Impact of consensus on appendable-block blockchain for IoT

Roben C. Lunardi*
IFRS and PUCRS
Brazil
roben.lunardi@acad.pucrs.br

Regio A. Michelin
CSE, UNSW and Cyber Security CRC
Australia
regio.michelin@cybersecuritycrc.org.au

Charles V. Neu
UNISC and PUCRS
Brazil
charlesneu@gmail.com

Henry C. Nunes
PUCRS
Brazil
henry.nunes@acad.pucrs.br

Avelino F. Zorzo
PUCRS
Brazil
avelino.zorzo@pucrs.br

Salil S. Kanhere
CSE, UNSW
Australia
salil.kanhere@unsw.edu.au

ABSTRACT

The Internet of Things (IoT) is transforming our physical world into a complex and dynamic system of connected devices on an unprecedented scale. Connecting everyday physical objects is creating new business models, improving processes and reducing costs and risks. Recently, blockchain technology has received a lot of attention from the community as a possible solution to overcome security issues in IoT. However, traditional blockchains (such as the ones used in Bitcoin and Ethereum) are not well suited to the resource-constrained nature of IoT devices and also with the large volume of information that is expected to be generated from typical IoT deployments. To overcome these issues, several researchers have presented lightweight instances of blockchains tailored for IoT. For example, proposing novel data structures based on blocks with decoupled and appendable data. However, these researchers did not discuss how the consensus algorithm would impact their solutions, *i.e.*, the decision of which consensus algorithm would be better suited was left as an open issue. In this paper, we improved an appendable-block blockchain framework to support different consensus algorithms through a modular design. We evaluated the performance of this improved version in different emulated scenarios and studied the impact of varying the number of devices and transactions and employing different consensus algorithms. Even adopting different consensus algorithms, results indicate that the latency to append a new block is less than 161ms (in the more demanding scenario) and the delay for processing a new transaction is less than 7ms, suggesting that our improved version of the appendable-block blockchain is efficient and scalable, and thus well suited for IoT scenarios.

---
*The first and second authors have the same contribution for the present research.

1 INTRODUCTION

The Internet of Things (IoT) refers to the tight integration of devices that are connected to sense, monitor and control processes encompassing various application domains, such as smart homes and smart cities [3]. The IoT is transforming our physical world into a complex and dynamic system of connected devices on an unprecedented scale. Also, it is expected that the widespread adoption of IoT will increase productivity, safety, efficiency and accuracy in different sectors, such as smart factors, supply chain, and health care [43].

Despite the expected benefits, IoT systems potentially present considerable safety and security risks, as they can be used in critical infrastructures such as energy, smart cities and health care. These systems are often a primary target for cybernetic attacks since it is possible to cause significant damage to critical infrastructure and even human lives. Thereby, IoT brings new challenges of network management, overhead in computation, data management and security requirements that need to be addressed efficiently for the large and sensitive amount of data being produced by an ever increasing number of devices, sensors and systems connected together [13].

In recent years, several researchers [30] [41] [18] [11] [36] [40] have proposed different solutions that use the blockchain technology in IoT to solve security issues. Some works propose novel blockchain architectures [18] [11], while others propose innovative blockchain data management solutions [36] [40]. However, few blockchain proposals for IoT present a modular framework that

can be adapted in different scenarios or easily changed to support different consensus algorithms.

Consequently, existing research has not addressed the following key challenges: (*i*) a blockchain solution that provides a fast response (few milliseconds) to insert and retrieve data from multiple devices; (*ii*) investigation into the impact of known blockchain attacks in IoT environments; and, (*iii*) deliberation about consensus algorithms and their impact on the IoT context.

In order to fill this gap, the goals of this work are: (*i*) a formalization for a modular lightweight blockchain that can be used in gateway-based IoT architecture; (*ii*) a discussion about main security issues in blockchain and how they impact the proposed blockchain considering an IoT scenario; and, (*iii*) an evaluation of the an improved version of an appendable-block blockchain [38] using two different consensus algorithms to demonstrate the trade-offs of different consensus in the IoT context.

The remainder of this work is organized as follows. Section 2 presents a background on recent works about blockchain in IoT and important discussions about consensus algorithms for blockchain in IoT. Section 3 discusses the main attacks on blockchains and how they affect and explore consensus algorithms mechanisms. Section 4 presents a formal definition for appendable-block blockchains. Section 4 also presents a modular solution to allow the usage of different consensus algorithms for IoT scenarios. A performance evaluation, comparing two consensus algorithms, is presented in Section 5, showing the impact of each one in the proposed framework. Section 6 presents the threats to validity of the proposed framework. Finally, we draw some conclusions in Section 7.

2 BACKGROUND

Blockchain was introduced by Bitcoin [39] to ensure a resilient and collaborative solution and to allow transactions between different peers with non-repudiation and tamper-resistance data [45]. In the last few years, several blockchain instances have been proposed [4] with different purposes, such as: Domain Name System [46], Supply Chain [9], Vehicular Networks [19] [47] , and Smart Grids [2] [27]. To be used in different domains, these blockchain solutions can consider the usage of different cryptography algorithms, consensus algorithms, data management and block structures.

2.1 Blockchain Layers

In order to help understanding these differences and how they impact a blockchain, Zorzo *et al.* [48] categorized blockchain components into four layers: Communication, Consensus, Data and Application. The "Communication" layer represents how the nodes in a blockchain communicate and exchange information. This layer defines the communication protocols, P2P architectures, and network infrastructure used by a blockchain.

Additionally, the "Consensus" layer encompasses the mechanisms for validating the candidate blocks before inserting them into the ledger and broadcasting that to other peers. The consensus algorithm is required in IoT context since the network is public, and usually there is no trust among peers.

