HyBloSE: Hybrid Blockchain for Secure-by-Design Smart Environments

Gaia Maselli
Sapienza, University of Rome
Rome, Italy
maselli@di.uniroma1.it

Mauro Piva
Sapienza, University of Rome
Rome, Italy
piva@di.uniroma1.it

Francesco Restuccia
Northeastern University
Boston, MA, USA
frestuc@northeastern.edu

ABSTRACT
Although smart environments are a key component of the Internet of Things (IoT), it is also clear that billions connected doors, washing machines, ovens and others will ultimately raise security and privacy concerns. Early work in this area, as well as most of commercial solutions, has adopted a centralized client/server approach, neglecting the multitude of risks that are induced by an unfair control of the server side. This has made the adoption of a decentralized and trust-less framework quintessential to guarantee devices security. Nevertheless, decentralized proposals are hardly applicable due to costs, slowness and privacy issues. In this paper, we make the use of blockchain practical for smart environments by designing HyBloSE, a secure-by-design and lightweight blockchain-based framework, able to run on low-power devices without additional hardware. HyBloSE is built by using Delegated Proof of Authority and a Moving Window Blockchain. We evaluate HyBloSE through a network emulator and real experiments with different Raspberry Pi platforms. Results show that HyBloSE guarantees a higher security level in terms of resiliency to internal and external attacks compared to centralized solutions, with overhead below 0.38s per operation and less than $4 per month for unlimited operations. Furthermore, we show how Proof of Authority is more adapt then Proof of Work in IoT private scenarios.

CCS CONCEPTS
• Security and privacy → Security protocols; Software and application security; • Networks → Network architectures.

KEYWORDS
Blockchain, Smart Building, Security, IoT

ACM Reference Format:

1 INTRODUCTION
Thanks to the Internet of Things (IoT), a fast-growing number of buildings is becoming smarter, connecting to the Internet an expected number of 55+ billion kitchen appliances, televisions, smartphones, utility meters, intra-body sensors, thermostats, etc. by 2025 [25, 26, 28, 32]. As homes and offices become more autonomous, we ask the following question: can a smart building be dangerous like a smart vehicle? Unfortunately, the security issues related to the development of a smart building are often undervalued [15]. This issues are further exacerbated by the centralized nature of the IoT. For example, smart speakers, which act mainly as echo devices, relaying traffic to/from the cloud [4]. However, delegating the full control of an autonomous place to devices and cloud services leads to security vulnerabilities [12].

Figure 1: IoT Devices inside a Smart Building

Figure 1 shows how the location of IoT devices can raise security and privacy concerns. For example, a smart plug may attack other devices or steal passwords [18], a widespread closed-source communication protocol may be exploited to allow attacks such as the one in [6], or an authority may trespass into smart devices to spy on people [5]. The key issue is that centralized systems controlling smart environments are susceptible to attacks due to incorrect behavior of employees or authorities [16]. Thus, it is quintessential to limit and control the capabilities of remote centralized control systems, introducing a decentralized and trust-less system, only manageable by the smart environment owners.

Existing work has tackled smart environments security through blockchain [2, 3, 8, 10, 17, 23]. However, speed of execution, privacy
and cost issues have limited the adoption of existing techniques in widespread commercial devices. The main limitations are (i) the need for additional hardware to be installed inside the smart environment; (ii) the need for IoT devices with high computation capabilities; (iii) the need to deal with blockchain related issues, like slowness, costs and privacy; (iv) the usage of private blockchain, which may be modified by the owner producing multiple forks [17].

This paper addresses the current research gap by proposing HyBloSE. Our paper’s goal is to make the use of blockchain practical for smart environments, with a lightweight solution able to run on low power devices without requiring additional hardware. HyBloSE is a secure-by-design blockchain framework based on a Delegated Proof of Authority and a Moving Window Blockchain.

