: Blockchain block explorer (2021). https://www.blockchain.com/pt/explorer
7. Bocek T, Rodrigues BB, Strasser T, Stiller B (2017) Blockchains everywhere - a use-case of blockchains in the pharma supply-chain. In: 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM). pp 772–777
8. Bonneau J (2016) Why buy when you can rent? In: Financial cryptography and data security. Springer, Berlin, pp 19–26
9. Boudguiga A, Bouzerna N, Granboulan L, Olivereau A, Quesnel F, Roger A, Sirdey R (2017) Towards better availability and accountability for IoT updates by means of a blockchain. In: 2017 IEEE European symposium on security and privacy workshops. pp 50–58
10. Cao B, Li Y, Zhang L, Zhang L, Mumtaz S, Zhou Z, Peng M (2019) When Internet of Things meets blockchain: Challenges in distributed consensus. IEEE Network 33(6):133–139
11. Chaudhary R, Aujla GS, Garg S, Kumar N, Rodrigues JJPC (2018) SDN-enabled multi-attribute-based secure communication for smart grid in IIoT environment. IEEE Transactions on Industrial Informatics 14:2629–2640
12. Christidis K, Devetsikiotis M (2016) Blockchains and smart contracts for the internet of things. IEEE Access 4:2292–2303
13. Conti M, E, SK, Lal C, Ruj S (2018) A survey on security and privacy issues of bitcoin. IEEE Communications Surveys Tutorials pp. 1–1
14. Dorri A, Kanhere SS, Jurdak R (2017) Towards an optimized blockchain for IoT. In: 2017 second international conference on internet-of-things design and implementation (IoTDI). ACM, pp. 173–178
15. Dorri A, Kanhere SS, Jurdak R, Gauravaram P (2017) Blockchain for IoT security and privacy: The case study of a smart home. In: 2017 IEEE international conference on pervasive computing and communications workshops (PerCom Workshops). pp 618–623
16. Douceur JR (2002) The sybil attack. In: Revised papers from the first international workshop on peer-to-peer systems, IPTPS '01. Springer-Verlag, London, pp 251–260
17. Esposito C, Ficco M, Gupta BB (2021) Blockchain-based authentication and authorization for smart city applications. Information Processing & Management 58(2):102468
18. Eyal I, Sirer EG (2014) Majority is not enough: Bitcoin mining is vulnerable. Financial cryptography and data security. Springer, Berlin, pp 436–454
19. Fan X, Chai Q (2018) Roll-dpos: A randomized delegated proof of stake scheme for scalable blockchain-based internet of things systems. 15th EAI international conference on mobile and ubiquitous systems: Computing, networking and services, MobiQuitous '18. ACM, New York, pp 482–484
20. Feng L, Zhang H, Lou L, Chen Y (2018) A blockchain-based collocation storage architecture for data security process platform of WSN. In: 2018 IEEE 22nd international conference on computer supported cooperative work in design ((CSCWD)). pp 75–80
21. Gervais A, Karame GO, Wüst K, Glykantzis V, Ritzdorf H, Capkun S (2016) On the security and performance of proof of work blockchains. 2016 ACM SIGSAC conference on computer and communications security, CCS '16. ACM, New York, pp 3–16
22. Guan Z, Si G, Zhang X, Wu L, Guizani N, Du X, Ma Y (2018) Privacy-preserving and efficient aggregation based on blockchain for power grid communications in smart communities. IEEE Communications Magazine 56(7):82–88
23. Han R, Gramoli V, Xu X (2018) Evaluating blockchains for IoT. In: 2018 9th IFIP international conference on new technologies, mobility and security (NTMS). pp 1–5
24. Heilman E, Kendler A, Zohar A, Goldberg S (2015) Eclipse attacks on bitcoin's peer-to-peer network. 24th USENIX security symposium. USENIX Association, Washington, D.C., pp 129–144
25. Henningsen S, Teunis D, Florian M, Scheuermann B (2019) Eclipsing ethereum peers with false friends. In: 2019 IEEE European symposium on security and privacy workshops (EuroS PW). pp 300–309
26. Huang J, Kong L, Chen G, Wu M, Liu X, Zeng P (2019) Towards secure industrial IoT: Blockchain system with credit-based consensus mechanism. IEEE Transactions on Industrial Informatics 15(6):3680–3689
27. Huh S, Cho S, Kim S (2017) Managing iot devices using blockchain platform. In: 2017 19th international conference on advanced communication technology (ICACT). pp 464–467
28. Jin J, Gubbi J, Marusic S, Palaniswami M (2014) An information framework for creating a smart city through internet of Things. IEEE Internet of Things Journal 1(2):112–121
29. Jing Q, Vasilakos AV, Wan J, Lu J, Qiu D (2014) Security of the Internet of Things: perspectives and challenges. Wireless Networks 20(8):2481–2501
30. Johnson B, Laszka A, Grossklags J, Vasek M, Moore T (2014) Game-theoretic analysis of ddos attacks against bitcoin mining pools. In: Böhme R, Brenner M, Moore T, Smith M (eds) Financial cryptography and data security. Springer, Berlin, pp 72–86
31. Kwon Y, Kim D, Son Y, Vasserman E, Kim Y (2017) Be selfish and avoid dilemmas: Fork after withholding (faw) attacks on bitcoin. In: Proceedings of the 2017 ACM SIGSAC conference on computer and communications security, CCS '17. ACM, New York, pp 195–209