The "Data" layer presents the blockchain information structure. This layer specifies the adopted cryptography algorithms, how data are stored, how the access to these data is performed, and how data are replicated. Additionally, there are some different approaches for the data types that are stored in a blockchain.

Moreover, there are different ways to use a blockchain. The "Application" layer defines the APIs for using data from a blockchain. For example, there are different ways to access data [11], to use coins [39], to generate tokens [24], to execute a distributed application [46], to use an identity management [35], to execute smart contracts [5].

Li *et al.* [34] presented a discussion and possible solutions to use blockchain in IoT. They proposed a solution focusing on the "Communication" layer [48] using P2P architecture that uses a mechanism called satellite chains, which use validating peers to share information between these chains. Furthermore, they propose integration with Hyperledger Fabric [12]. However, they do not present evaluation of the performance results, nor security analysis of the proposed solution. Consequently, it is hard to evaluate in which scenario their work could be applied to.

Boudguiga *et al.* [11] focused on the "Application" layer of the blockchain, employing blockchain to perform access control in the context of IoT. Moreover, they present a discussion about the application of their proposal in different scenarios in which IoT is used, such as Smart Homes, Smart Grids, and Industry 4.0. They also presented an infrastructure based in a Blockchain-as-a-Service (BaaS) that is able to improve the application performance. However, their paper does not present practical experiments to support the evaluation, nor the blockchain data management is considered in the research.

Focusing on the "Consensus" layer, Feng *et al.* [23] proposed an Hierarchical Byzantine Fault Tolerant consensus algorithm in order to solve the scale issues presented by 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 [18][36]. However, they do not present the evaluated architecture nor present how they implemented their solution.

Focusing on the architecture of the "Communication" layer, Dorri *et al.* [18] proposed a solution where overlays control the access to data stored in a blockchain shared among different overlays. In this architecture, an overlay has enough computing power to maintain a blockchain and IoT devices are not exposed to common attacks such as Distributed Denial of Service (DDoS) and Dropping Attack [32].

In a similar architecture, Lunardi *et al.* [36] proposed the adoption of gateways (limited hardware, however with enough power to maintain a blockchain). Additionally, they presented a different solution for the "Data" layer, where they introduced the concept of appendable blocks, *i.e.*, a block can continue to be appended with information after it has been inserted into the blockchain. Also considering the "Data" layer, in a different work, Dorri *et al.* [17] proposed deletion of blocks in the blockchain. These two works [36] [17] can help to reduce the amount of data that is managed by nodes in a blockchain, which is important in environments that produce large amount of data.

Additionally, a framework called SpeedyChain [38] presents a blockchain to be used in Smart Cities scenarios. Also, to improve the "Data" layer, SpeedyChain contains a mechanism to control

the amount of information inside a block (using expiration of the public key) and a mechanism to detach the payload from the block.

2.2 Consensus for Blockchain in IoT

The consensus algorithm plays a crucial role for ensuring that each new block contains valid information, and any peer is able to verify the information in the blockchain. The unreliable peer environment, where a blockchain is being executed, could be considered in order to provide a solution to a common authentication problem, which is related to have a third party involved. In an authentication context, the third party is responsible to assure the trust in each party involved in the authentication process.

A important discussion about consensus algorithms is presented in Christidis *et al.* [14] research. This research investigated different consensus algorithms such as Sieve, Practical Byzantine Fault Tolerant (PBFT), Proof-of-Stake and Proof-of-Work. Christidis *et al.* consider that both the network in which it will be used and the attack vector that is intended to be mitigated 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.* research presented blockchain in IoT as a facilitator service to share and to disseminate cryptography as a workflow. Despite the application examples evaluated in their research, none of them applied blockchain as an identity manager or authentication service.

PBFT can have some interesting characteristics for IoT environments: it does not require devices with high processing power and it does not require coins or tokens. In order to evaluate the performance of PBFT in IoT scenario, Sukhwani *et al.* [44] modeled PBFT using Stochastic Reward Nets. In this study (performed with up to 100 nodes), initial results showed that PBFT can be a problem in a large scenario and is affected both by the number of nodes and by the number of transactions.

Han *et al.* [28] presented an evaluation of different Byzantine based consensus algorithms used in blockchain for IoT scenarios. They evaluated PBFT (in Hyperledger Fabric v0.6 blockchain), Ripple, and Byzantine Fault-Tolerant State Machine Replication (BFT-SMaRt) consensus algorithms using the same workload. The results shown that for high a number of requests and high number of nodes (simulating an IoT scenario) the throughput is drastically affected.

As mentioned previously, Feng *et al.* [23] proposed an Hierarchical Byzantine Fault Tolerant consensus algorithm. Although it is an interesting approach, it can be also expressed by Gateways of networks performing a BFT based consensus algorithm (that was also proposed by other works [16][36]).

Fan *et al.* [22] proposed Roll-DPoS, an adapted version of Distributed Proof-of-Stake (DPoS) designed for IoT environments. In their algorithm, for each block generation, a node is responsible to produce the block, and send it to be validate by the other nodes that are participating in consensus. 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 made about performance, security issues or its applicability in dynamic scenarios.

While existing research presented important improvements in the state of the art, few discussions were presented in relation to the security analysis of the proposed solutions and how the "Consensus" layer choice impact the performance of a blockchain for IoT. Additionally, there are no discussion about the impact of consensus algorithms in appendable-block blockchains. Consequently, we present, in the next sections, some advances to fill these gaps.

3 SECURITY ISSUES REGARDING CONSENSUS ALGORITHMS

In this section, we present a discussion about known attacks that could be performed on blockchains and analyze their impact in an IoT setting. In order to analyze these attacks, we classified them using the stack model proposed by Zorzo *et al.* [48], as described in Section 2, and arranged the model to different threats in Table 1. Even though we mention different attacks, regarding this paper, we focus on the main attacks that 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** [15], a malicious user sends multiple transactions to reachable peers in order to spend the same coin more than once. Alternatively, **Finney attack** [15] 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** [15] 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 conflict in the longest chain).