To tackle the above limitations, HyBloSE introduces several novel features for smart buildings:

- HyBloSE makes a smart environment secure-by-design by splitting up security from application logic, and introducing an additional layer of security. This approach maintains the benefits of cloud-based logic systems (e.g., speed) limiting the drawbacks introduced by blockchain;
- HyBloSE introduces a Public Smart Building Contract, which clarifies and defines the functionalities of each device or system of a smart environment, restricting its capabilities even in case of exploitation. HyBloSE is based on a Delegated Proof of Authority. In this way HyBloSE does not involve any significant overhead, and differently form other blockchain based proposals, it does not require any additional hardware (e.g., powerful server inside homes);
- Finally, to allow traceability and be compliant with privacy regulations, HyBloSE presents a new approach to hybrid blockchain, introducing a moving window of signed contents, which are periodically deleted.

The remainder of the paper is structured as follows. Section 2 presents a description of the challenges which are still to be tackled to implement a blockchain system inside a smart building. Section 3 presents HyBloSE, and how these challenges are addressed. Section 4 assesses the effectiveness and the efficiency of the proposed architecture through emulations and real experiments, Section 5 presents the related work and Section 6 draws some conclusions.

2 CHALLENGES

The objective of this work is to make the use of blockchain practical for smart buildings. This involves several challenges, as we explain.

Speed: Blockchain is slow. How can we make it fast enough to meet expectations of smart building users? Well-known public blockchains are very slow for smart building applications. Bitcoin accepts up to 10 transaction per second, with a delay between each block of 15 minutes [16]. The faster Ethereum accepts up to 30 transaction per second, with a delay between each block of 20 seconds [29]. This means that when a user issues a new transaction, e.g., a transfer of funds or the execution of a function on the blockchain, she has to wait for at least 20 seconds for the request confirmation. However, users expect fast response, e.g., a light starts emitting light after touching a switch. A way to reduce this delay with the Proof of Work (PoW) consensus is to reduce the puzzle complexity [29]. However, this approach is faulty as the reduction makes the system open to attacks like the 51%.

Cost: Writing on a public blockchain has an economic cost. How can we make this cost low and independent on the building size? A public blockchain, like Bitcoin or Ethereum, is a distributed network of systems all running the same protocol and all owning a copy of the contents of the blockchain. The spread of new data through all the members of a blockchain requires time, thus public blockchains are slow in confirming transactions. Furthermore, writing on a public blockchain costs monetary resources, then used to refund the users that participate into the blockchain. On Ethereum, this fee increases depending on how much we want to write, and is around $0.10 per kilobyte [7]. Also, for each operation we want to perform there is a minimum fee of around $0.05. This means that for each proposal based on the Ethereum public blockchain, users are expected to pay at least $0.05 for each operation they perform.

Low power devices: How can we connect low power devices to a blockchain? There are multiple ways for a system to connect to a public blockchain. The first one involves a full node that connects itself to the other nodes running the public blockchain, obtains from them the entire blockchain, verifies it, and continuously receives new blocks from other nodes. A full node is also able to mine. However, this node must have a large amount of space available, e.g., more than 140 gigabytes for Ethereum. Also, if the system wants to mine it must be provided with at least a 64 bits processor, and with a powerful GPU. It is not realistic to expect such power on many IoT devices. The second way is through a light node: instead of maintaining the full blockchain, it stores only the blocks headers, reducing the space required. While this node can run on modest devices, it is still not able to run on low power devices, as it should be always connected to the peer to peer network. Another way to reach a public blockchain is through a gateway, like Infura [11]. These systems are free and publicly available, and offer classical REST API which devices can contact to interact with the public blockchain. These systems have full nodes running, and translates REST requests to blockchain transactions. However, the adoption of such systems creates a new single point of centralization.