32. Lao L, Li Z, Hou S, Xiao B, Guo S, Yang Y (2020) A survey of IoT applications in blockchain systems: Architecture, consensus, and traffic modeling. ACM Comput Surv 53(1):1–32
33. Lei A, Cruickshank H, Cao Y, Asuquo P, Ogah CPA, Sun Z (2017) Blockchain-based dynamic key management for heterogeneous intelligent transportation systems. IEEE Internet of Things Journal 4(6):1832–1843
34. Li C, Zhang J, Yang X, Youlong L (2021) Lightweight blockchain consensus mechanism and storage optimization for resource-constrained IoT devices. Information Processing & Management 58(4):102602

35. Li C, Zhang L (2017) A blockchain based new secure multi-layer network model for internet of things. In: 2017 IEEE International Congress on Internet of Things (ICIOT). pp 33–41
36. Linux Foundation: Hyperledger (2020). https://github.com/hyperledger
37. Lunardi RC, Alharby M, Nunes HC, Dong C, Zorzo AF, van Moorsel A (2020) Context-based consensus for appendable-block blockchains. In: 2020 IEEE international conference on blockchain (Blockchain). pp 401–408
38. Lunardi RC, Michelin RA, Neu CV, Nunes HC, Zorzo AF, Kanhere SS (2019) Impact of consensus on appendable-block blockchain for IoT. In: 16th EAI international conference on mobile and ubiquitous systems: Computing, networking and services (MobiQuitous). Association for Computing Machinery, pp 228–237
39. Lunardi RC, Michelin RA, Neu CV, Zorzo AF (2018) Distributed access control on IoT ledger-based architecture. In: 2018 IEEE/IFIP network operations and management symposium (NOMS). pp 1–7
40. Maitra S, Yanambaka VP, Abdelgawad A, Puthal D, Yelamarthi K (2020) Proof-of-authentication consensus algorithm: Blockchain-based IoT implementation. In: 2020 IEEE 6th World Forum on Internet of Things (WF-IoT). pp 1–2
41. Malik S, Dedeoglu V, Kanhere SS, Jurdak R (2019) Trustchain: Trust management in blockchain and iot supported supply chains. In: 2019 IEEE international conference on blockchain (Blockchain). pp 184–193
42. Michelin RA, Ahmed N, Kanhere SS, Seneviratne A, Jha S (2020) Leveraging lightweight blockchain to establish data integrity for surveillance cameras. In: 2020 IEEE international conference on blockchain and cryptocurrency (ICBC). pp 1–3
43. Michelin RA, Dorri A, Steger M, Lunardi RC, Kanhere SS, Jurdak R, Zorzo AF (2018) Speedychain: A framework for decoupling data from blockchain for smart cities. 2018 15th EAI international conference on mobile and ubiquitous systems: Computing, networking and services (MobiQuitous). ACM, New York, pp 145–154
44. Mubarakali A (2020) Healthcare services monitoring in cloud using secure and robust healthcare-based blockchain(srhb) approach. Mobile Networks and Applications 25:1330–1337
45. Novo O (2018) Blockchain meets IoT: An architecture for scalable access management in IoT. IEEE Internet of Things Journal 5(2):1184–1195
46. Nunes HC, Lunardi RC, Zorzo AF, Michelin RA, Kanhere SS (2020) Context-based smart contracts for appendable-block blockchains. In: 2020 IEEE international conference on blockchain and cryptocurrency (ICBC). pp 1–9
47. Pinno OJA, Gregio ARA, Bona LCED (2017) Controlchain: Blockchain as a central enabler for access control authorizations in the IoT. In: GLOBECOM 2017 - 2017 IEEE global communications conference. pp 1–6
48. Puthal D, Mohanty SP (2019) Proof of authentication: IoT-friendly blockchains. IEEE Potentials 38(1):26–29
49. Qiu T, Chen N, Li K, Atiquzzaman M, Zhao W (2018) How can heterogeneous internet of things build our future: A survey. IEEE Communications Surveys Tutorials 20(3):2011–2027
50. Rathee G, Ahmad F, Sandhu R, Kerrache CA, Azad MA (2021) On the design and implementation of a secure blockchain-based hybrid framework for industrial internet-of-things. Information Processing & Management 58(3):102526
51. Sukhwani H, Martínez JM, Chang X, Trivedi KS, Rindos A (2017) Performance modeling of pbft consensus process for permissioned blockchain network (hyperledger fabric). In: 2017 IEEE 36th symposium on reliable distributed systems (SRDS). pp 253–255
52. Wang S, Ouyang L, Yuan Y, Ni X, Han X, Wang F (2019) Blockchain-enabled smart contracts: Architecture, applications, and future trends. IEEE Transactions on Systems, Man, and Cybernetics: Systems 49(11):2266–2277
53. Yang J, Lu Z, Wu J (2018) Smart-toy-edge-computing-oriented data exchange based on blockchain. Journal of Systems Architecture 87:36–48
54. Zhao N, Wu H, Zhao X (2020) Consortium blockchain-based secure software defined vehicular network. Mobile Networks and Applications 25(1):314–327
55. Zorzo AF, Nunes HC, Lunardi RC, Michelin RA, Kanhere SS (2018) Dependable IoT using blockchain-based technology. In: 2018 Eighth latin-american symposium on dependable computing (LADC), pp 1–9

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.