Many proposals that adopt blockchain in IoT scenarios [34] [11] [18] [36] [38] do not use cryptocurrencies. Consequently, Double spending, Finney, and Vector 76 attacks are not attractive for malicious users. For example, in the case of SpeedyChain [38], an appendable-block blockchain, these attacks do not represent a threat as sending multiple transactions with the same timestamp, signature, and information will be discarded in case of collision or in case of incorrect order, the transaction will be discarded.

There are different attacks that explores vulnerabilities in the mining mechanism of Proof-of-Work (consensus algorithm), 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 behavior [26]. Similarly, **Selfish Mining** attack consists of a malicious user (or a pool), keeping own mined blocks private until its chain reach a longer length than the main blockchain. As per the fork rule, the attacker chain will now become the main chain [21]. **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 [6]. 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

Table 1: Most common security issues for blockchains

Threat	Layer	Cause
Double Spending	Consensus, Data, Application	Concurrency and delay to insert new transactions
Finney Attack	Consensus, Data, Application	Concurrency and consensus algorithm
Vector76 Attack	Consensus, Data, Application	Concurrency, mining process and consensus algorithm
51% Attack	Consensus	Consensus algorithm based on computing power
Selfish Mining	Consensus, Communication	Fork solving algorithm
Block-withholding	Consensus	Mining pool reward mechanism
Fork-after-withhold (FAW)	Consensus	Fork solving and mining pool reward mechanism
Bribery Attack	Consensus	Consensus and fork solving algorithm
Deanonymization	Communication, Data	P2P connections and public key reuse
DDoS Attack	Communication	Consume target resources
Transaction Malleability	Data	Bitcoin blockchain transaction id usage
Sybil Attack	Consensus, Communication	P2P network and the ability to create multiple identities
Eclipse Attack	Communication	Network isolation
Smart Contracts Vulnerabilities	Application	Bad programming practices and smart contract errors

do multiples pool in order to increase its reward) [33]. **Bribery attack** [10] 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 multiples identities in the network with the ultimate goal of influencing the network [20]. The **Eclipse attack** consists of a malicious user aiming to monopolize the incoming and outgoing connections of a victim, thus isolating the victim from the main blockchain network [29].

Selfish Mining, 51%, Block-Withholding, FAW and Bribery attacks are based on strategies adopted by PoW (Proof-of-Work) consensus algorithms. Consequently, choosing a solution for IoT that uses a different consensus algorithm (*e.g.*, PBFT) 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. In order to solve these issues, we proposed

(see Section 4.3.4) and evaluated (see Section 5) in this paper the use of two different consensus algorithms for IoT environments.

4 APPENDABLE-BLOCK BLOCKCHAIN IN IOT

In this section, we present the fundamental concepts of a blockchain architecture that underpins our proposed framework.

The proposed framework was designed using a layer-based IoT architecture [31] - similar to that is adopted in Lunardi *et al.* [36], Dorri *et al.* [18] and Michelin *et al.* [38] - that is composed by: (*i*) devices (*D* in Figure 1) in the Perception Layer; (*ii*) Gateways (*G*) in the Transportation Layer; and (*iii*) Service Providers (*SP*) in the Application Layer. Therefore, each device can produce information and send to the gateways to append data to its own block. Consequently, devices can keep producing and appending information into blockchain independently to the other devices operations. Service Providers can access the information from a device (that it is stored in the blockchain) through the gateways.
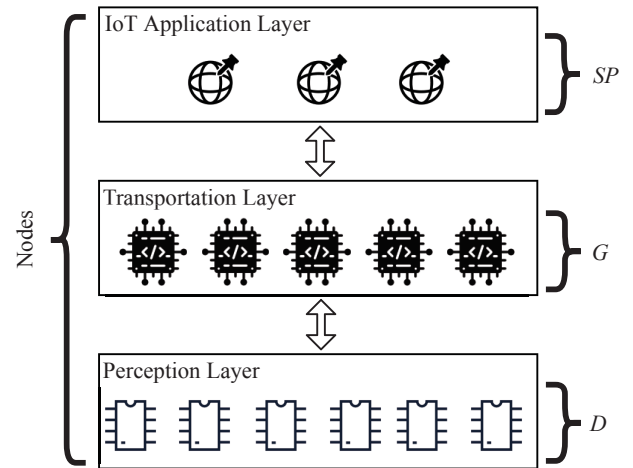
Figure 1: Main IoT nodes

It is important to note that this work focuses on the blockchain that is maintained in the Transportation Layer presented in the IoT architecture (Figure 1). Also, it is important to note that this work uses concepts that were presented in other works [36] [38], but adapted them for a more dynamic scenario using a modular blockchain. Consequently, consensus algorithms can be used based on each IoT requirement. 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). The proposed solution provides an API to applications for internal (*e.g.*: logs, alerts, and logistics) or external usage (*e.g.* providing APIs for partners applications).

4.1 Architecture

Let $N = \{N_1, ..., N_n\}$ be the set of n nodes in the system with public-private key pairs (NPK_i, NSK_i). Also, consider that these nodes can

have different roles in the architecture. Consequently, this system is composed by d devices, where $D = \{D_1, ..., D_d\}$, that usually produce information and could be controlled remotely; g gateways, where $G = \{G_1, ..., G_g\}$, that manage the access to information in a blockchain; not limited to this, different kind of nodes are supported such as s service providers $SP = \{SP_1, ..., SP_s\}$. Therefore, $N = D \cup G \cup SP$.