Integrity: How can we guarantee that a private blockchain has not been altered? A way to solve the cost and speed issues related to the use of a public blockchain is the adoption of a private blockchain. Instead of having a public peer-to-peer network of miners, we have a close set of nodes connected together which operate on a blockchain disconnected from the public one. The owner of the nodes in this case can modify the underlying protocol, for example changing the way in which the Proof of Work puzzle complexity is defined, reducing or increasing it. However, as the owner can control all the nodes, she may easily change their content. For example, in case of an inappropriate event, the owner may induce some nodes in producing a fork, maintaining two copies of the blockchain, one with the event and the other one without.

How can we combine the blockchain immutability with the privacy right to be forgotten? In the last years the need for users’ privacy protection has become evident [9]. Two of the most discussed principles of privacy are the temporality of data and the right to be forgotten. The first principle defines that collected data must be
stored for a limited amount of time, depending on its specific utilization. The second defines that each user must be able to ask for the removal of data that regards itself. Both these principles are not obtainable on a public blockchain, and requires specific changes in a private one.

3 THE HYBLOSE SYSTEM

We propose HyBloSE (see Figure 2), a practical solution that enhances smart environments security and maintains the advantages of a centralized orchestration, involving a public and a private moving blockchain. HyBloSE addresses the above challenges as follows. To address speed HyBloSE introduces a Delegated Proof of Authority (DPoA) and a Private Moving Blockchain. To reduce cost HyBloSE uses a combination of the Ethereum public Blockchain and a private moving Blockchain. To integrate low-power devices HyBloSE exploits powerful nodes as a gateway for low-power ones, in addition to the DPoA. To ensure integrity and to guarantee privacy, HyBloSE exploits the characteristics of the private moving window blockchain.

Delegated Proof of Authority. The Proof of Authority is a consensus mechanism where only a set of authorized nodes can mine. In a smart building, the owner can indicate in a Smart Contract the authorized nodes. To limit power consumption, and to let mine only part of the nodes, in the private moving blockchain a Delegated Smart Contract elects two miners to mine for the next hour.

At the end of each hour elected miners invoke the Delegates Smart Contract, and elects the two miners which will be active for the next hour. This election is performed by calculating the number of operations that happened in the last our, excluded the ones produced by the active miners, and with a modulo operation on the set of potential authorized nodes. The two miners that are currently mining can not mine in the successive hour. The number of operations is required because current nodes may influence a random number, becoming able to deterministically define the next nodes that will mine.

The number of operations is hard to be forged by the miners, as the only way they have would be to refuse some devices requests. However, refusing requests automatically activates the other miners which were not mining, activating a security procedure, in which the other nodes produce a new fork and the exclusion of the misbehaving node. Non elected nodes perform two actions: i) verify the operated of elected nodes, and in case of misbehavior start mining; ii) verify that the elected nodes include in the private blockchain all the requests from other nodes; otherwise, in case elected nodes are excluding some device, not elected nodes start mining. In this way only two nodes per time have high energy consumption, but an attacker still have to exploit the majority of all the nodes with mining capabilities to control the building. Differently from the PoW, the DPoA does not requires the ethash DAG computation. This operation requires a huge amount of memory and a 64bits processor. Removing this operation, the DPoA can be executed even on low power nodes with reduced capabilities.

Private Moving Blockchain. Each device with a fixed source of power and with modest computation capabilities, for reference a Raspberry Pi Zero is enough, can run the Private Moving Blockchain. This blockchain follows the Ethereum protocols, while differing for three reasons. First, it implements the Delegated Proof of Authority consensus, meaning that only authorized devices, the ones contained in the set M of the Public Smart Building Contract, can mine and create new blocks. This blockchain manages the Delegated Smart Contract and a perfect copy of the Public Smart Building Contract, that is continuously updated. Second, periodically the hash of some blocks is written on the public blockchain. Third, periodically the oldest part of the private blockchain is removed. The remaining part always starts from the header of a node whose hash has been recorded on the public blockchain. In this way HyBloSE is able to satisfy the privacy requirements of data temporarily and right to be forgotten. The privacy is also increased as the private moving blockchain is only local to the smart building, and not accessible from outside.

Secure Oracle. Low power nodes can not run a Blockchain node. In order to interact with the moving blockchain Blockchain-Enable actuators act as gateways for low power nodes. However, low power nodes sign the requests they send to the gateways with their ethereum account. In this way the Blockchain-Enable actuators are not able to impersonate other nodes, but can only receive their requests and write them on the moving blockchain. The entire system, as depicted in Fig. 2, is based on a mesh network. This avoids disconnections of low power nodes due to a single Blockchain-Enable actuators failure.

Blockchain-Enabled Actuators. All smart building actuators connected to a fixed source of power can be miner. The computational capability required to contribute to the Moving Window Blockchain is not high, as it is based on the Proof of Authority consensus. These actuators receive information, i.e. sensed data, from sensors or remote services. For example, when a switch is enabled by an user, the switch sends the information to at least one Blockchain-Enable Actuator. Active miners will take the information and insert it in the first available block. Depending on the sensed information, an actuator may be required by an external logic system to perform an action. This request is written on the Moving Blockchain. Once that the actuator reads the request, it verifies the request signatures and the request’s maker authorizations.
Table 1: Qualitative Analysis of HyBloSE

<table>
<thead>
<tr>
<th>Layer</th>
<th>Threat</th>
<th>Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>Unauthorized access to an Exchange Server</td>
<td>Only nodes authorized by the smart building owner can access the private moving blockchain. If the intruder controls less than the half of the nodes running the private blockchain it cannot perform attacks.</td>
</tr>
<tr>
<td></td>
<td>Exchange Denial of Service</td>
<td>The private network is based on a mesh, and has no single point of failure. In case of misbehaving nodes, these nodes are excluded from mining or from the entire system.</td>
</tr>
<tr>
<td></td>
<td>Employees Host Security</td>
<td>No personal data is collected in the public or the private blockchain. The actions which happen inside the smart building are only temporarily recorded on the private blockchain.</td>
</tr>
<tr>
<td></td>
<td>Malicious Program Infection</td>
<td>A malicious agent must compromise at least the majority of the mining nodes, otherwise the other nodes can exclude compromised ones automatically.</td>
</tr>
<tr>
<td></td>
<td>Reentrancy Attack</td>
<td>Each device to device authorization is verified as the first step of each action execution, making the reentrancy attack not feasible.</td>
</tr>
<tr>
<td></td>
<td>Unauthorized Access Attack</td>
<td>Since each authorized miner is directly recorded in the private blockchain and managed by voting, an attacker in order to become dangerous must first control more than the half of the mining nodes.</td>
</tr>
<tr>
<td></td>
<td>Solidity Development Security</td>
<td>Not applicable. Each node is robust to the mining pool attack, as only authorized nodes can mine.</td>
</tr>
<tr>
<td>Smart Contract</td>
<td>Mining Pool Attack</td>
<td>The private network is based on a mesh, and has no single point of failure. In case of misbehaving nodes, these nodes are excluded from mining or from the entire system.</td>
</tr>
<tr>
<td></td>
<td>Reentrancy Attack</td>
<td>The private network is based on a mesh, and has no single point of failure. In case of misbehaving nodes, these nodes are excluded from mining or from the entire system.</td>
</tr>
<tr>
<td></td>
<td>Unauthorized Access Attack</td>
<td>Since each authorized miner is directly recorded in the private blockchain and managed by voting, an attacker in order to become dangerous must first control more than the half of the mining nodes.</td>
</tr>
<tr>
<td></td>
<td>Solidity Development Security</td>
<td>The entire system is based on a set of standard smart contracts.</td>
</tr>
<tr>
<td>Consensus</td>
<td>Bribe attack</td>
<td>Validators do not follow an economic logic. This makes the bribe attack not applicable.</td>
</tr>
<tr>
<td></td>
<td>Long range attack</td>
<td>The private key of the authorized miners is known only to the miners. For this reason an attacker should first exploit the nodes before making the attack.</td>
</tr>
<tr>
<td></td>
<td>Coin Age Accumulation Attack</td>
<td>Not applicable.</td>
</tr>
<tr>
<td></td>
<td>Precomputing Attack</td>
<td>If a node performs a Precomputing, other sleeping powerful nodes would reveal it. These nodes would then start mining and mitigate the attack.</td>
</tr>
<tr>
<td></td>
<td>Sybil Attack</td>
<td>As the network or nodes is a mesh, it is hard to disconnect it or to hide parts of it. However, all systems which provide a private blockchain are vulnerable in case of network disjunction.</td>
</tr>
<tr>
<td>Network</td>
<td>Eclipse Attack</td>
<td>The mesh network limits this attack feasibility. Furthermore, a node performing dos or behaving inconsistently may be excluded by the others with a vote.</td>
</tr>
<tr>
<td>Data</td>
<td>Block Data</td>
<td>The private moving blockchain may only collect structured data coming from authorized nodes.</td>
</tr>
<tr>
<td></td>
<td>Signature and Encryption Method</td>
<td>As far as now the SHA256 algorithm is used, but it can be updated with a new building smart contract.</td>
</tr>
</tbody>
</table>