Assume that all nodes in N can use the same cryptography algorithms. Moreover, every NPK_i should be different and accessible by any participant in this system. Also, assume that a key pair (public and secret keys) from a device will be represented as (DPK_j, DSK_j) and a key pair from gateway will be represented as (GPK_h, GSK_h). Consider that each device in D (Perception Layer) should be connected to a gateway in G (Transportation Layer) through different (wired or wireless) network devices. Additionally, the gateways are responsible to manage the device access and provide an API that allows to manage the blockchain.

4.2 Blockchain Definition

Based on the IoT architecture presented in Figure 1, the blockchain will be maintained by gateways in G (Transportation Layer in Figure 1). To ensure that every participant can access any NPK_i (e.g., DPK_j or GPK_h) and information stored in a Gateway was not tampered with, let a blockchain $B = \{B_1, ..., B_b\}$ be a set of b blocks. Each B_k has a pair of different information (BH_k, BL_k), where BH_k is responsible to maintain the block header of B_k and the BL_k stores the block ledger, i.e., the set of transactions of B_k as shown in details in Figure 2.
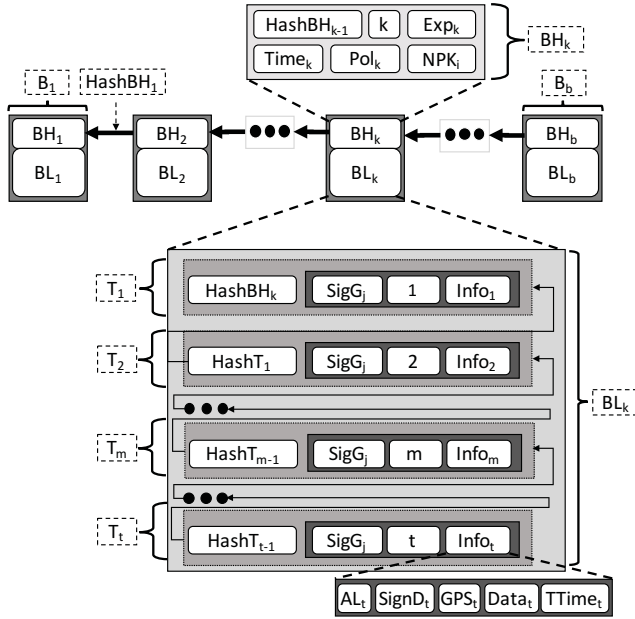

Figure 2: Main blockchain components.

Therefore, BH_k is composed by ($HashBH_{k-1}$, k, $Time_k$, Exp_k, Pol_k, NPK_i), where

$$HashBH_{k-1} = \begin{cases} 0 & \text{, when } k = 1 \\ \text{hash digest of } BH_{k-1} & \text{, when } k \geq 2 \end{cases}$$

where hash digest is obtained through a hash function, i.e., $HashBH_{k-1}$ contains the hash digest of previous block header (or zero when it is the first block); k is equal to the index of the block B_k in the blockchain; $Time_k$ is the timestamp from when the block was generated; Exp_k presents the threshold time to insert a new transaction in its block ledger, for example, after this time a device should create a new key pair (NPK, NSK) and create a new block; 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 every NPK should be available in the blockchain.

Let $BL_k = \{T_1, ..., T_t\}$ be the set of t transactions on the block ledger of the block B_k. T_m is composed by ($HashT_{m-1}$, m, $SigG_m$, $Info_m$), where

$$HashT_{m-1} = \begin{cases} \text{hash digest of } BH_k & \text{, when } m = 1 \\ \text{hash digest of } T_{m-1} & \text{, when } m \geq 2 \end{cases}$$

where the $HashT_{m-1}$ 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 is equal to the index of the transaction T_m in the block ledger BL_k; $SigG_m$ represents the result of the cryptography using the GPK_h to sign $Info_m$.

The $Info_m$ can be different for each type of node. Devices provide a set of information ($SigD_m$, AL_m, GPS_m, $Data_m$, $TTime_m$), 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 $Data_m$ 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.

4.3 Main Operations

The main operations that can be performed in the proposed blockchain are: appending blocks, appending transactions, key update and consensus algorithm. They are detailed in the next subsections.

4.3.1 Appending blocks. Insertion of a new block B_k in blockchain B is started by a gateway (present in Transportation Layer) with the objective to include a new node (N_i) public key (NPK_i). This algorithm is performed every time that a node N_i requests a connection and its Public Key (NPK_i) is not present in the blockchain (line 1 in Algorithm 1).

After verifying that a NPK_i is not present in the blockchain, the gateway should send this new public key to perform a consensus to insert the new block (line 2). It is important to note that the consensus is performed by a Leader elected in the blockchain (see details in Section 4.3.4).

Algorithm 1 Insertion of new blocks in the blockchain

Require: Connection request and requester NPK_i
1: **if** NPK_i is not present in any BH$_j$ **then**
2:    **sendBlockToConsensus**(NPK_i)
3: **end if**

4.3.2 Appending transaction. Every time a node N_i produces new information $Info_m$ to be inserted in the blockchain, it 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) is present in a block header BH_i from blockchain B (line 1 in Algorithm 2). When, a gateway receives a new information $Info_m$, the digital signature $SigD_m$ present in $Info_m$ should be validated (lines 2 and 3) using the 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 $HashT_{m-1}$ (line 6), the index of the transaction (based on the last transaction) m (line 7), and the digital signature from the gateway that is processing the transaction $SigG_m$ (line 8) using its secret key GSK_h.

After that, the gateway creates the new transaction T_m (line 9), and the transaction can be broadcast to the other gateways (line 10).

Algorithm 2 Appending new transactions into the block ledger

Require: $Info_m$ and device NPK_i
1: **if** NPK_i is present in any BH_j **then**
2:   $result \leftarrow$ **verifySign**($NPK_i, Info_m$)
3:   **if** $result$ is **true then**
4:     $b \leftarrow$ **blockIndex**(B, NPK_i)
5:     $t \leftarrow$ **lastTransaction**(BL_b)
6:     $HashT_{m-1} \leftarrow$ **hash**(T_t)
7:     $m \leftarrow t + 1$
8:     $SigG_m \leftarrow$ **sign**($GSK_h, Info_m$)
9:     $T_m \leftarrow \{HashT_{m-1}, m, SigG_m, Info_m\}$
10:     **broadcast**(T_m, BH_b)
11:   **end if**
12: **end if**

4.3.3 Key Update. Anytime that a gateway receives a transaction with its timestamp $TTime_m$ with a higher value than the expiration time present in the origin node N_i expiration time Exp_k the gateway will execute the key update algorithm (Algorithm 3). Also, the node N_i can send to the gateway a request to update its public key NPK_i'.

In both situations, a gateway will request the node N_i its new public key NPK_i' (line 1 in the Algorithm 3). After the key validation (*e.g.*, if the key is not already in the blockchain), the gateway will append a new block into the blockchain with the new NPK_i' from node N_i (line 3).

In order to append a new block, a gateway will use Algorithm 1 presented previously. Consequently, each node will receive a new block with the new public key NPK_i' of the node N_i.

Algorithm 3 Algorithm for key update

Require: $TTime_m \geq Exp_k$ or requested by node N_i
1: $NPK_i' \leftarrow$ **requestNewKey**(NPK_i)
2: **if** NPK_i' is valid **then**
3:   **appendBlock**(NPK_i') {see Algorithm 1}
4: **end if**

4.3.4 Consensus. Usually, a blockchain was designed 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.

Currently, there are different consensus algorithms used by blockchains, such as: Proof-of-Work (PoW), Proof-of-Stake (PoS), Byzantine Fault-Tolerance (PBFT), Federated Byzantine Agreement (FBA) or delegated Byzantine Fault-Tolerance (dBFT). Furthermore, it is not possible to define a single solution that will perform better than others for any scenario.

Two different consensus algorithms are proposed, but not limited to them : (*i*) validation based on a specific number of witness, where every block should be signed by at least a predefined number of witness; and (*ii*) adapted PBFT algorithm, where more than 2/3 of the active gateways should validate and sign the block. Both consensus algorithms could be summarized in Algorithm 4.

Algorithm 4 Generic consensus algorithm

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 is 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 4). 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). It is important to note that 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).

We presented a simplified version of an algorithm to represent both PBFT and Witness-based consensus algorithms. However, we intend to evaluate other consensus algorithms in a future work, such as dBFT and FBA.

Next section presents a discussion about overhead introduced by the consensus algorithms in the improved appendable-block blockchain.

5 PERFORMANCE EVALUATION

In order to evaluate the performance of the proposed blockchain in IoT scenarios, the CORE emulator platform [1] was used. The evaluation was run on a VMware Fusion 8.5.10 with 6 processors and *12GB* of *RAM* on an Intel *i7@2.8Ghz* and *16GB* of *RAM*. We performed the evaluation using 10 gateways, where each gateway runs in a container based-virtualized machine; in 9 different scenarios (as presented in Table 2) using 100, 500 and 1000 devices connected through theses gateways (10, 50 and 100 per gateway) and 100, 500 and 1,000 transactions per device (*e.g.*, 1,000,000 transactions in Scenario I). All times presented in Table 2 represent the average time considering the whole execution in all gateways.

The Witness-based consensus was used as a baseline in terms of time to append blocks and information. As expected, it can be observed in Table. 2 that varying the consensus algorithm has impact in the performance in the task to achieve consensus on inserting a block (used to insert block header with public key of each device). For example, in Scenario A, witness-based consensus takes 58.20ms to achieve the consensus against 102.82ms using PBFT and in Scenario I (scenario with highest number of devices and transactions), witness-based consensus takes 72.47ms against 160.35ms using PBFT (more than twice the time). However, witness-based consensus is more likely to be affected by different attacks (*e.g.*, Eclipse and Sybil attacks) in comparison to PBFT.

In the other blockchain operations - for instance, time to add a new block in the leader gateway (gateway that started the consensus), as well as the time to update the blockchain, to append a new transaction in a gateway (where devices are connected to) and to update the blockchain with the new transaction - presented few or no impact using both consensus algorithms. However, the number of transactions and nodes influenced in the processing time to append a transaction in the most demanding scenario (Scenario I) takes less than 7ms to both append the transaction (4.28ms in Witness-based and 4.55ms in PBFT) and to update a new transaction in the other gateways (2.33ms in Witness-based and 2.39ms in PBFT). Also, consensus algorithm has few impact in the average time to append transactions in the block, as can be observed in Figure 3.

Additionally, it can be observed that growing the number of transactions (overload of processing in gateways) has more impact than the number of devices that a gateway is handling. For example, scenario D has half of transactions and 5 times more nodes than C, but takes almost the same time to reach the consensus for a
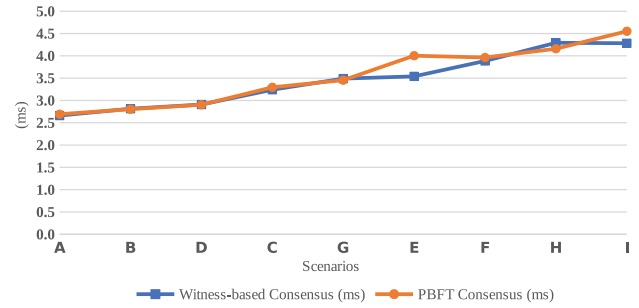