Public Smart Building Contract. The Smart Contract is deployed on the public Ethereum blockchain, and contains:

- \( U = \{ u_1, ..., u_n \} \) a set containing the Ethereum addresses of the users allowed to interact with the contract;
- \( M = \{ m_1, ..., m_n \} \) a set containing the Ethereum addresses of the authorized miners inside a smart building;
- \( D = \{ d_1, ..., d_n \} \) a set containing the Ethereum addresses of the entities, devices and remote services, related to the smart building;
- \( A = \{ a_1, ..., a_k \} \) a set of encoded strings defining all the actions which can be performed by actuators;
- \( R(d_a, d_b, a) = 0, 1 \) a function which takes as input the addresses of two devices and an action, and defines if the device \( d_a \) can trigger the action \( a \) of device \( d_b \).

Each time an user wants to define a new building, he deploys a new Public Smart Building Contract. For each new device, the user updates the smart contract with identities and rules.

Users. In order to start HyBloSE an user has to perform the following actions: i) she deploys the Public Smart Building Contract to the Ethereum public blockchain; ii) she sets the Public Smart Building Contract in the private moving window blockchain; iii) the private moving window creates a copy of the Public Smart Building Contract and asks to the user to set the list of authorized miners, the list of smart building devices and relative authorizations. With HyBloSE the owner of the smart building is able to define the rules for his autonomous place, and even in case of a cloud service or device attack, actuators would not be able to perform actions not already allowed, making the autonomous place secure by design. Through the use of blockchain, HyBloSE guarantees that a cloud service or an IoT device, even if compromised, cannot perform unforeseen actions. Also, with HyBloSE it is always possible to track who or what triggered an action, both for statistics and accountability.

4 PERFORMANCE EVALUATION

To study the effectiveness of HyBloSE we perform a qualitative analysis, while to assess efficiency we carry out a quantitative analysis that includes both emulation and real experiments. We consider two system architectures. The first one, named cloud logic, considers a cloud-based management logic in which all sensor packets are routed to actuators through the cloud service. The second architecture, named local logic, keeps all decisions local and sensor packets are routed directly to actuators, according to a local management system. These architectures are considered in standalone fashion as well as integrated with the HyBloSE system, so that we can evaluate the impact of introducing the cryptographic operations.