Figure 3: Average time to append a transaction in the Gateway connected to the producer Device

block. Differently, scenario F has half of nodes and 5 times more transactions than scenario G, resulting in F spending around 3% more time to achieve the consensus than G. Figure 4 presents a comparison of the time to achieve consensus of a block in different scenarios.
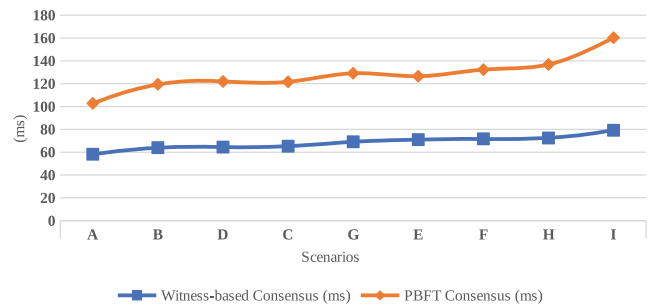

Figure 4: Average time to perform the consensus for a block

As a comparison, the Bitcoin network has around 10,000 [7] active nodes in a 24-hour slice, consequently, the experiment in Scenario I represents approximately 10% of the Bitcoin network. As a comparison, the Bitcoin has more than 150,000 confirmed transactions per day [15] with a peak of 490,644 confirmed transaction in a day [8], which means that the evaluation in Scenario I, at least represents more than twice the transactions in the Bitcoin blockchain in a day. A more effective comparison could be made with IOTA [25] - a blockchain developed for IoT - which has around 8.7 transactions per second [42]. This means around 750,000 transactions processed in a day (around 75% of the transactions processed in Scenario I). Also, it represents that IOTA transaction processing time is around 115ms. Consequently, the transactions processing time in our solution represents less than 6% of the time that is spent in IOTA - 115ms in IOTA and 7ms in our solution (4.55ms to append a transaction in a gateway summed with 2.39ms to update the entire blockchain using PBFT).

The evaluation performed in this paper presented good results in the emulated IoT scenarios with different number of devices and transactions. It is important to note that the code that implements the proposed blockchain was developed using the Python

Table 2: Performance Evaluation in 8 different scenarios

	A	B	C	D	E	F	G	H	I
Number of connected Devices per Gateway	10	10	10	50	50	50	100	100	100
Number of produced transactions per Device	100	500	1,000	100	500	1,000	100	500	1,000
Number of blocks produced (one per Device)	100	100	100	500	500	500	1,000	1,000	1,000
Number of transactions produced (total)	10,000	50,000	100,000	50,000	250,000	500,000	100,000	500,000	1,000,000
Average time to perform Witness-based consensus for each block	58.20ms	64.01ms	65.25ms	64.51ms	71.02ms	71.73ms	69.13ms	72.47ms	79.22ms
Average time to perform PBFT consensus for each block	102.82ms	119.53ms	121.68ms	121.98ms	126.56ms	132.37ms	129.14ms	136.86ms	160.35ms
Average time to insert a Block in the ledger of the Leader Gateway using Witness-based consensus	3.72ms	3.56ms	4.42ms	4.66ms	4.82ms	5.81ms	5.33ms	5.95ms	6.28ms
Average time to insert a Block in the ledger of the Leader Gateway using PBFT consensus	3.40ms	4.45ms	5.16ms	4.21ms	4.87ms	5.88ms	5.29ms	5.93ms	6.52ms
Average time to insert a Block in the ledger of the other Gateways using Witness-based consensus	0.22ms	0.22ms	0.23ms	0.22ms	0.23ms	0.23ms	0.23ms	0.24ms	0.25ms
Average time to insert a Block in the ledger of the other Gateways using PBFT consensus	0.22ms	0.22ms	0.23ms	0.23ms	0.23ms	0.26ms	0.24ms	0.24ms	0.27ms
Average time to append a Transaction in the Gateway connected to the producer Device using Witness-based consensus	2.66ms	2.82ms	2.91ms	3.24ms	3.49ms	3.54ms	3.89ms	4.29ms	4.28ms
Average time to append a Transaction in the Gateway connected to the producer Device using PBFT consensus	2.69ms	2.80ms	2.90ms	3.30ms	3.46ms	4.00ms	3.96ms	4.16ms	4.55ms
Average time to append a Transaction in each Gateway using Witness-based consensus	0.94ms	1.18ms	1.48ms	1.30ms	1.58ms	1.89ms	1.73ms	2.11ms	2.33ms
Average time to append a Transaction in each Gateway using PBFT consensus	0.94ms	1.17ms	1.47ms	1.31ms	1.55ms	2.03ms	1.73ms	2.03ms	2.39ms

programming language and a set of libraries. The code is available at GitHub [37] and could be used to replicate the experiments. In a future work, we intend to evaluate the solution in a real IoT scenario, composed by different hardware with different number of gateways and mission critical devices.

6 THREATS TO VALIDITY

In this section, we describe the threats to the validity of the results presented in the evaluation. The first threat is related to hardware capability. In this work, we did not present an evaluation with real IoT devices. However, we used the same cryptography algorithms and methods than those that were adopted by Lunardi *et al.* [36] in their experiments (using real hardware). Consequently, devices using IoT hardware should be capable to execute the same operations, but probably with a different performance.

The second threat is related to the architecture adopted and possible malicious gateways performing an Eclipse attack against some devices. Although we assumed that a device can connect to another gateway, we did not discuss this situation in this paper, and therefore, at the moment, our solution is susceptible to an Eclipse attack. This specific threat should be better addressed in future work.

Another threat that can affect the evaluation is the mobility of nodes. We did not consider in our evaluation the problems that a mobile device or gateway can produce. Hence, this threat can be further discussed in future work.

7 CONCLUSION

Industry 4.0 and Smart Cities are increasing the number of devices and the intelligence in these devices. This leads to the need for a data handling that is able to run in a decentralized scenario and at the same time to keep its integrity and resilience with a very fast response time (milliseconds), using consensus algorithms that can be adapted for IoT scenarios. To fulfill this need the proposed blockchain presents promising results (using both a simplified Witness-based and PBFT consensus) based on the evaluation performed in Section 5.