4.1 Effectiveness

We now look at the effectiveness of HyBloSE through a qualitative security analysis, which follows the approach described in [16, 27]. Table 1 shows how HyBloSE is resilient to different types of threats in each layer of the blockchain. The application layer is mainly focused on issues related to the centralization points of a system. HyBloSE tackles most of them with DPoA, as only authorized nodes can take part in the system, and through its mesh network. The mesh structure makes also HyBloSE robust to network layer attacks. The Smart Contract layer examines the application source code. As HyBloSE Smart Contracts have the same source for every building, the surface of potentially vulnerable code is reduced. The third layer refers to the consensus mechanism, and the Delegated Proof of Authority is able to contrast all the reported security risks. The fourth investigates the network, and the last investigate the Data layer, where the private moving blockchain is fundamental, as able to guarantee confidentiality, as the data are local to the smart building, and privacy, as data are periodically removed.
4.2 Efficiency: Emulation

We evaluate HyBloSE by using SMARTEEX [19], a software tool to emulate complex network scenarios and collecting statistics. We consider a smart home scenario with up to 40 devices. We measure the overhead — the time incurred by HyBloSE operations — and the reaction time — the time since when a sensor detects a new event to when the actuator takes the corresponding action.

Figure 3 shows the results on overhead by varying the type of devices. The additional time incurred by applying HyBloSE respectively to local and cloud logic architectures, is very short (between 100ms and 150 ms), and is negligible from a user perspective. Furthermore, the overhead keeps quite constant independently of the type of device. We then evaluate the impact of diminishing the number of devices, as a measure of scalability. We consider two different settings: (i) a simple case including 10 devices (5 sensors and 5 actuators), and (ii) a more complex case, with 40 devices (20 sensors and 20 actuators). In both settings, the number of devices does not have an impact on reaction time. Results with 10 and 40 devices are comparable, independently of the type of devices. Thus, our first set of results clearly show the efficiency of HyBloSE, which results lightweight (i.e., contained overhead) and scalable.

4.3 Efficiency: Experiments

We now present the results of our experimental evaluation with 10 devices: 3 Raspberry Pi Zero W, 3 Raspberry Pi 4, 1 Raspberry Pi 3b+, 1 Raspberry Pi 2 and 2 computers, with i7 and i9 processor family. Experimental results (see Fig. 5) show that in both architectures the overhead introduced by HyBloSE is limited, i.e., 370ms for the local logic architecture and 140ms for the cloud logic counterpart. These values confirm the outcome of the emulation analysis. We also investigate how much time the private blockchain employs to create the first block, i.e. the time required between the system startup and the first action in the smart building. We compare it with SWBC [16], the most recent state of the art blockchain framework for smart building (based on a modification of PoW), and a classical private blockchain based on PoW. We found that the production of the first block in PoW systems has an high computational cost due to the generation of the Ethash DAG, requisite for mining [29], that requires a long time, 220seconds like shown in Figure 6, even on a Pi 4. This operation is impossible on 32 bit processors, due to RAM address space limits. This makes PoW based systems not work 32 bit and low power devices, like the Pi Zero. The Prof of Authority instead does not involve a DAG, making it suitable even for less powerful devices. Thus, HyBloSE can run on low power and 32 bit devices, requiring always less than 2 seconds to start, being more than 100 times faster then solutions based on PoW, as shown in Figure 6.

**Economic feasibility.** We report an effective cost analysis of the operations related to the public blockchain, presented in Table 2. This test has been performed with the real contract on the Ethereum main net, performing effective operations. We defined different gas costs depending on each operation. e.g. we want to be as fast as possible while deactivating rules but we accept a delay while removing a device which rules have already been deactivated. A lower amount of gas paid produces higher delay in operation execution. The most expensive operation is the deployment of the Smart Building Contract, which requires 4.60$, but which is executed only once per building. Other operations have a really low cost, always less then 0.20$. These operations are executed rarely, only when a user introduces or removes devices from the smart building. The snapshot is instead performed once per day, with a monthly cost of about 3$. We recall that all the other operations are performed on the private moving blockchain, and do not involve any cost.