This paper also presented a modular definition of the proposed blockchain and its main operations, which is the capability to handle transactions, appending them to an existing block, and still keep data integrity. Due to this feature, the time to add transactions in a block is kept in a few milliseconds. In comparison to blockchain such as, the ones used in Bitcoin and IOTA, in the proposed blockchain time to include a transaction is considerable lower. Also, due to the proposed modularization, the evaluation was performed using two different consensus algorithms in 9 different scenarios.

A security analysis was conducted in order to discuss the most common attacks that could affect a blockchain consensus layer. It was observed that malicious gateways could interfere or delay the transaction inclusion in the blockchain. Thus, leading to an open issue, *i.e.*, to improve and mitigate attacks such as Eclipse and Sybil.

As future work, we intend to scale the presented scenario varying the number of gateways that are available. As pointed in the evaluation section, depending on the gateway processing load, the transaction processing time increases. Further discussion should be performed considering different consensus algorithms.

REFERENCES

[1] J. Ahrenholz, C. Danilov, T. R. Henderson, and J. H. Kim. 2008. CORE: A real-time network emulator. In *27th IEEE Military Communications Conference (MILCOM 2008)*. 1–7.

[2] N. Zhumabekuly Aitzhan and D. Svetinovic. 2016. Security and Privacy in Decentralized Energy Trading through Multi-signatures, Blockchain and Anonymous Messaging Streams. *IEEE Transactions on Dependable and Secure Computing* (2016), 1–1.

[3] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash. 2015. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys Tutorials* 17, 4 (Fourthquarter 2015), 2347–2376. https://doi.org/10.1109/COMST.2015.2444095

[4] Muneeb Ali, Jude Nelson, Ryan Shea, and Michael J. Freedman. 2016. Blockstack: A Global Naming and Storage System Secured by Blockchains. In *2016 USENIX Annual Technical Conference (USENIX ATC 16)*. USENIX Association, 181–194.

[5] Nicola Atzei, Massimo Bartoletti, and Tiziana Cimoli. 2017. A Survey of Attacks on Ethereum Smart Contracts SoK. In *Proceedings of the 6th International Conference on Principles of Security and Trust - Volume 10204*. 164–186.

[6] S. Bag, S. Ruj, and K. Sakurai. 2017. Bitcoin Block Withholding Attack: Analysis and Mitigation. *IEEE Transactions on Information Forensics and Security* 12, 8 (2017), 1967–1978.

[7] BITONODES. 2019. Global Bitcoin nodes distribution. https://bitnodes.earn.com/

[8] Blockchain. 2019. Confirmed Transactions Per Day in Bitcoin. https://www.blockchain.com/charts/n-transactions

[9] T. Bocek, B. B. Rodrigues, T. Strasser, and B. Stiller. 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)*. 772–777.

[10] Joseph Bonneau. 2016. Why Buy When You Can Rent?. In *Financial Cryptography and Data Security*, Jeremy Clark, Sarah Meiklejohn, Peter Y.A. Ryan, Dan Wallach, Michael Brenner, and Kurt Rohloff (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 19–26.

[11] A. Boudguiga, N. Bouzerna, L. Granboulan, A. Olivereau, F. Quesnel, A. Roger, and R. Sirdey. 2017. Towards Better Availability and Accountability for IoT Updates by Means of a Blockchain. In *2017 IEEE European Symposium on Security and Privacy Workshops (EuroS PW)*. 50–58.

[12] C. Cachin. 2016. Architecture of the Hyperledger blockchain fabric. In *Workshop on Distributed Cryptocurrencies and Consensus Ledgers, 2016.*

[13] R. Chaudhary, G. S. Aujla, S. Garg, N. Kumar, and J. J. P. C. Rodrigues. 2018. SDN-Enabled Multi-Attribute-Based Secure Communication for Smart Grid in IIoT Environment. *IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS* 14 (2018), 2629–2640.

[14] K. Christidis and M. Devetsikiotis. 2016. Blockchains and Smart Contracts for the Internet of Things. *IEEE Access* 4 (2016), 2292–2303. https://doi.org/10.1109/ACCESS.2016.2566339

[15] M. Conti, S. K. E, C. Lal, and S. Ruj. 2018. A Survey on Security and Privacy Issues of Bitcoin. *IEEE Communications Surveys Tutorials* (2018), 1–1.

[16] Ali Dorri, Salil S. Kanhere, and Raja Jurdak. 2017. Towards an Optimized Block-Chain for IoT. In *Second International Conference on Internet-of-Things Design and Implementation (IoTDI '17)*. ACM, New York, NY, USA, 173–178.

[17] Ali Dorri, Salil S. Kanhere, and Raja Jurdak. 2019. MOF-BC: A memory optimized and flexible blockchain for large scale networks. *Future Generation Computer Systems* 92 (2019), 357 – 373.

[18] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram. 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)*. 618–623.

[19] A. Dorri, M. Steger, S. S. Kanhere, and R. Jurdak. 2017. BlockChain: A Distributed Solution to Automotive Security and Privacy. *IEEE Communications Magazine* 55, 12 (2017), 119–125.

[20] John R. Douceur. 2002. The Sybil Attack. In *Revised Papers from the First International Workshop on Peer-to-Peer Systems (IPTPS '01)*. 251–260.

[21] Ittay Eyal and Emin Gün Sirer. 2014. Majority Is Not Enough: Bitcoin Mining Is Vulnerable. In *Financial Cryptography and Data Security*. Springer Berlin Heidelberg, 436–454.

[22] Xinxin Fan and Qi Chai. 2018. Roll-DPoS: A Randomized Delegated Proof of Stake Scheme for Scalable Blockchain-Based Internet of Things Systems. In *15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous '18)*. ACM, New York, NY, USA, 482–484. https://doi.org/10.1145/3286978.3287023

[23] L. Feng, H. Zhang, L. Lou, and Y. Chen. 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))*. 75–80. https://doi.org/10.1109/CSCWD.2018.8465319

[24] G. Fenu, L. Marchesi, M. Marchesi, and R. Tonelli. 2018. The ICO phenomenon and its relationships with ethereum smart contract environment. In *2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE)*. 26–32.

[25] IOTA Foundation. 2018. IOTA - Next Generation Blockchain. https://iota.org/

[26] Arthur Gervais, Ghassan O. Karame, Karl Wüst, Vasileios Glykantzis, Hubert Ritzdorf, and Srdjan Capkun. 2016. On the Security and Performance of Proof of Work Blockchains. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16)*. 3–16.

[27] Z. Guan, G. Si, X. Zhang, L. Wu, N. Guizani, X. Du, and Y. Ma. 2018. Privacy-Preserving and Efficient Aggregation Based on Blockchain for Power Grid Communications in Smart Communities. *IEEE Communications Magazine* 56, 7 (2018), 82–88.

[28] R. Han, V. Gramoli, and X. Xu. 2018. Evaluating Blockchains for IoT. In *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*. 1–5. https://doi.org/10.1109/NTMS.2018.8328736

[29] Ethan Heilman, Alison Kendler, Aviv Zohar, and Sharon Goldberg. 2015. Eclipse Attacks on Bitcoin's Peer-to-Peer Network. In *24th USENIX Security Symposium (USENIX Security 15)*. Washington, D.C., 129–144.

[30] S. Huh, S. Cho, and S. Kim. 2017. Managing IoT devices using blockchain platform. In *2017 19th International Conference on Advanced Communication Technology (ICACT)*. 464–467.

[31] Qi Jing, Athanasios V. Vasilakos, Jiafu Wan, Jingwei Lu, and Dechao Qiu. 2014. Security of the Internet of Things: perspectives and challenges. *Wireless Networks* 20, 8 (01 Nov 2014), 2481–2501.

[32] Benjamin Johnson, Aron Laszka, Jens Grossklags, Marie Vasek, and Tyler Moore. 2014. Game-Theoretic Analysis of DDoS Attacks Against Bitcoin Mining Pools. In *Financial Cryptography and Data Security*, Rainer Böhme, Michael Brenner, Tyler Moore, and Matthew Smith (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 72–86.

[33] Yujin Kwon, Dohyun Kim, Yunmok Son, Eugene Vasserman, and Yongdae Kim. 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)*. 195–209.

[34] Wenting Li, Alessandro Sforzin, Sergey Fedorov, and Ghassan O. Karame. 2017. Towards Scalable and Private Industrial Blockchains. In *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*.

[35] C. Lin, D. He, X. Huang, M. Khurram Khan, and K. K. R. Choo. 2018. A New Transitively Closed Undirected Graph Authentication Scheme for Blockchain-Based Identity Management Systems. *IEEE Access* 6 (2018), 28203–28212.

[36] R. C. Lunardi, R. A. Michelin, C. V. Neu, and A. F. Zorzo. 2018. Distributed access control on IoT ledger-based architecture. In *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*. 1–7.

[37] Roben C. Lunardi, Regio A. Michelin, and Avelino F. Zorzo. 2018. SpeedyChain Plataform. https://github.com/regio/r2ac/tree/2018consensus.

[38] Regio A. Michelin, Ali Dorri, Marco Steger, Roben C. Lunardi, Salil S. Kanhere, Raja Jurdak, and Avelino F. Zorzo. 2018. SpeedyChain: A Framework for Decoupling Data from Blockchain for Smart Cities. In *15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous '18)*. 145–154.

[39] Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system, http://bitcoin.org/bitcoin.pdf, Retrieved 26 Feb 2019.

[40] O. Novo. 2018. Blockchain Meets IoT: An Architecture for Scalable Access Management in IoT. *IEEE Internet of Things Journal* 5, 2 (2018), 1184–1195.

[41] O. J. A. Pinno, A. R. A. Gregio, and L. C. E. De Bona. 2017. ControlChain: Blockchain as a Central Enabler for Access Control Authorizations in the IoT. In *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*. 1–6.

[42] IOTA Search. 2019. IOTA Search - Transactions Overview. https://iotasear.ch/live-transactions

[43] V. Sharma, G. Choudhary, Y. Ko, and I. You. 2018. Behavior and Vulnerability Assessment of Drones-Enabled Industrial Internet of Things (IIoT). *IEEE Access* 6 (2018), 43368–43383.

[44] H. Sukhwani, J. M. Martἦnez, X. Chang, K. S. Trivedi, and A. Rindos. 2017. Performance Modeling of PBFT Consensus Process for Permissioned Blockchain Network (Hyperledger Fabric). In *2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS)*. 253–255. https://doi.org/10.1109/SRDS.2017.36

[45] F. Tschorsch and B. Scheuermann. 2016. Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies. *IEEE Communications Surveys Tutorials* 18, 3 (2016), 2084–2123.

[46] X. Wang, K. Li, H. Li, Y. Li, and Z. Liang. 2017. ConsortiumDNS: A Distributed Domain Name Service Based on Consortium Chain. In *2017 IEEE 19th International Conference on High Performance Computing and Communications; IEEE 15th International Conference on Smart City; IEEE 3rd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*. 617–620.

[47] Z. Yang, K. Yang, L. Lei, K. Zheng, and V. C. M. Leung. 2018. Blockchain-based Decentralized Trust Management in Vehicular Networks. *IEEE Internet of Things Journal* (2018), 1–1.

[48] A. F. Zorzo, H. C. Nunes, R. C. Lunardi, R. A. Michelin, and S. S. Kanhere. 2018. Dependable IoT Using Blockchain-Based Technology. In *2018 Eighth Latin-American Symposium on Dependable Computing (LADC)*. 1–9.