<table>
<thead>
<tr>
<th>Operation</th>
<th>GAS</th>
<th>GWei</th>
<th>Time(s)</th>
<th>$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smart Building Contract Deployment</td>
<td>760022</td>
<td>25</td>
<td>226</td>
<td>4.60$</td>
</tr>
<tr>
<td>Add a new Device</td>
<td>21800</td>
<td>29</td>
<td>64</td>
<td>0.15$</td>
</tr>
<tr>
<td>Remove a Device</td>
<td>21000</td>
<td>18</td>
<td>6598</td>
<td>0.095$</td>
</tr>
<tr>
<td>Create a new Rule</td>
<td>21204</td>
<td>29</td>
<td>64</td>
<td>0.145$</td>
</tr>
<tr>
<td>Deactivate a Rule</td>
<td>21000</td>
<td>32</td>
<td>27</td>
<td>0.168$</td>
</tr>
<tr>
<td>Private Blockchain Snapshot</td>
<td>21800</td>
<td>18</td>
<td>6598</td>
<td>0.105$</td>
</tr>
</tbody>
</table>

5 RELATED WORK

The dilemma of how to make IoT secure has raised significant interest from the research community over the last years [14, 22, 24, 30]. Several survey has promoted the adoption of blockchain for IoT security [2, 3, 8, 10, 17, 23]. However, current approaches are still far from being adopted because several practical issues. Dorri et al., [13] propose the use of local private blockchain for smart-home, but they require an additional high power system
inside the smart-home, which creates a central private blockchain. Approaches with only a private blockchain fail in the traceability feature, as losing the distributed ledger allows modification of the blockchain contents. HyBloSE instead relies on both public and private blockchain, guaranteeing security and transparency. The need for additional hardware is also present in [20], where authors propose to use the blockchain for an access control system based on IoT devices. In this case an Ethereum Smart Contract contains the list of managers, i.e. people able to define specific access control permissions for devices. Adler et al. tackled the problem of low power IoT devices proposing an oracle management hub [1], an hub that translates messages from IoT devices, encoded in CoAP, into JSON-RPC messages understandable by blockchain nodes. As recognized by authors, this proposal has low applicability in IoT scenarios due to public blockchain transaction costs and delays. HyBloSE instead exploits a private blockchain for permissions management, reducing blockchain related costs and decreasing delays.

The access management problem has also been tackled by Zhang et al., [31], which extended the problem to generic rule-based resources management, introducing one Ethereum Smart Contract for each rule. Interactions between entities, i.e. access to resources, are checked by another smart contract called judge that controls and penalizes devices having unfair behaviour. Even in this work authors acknowledge that the proposal is hard to be applied: the average time for each operation is around 30 seconds (unacceptable for the user), while costs related to blockchain are not analyzed. Also Pinno et Al. [21] worked on permission management through blockchain, presenting ControlChain, a framework composed by four blockchains which supports access management models like RBAC, OrBAC, ABAC, UCON, CapBAC. However, ControlChain has not been publicly deployed, and the proposal does not evaluate costs and delays of the new blockchain. HyBloSE instead is deployed on a public, trustworthy and active blockchain. This allowed us to evaluate both the HyBloSE costs and induced delays, demonstrating the negligible impact of our proposal.

6 CONCLUSION

In this paper, we have presented HyBloSE, a system able to secure smart buildings even in case of attack to the cloud management system. Our approach is based on a Delegated Proof of Authority consensus and a Moving Window Blockchain. HyBloSE decouples the security aspects from the application logic of IoT, saving the centralized logic and distributing the security management. HyBloSE overcomes related work limitations, especially in terms of practicality: HyBloSE is able to run on low power devices, and without requiring any additional hardware. Results demonstrate that HyBloSE induces negligible overhead (below 0.4ms), resulting fully compatible with already deployed hardware.

ACKNOWLEDGEMENTS

This work was supported in part by Sapienza under grant “Avvio Alla Ricerca - AR1916B891EB23”.

REFERENCES


